Durham University

Durham E-Theses

A computer based college record system providing for record interrogation and data interchange

Bell, D. M. R.

How to cite:

Bell, D. M. R. (1973) *A computer based college record system providing for record interrogation and data interchange*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/10066/

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full Durham E-Theses policy for further details.

Academic Support Office, The Palatine Centre, Durham University, Stockton Road, Durham, DH1 3LE
e-mail: e-theses.admin@durham.ac.uk Tel: +44 0191 334 6107
http://etheses.dur.ac.uk

BCRS.....BEDE COLLEGE RECORD SYSTEM

PROGRAMME DOCUMENTATION

(PART 2)

ataset Limited

+M.Sc.200

BCRS.....BEDE COLLEGE RECORD SYSTEM

PROGRAMME DOCUMENTATION

(PART 2)

Dataset Limited

ABSTRACT

THE DATA MANAGEMENT SYSTEM BCRS RECORDS AND MAINTAINS STUDENT RECORDS IN DATA BASE FILES.

THE SYSTEM CONSISTS OF A SUITE OF PL/1 (PROGRAMMING LANGUAGE 1) PROGRAMMES DESIGNED TO RUN ON THE NUMAC IBM 360/67 COMPUTER UNDER MTS (MICHIGAN TERMINAL SYSTEM) IN BATCH MODE.

THIS REPORT TOGETHER WITH 'BCRS.....BEDE COLLEGE RECORD SYSTEM, PROGRAMME DOCUMENTATION, (PART 1)' DESCRIBES THE PROGRAMMES AND PROVIDES LISTINGS OF THEIR SOURCE CODE, THE INPUT EXPECTED AND THE OUTPUT PROVIDED.

PROGRAMME DOCUMENTATION (PART 2)

CONTENTS.

		PART
1.	INTRODUCTION.	1
2.	PROGRAMME VARIABLES.	1
	2.1 THE RECORD STRUCTURE	1
	2.2 PROGRAMME VARIABLES	1
	2.3 PROGRAMME FILES	1
3.	PROGRAMME R.	
	3.1 PROGRAMME INTRODUCTION	1
	3.2 PROGRAMME LISTING	2
	3.3 COMPILER LISTING	2
4.	PROGRAMME F.	
	4.1 PROGRAMME INTRODUCTION	1
	4.2 PROGRAMME LISTING	2
	4.3 COMPILER LISTING	2
5.	PROGRAMME I.	
	5.1 PROGRAMME INTRODUCTION	1
	5.2 PROGRAMME LISTING	2
	5.3 COMPILER LISTING	2
6.	PROGRAMME D.	
	6.1 PROGRAMME INTRODUCTION	1
	6.2 PROGRAMME LISTING	2
	6.3 COMPILER LISTING	2
7.	THE DATA CONSTANTS FILE.	1
	7.1 FILE LISTING	2
8.	INPUT DATA.	1
	8.1 DATA LISTINGS	2
9.	OUTPUT LISTINGS.	1
	9.1 OUTPUT LISTINGS	2
10.	D.E.S. OUTPUT LISTINGS.	1
	10.1 OUTPUT LISTINGS	2
APPENDIX 1.	THE FILE SEARCH ROUTINES.	1
APPENDIX 2.	THE EVALUATION OF A 'REQUEST'.	1

3. PROGRAMME R.

3.2 PROGRAMME LISTING

THIS SECTION CONTAINS A LISTING OF THE SOURCE CODE OF PROGRAMME R.

Dataset Limited

```
 1                                                                  /*****************************************/
 2    (SUBRG):(STRG):                                               /*                                       */
 3    R:PROCEDURE OPTIONS(MAIN);                                    /*            BCRS....PROGRAMME R        */
 4         DCL 1 RECORD BASED(PTR),                                 /*                                       */
 5               2 (DESNO CHAR(8), NAME CHAR(40)),                  /*****************************************/
 6               2 GENERAL,                                         /*                                       */
 7                 3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),  /* DECLARATION OF STRUCTURE 'RECORD'  */
 8                 3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),  /* -> MINOR STRUCTURES OF 'RECORD'  */
 9                 3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)),  /*    -> GENERAL (GENERAL INFORMATION)  */
10                 3 FPOS CHAR(4),                                  /*    -> EDUCATION (PREVIOUS EDUCATION)  */
11               2 EDUCATION,                                       /*    -> COURSE (COURSE DETAILS)  */
12                 3 (SCHNO CHAR(2), SCH(10) CHAR(5)),              /*    -> FINALS (FINAL RESULT)  */
13                 3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),           /*    -> ANNPROG (ANNUAL PROGRESS)  */
14                 3 (UNCOL(2) CHAR(21), FPT(2) CHAR(1), FT(2,2) CHAR(6)),  /*    -> INTEL (INTELLIGENCE TEST SCORES)  */
15                 3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),  /*
16                 3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),            /*
17                 3 (ADIP CHAR(30), ADIPC CHAR(1)),                /*
18                 3 DDIPC CHAR(1),                                 /*
19                 3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),              /*
20                 3 ATO CHAR(1),                                   /*
21               2 COURSE,                                          /*
22                 3 (LEN CHAR(1), CODES(4) CHAR(3)),               /*
23                 3 ((LE,DC(4,24:64)) CHAR(1)),                    /*
24                 3 (DC1(4,26:30) CHAR(2)),                        /*
25                 3 ((DC46(4),DC49(4)) CHAR(3)),                   /*
26               2 FINALS,                                          /*
27                 3 ((TC,WS,BE,SUBCAR) CHAR(1)),                   /*
28               2 ANNPROG,                                         /*
29                 3 AP CHAR(802),                                  /*
30                 3 ((LEAP,FK(100,2)) FIXED BINARY),               /*
31               2 INTEL,                                           /*
32                 3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY),  /*
33                 3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),          /*
34                 3 STEN(2,16) FIXED BINARY;                       /*
35    /*-----------------------------------------------------------*/  /*-----------------------------------------*/
36         DCL (NORM(2,16,13) FIXED BINARY, NORM2(2,16,2)           /* DECLARATION OF PROGRAMME VARIABLES  */
37         FLOAT DECIMAL, (OPSOK(6) CHAR(40), PSOK(5) CHAR(65), GRC CHAR(11),  /*
38         STK(6) CHAR(50),MK(3) CHAR(12),CK(10) CHAR(35),RK(5) CHAR(30),  /*
39         GRK(11) CHAR(30),OAK(4) CHAR(40),ORK(6) CHAR(30)) VARYING  /*
40         ) EXTERNAL;                                              /*
41         DCL 1 RECL(500),                                         /*
42               2 NAML CHAR(40),                                   /*
43               2 DESL CHAR(8),                                    /*
44               2 (DOBL,START) CHAR(6),                            /*
45               2 FIN CHAR(7),                                     /*
46               2 TYPE CHAR(1),                                    /*
47               2 DHT CHAR(3),                                     /*
48               2 DHY CHAR(2),                                     /*
49               2 DHB CHAR(21);                                    /*
50         DCL (FILN1,FILN2,DESNO2,DT,DES) CHAR(8),YE CHAR(4),TRANS CHAR(2),  /*
51         (PI,WC,YT,NTT) CHAR(1), (B,DESB) BIT(1), (OPT,KT,YR,NT)  /*
52         FIXED BINARY, BCON BIT(1) INITIAL('1'B), BMAX BIT(1), (VP,CSC)  /*
53         CHAR(1), VPB BIT(1), (VPI,I) FIXED BINARY, DESNO3 CHAR(8);  /*
54         DCL L(10) LABEL INITIAL(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10);  /*
55         DCL CON ENTRY(CHAR(8)), DESOP BIT(1) INITIAL('0'B);      /*
```

Dataset Limited

```
56    /*-----------------------------------------------------------------*/
57         DCL STUDENT FILE RECORD ENV(F(2416));
58         DCL DIC FILE;
59    /*****************************************************************/
60    RECIN:PROCEDURE;
61         DCL CONT CHAR(1), (S,Q,K) FIXED BINARY, X5 CHAR(5), X10 CHAR(10),
62         X2 CHAR(2);
63         GET EDIT(NAME,TITLE,SEX,DOB,DOBC,TTFT(1,1),MART,REG,SET,FPOS,POCC,
64         SOCC,OV25,REL,CONF,YC,AUTHNC)(COL(9),A(40),X(1),A(2),
65         2(A(1),A(6)),3(A(1)),A(4),5(A(1)),A(2),A(1));
66         GET EDIT(PNAME,AUTH,LEN,CODES(1),CODES(2),DC(1,24),DC(1,25),
67         DC1(1,26),DC1(1,27),DC1(1,29),DC1(1,30),DC(1,32),DC(1,34),DC(1,35)
68         ,DC(1,36),DC(1,37),DC(1,38),DC(1,41),DC(1,42),DC(1,43),DC(1,44),
69         DC(1,45),DC46(1),DC49(1),DC(1,59),DC(1,64))
70         (COL(9),A(17),A(19),A(1),2(A(3)),2(A(1)),4(A(2)),11(A(1)),2(A(3)),
71         2(A(1)));
72    L2:  GET EDIT(CONT)(COL(9),A(1));
73         IF CONT¬='C' THEN
74         DO; GET EDIT(ATO,SCHNO,QUALNO)(A(1),X(1),2(A(2)));
75             S=SCHNO;                                        /* CONVERSION */
76             DO K=1 TO MIN(S,10);
77                GET EDIT(SCH(K))(A(5));
78             END;
79             IF S>10 THEN
80             DO; PUT EDIT(DESNO,'ERROR.....SCHOOL TYPE STORAGE EXCEEDED')
81                 (COL(1),A,X(1),A);
82                 DO K=11 TO S;
83    L222:            GET EDIT(X5)(A(5));
84                     IF X5=SUBSTR(DESNO,1,5) THEN
85                     DO; GET EDIT(X5)(A(5));
86                         GOTO L222;
87                     END;
88                 END;
89             END;
90             IF MOD(S,2)=0 THEN GET EDIT(X5)(A(5));
91             Q=QUALNO;                                       /* CONVERSION */
92             DO K=1 TO MIN(Q,30);
93    L1:         GET EDIT(QUAL(K))(A(10));
94                IF QUAL(K)=DESNO || '  ' THEN GOTO L1;
95             END;
96             IF Q>30 THEN
97             DO; PUT EDIT(DESNO,'ERROR.....QUALIFICATION STORAGE EXCEEDED')
98                 (COL(1),A,X(1),A);
99                 DO K=31 TO Q;
100   L223:            GET EDIT(X10)(A(10));
101                    IF X10=DESNO || '  ' THEN GOTO L223;
102                END;
103            END;
104        END; ELSE
105        DO; GET EDIT(CONT)(A(1));
106            IF CONT='1' THEN
107            DO; DO K=1 TO 2;
108                    GET EDIT(UNCOL(K),FPT(K),FT(K,1),FT(K,2))
109                    (A(21),A(1),2(A(6)));
110                END;
111                GOTO L2;
112            END; ELSE
113            IF CONT='2' | CONT='3' THEN
114            DO; K=CONT;                                     /* CONVERSION */
115                K=K-1;
```

```
/*-----------------------------------------------------------------*/
/* DECLARATION OF FILE (STUDENT)                                   */
/* DECLARATION OF FILE (DIC)                                       */
/*****************************************************************/
/* RECORD INPUT                                                    */
/*****************************************************************/
/* VARIABLE DECLARATIONS                                           */
/* INPUT GENERAL INFORMATION                                       */
/*                                                                 */
/*                                                                 */
/* INPUT COURSE DETAILS                                            */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/* INPUT CONTINUATION MARKER                                       */
/* TEST FOR CONTINUATION                                           */
/* INPUT PREVIOUS EDUCATION                                        */
/* -> SCHOOL ATTENDANCES                                           */
/*                                                                 */
/* -> INPUT SCHOOL INFORMATION                                     */
/*                                                                 */
/* -> NO. ATTENDED >10                                             */
/*    -> CLASS Q ERROR/WARNING                                     */
/*                                                                 */
/*    -> IGNORE REMAINING SCHOOLS                                  */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/* -> QUALIFICATIONS RECORDED                                      */
/*                                                                 */
/* -> INPUT QUALIFICATIONS                                         */
/*                                                                 */
/*                                                                 */
/* -> NO. OF QUALIFICATIONS >30                                    */
/*    -> CLASS Q ERROR/WARNING                                     */
/*                                                                 */
/*    -> IGNORE REMAINING QUALIFICATIONS                           */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*                                                                 */
/* INPUT CONTINUATION NUMBER                                       */
/* CONTINUATION ROUTINE 1                                          */
/*                                                                 */
/* -> INPUT UNIV. OR COLL. ATTENDED                                */
/*                                                                 */
/*                                                                 */
/* -> BRANCH FOR NEXT CONTINUATION MARKER                          */
/*                                                                 */
/* CONTINUATION ROUTINE 2 OR 3                                     */
/*                                                                 */
/*                                                                 */
```

Dataset Limited

```
116                 GET EDIT(UNOTH(K),DEG(K),HPCD(K),SUB(K),CATT(K))          /* -> INPUT UNIV. OR OTHER AWARD                */
117                 (A(21),A(3),A(4),A(36),A(6));                             /*                                              */
118                 GOTO L2;                                                  /* -> BRANCH FOR NEXT CONTINUATICN MARKER       */
119             END; ELSE                                                     /*                                              */
120             IF CONT='4' THEN                                              /* CCNTINUATION ROUTINE 4                       */
121             DO; GET EDIT(ADIP,ADIPC,DDIPC,OQ(1),OQC(1),OQ(2),OQC(2))      /* -> INPUT CTHER QUALIFICATICNS                */
122                 (A(21),2(A(1)),2(A(19),A(4)));                            /*                                              */
123                 GOTO L2;                                                  /* -> BRANCH FOR NEXT CONTINUATION MARKER       */
124             END;                                                          /*                                              */
125         END;                                                              /*                                              */
126         LE='1';                                                           /* SET LATEST ENTRY                             */
127     END;                                                                  /*                                              */
128     /*****************************************************************/   /*******************************************/
129     CHECKREC:PROCEDURE;                                                   /* RECORD CHECKING ROUTINE                      */
130         DCL X2 CHAR(2), (K,Q) FIXED BINARY, OKB BIT(1);                   /*******************************************/
131         IF SUBSTR(TTFT(1,1),5,2)<'72' THEN OKB='1'B; ELSE OKB='0'B;       /* SET OLD/NEW CODES BIT                        */
132         X2=SUBSTR(TTFT(1,1),5,2);                                         /* CALCULATE YEAR OF ADMISSION                  */
133         K=X2;                                          /* CONVERSION */   /*                                              */
134         X2=SUBSTR(DOB,5,2);                                               /* CALCULATE YEAR OF BIRTH                      */
135         Q=X2;                                          /* CONVERSION */   /*                                              */
136         IF K-Q<25 THEN OV25=' '; ELSE IF OV25=' ' THEN OV25='0';          /* SET OVER 25 OCCUPATION IF NCT CODED          */
137         IF Q<17 | Q>60 THEN                                               /* AGE                                          */
138         PUT EDIT(DESNO,'AGE CHECK, RECORDED AS',Q)                        /* -> CLASS E ERROR/WARNING                     */
139         (COL(1),A,X(1),A,X(1),F(2));                                      /*                                              */
140         IF SEX¬='M' & SEX¬='F' THEN                                       /* SEX                                          */
141         PUT EDIT(DESNO,'ERROR RETURNED.....SEX')(SKIP,A(8),X(1),A);       /* -> CLASS E ERROR/WARNING                     */
142         IF MART<'1' | MART>'3' THEN                                       /* MARITAL STATUS                               */
143         PUT EDIT(DESNO,'ERROR RETURNED.....MARITAL STATUS')               /* -> CLASS E ERROR/WARNING                     */
144         (SKIP,A(8),X(1),A);                                               /*                                              */
145         IF REL<'1' | ((REL>'5' & ¬OKB) | (REL>'6' & OKB)) THEN            /* RELIGION                                     */
146         PUT EDIT(DESNO,'ERROR RETURNED.....RELIGION')(SKIP,A(8),X(1),A);  /* -> CLASS E ERROR/WARNING                     */
147         IF SET<'1' | ((SET>'4' & OKB) | (SET>'6' & ¬OKB)) THEN            /* SETTLEMENT                                   */
148         PUT EDIT(DESNO,'ERROR RETURNED.....SETTLEMENT')                   /* -> CLASS E ERROR/WARNING                     */
149         (SKIP,A(8),X(1),A);                                               /*                                              */
150         IF ¬OKB & (REG<'A' | REG>'K') THEN                                /* GEOGRAPHICAL REGION                          */
151         PUT EDIT(DESNO,'ERROR RETURNED.....REGION')(SKIP,A(8),X(1),A);    /* -> CLASS E ERROR/WARNING                     */
152         IF POCC<'1' | ((POCC>'5' & ¬OKB) | (POCC>'6' & OKB)) THEN         /*                                              */
153         PUT EDIT(DESNO,'ERROR RETURNED.....PARENT/GUARDIAN OCCUPATION')   /* -> CLASS E ERROR/WARNING                     */
154         (SKIP,A(8),X(1),A);                                               /*                                              */
155         IF SOCC¬=' ' & (SOCC<'1' | ((SOCC>'5' & ¬OKB) | (SOCC>'6' & OKB))) /* STUDENT OCCUPATION                           */
156         THEN PUT EDIT(DESNO,'ERROR RETURNED.....STUDENT OCCUPATION')      /* -> CLASS E ERROR/WARNING                     */
157         (SKIP,A(8),X(1),A);                                               /*                                              */
158         IF OV25¬=' ' & (OV25<'0' | OV25>'9') THEN                         /* OVER 25 OCCUPATION                           */
159         PUT EDIT(DESNO,'ERROR RETURNED.....OVER 25 OCCUPATICN')           /* -> CLASS E ERROR/WARNING                     */
160         (SKIP,A(8),X(1),A);                                               /*                                              */
161         IF SUBSTR(FPOS,1,2)>SUBSTR(FPOS,3,2) THEN                         /* FAMILY POSITION                              */
162         PUT EDIT(DESNO,'ERROR RETURNED.....FAMILY POSITION')              /* -> CLASS E ERROR/WARNING                     */
163         (SKIP,A,X(1),A);                                                  /*                                              */
164         IF SUBSTR(TTFT(1,1),3,4)¬=DC1(1,26) || DC1(1,27) THEN             /* START DATE                                   */
165         PUT EDIT(DESNO,'ERROR RETURNED.....START DATE CHECK')             /* -> CLASS E ERROR/WARNING                     */
166         (COL(1),A,X(1),A);                                                /*                                              */
167         IF ¬OKB & (AUTHNO<'1' | AUTHNO>'5') THEN                          /* AUTHORITY/COUNTRY NUMBER                     */
168         PUT EDIT(DESNO,'ERROR RETURNED.....AUTHORITY/COUNTRY NUMBER')     /* -> CLASS E ERROR WARNING                     */
169         (COL(1),A,X(1),A);                                                /*                                              */
170     END;                                                                  /*                                              */
171     /*****************************************************************/   /*******************************************/
172     ITRECIN:PROCEDURE;                                                    /* INTELLIGENCE TEST INPUT ROUTINE              */
173         DCL MNT(16) FIXED BINARY INITIAL(20,13,26,26,26,20,26,20,20,26,   /*******************************************/
174         20,26,20,20,20,26);                                               /* VARIABLE DECLARATIONS                        */
175         DCL (I,J) FIXED BINARY, SW BIT(1) INITIAL('0'B);                  /*                                              */
```

Dataset Limited

```
176          GET EDIT(VALUES(YR,*),DSCALE(YR),AH4(YR,*),PF16(YR,*))                  /* INPUT TEST SCORES                          */
177          (6(F(2)),F(3),2(F(2)),F(3),16(F(2)));                                   /*                                            */
178          DO K=1 TO 6;                                                            /*                                            */
179             IF VALUES(YR,K)<0 | VALUES(YR,K)>60 THEN                             /* TEST VALUES SCORES                         */
180             PUT EDIT(DESNO,'ERROR RETURNED FOR VALUES SCORE',K)                  /* -> CLASS F ERROR/WARNING                   */
181             (SKIP,COL(1),A,X(1),A,X(1),F(1));                                    /*                                            */
182          END;                                                                    /*                                            */
183          IF DSCALE(YR)<0 | DSCALE(YR)>300 THEN                                   /* TEST D.SCALE SCORE                         */
184          PUT EDIT(DESNO,'ERROR RETURNED FOR D.SCALE SCORE')                      /* -> CLASS F ERROR/WARNING                   */
185          (SKIP,COL(1),A,X(1),A);                                                 /*                                            */
186          DO K=1 TO 3;                                                            /*                                            */
187             IF AH4(YR,K)<0 | AH4(YR,K)>130 THEN                                  /* TEST A.H.4 SCORES                          */
188             PUT EDIT(DESNO,'ERROR RETURNED FOR A.H.4 SCORE',K)                   /* -> CLASS F ERROR/WARNING                   */
189             (SKIP,COL(1),A,X(1),A,X(1),F(2));                                    /*                                            */
190          END;                                                                    /*                                            */
191          DO K=1 TO 16;                                                           /*                                            */
192             IF PF16(YR,K)<0 | PF16(YR,K)>MNT(K) THEN                             /* TEST P.F.16 SCORES                         */
193             DO; PUT EDIT(DESNO,'ERROR RETURNED FOR P.F.16 SCORE',K)              /* -> CLASS F ERROR/WARNING                   */
194                 (SKIP,COL(1),A,X(1),A,X(1),F(2));                                /*                                            */
195                 SW='1'B;                                                         /* -> SET ERROR SWITCH                        */
196             END;                                                                 /*                                            */
197          END;                                                                    /*                                            */
198          IF SW THEN GOTO L2;                                                     /* BRANCH IF ERROR SWITCH ON                  */
199          DO I=1 TO 16;                                                           /* CALCULATE STEN SCORES                      */
200             DO J=1 TO 9;                                                         /*                                            */
201                IF PF16(YR,I)<=NORM(NT,I,1) THEN                                  /*                                            */
202                DO; STEN(YR,I)=J;                                                 /*                                            */
203                    GOTO L1;                                                      /*                                            */
204                END; ELSE                                                         /*                                            */
205                IF PF16(YR,I)>NORM(NT,I,J) & PF16(YR,I)<=NORM(NT,I,J+1) THEN      /*                                            */
206                DO; STEN(YR,I)=J+1;                                               /*                                            */
207                    GOTO L1;                                                      /*                                            */
208                END;                                                              /*                                            */
209             END;                                                                 /*                                            */
210    L1:   END;                                                                    /*                                            */
211    L2:   SUBSTR(INTELB,YR,1)='*';                                                /* SET INTELLIGENCE TEST BIT                  */
212    END;                                                                          /*                                            */
213    /*****************************************************************/           /*******************************************/
214    UPDATE:PROCEDURE;                                                             /* RECORD UPDATE ROUTINE                      */
215          DCL INST CHAR(70), X1 CHAR(1), XVAR CHAR(4) VARYING,                    /*******************************************/
216          (I,K,POSEND,POSEQ,J) FIXED BINARY;                                      /* VARIABLE DECLARATIONS                      */
217          GET EDIT(X1)(COL(9),A(1));                                              /* INPUT YEAR OF UPDATE                       */
218          IF X1>LEN THEN PUT EDIT(DESNO,'WARNING....YEAR OF UPDATE > COURSE'      /* TEST COURSE LENGTH                         */
219          ,' LENGTH')(COL(1),A,X(1),A,A);                                         /* -> CLASS H ERROR/WARNING                   */
220          GET EDIT(X1)(COL(10),A(1));                                             /* INPUT CONTINUATION                         */
221          IF X1='C' THEN                                                          /* TEST FOR FOURTH YEAR (IE B.ED. YEAR)       */
222          DO; PUT EDIT(DESNO,'PROCEEDING TO A FOURTH YEAR')(COL(1),A,             /* -> CLASS R ERROR/WARNING                   */
223              X(1),A);                                                            /*                                            */
224              LEN='4';                                                            /* -> ALTER COURSE LENGTH                     */
225          END; ELSE IF X1='F' THEN GOTO L2;                                       /* BRANCH IF FINAL RESULT                     */
226    L1:   GET EDIT(INST)(COL(11),A(70));                                          /* INPUT FIRST/NEXT CARD                      */
227          DO I=1 BY 10 TO 70;                                                     /*                                            */
228             IF SUBSTR(INST,I,2)='*A' THEN                                        /* -> LOCATE START OF ENTRY                   */
229             DO; POSEQ=I+INDEX(SUBSTR(INST,I+1),'=');                             /* -> LOCATE POSITION OF EQUALS SYMBOL        */
230                 POSEND=I+INDEX(SUBSTR(INST,I+1),'*');                            /* -> LOCATE END OF ENTRY                     */
231                 IF POSEQ=I THEN GOTO L3;                                         /* -> BRANCH IF EQUALS SYMBOL NOT FOUND       */
232                 IF POSEND=I THEN                                                 /* -> END OF ENTRY NOT FOUND                  */
233                 POSEND=I+INDEX(SUBSTR(INST,I+1),' ');                            /*    -> LOCATE END BLANK                     */
234                 XVAR=SUBSTR(INST,I+2,POSEQ-I-2);                                 /* -> CALCULATE FIELD KEY (LABEL) NUMBER      */
235                 K=XVAR;                                  /* CONVERSION */        /*                                            */
```

Dataset Limited

```
236                 IF FK(K,1)¬=0 THEN                                  /* -> TEST FOR RECORDED ENTRY                  */
237                 DO; SUBSTR(AP,FK(K,1))=SUBSTR(AP,FK(K,1)+FK(K,2));  /*    -> UPDATE 'AP' VECTOR                    */
238                     LEAP=LEAP-FK(K,2);                              /*    -> UPDATE 'LEAP' COUNTER                 */
239                     DO J=1 TO 100;                                  /*    -> ALTER SUBSEQUENT FIELD KEYS           */
240                        IF FK(J,1)¬=0 & FK(J,1)>FK(K,1) THEN         /*                                             */
241                        FK(J,1)=FK(J,1)-FK(K,2);                     /*                                             */
242                     END;                                            /*                                             */
243                     FK(K,*)=0;                                      /*    -> INITIALISE CURRENT FIELD KEY          */
244                 END;                                                /*                                             */
245                 IF POSEND=POSEQ+1 THEN GOTO L3;                     /* -> BRANCH IF NO ENTRY                       */
246                 IF (LEAP+POSEND-POSEQ)>802 THEN                     /* -> TEST 'AP' VECTOR SIZE                    */
247                 DO; PUT EDIT(DESNO,'ANNUAL PROGRESS VECTOR SIZE EXCEEDED',  /*    -> CLASS P ERROR/WARNING                 */
248                     '...INFORMATION IGNORED')(COL(1),A,X(1),A,A);   /*                                             */
249                     DO J=1 BY 1;                                    /*    -> IGNORE REMAINING INFORMATION          */
250                        IF SUBSTR(INST,70,1)='C' THEN GET SKIP; ELSE RETURN;  /*    -> RETURN FOR NEXT UPDATE                */
251                     END;                                            /*                                             */
252                 END;                                                /*                                             */
253                 SUBSTR(AP,LEAP,POSEND-POSEQ)=SUBSTR(INST,POSEQ+1,   /* -> ADD ENTRY TO 'AP' VECTOR                 */
254                 POSEND-POSEQ);                                      /*                                             */
255                 FK(K,1)=LEAP;                                       /* -> SET FIELD KEY POSITION POINTER           */
256                 LEAP=LEAP+POSEND-POSEQ;                             /* -> SET 'LEAP'                               */
257                 FK(K,2)=POSEND-POSEQ;                               /* -> SET FIELD KEY LENGTH POINTER             */
258             END;                                                    /*                                             */
259     L3:  END;                                                       /* BRANCH FOR NEXT ENTRY                       */
260          IF SUBSTR(INST,70,1)='C' THEN GOTO L1; ELSE RETURN;        /* BRANCH IF CONTINUATION REQUIRED             */
261     L2:  GET EDIT(TC,WS,BE,SUBCAR,TTFT(1,2))                        /* INPUT FINAL RESULTS                         */
262          (COL(20),A(1),COL(30),A(1),COL(40),A(1),COL(50),A(1),      /*                                             */
263          COL(60),A(6));                                             /*                                             */
264          TTFT(2,1)='LEFT  ';                                        /* -> RECORD STUDENT AS 'LEFT'                 */
265     END;                                                            /*                                             */
266     /*****************************************************************/  /*****************************************/
267     SPNT:PROCEDURE;                                                 /* FILE OUTPUT ROUTINE 1                       */
268          DCL (I,N) FIXED BINARY, PB BIT(1) INITIAL('0'B), SW BIT(1);  /*****************************************/
269          IF PI='2' THEN                                             /* PRINT INDICATOR 2                           */
270          DO; PUT SKIP(3);                                           /*                                             */
271              DO I=1 TO KT;                                          /*                                             */
272                 IF MOD(I-1,10)=0 THEN PUT SKIP;                     /* -> OUTPUT DES.NO., 10 PER LINE              */
273                 PUT EDIT(DESL(I))(X(2),A(8));                       /*                                             */
274                 IF FIN(I)¬='        ' THEN IF SUBSTR(FIN(I),7,1)='*' THEN  /* -> OUTPUT '*' IF WITHDRAWN                  */
275                 PUT EDIT('*')(A); ELSE PUT EDIT('+')(A); ELSE PUT EDIT(' ')  /* -> OUTPUT '+' IF LEFT                       */
276                 (A);                                                /*                                             */
277              END;                                                   /*                                             */
278              RETURN;                                                /*                                             */
279          END; ELSE                                                  /*                                             */
280          IF PI='3' THEN                                             /* PRINT INDICATOR 3                           */
281          DO; ON ENDPAGE(SPRINT)                                     /* -> SET ENDPAGE ROUTINE                      */
282              BEGIN; IF PB THEN PUT PAGE;                            /*    -> NEW PAGE WHEN REQUIRED                */
283                     PUT EDIT('NAME','D.E.S.NO.','BIRTH','START','FINISH')  /*    -> OUTPUT TITLES                         */
284                     (SKIP(2),COL(1),A,COL(42),A,3(X(5),A));         /*                                             */
285                     PUT SKIP(2);                                    /*                                             */
286              END;                                                   /* -> END ENDPAGE ROUTINE                      */
287              N=KT;                                                  /* -> BUBBLE SORT ROUTINE                      */
288     L1:      SW='0'B;                                               /*                                             */
289              N=N-1;                                                 /*                                             */
290              DO I=1 TO N;                                           /*                                             */
291                 IF NAML(I)>NAML(I+1) THEN                           /*                                             */
292                 DO; RECL(KT+1)=RECL(I);                             /*                                             */
293                     RECL(I)=RECL(I+1);                              /*                                             */
294                     RECL(I+1)=RECL(KT+1);                           /*                                             */
295                     SW='1'B;                                        /*                                             */
```

Dataset Limited

```
296                 END;                                                    /*                                          */
297              END;                                                       /*                                          */
298              IF SW THEN GOTO L1;                                        /*                                          */
299              SIGNAL ENDPAGE(SPRINT);                                    /* -> SIGNAL ENDPAGE ROUTINE                */
300              PB='1'B;                                                   /* -> SET NEW PAGE BIT                      */
301              DO I=1 TO KT;                                              /* -> OUTPUT RECORDS                        */
302                 PUT EDIT(NAML(I),DESL(I),SUBSTR(DOBL(I),1,2),'/',SUBSTR(  /*                                        */
303                 DOBL(I),3,2),'/',SUBSTR(DOBL(I),5,2),SUBSTR(START(I),1,2),  /*                                       */
304                 '/',SUBSTR(START(I),3,2),'/',SUBSTR(START(I),5,2))         /*                                        */
305                 (SKIP,COL(1),A(40),X(2),A(8),2(X(2),5(A)));                /*                                        */
306                 IF FIN(I)¬='        ' THEN PUT EDIT(SUBSTR(FIN(I),1,2),    /*                                        */
307                 '/',SUBSTR(FIN(I),3,2),'/',SUBSTR(FIN(I),5,2))(X(3),5(A));  /*                                       */
308              END;                                                       /*                                          */
309              ON ENDPAGE(SPRINT) SYSTEM;                                 /* -> RESET ENDPAGE ROUTINE                 */
310           END;                                                          /*                                          */
311        END;                                                             /*                                          */
312        /*****************************************************************/  /*****************************************/
313        PRINT:PROCEDURE;                                                 /* FULL PRINT ROUTINE                       */
314           DCL (I,K) FIXED BINARY;                                       /* VARIABLE DECLARATIONS                    */
315           /**************************************************************/  /*****************************************/
316        CDATE:PROCEDURE(X6) RETURNS(CHAR(8));                            /* DATE PRINT ROUTINE                       */
317           DCL X6 CHAR(6);                                               /*****************************************/
318           IF X6¬='        ' THEN                                        /* ACCEPTS 6 CHARACTER DATE                 */
319           RETURN(SUBSTR(X6,1,2) || '/' || SUBSTR(X6,3,2) || '/' || SUBSTR(  /* RETURNS 8 CHARACTER DATE               */
320           X6,5,2)); ELSE RETURN('         ');                           /*                                          */
321        END;                                                             /*                                          */
322        DCL CDATE ENTRY(CHAR(6)) RETURNS(CHAR(8));                       /*                                          */
323           /**************************************************************/  /*****************************************/
324        GENPNT:PROCEDURE;                                                /* GENERAL INFORMATION PRINT ROUTINE        */
325           DCL OKB BIT(1), X6 CHAR(6), X1 CHAR(1) DEFINED X6 POS(6);     /*****************************************/
326           IF SUBSTR(TTFT(1,1),5,2)<'72' THEN OKB='1'B; ELSE OKB='0'B;   /* SET OLD/NEW CODES BIT                    */
327           IF SEX='M' THEN PUT EDIT('MALE,')(COL(10),A); ELSE            /* SEX                                      */
328           PUT EDIT('FEMALE,')(COL(10),A);                               /*                                          */
329           IF DOB¬='        ' THEN PUT EDIT('BORN: ',CDATE(DOB))(X(1),A,A);  /* DATE OF BIRTH                          */
330           IF TITLE='M1' THEN PUT EDIT('TITLE: MR')(COL(10),A); ELSE     /* TITLE                                    */
331           IF TITLE='M2' THEN PUT EDIT('TITLE: MRS')(COL(10),A); ELSE    /*                                          */
332           IF TITLE='M3' THEN PUT EDIT('TITLE: MISS')(COL(10),A);        /*                                          */
333           PUT SKIP(2);                                                  /*                                          */
334           X6=TTFT(1,1);                                                 /* TEACHERS' TRAINING: FROM                 */
335           PUT EDIT('TEACHERS'' TRAINING: FROM',CDATE(X6))(COL(10),A,X(1),A);  /*                                  */
336           X6=TTFT(1,2);                                                 /* TEACHERS' TRAINING: TO                   */
337           IF X6¬='        ' THEN PUT EDIT('TEACHERS'' TRAINING: TO  ',  /*                                          */
338           CDATE(X6))(COL(10),A,X(1),A);                                 /*                                          */
339           X6=TTFT(2,1);                                                 /* REASON FOR WITHDRAWAL                    */
340           IF SUBSTR(X6,1,4)='WITH' THEN PUT EDIT('STUDENT WITHDREW, REASON',  /*                                  */
341           'CODED AS',X1)(COL(10),A,2(X(1),A));                          /*                                          */
342           PUT SKIP(2);                                                  /*                                          */
343           IF PNAME¬='    ' THEN PUT EDIT('PREVIOUS SURNAME',PNAME)      /* PREVIOUS SURNAME                         */
344           (COL(10),A,X(1),A);                                           /*                                          */
345           IF MART>='1' & MART<='3' THEN                                 /* MARITAL STATUS                           */
346           DO; I=MART;                                   /* CONVERSION */  /*                                          */
347              PUT EDIT('MARITAL STATUS:',MK(I))(COL(10),A,X(1),A);       /*                                          */
348           END;                                                          /*                                          */
349           IF DOBC='1' THEN PUT EDIT('DATE OF BIRTH CERTIFICATE CONFIRMED')  /* DATE OF BIRTH CERTIFICATE            */
350           (COL(10),A);                                                  /*                                          */
351           IF FPOS¬='     ' THEN PUT EDIT('POSITION IN FAMILY: ',SUBSTR(FPOS,  /* POSITION IN FAMILY                  */
352           1,2),'/',SUBSTR(FPOS,3,2))(COL(10),4(A));                     /*                                          */
353           IF SET>='1' & ((SET<='4' & OKB) | (SET<='6' & ¬OKB)) THEN     /* SETTLEMENT                               */
354           DO; I=SET;                                    /* CONVERSION */  /*                                          */
355              PUT EDIT('SETTLEMENT: ')(COL(10),A);                       /*                                          */
```

Dataset Limited

```
356             IF OKB THEN PUT EDIT(OAK(I))(A); ELSE PUT EDIT(STK(I))(A);          /*                                          */
357         END;                                                                    /*                                          */
358         I=INDEX(GRC,REG);                                                       /* GEOGRAPHICAL REGION                      */
359         IF REG>='1' & REG<='11' & ¬OKB THEN                                     /*                                          */
360         PUT EDIT('REGION: ',GRK(I))(COL(10),A,A); ELSE                          /*                                          */
361         IF OKB THEN PUT EDIT('REGION: NOT CODED PRIOR TO 1972')                 /*                                          */
362         (COL(10),A);                                                            /*                                          */
363         IF SOCC>='1' & ((SOCC<='6' & OKB) | (SOCC<='5' & ¬OKB)) THEN            /* STUDENT OCCUPATION                       */
364         DO; I=SOCC;                                          /* CONVERSION */   /*                                          */
365             PUT EDIT('STUDENT OCCUPATION: ')(COL(10),A);                        /*                                          */
366             IF OKB THEN PUT EDIT(OPSOK(I))(A); ELSE PUT EDIT(PSOK(I))(A);       /*                                          */
367         END; ELSE                                                               /*                                          */
368         IF SOCC=' ' THEN PUT EDIT('STUDENT OCCUPATION: AT SCHOOL')(             /*                                          */
369         COL(10),A);                                                             /*                                          */
370         IF POCC>='1' & ((POCC<='6' & OKB) | (POCC<='5' & ¬OKB)) THEN            /* PARENT/GUARDIAN OCCUPATION               */
371         DO; I=POCC;                                          /* CONVERSION */   /*                                          */
372             PUT EDIT('PARENT OCCUPATION: ')(COL(10),A);                         /*                                          */
373             IF OKB THEN PUT EDIT(OPSOK(I))(A); ELSE PUT EDIT(PSOK(I))(A);       /*                                          */
374         END;                                                                    /*                                          */
375         IF REL>='1' & ((REL<='6' & OKB) | (REL<='5' & ¬OKB)) THEN               /* RELIGION                                 */
376         DO; I=REL;                                           /* CONVERSION */   /*                                          */
377             PUT EDIT('RELIGION: ')(COL(10),A);                                  /*                                          */
378             IF OKB THEN PUT EDIT(ORK(I))(A); ELSE PUT EDIT(RK(I))(A);           /*                                          */
379         END;                                                                    /*                                          */
380         IF CONF='Y' THEN PUT EDIT('CONFIRMED WHEN ADMITTED')(COL(10),A);        /* CONFIRMATION                             */
381         ELSE PUT EDIT('NOT CONFIRMED WHEN ADMITTED')(COL(10),A);                /*                                          */
382         IF OV25=' ' THEN PUT EDIT('OVER 25 QUESTION: STUDENT UNDER 25 ',        /* OVER 25 OCCUPATION                       */
383         'WHEN ADMITTED')(COL(10),A,A); ELSE IF OV25>='0' & OV25<='9' THEN       /*                                          */
384         DO; I=OV25;  IF I=0 THEN I=10;                       /* CONVERSION */   /*                                          */
385             PUT EDIT('OVER 25 QUESTION: ',OK(I))(COL(10),2(A));                 /*                                          */
386         END;                                                                    /*                                          */
387         IF AUTH¬='  ' THEN PUT EDIT('AUTHORITY/COUNTRY: ',AUTH,'AUTHOR',        /* AUTHORITY/COUNTRY NAME & NO.             */
388         'ITY CODE: ',AUTHNO)(COL(10),A,A,COL(10),A,A,A);                        /*                                          */
389     END;                                                                        /*                                          */
390         /*****************************************************************/     /*****************************************/
391     COUPNT:PROCEDURE;                                                           /* COURSE DETAILS PRINT ROUTINE             */
392         PUT EDIT('LENGTH OF COURSE: ',LEN)(COL(10),A,A);                        /*****************************************/
393         IF CODES(1)¬='   ' THEN PUT EDIT('MAIN SUBJECT CODE: ',CODES(1))        /* LENGTH, MAIN SUBJECT CODE                */
394         (X(3),A,A);                                                             /*                                          */
395         IF CODES(2)¬='   ' THEN PUT EDIT('2ND. PROV CODE: ',CODES(2))           /* PROVISIONAL SUBJECT CODE                 */
396         (X(3),A,A);                                                             /*                                          */
397         PUT EDIT('DETAILS OF COURSE:  24 25   26 27 29 30   32   34 35 ',       /* COURSE DETAILS                           */
398         '36 37 38 41 42   43 44 45   46-48 49-51   59   64')(COL(10),A,A);      /* -> TITLES                                */
399         K=LE;                                                /* CONVERSION */   /*                                          */
400         DO I=1 TO K;                                                            /* -> OUTPUT DETAILS                        */
401            PUT EDIT(DC(I,24),DC(I,25),DC1(I,26),'/',DC1(I,27),DC1(I,29),        /*                                          */
402            '/',DC1(I,30),DC(I,32),DC(I,34),DC(I,35),DC(I,36),DC(I,37),          /*                                          */
403            DC(I,38),DC(I,41),DC(I,42),DC(I,43),DC(I,44),DC(I,45),DC46(I),       /*                                          */
404            DC49(I),DC(I,59),DC(I,64))(COL(31),A,X(2),A,X(3),3(A),X(1),3(A)      /*                                          */
405            ,X(4),A,X(2),7(X(2),A),X(2),3(X(2),A),X(5),A,X(3),A,2(X(4),A));      /*                                          */
406         END;                                                                    /*                                          */
407     END;                                                                        /*                                          */
408         /*****************************************************************/     /*****************************************/
409     SAQPNT:PROCEDURE;                                                           /* PREVIOUS EDUCATION PRINT ROUTINE         */
410         PUT EDIT('NO. OF SCHOOLS ATTENDED: ',SCHNO,'  CODED: ')(COL(10),        /*****************************************/
411         3(A));                                                                  /* NO. OF SCHOOLS ATTENDED                  */
412         K=SCHNO;                                             /* CONVERSION */   /*                                          */
413         DO I=1 TO K;                                                            /*                                          */
414            PUT EDIT(SCH(I))(X(1),A);                                            /* -> SCHOOL ATTENDANCES                    */
415         END;                                                                    /*                                          */
```

Dataset Limited

```
416          PUT EDIT('NO. OF QUALIFICATIONS RECORDED: ',QUALNO)                    /* NO. OF QUALIFICATIONS RECORDED          */
417          (SKIP(2),COL(10),A,A);                                                 /*                                         */
418          K=QUALNO;                                          /* CONVERSION */    /*                                         */
419          PUT EDIT('QUALIFICATIONS: ')(COL(10),A);                               /*                                         */
420          DO I=1 TO K;                                                           /*                                         */
421             PUT EDIT(QUAL(I))(X(4),A(10));                                      /* -> QUALIFICATIONS RECORDED              */
422             IF MOD(I,6)=0 THEN PUT EDIT('  ')(COL(24),A);                       /* -> PRINTED 6 PER LINE                   */
423          END;                                                                   /*                                         */
424          IF UNCOL(1)¬='    ' | UNCOL(2)¬='    ' THEN                            /* UNIV. OR COLL. ATTENDED                 */
425          DO; PUT EDIT('UNIV. OR COLL. ATTENDED FOR DEGREE STUDY','F/P TIME'     /* -> TITLES                               */
426              ,'FROM','TO')(SKIP(2),COL(10),A,X(4),A,X(8),A,X(10),A);            /*                                         */
427              PUT EDIT(UNCOL(1),FPT(1),CDATE(FT(1,1)),CDATE(FT(1,2)),            /* -> UNIV. OR COLL.                       */
428              UNCOL(2),FPT(2),CDATE(FT(2,1)),CDATE(FT(2,2)))(2(COL(10),          /*                                         */
429              A,COL(58),A,COL(66),A,COL(78),A));                                 /*                                         */
430          END;                                                                   /*                                         */
431          IF UNOTH(1)¬='    ' | UNOTH(2)¬='    ' THEN                            /* UNIV. OR OTHER AWARDS                   */
432          DO; PUT EDIT('UNIV. OR OTHER AWARDING BODY','TITLE','F/P,C/D',         /* -> TITLES                               */
433              'SUBJECTS','DATE')(SKIP(2),COL(10),A,COL(43),A,COL(50),A,          /*                                         */
434              COL(59),A,COL(103),A);                                             /*                                         */
435              PUT EDIT(SUBSTR(UNOTH(1),2),DEG(1),HPCD(1),SUB(1),                 /* -> UNIV. OR OTHER                       */
436              CDATE(DATT(1)),SUBSTR(UNOTH(2),2),DEG(2),HPCD(2),                  /*                                         */
437              SUB(2),CDATE(DATT(2)))(2(COL(10),A,COL(45),A,                      /*                                         */
438              COL(53),A,COL(59),A,COL(99),A));                                   /*                                         */
439          END;                                                                   /*                                         */
440          IF ADIPC¬=' ' THEN PUT EDIT('DIPLOMA IN ART AND DESIGN',ADIP,          /* ART DIPLOMA                             */
441          ADIPC)(SKIP(2),COL(10),A,X(1),A,X(1),A);                               /*                                         */
442          IF DDIPC¬=' ' THEN PUT EDIT('NATIONAL DIPLOMA IN DESIGN',DDIPC)        /* NATIONAL DIPLOMA                        */
443          (SKIP(2),COL(10),A,X(1),A);                                            /*                                         */
444          IF OQ(1)¬='  ' | OQ(2)¬='  ' THEN                                      /* OTHER ART QUALIFICATIONS                */
445          DO; PUT EDIT('OTHER ART QUALIFICATIONS: ')(SKIP(2),COL(10),A);         /* -> TITLES                               */
446              PUT EDIT(OQ(1),OQC(1),OQ(2),OQC(2))(2(COL(36),A,X(1),A));          /* -> QUALIFICATIONS                       */
447          END;                                                                   /*                                         */
448          IF ATO¬=' ' THEN PUT EDIT('STUDENT EXCEPTED BY A.T.O. FROM',           /* A.T.O. EXCEPTED                         */
449          ' EXAMINATION REQUIREMENTS')(SKIP(2),COL(10),A,A);                     /*                                         */
450      END;                                                                       /*                                         */
451          /*************************************************************/        /*****************************************/
452      APPNT:PROCEDURE;                                                           /* ANNUAL PROGRESS PRINT ROUTINE           */
453          PUT EDIT('ANNUAL PROGRESS VECTOR:  ')(COL(10),A);                      /*****************************************/
454          K=35;                                                                  /* SET COLUMN COUNTER                      */
455          DO I=1 TO 100;                                                         /*                                         */
456             IF FK(I,1)¬=0 THEN                                                  /* TEST FIELD KEYS                         */
457             DO; PUT EDIT('*A',I,'=',SUBSTR(AP,FK(I,1),FK(I,2)))                 /* -> OUTPUT ENTRY                         */
458                 (COL(K),A,P'999',A,A);                                          /*                                         */
459                 IF K+FK(I,2)+5>=95 | K=95 THEN K=35; ELSE                       /* -> RESET COLUMN COUNTER                 */
460                 IF FK(I,2)+5>=15 THEN K=K+30; ELSE K=K+15;                      /*                                         */
461             END;                                                                /*                                         */
462          END;                                                                   /*                                         */
463          IF TTFT(2,1)¬='       ' & SUBSTR(TTFT(2,1),1,4)¬='WITH' THEN            /* OUTPUT FINAL RESULTS                    */
464          DO; PUT EDIT('FINAL RESULT:','TEACHER''S CERTIFICATE',TC,              /* -> TEACHERS' CERTIFICATE                */
465              'WING SCIENCE',WS,'B.ED.',BE)(COL(10),A,3(COL(25),A,               /* -> WING SCIENCE, B.ED.                  */
466              COL(50),A));                                                       /*                                         */
467              PUT EDIT('SUBSEQUENT CAREER:',SUBCAR)(COL(10),A,X(2),A);           /* -> SUBSEQUENT CAREER                    */
468          END;                                                                   /*                                         */
469      END;                                                                       /*                                         */
470          /*************************************************************/        /*****************************************/
471          IF VPI=3 THEN PUT PAGE;                                                /* NEW PAGE IF FULL PRINT                  */
472          PUT EDIT(DESNO,NAME)(COL(1),A,X(1),A);                                 /* OUTPUT DES.NO. & NAME                   */
473          IF VPI=0 THEN                                                          /* WITHDRAWAL VERIFICATION                 */
474          DO; PUT EDIT('RECORDED AS HAVING WITHDRAWN ON',CDATE(TTFT(1,2)),       /* -> DATE OF WITHDRAWAL                   */
475              'REASON CODED AS',SUBSTR(TTFT(2,1),6,1))(COL(10),A,                /* -> REASON FOR WITHDRAWAL                */
```

Dataset Limited

```
476             3(X(1),A));                                                  /*                                          */
477             RETURN;                                                      /*                                          */
478         END; ELSE                                                        /*                                          */
479         IF VPI=1 THEN                                                    /* CHANGE COURSE/SUBJECT VERIFICATION       */
480         DO; PUT EDIT('RECORDED AS HAVING CHANGED ')(COL(10),A);          /* -> TITLES                                */
481             IF CSC='C' THEN PUT EDIT('COURSE')(A); ELSE                  /* -> COURSE                                */
482             PUT EDIT('SUBJECT')(A);                                      /* -> SUBJECT                               */
483             CALL COUPNT;                                                 /* -> CALL COURSE DETAILS PRINT ROUTINE     */
484             RETURN;                                                      /*                                          */
485         END; ELSE                                                        /*                                          */
486         IF VPI=2 THEN                                                    /* ANNUAL UPDATE VERIFICATION               */
487         DO; PUT EDIT('RECORD UPDATED, ANNUAL PROGRESS VECTOR OR ',       /* -> TITLES                                */
488             'FINAL RESULT')(COL(10),A,A);                                /*                                          */
489             CALL APPNT;                                                  /* -> CALL ANNUAL PROGRESS PRINT ROUTINE    */
490             RETURN;                                                      /*                                          */
491         END; ELSE                                                        /*                                          */
492         IF VPI=3 THEN                                                    /* FULL PRINT                               */
493         DO; PUT SKIP(2);                                                 /*                                          */
494             CALL GENPNT;                                                 /* -> GENERAL INFORMATION PRINT ROUTINE     */
495             PUT SKIP(2);                                                 /*                                          */
496             CALL SAQPNT;                                                 /* -> PREVIOUS EDUCATION PRINT ROUTINE      */
497             PUT SKIP(2);                                                 /*                                          */
498             CALL COUPNT;                                                 /* -> COURSE DETAILS PRINT ROUTINE          */
499             PUT SKIP(2);                                                 /*                                          */
500             CALL APPNT;                                                  /* -> ANNUAL PROGRESS PRINT ROUTINE         */
501         END;                                                             /*                                          */
502     END;                                                                 /*                                          */
503     /*****************************************************************/  /*******************************************/
504     SPNT1:PROCEDURE;                                                     /* FILE OUTPUT ROUTINE 2                    */
505         DCL (I,N) FIXED BINARY, COD CHAR(3), B BIT(1);                   /*******************************************/
506         N=KT;                                                            /* BUBBLE SORT ROUTINE                      */
507     L1: B='0'B;                                                          /*                                          */
508         N=N-1;                                                           /*                                          */
509         DO I=1 TO N;                                                     /*                                          */
510            IF DHT(I) || TYPE(I) || NAML(I)>DHT(I+1) || TYPE(I+1) ||      /*                                          */
511            NAML(I+1) THEN                                                /*                                          */
512            DO; RECL(KT+1)=RECL(I);                                       /*                                          */
513                RECL(I)=RECL(I+1);                                        /*                                          */
514                RECL(I+1)=RECL(KT+1);                                     /*                                          */
515                B='1'B;                                                   /*                                          */
516            END;                                                          /*                                          */
517         END;                                                             /*                                          */
518         IF B THEN GOTO L1;                                               /*                                          */
519         COD=DHT(1); B='0'B;                                              /* RECORD FIRST COURSE CODE                 */
520         IF COD='   ' THEN                                                /* -> OUTPUT TITLES                         */
521         DO; PUT EDIT('NO MAIN COURSE RECORDED','DATE OF LISTING',DT,     /*                                          */
522             'TYPE','D.E.S.NO','NAME')(PAGE,COL(1),A,SKIP(3),COL(1),A,X(1),  /*                                       */
523             A,SKIP(3),COL(1),A,COL(7),A,COL(17),A);                      /*                                          */
524             PUT SKIP;                                                    /*                                          */
525             B='1'B;                                                      /* -> SET BIT SWITCH                        */
526         END;                                                             /*                                          */
527         DO I=1 TO KT;                                                    /* OUTPUT RECORDS                           */
528     L2:    IF ¬B THEN                                                    /* -> OUTPUT NEW TITLES                     */
529            DO; PUT EDIT('COURSE CODE',COD,'DATE OF LISTING',DT,'TYPE',   /*                                          */
530                'D.E.S.NO','NAME')(PAGE,COL(1),A,X(1),A,SKIP(3),COL(1),A,  /*                                          */
531                X(1),A,SKIP(3),COL(1),A,COL(7),A,COL(17),A);              /*                                          */
532                PUT SKIP;                                                 /*                                          */
533                B='1'B;                                                   /*                                          */
534            END;                                                          /*                                          */
535            IF DHT(I)¬=COD THEN                                           /* -> TEST COURSE                           */
```

Dataset Limited

```
536            DO; COD=DHT(I);                                          /*    -> RECORD CHANGE OF COURSE                  */
537                B='0'B;                                              /*    -> RESET BIT SWITCH                         */
538                GOTO L2;                                             /*    -> BRANCH FOR NEW TITLES                    */
539            END;                                                     /*                                                */
540            IF TYPE(I)='P' THEN PUT EDIT('PGCE')(SKIP,COL(1),A); ELSE /* -> OUTPUT TYPE OF STUDENT                   */
541            PUT EDIT(TYPE(I))(SKIP,COL(4),A);                        /*                                                */
542            PUT EDIT(DESL(I),NAML(I))(COL(7),A,COL(17),A);           /* -> OUTPUT DES.NO. & NAME                       */
543         END;                                                        /*                                                */
544      END;                                                           /*                                                */
545      /*****************************************************************/ /*****************************************/
546      DESPNT:PROCEDURE;                                              /* D.E.S. OUTPUT LISTING ROUTINE                  */
547         DCL X2 CHAR(2), (I,N) FIXED BINARY, SW BIT(1);              /*****************************************/
548         ON ENDPAGE(SPRINT)                                          /* SET ENDPAGE ROUTINE                            */
549         BEGIN; PUT EDIT('.....BEDE COLLEGE  DURHAM',YE,'.....','TYPE',  /* -> OUTPUT TITLES                            */
550                'D.E.S.NO','NAME','DATE OF BIRTH','COURSE___DATES',  /*                                                */
551                'DEGREES HELD')(PAGE,COL(38),A,X(2),A,A,SKIP(2),COL(1),A,  /*                                          */
552                COL(7),A,COL(17),A,COL(56),A,COL(71),A,COL(88),A);   /*                                                */
553                PUT SKIP(2);                                         /*                                                */
554         END;                                                        /* END ENDPAGE ROUTINE                            */
555         IF KT=0 THEN GOTO L21;                                      /* BRANCH IF KT=0                                 */
556         N=KT;                                                       /* BUBBLE SORT ROUTINE                            */
557      L20: SW='0'B;                                                  /*                                                */
558         N=N-1;                                                      /*                                                */
559         DO I=1 TO N;                                                /*                                                */
560            IF TYPE(I) || NAML(I)>TYPE(I+1) || NAML(I+1) THEN        /*                                                */
561            DO; RECL(KT+1)=RECL(I);                                  /*                                                */
562                RECL(I)=RECL(I+1);                                   /*                                                */
563                RECL(I+1)=RECL(KT+1);                                /*                                                */
564                SW='1'B;                                             /*                                                */
565            END;                                                     /*                                                */
566         END;                                                        /*                                                */
567         IF SW THEN GOTO L20;                                        /*                                                */
568         N=0;                                                        /* FIND NUMBER OF PGCE STUDENTS                   */
569         DO I=1 TO KT;                                               /*                                                */
570            IF TYPE(I)='P' THEN                                      /* -> LOCATE STUDENT                              */
571            DO; N=N+1;                                               /* -> UPDATE COUNTER                              */
572                IF SUBSTR(DHB(I),1,1)='F' THEN SUBSTR(DHB(I),1,1)=' ';  /* -> RESET FOREIGN DEGREE STUDENTS            */
573            END; ELSE GOTO L2;                                       /*                                                */
574         END;                                                        /*                                                */
575      L2: SW='0'B;                                                   /* BUBBLE SORT ROUTINE OF PGCE STUDENTS           */
576         N=N-1;                                                      /*                                                */
577         DO I=1 TO N;                                                /*                                                */
578            IF SUBSTR(DHB(I),1,1) || NAML(I) > SUBSTR(DHB(I+1),1,1) ||  /*                                          */
579            NAML(I+1) THEN                                           /*                                                */
580            DO; RECL(KT+1)=RECL(I);                                  /*                                                */
581                RECL(I)=RECL(I+1);                                   /*                                                */
582                RECL(I+1)=RECL(KT+1);                                /*                                                */
583                SW='1'B;                                             /*                                                */
584            END;                                                     /*                                                */
585         END;                                                        /*                                                */
586         IF SW THEN GOTO L2;                                         /*                                                */
587         X2=TYPE(1) || SUBSTR(DHB(1),1,1);                           /* RECORD FIRST TYPE AND DEGREE                   */
588         IF X2='P ' THEN                                             /* LOCATE FOREIGN DEGREE OR OTHER QUAL.           */
589         DO; SIGNAL ENDPAGE(SPRINT);                                 /* -> OUTPUT TITLES                               */
590             PUT EDIT('PGCE STUDENT(S): FOREIGN DEGREES OR EQUIVALENT')  /*                                           */
591             (SKIP(2),COL(1),A);                                     /*                                                */
592             PUT SKIP(2);                                            /*                                                */
593         END; ELSE SIGNAL ENDPAGE(SPRINT);                           /* OUTPUT TITLES                                  */
594         DO I=1 TO KT;                                               /* OUTPUT RECORDS                                 */
595            IF TYPE(I) || SUBSTR(DHB(I),1,1)¬=X2 THEN                /* -> TEST TYPE AND DEGREE                        */
```

Dataset Limited

```
596            DO; SIGNAL ENDPAGE(SPRINT);                                  /*    -> OUTPUT TITLES                          */
597                X2=TYPE(I) || SUBSTR(DHB(I),1,1);                       /*    -> RECORD NEW TYPE AND DEGREE             */
598            END;                                                         /*                                              */
599            IF TYPE(I)='P' THEN PUT EDIT('PGCE')(SKIP,COL(1),A); ELSE    /* -> OUTPUT TYPE                               */
600            PUT EDIT(TYPE(I))(SKIP,COL(4),A);                            /*                                              */
601            PUT EDIT(DESL(I),NAML(I),SUBSTR(DOBL(I),1,2),'/',SUBSTR(     /* -> CUTPUT DES.NO., NAME, DCB, AND            */
602            DOBL(I),3,2),'/',SUBSTR(DOBL(I),5,2),START(I),'TO',FIN(I))   /* -> OUTPUT FROM-TO DATES                      */
603            (COL(7),A,COL(17),A,COL(61),5(A),COL(71),A(5),X(1),A,X(1),A(5)  /*                                          */
604            );                                                           /*                                              */
605            IF DHB(I)¬='    ' THEN                                       /* -> TEST FOR DEGREE                           */
606            PUT EDIT(DHT(I),'19' || DHY(I),SUBSTR(DHB(I),2))             /*    -> CUTPUT DEGREE INFORMATION              */
607            (COL(88),A,2(X(2),A));                                       /*                                              */
608        END;                                                             /*                                              */
609    L21: PUT EDIT('NO. OF STUDENTS',KT)(SKIP(2),COL(1),A,X(2),F(3));     /* OUTPUT SAMPLE SIZE 'KT'                      */
610        ON ENDPAGE(SPRINT) SYSTEM;                                       /* RESET ENDPAGE ROUTINE                        */
611    END;                                                                 /*                                              */
612    /*****************************************************************/   /*******************************************/
613        ON CONVERSION                                                    /* SET DATE CONVERSION ERROR ROUTINE            */
614        BEGIN; PUT EDIT('********DATA CONVERSION ERROR********',          /* -> OUTPUT MESSAGE                            */
615                'CURRENT D.E.S.NO.',DESNO)(SKIP(2),A,X(1),A,X(1),A);     /* -> OUTPUT CURRENT DES. NO.                   */
616        END;                                                             /* END CONVERSICN ROUTINE                       */
617        ON ENDFILE(SCARDS) GOTO L09;                                     /* BRANCH ON ENDFILE (SCARDS)                   */
618    L39: GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));      /* INPUT OPTION PARAMETER CARD                  */
619        GET LIST(OPT);                                                   /* INPUT OPTION                                 */
620        IF OPT<1 | OPT>10 THEN                                           /* TEST OPTION                                  */
621        DO; PUT EDIT('OPTION INVALID.....PROGRAM TERMINATED')            /* -> CLASS A EPRCR/WARNING                     */
622            (SKIP,COL(1),A);                                             /*                                              */
623            GOTO L09;                                                    /* -> BRANCH TO PROGRAMME TERMINATION           */
624        END;                                                             /*                                              */
625        TRANS='00'; B='0'B; PI='C'; YT='0'; NTT='0'; VP='0';             /* INITIALISE OPTION PARAMETERS                 */
626        DES='        '; BMAX='0'B;                                       /*                                              */
627        GET DATA;                                                        /* INPUT CPTION PARAMETERS                      */
628        IF DES¬='        ' THEN                                          /* TEST FOR D.E.S. OUTPUT                       */
629        DO; DESB='1'B;                                                   /* -> SET D.E.S. OUTPUT BIT                     */
630            IF ¬DESOP THEN                                               /* -> TEST TRANSACTICN FILE STATUS              */
631            DO; OPEN FILE(DIC) OUTPUT TITLE(DES);                        /* -> OPEN TRANSACTION FILE                     */
632                DESOP='1'B;                                              /*    -> SET STATUS BIT                         */
633            END;                                                         /*                                              */
634        END; ELSE DESB='0'B;                                             /*                                              */
635        IF VP='1' THEN VPB='1'B; ELSE VPB='0'B;                          /* VERIFY OPTICN PARAMETERS                     */
636        IF NTT='1' THEN NT=1; ELSE IF NTT='2' THEN NT=2; ELSE NT=1;      /* PARAMETER NTT                                */
637        IF YT='F' THEN YR=1; ELSE IF YT='L' THEN YR=2; ELSE YR=1;        /* PARAMETER YT                                 */
638        PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME R',     /* OUTPUT TITLES                                */
639        'OPTION REQUESTED',OPT,'YEAR REQUESTED',YE)(PAGE,COL(1),A,       /*                                              */
640        SKIP(2),COL(1),A,COL(1),A,COL(32),F(2),COL(1),A,COL(30),A);      /*                                              */
641        PUT SKIP(2);                                                     /*                                              */
642        IF DES¬='        ' THEN PUT EDIT('DES O/P REQUIRED')(COL(1),A);  /* OUTPUT OPTICN PARAMETERS IN EFFECT, DES      */
643        IF PI¬='0' THEN PUT EDIT('PRINT INDICATOR',PI)                   /* -> PI                                        */
644        (COL(1),A,COL(33),A);                                            /*                                              */
645        IF YT¬='0' THEN PUT EDIT('YEAR OF TEST',YT)(COL(1),A,COL(33),A); /* -> YT                                        */
646        IF NTT¬='0' THEN PUT EDIT('NORM TABLE REQUIRED',NTT)             /* -> NTT                                       */
647        (COL(1),A,COL(33),A(1));                                         /*                                              */
648        IF VP¬='0' THEN PUT EDIT('VERIFICATICN PRINT REQUIRED')(COL(1),A);  /* -> VP                                      */
649        PUT SKIP(4);                                                     /*                                              */
650        IF BCON & (OPT=2 | (OPT=6 & PI='1') | (OPT=9 & VPB)) THEN        /* -> TEST FCR PROCEDURE CON                    */
651        DO; CALL CON(FILN2);                                             /* -> CALL CON (INPUT OF DATA CONSTANTS)        */
652            BCON='0'B;                                                   /* -> SET CON BIT                               */
653        END;                                                             /*                                              */
654        GOTO L(OPT);                                                     /* BRANCH TO REQUIRED OPTION                    */
655    /*-----------------------------------------------------------------*/ /*-----------------------------------------*/
```

Dataset Limited

```
656    L1:L7:OPEN FILE(STUDENT) OUTPUT TITLE(FILN1);                          /* OPTIONS 1 AND 7                          */
657          ALLOCATE RECORD;                                                 /* ALLOCATE STORAGE FOR RECORD              */
658    L11: GENERAL=' '; EDUCATION=' '; COURSE=' '; AP=' '; INTEL=0;          /* INITIALISE RECORD VARIABLES              */
659          INTELB='  '; LEAP=1; FK=0;                                       /*                                          */
660          GET EDIT(DESNO)(COL(1),A(8));                                    /* INPUT DES.NO.                            */
661          IF DESNO='********' THEN                                         /* TEST FOR INPUT TERMINATOR                */
662          DO; CLOSE FILE(STUDENT);                                         /* -> CLOSE FILE                            */
663              FREE RECORD;                                                 /* -> FREE STORAGE                          */
664              GOTO L39;                                                    /* -> BRANCH TO PROGRAMME CONTROL SECTION   */
665          END;                                                             /*                                          */
666          CALL RECIN;                                                      /* CALL RECORD INPUT                        */
667          CALL CHECKREC;                                                   /* CALL RECORD CHECKING ROUTINE             */
668          IF OPT=7 & DESB THEN PUT FILE(DIC) EDIT(DESNO || '15')(COL(1),A);/* OUTPUT TO TRANSACTION FILE               */
669          WRITE FILE(STUDENT) FROM(RECORD);                                /* WRITE RECORD TO FILE                     */
670          GOTO L11;                                                        /* BRANCH FOR NEXT RECORD                   */
671    /*-----------------------------------------------------------------*/  /*------------------------------------------*/
672    L2:L3:L4:L8:L9:OPEN FILE(STUDENT) UPDATE TITLE(FILN1);                 /* OPTIONS 2,3,4,8 AND 9                    */
673          ON ENDFILE(STUDENT)                                              /* SET ENDFILE (STUDENT) ROUTINE            */
674          BEGIN; CLOSE FILE(STUDENT);                                      /* -> CLOSE FILE                            */
675                 OPEN FILE(STUDENT) UPDATE TITLE(FILN1);                   /* -> OPEN FILE                             */
676                 GOTO L41;                                                 /* -> BRANCH FOR FIRST RECORD               */
677          END;                                                             /* END ENDFILE ROUTINE                      */
678    L42: GET EDIT(DESNO2)(COL(1),A(8));                                    /* INPUT REQUIRED DES.NO.                   */
679          IF DESNO2='********' THEN                                        /* TEST FOR INPUT TERMINATOR                */
680          DO; CLOSE FILE(STUDENT);                                         /* -> CLOSE FILE                            */
681              GOTO L39;                                                    /* -> BRANCH TO PROGRAMME CONTROL SECTION   */
682          END;                                                             /*                                          */
683    L41: READ FILE(STUDENT) SET(PTR);                                      /* READ FIRST/NEXT RECORD                   */
684          IF DESNO=DESNO2 THEN                                             /* TEST RECORD DES.NO. WITH REQUIRED DES.NO */
685          DO; IF B THEN B='0'B;                                            /* RECORD 'FOUND' ROUTINE, RESET BIT        */
686              IF OPT=2 THEN CALL ITRECIN; ELSE                             /* -> OPTION 2, INTEL.TEST INPUT            */
687              IF OPT=3 THEN                                                /* -> OPTION 3                              */
688              DO; CALL UPDATE;                                             /* -> CALL ANNUAL UPDATE ROUTINE            */
689                  IF VPB THEN                                              /* -> TEST FOR VERIFICATION PRINT           */
690                  DO; VPI=2;                                               /*    -> SET VERIFICATION PRINT INDICATOR   */
691                      CALL PRINT;                                          /*    -> CALL FULL PRINT ROUTINE            */
692                  END;                                                     /*                                          */
693                  IF DESB & SUBSTR(TTFT(2,1),1,4)='LEFT' THEN              /* -> TEST FOR DES. OUTPUT                  */
694                  PUT FILE(DIC) EDIT(DESNO || '99')(COL(1),A);             /*   -> OUTPUT TO TRANSACTION FILE          */
695              END; ELSE                                                    /*                                          */
696              IF OPT=4 THEN                                                /* -> OPTION 4                              */
697              DO; GET EDIT(TTFT(1,2),WC)(COL(10),A(6),X(1),A(1));          /* -> INPUT DATE & REASON FOR WITHDRAWAL    */
698                  TTFT(2,1)='WITH ' || WC;                                 /* -> RECORD STUDENT WITHDRAWAL             */
699                  IF VPB THEN                                              /* -> TEST FOR VERIFICATION PRINT           */
700                  DO; VPI=0;                                               /*    -> SET VERIFICATION PRINT INDICATOR   */
701                      CALL PRINT;                                          /*    -> CALL FULL PRINT ROUTINE            */
702                  END;                                                     /*                                          */
703                  IF DESB THEN PUT FILE(DIC) EDIT(DESNO || '12')(COL(1),A);/* -> OUTPUT TO TRANSACTION FILE            */
704              END; ELSE                                                    /*                                          */
705              IF OPT=8 THEN                                                /* -> OPTION 8                              */
706              DO; I=LE;                                   /* CONVERSION */ /* -> SET I=LATEST ENTRY                    */
707                  IF LE¬='4' THEN LE=SUBSTR('1234',I+1,1);                 /* -> RESET LATEST ENTRY                    */
708                  I=LE;                                   /* CONVERSION */ /* -> SET I=NEW LATEST ENTRY                */
709                  GET EDIT(CSC,DC(I,24),DC(I,25),DC1(I,26),DC1(I,27),      /* -> INPUT NEW DETAILS OF COURSE           */
710                  DC1(I,29),DC1(I,30),DC(I,32),DC(I,34),DC(I,35),DC(I,36), /*                                          */
711                  DC(I,37),DC(I,38),DC(I,41),DC(I,42),DC(I,43),DC(I,44),   /*                                          */
712                  DC(I,45),DC46(I),DC49(I),DC(I,59),DC(I,64))(COL(10),A(1), /*                                          */
713                  COL(52),2(A(1)),4(A(2)),11(A(1)),2(A(3)),2(A(1)));       /*                                          */
714                  CODES(1)=DC46(I);                                        /* -> RECORD NEW MAIN SUBJECT               */
715                  CODES(2)=DC49(I);                                        /* -> RECORD NEW PROVISIONAL SUBJECT        */
```

Dataset Limited

```
716                 IF VPB THEN                                          /* -> TEST FOR VERIFICATION PRINT              */
717                 DO; VPI=1;                                           /*    -> SET VERIFICATION PRINT INDICATOR      */
718                     CALL PRINT;                                      /*    -> CALL FULL PRINT ROUTINE               */
719                 END;                                                 /*                                             */
720                 IF DESB THEN                                         /* -> TEST FOR DES. OUTPUT                     */
721                 PUT FILE(DIC) EDIT(DESNO || '11')(COL(1),A);         /*    -> OUTPUT TO TRANSACTION FILE            */
722             END; ELSE                                                /*                                             */
723             IF OPT=9 THEN                                            /* -> OPTION 9                                 */
724             DO; GET DATA;                                            /* -> INPUT SPECIFIC ALTERATION(S)             */
725                 IF VPB THEN                                          /* -> TEST FOR VERIFICATION PRINT              */
726                 DO; VPI=3;                                           /*    -> SET VERIFICATION PRINT INDICATOR      */
727                     CALL PRINT;                                      /*    -> CALL FULL PRINT ROUTINE               */
728                 END;                                                 /*                                             */
729             END;                                                     /*                                             */
730             REWRITE FILE(STUDENT) FROM(RECORD);                      /* -> REWRITE RECORD TO FILE                   */
731             GOTO L42;                                                /* -> BRANCH FOR NEXT REQUIRED DES.NO.         */
732         END; ELSE                                                    /* END RECORD FOUND ROUTINE                    */
733         DO; IF ¬B THEN                                               /* RECORD 'NOT FOUND' ROUTINE                  */
734             DO; DESNO3=DESNO;                                        /* -> RECORD CURRENT DES.NO.                   */
735                 B='1'B;                                              /* -> SET 'RECORD FOUND' BIT                   */
736             END; ELSE                                                /*                                             */
737             DO; IF DESNO=DESNO3 THEN                                 /* -> TEST FOR COMPLETE FILE CYCLE             */
738                 DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')  /* -> CLASS C ERROR/WARNING                 */
739                     (COL(1),A,X(1),A);                               /*                                             */
740                     B='0'B;                                          /* -> RESET 'RECORD FOUND' BIT                 */
741                     GOTO L42;                                        /* -> BRANCH FOR NEXT REQUIRED DES.NO.         */
742                 END;                                                 /*                                             */
743             END;                                                     /*                                             */
744             GOTO L41;                                                /* -> BRANCH FOR NEXT RECORD                   */
745         END;                                                         /* END RECORD NOT FOUND ROUTINE.               */
746     /*-----------------------------------------------------------*/  /*-------------------------------------------*/
747     L5:L6:L10:OPEN FILE(STUDENT) INPUT TITLE(FILN1);                 /* OPTIONS 5,6 AND 10                          */
748         ON ENDFILE(STUDENT)                                          /* SET ENDFILE (STUDENT) ROUTINE               */
749         BEGIN; DT=DATE;                                              /* -> CALCULATE DATE                           */
750                DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||  /*                                             */
751                SUBSTR(DT,7,2);                                       /*                                             */
752                IF OPT=5 THEN CALL DESPNT; ELSE                       /* -> OPTION 5, CALL DES. LISTING ROUTINE      */
753                IF OPT=6 THEN                                         /* -> OPTION 6                                 */
754                DO; IF PI='2' THEN                                    /* -> PRINT INDICATOR 2                        */
755                    DO; PUT EDIT(YE,'FILE.....',KT,'STUDENTS INCLUDED',  /*    -> OUTPUT TITLES                      */
756                        '.....NUMBERS FOLLOWED BY ''*'' SHOW THAT THE ',  /*    -> OUTPUT SAMPLE SIZE 'KT'            */
757                        'STUDENT HAS WITHDRAWN FROM THE COLLEGE','NUMBERS',  /*                                    */
758                        ' FOLLOWED BY ''+'' SHOW THAT THE STUDENT HAS',  /*                                        */
759                        ' COMPLETED HIS COURSE AND LEFT')             /*                                             */
760                        (PAGE,COL(1),A,X(1),A,F(3),X(1),3(A),         /*                                             */
761                        SKIP,COL(41),3(A));                           /*                                             */
762                        IF KT¬=0 THEN CALL SPNT; ELSE GOTO L38;       /*    -> TEST KT, BRANCH OR OUTPUT ROUTINE 1   */
763                    END; ELSE                                         /*                                             */
764                    IF PI='3' THEN                                    /* -> PRINT INDICATOR 3                        */
765                    DO; PUT EDIT(YE,'FILE.....',KT,'STUDENTS INCLUDED')  /*    -> OUTPUT TITLES & SAMPLE SIZE 'KT'   */
766                        (PAGE,COL(1),A(4),X(1),A,F(3),X(1),A);        /*                                             */
767                        IF KT¬=0 THEN CALL SPNT; ELSE GOTO L38;       /*    -> TEST KT, BRANCH OR OUTPUT ROUTINE 1   */
768                    END; ELSE                                         /*                                             */
769                    IF PI='4' THEN                                    /* -> PRINT INDICATOR 4                        */
770                    DO; PUT EDIT(YE,'FILE.....',KT,'STUDENTS INCLUDED')  /*    -> OUTPUT TITLES & SAMPLE SIZE 'KT'   */
771                        (PAGE,COL(1),A(4),X(1),A,F(3),X(1),A);        /*                                             */
772                        IF KT¬=0 THEN CALL SPNT1;                     /*    -> TEST KT, CALL OUTPUT ROUTINE 2        */
773                    END;                                              /*                                             */
774                END;                                                  /*                                             */
775     L38:       PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(2),A);  /* -> OUTPUT DATE OF LISTING                */
```

Dataset Limited

```
776                 IF BMAX THEN                                        /* -> TEST FILE DIVIDED BIT                    */
777                 DO; BMAX='0'B;                                      /*    -> RESET BIT                             */
778                     PECL=' '; KT=0;                                 /*    -> INITIALISE VARIABLES                  */
779                     GOTO L78;                                       /*    -> BRANCH TO REQUIRED OPTION             */
780                 END;                                                /*                                             */
781                 CLOSE FILE(STUDENT);                                /* -> CLOSE FILE                               */
782                 GOTO L39;                                           /* -> BRANCH TO PROGRAMME CONTROL SECTION      */
783             END;                                                    /* END ENDFILE ROUTINE                         */
784             IF OPT=6 & PI='1' THEN VPI=3;                           /* SET VERIFICATION PRINT INDICATOR, OPT 6     */
785             IF OPT=5 | OPT=6 THEN                                   /* OPTIONS 5 AND 6                             */
786             DO; RECL=' ';                                           /* -> INITIALISE VARIABLES                     */
787                 KT=0;                                               /*                                             */
788             END;                                                    /*                                             */
789     L12:    READ FILE(STUDENT) SET(PTR);                            /* READ FIRST/NEXT RECORD                      */
790             IF OPT=5 | OPT=10 THEN                                  /* OPTIONS 5 AND 10                            */
791             DO; IF SUBSTR(DESNO,1,2)¬=SUBSTR(TTFT(1,1),5,2) THEN    /* -> TEST YEAR OF ADMISSION                   */
792                 DO; PUT EDIT(DESNO,'ADMITTED IN YEAR','19' ||       /*    -> CLASS N ERROR/WARNING                 */
793                     SUBSTR(DESNO,1,2))(SKIP,COL(1),A,X(1),A,X(2),A);/*                                             */
794                     GOTO L12;                                       /*    -> BRANCH FOR NEXT RECORD                */
795                 END;                                                /*                                             */
796             END;                                                    /*                                             */
797     L78:    IF OPT=5 THEN                                           /* OPTION 5                                    */
798             DO; IF KT=499 THEN                                      /* -> TEST FOR MAXIMUM FILE SIZE               */
799                 DO; PUT EDIT('********WARNING******** MAXIMUM SIZE REACHED',/*    -> OUTPUT WARNING                 */
800                     '...FILE DIVIDED')(COL(1),A,A);                 /*                                             */
801                     BMAX='1'B;                                      /*    -> SET FILE DIVIDED BIT                  */
802                     SIGNAL ENDFILE(STUDENT);                        /*    -> SIGNAL ENDFILE (STUDENT) ROUTINE      */
803                 END;                                                /*                                             */
804                 KT=KT+1;                                            /* -> UPDATE COUNTER                           */
805                 DESL(KT)=DESNO;                                     /* -> RECORD DES.NO.                           */
806                 NAML(KT)=NAME;                                      /* -> RECORD NAME                              */
807                 DOBL(KT)=DOB;                                       /* -> RECORD DATE OF BIRTH                     */
808                 START(KT)=DC1(1,26) || '/' || DC1(1,27);            /* -> RECORD START DATE                        */
809                 FIN(KT)=DC1(1,29) || '/' || DC1(1,30);              /* -> RECORD FINISH DATE                       */
810                 IF DC(1,59)='1' THEN                                /* -> LOCATE PGCE STUDENTS                     */
811                 DO; TYPE(KT)='P';                                   /*    -> RECORD TYPE                           */
812                     IF UNOTH(1)¬=' ' THEN                           /*    -> TEST FOR UNIV. OR OTHER BODY          */
813                     DO; DHT(KT)=DEG(1);                             /*    -> RECORD DEGREE                         */
814                         DHY(KT)=SUBSTR(DATT(1),5,2);                /*    -> RECORD YEAR                           */
815                         DHB(KT)=UNOTH(1);                           /*    -> RECORD UNIV. OR OTHER BODY            */
816                     END; ELSE                                       /*                                             */
817                     IF UNCOL(1)¬=' ' THEN                           /*    -> TEST FOR UNIV. OR COLLEGE ATTENDED    */
818                     DO; DHY(KT)=SUBSTR(FT(1,2),5,2);                /*    -> RECORD YEAR LEFT                      */
819                         DHB(KT)=' ' || UNCOL(1);                    /*    -> RECORD UNIV. OR COLLEGE               */
820                     END;                                            /*                                             */
821                 END; ELSE                                           /*                                             */
822                 IF LEN='1' | LEN='2' | LEN='3' THEN TYPE(KT)=LEN; ELSE /* -> RECORD COURSE LENGTH, TYPE             */
823                 DO; PUT EDIT(DESNO,'TYPE WARNING.....SET TO 3')     /*    -> CLASS D ERROR/WARNING                 */
824                     (SKIP,COL(1),A,X(1),A);                         /*                                             */
825                     TYPE(KT)='3';                                   /*    -> TYPE RECORDED AS 3                    */
826                 END;                                                /*                                             */
827             END; ELSE                                               /*                                             */
828             IF OPT=6 THEN                                           /* OPTION 6                                    */
829             DO; IF PI='1' THEN CALL PRINT; ELSE                     /* -> PRINT INDICATOR 1, CALL FULL PRINT       */
830                 IF PI='2' | PI='3' | (PI='4' & TTFT(1,2)='        ') THEN /* -> PI 2,3 OR (4 & STUDENT AT COLLEGE)   */
831                 DO; IF KT=499 THEN                                  /* -> TEST FOR MAXIMUM FILE SIZE               */
832                     DO; PUT EDIT('********WARNING******** MAXIMUM SIZE REAC',/*    -> OUTPUT WARNING                 */
833                         'HED...FILE DIVIDED')(COL(1),A,A);          /*                                             */
834                         BMAX='1'B;                                  /*    -> SET FILE DIVIDED BIT                  */
835                         SIGNAL ENDFILE(STUDENT);                    /*    -> SIGNAL ENDFILE (STUDENT) ROUTINE      */
```

Dataset Limited

```
836                 END;                                                    /*                                          */
837                 KT=KT+1;                                                /* -> UPDATE COUNTER                        */
838                 DESL(KT)=DESNO;                                         /* -> RECORD DES.NO.                        */
839                 IF TTFT(1,2)¬='       ' THEN                            /* -> TEST FINISH DATE                      */
840                 FIN(KT)=TTFT(1,2);                                      /*    -> RECORD FINISH DATE                 */
841                 IF TTFT(2,1)¬='       ' THEN                            /* -> TEST WITHDRAWAL                       */
842                 IF SUBSTR(TTFT(2,1),1,4)='WITH' THEN                    /*    -> RECORD '*' FOR WITHDRAWAL          */
843                 SUBSTR(FIN(KT),7,1)='*'; ELSE SUBSTR(FIN(KT),7,1)='+';  /*    -> RECORD '+' FOR LEFT                */
844                 IF PI='3' THEN                                          /* -> PRINT INDICATOR 3                     */
845                 DO; NAML(KT)=NAME;                                      /*    -> RECORD NAME                        */
846                     DOBL(KT)=DOB;                                       /*    -> RECORD DATE OF BIRTH               */
847                     START(KT)=TTFT(1,1);                                /*    -> RECORD START DATE                  */
848                 END; ELSE                                               /*                                          */
849                 IF PI='4' THEN                                          /* -> PRINT INDICATOR 4                     */
850                 DO; NAML(KT)=NAME;                                      /*    -> RECORD NAME                        */
851                     DHT(KT)=CODES(1);                                   /*    -> RECORD MAIN SUBJECT                */
852                     I=LE;                                /* CONVERSION */ /*    -> SET I=LATEST ENTRY                 */
853                     IF DC(I,59)='1' THEN TYPE(KT)='P'; ELSE             /*    -> LOCATE PGCE STUDENTS               */
854                     IF LEN>='1' & LEN<='4' THEN TYPE(KT)=LEN; ELSE      /*    -> RECORD COURSE LENGTH, TYPE         */
855                     DO; PUT EDIT(DESNO,'TYPE WARNING.....SET TO 3')     /*    -> CLASS D ERROR/WARNING              */
856                         (COL(1),A,X(1),A);                              /*                                          */
857                         TYPE(KT)='3';                                   /*    -> TYPE RECORDED AS 3                 */
858                     END;                                                /*                                          */
859                 END;                                                    /*                                          */
860             END;                                                        /*                                          */
861         END; ELSE                                                       /*                                          */
862         IF OPT=10 & DESB THEN PUT FILE(DIC) EDIT(DESNO || '01')(COL(1),A); /* OPTION 10, OUTPUT TO TRANSACTION FILE */
863         GOTO L12;                                                       /* BRANCH FOR NEXT RECORD                   */
864     /*-----------------------------------------------------------------*/ /*-----------------------------------------*/
865     L09: IF DESOP THEN CLOSE FILE(DIC);                                 /* TEST TRANSACTION FILE STATUS AND CLOSE   */
866         PUT PAGE;                                                       /*                                          */
867     END;                                                                /* END PROGRAMME R                          */
868                                                                         /*******************************************/
869     %PROCESS;                                                           /*                                          */
870                                                                         /*******************************************/
871     CON:PROC(FILN2);                                                    /* PROCEDURE CON                            */
872         DCL FILN2 CHAR(8);                                              /*                                          */
873         DCL (I,J,K) FIXED BINARY, AK(3,26) FIXED BINARY EXTERNAL,       /*                                          */
874         (AUTR(*) CHAR(19), AUTR1(*) CHAR(3), ATAB(*) CHAR(3), ATAB1(*)  /*                                          */
875         FIXED BINARY) CONTROLLED EXTERNAL, (ALPHA CHAR(26), NALC FIXED  /*                                          */
876         BINARY) EXTERNAL;                                               /* VARIABLE DECLARATIONS                    */
877         DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),        /*                                          */
878         MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),  /*                                          */
879         GRK(11) CHAR(30),OAK(4) CHAR(40), ORK(6) CHAR(30), OPSOK(6)     /*                                          */
880         CHAR(40)) VARYING EXTERNAL;                                     /*                                          */
881         DCL NORM(2,16,10) FIXED BINARY EXTERNAL;                        /*                                          */
882         DCL NORM2(2,16,2) FLOAT DECIMAL EXTERNAL;                       /*                                          */
883     /*-----------------------------------------------------------------*/ /*-----------------------------------------*/
884         DCL CONST FILE;                                                 /* DECLARATION OF FILE (CONST)              */
885     /*-----------------------------------------------------------------*/ /*-----------------------------------------*/
886         OPEN FILE(CONST) INPUT TITLE(FILN2);                            /* OPEN CONST FOR INPUT                     */
887         DO I=1 TO 26;                                                   /*                                          */
888            GET FILE(CONST) LIST(AK(1,I));                               /* INPUT AUTHORITY/COUNTRY NUMBERS          */
889         END;                                                            /*                                          */
890         AK(2,1)=1;     AK(3,1)=AK(2,1)+AK(1,1)-1;                       /* -> CALCULATE AK KEYS                     */
891         DO I=2 TO 26;                                                   /*                                          */
892            AK(2,I)=AK(3,I-1)+1;                                         /* -> START POSITIONS                       */
893            AK(3,I)=AK(1,I)+AK(2,I)-1;                                   /* -> LAST POSITIONS                        */
894         END;                                                            /*                                          */
895         N=SUM(AK(1,*));                                                 /* -> CALCULATE STORAGE REQUIRED            */
```

Dataset Limited

```
896          ALLOCATE AUTR(N),AUTR1(N);                                  /* -> ALLOCATE STCRAGE                          */
897          DO K=1 TO 26;                                               /*                                              */
898            DO I=AK(2,K) TO AK(3,K);                                  /*                                              */
899             GET FILE(CONST) EDIT(AUTR(I),AUTR1(I))                   /* -> INPUT AUTHORITY/COUNTRY & CODES           */
900             (COL(1),A(19),COL(40),A(3));                             /*                                              */
901             END;                                                     /*                                              */
902          END;                                                        /*                                              */
903          GET FILE(CONST) EDIT((OAK(K) DO K=1 TO 4))(4(COL(1),A(40))); /* INPUT AREA CODES (PRE 1972)                  */
904          GET FILE(CONST) EDIT((ORK(K) DO K=1 TO 6))(6(COL(1),A(30))); /* INPUT RELIGICN CODES (PRE 1972)              */
905          GET FILE(CONST) EDIT((OPSOK(K) DO K=1 TO 6))(6(COL(1),A(40))); /* INPUT PARENT/STUDENT OCC.CODES (PRE 1972) */
906          GET FILE(CONST) EDIT((SCK(I) DO I=1 TO 9))(9(COL(1),A(60))); /* INPUT SCHOOL CODES                           */
907          GET FILE(CONST) EDIT((PSOK(I) DO I=1 TO 5))(5(COL(1),A(65))); /* INPUT PARENT/STUDENT OCC. CODES              */
908          GET FILE(CONST) EDIT((STK(I) DO I=1 TO 6))(6(COL(1),A(50))); /* INPUT SETTLEMENT CODES                       */
909          GET FILE(CONST) EDIT((MK(I) DO I=1 TO 3))(3(COL(1),A(12))); /* INPUT MARITAL STATUS CODES                   */
910          GET FILE(CONST) EDIT((OK(I) DO I=1 TO 10))(10(COL(1),A(35))); /* INPUT CVER 25 OCC. CODES                    */
911          GET FILE(CONST) EDIT((RK(I) DO I=1 TO 5))(5(CCL(1),A(30))); /* INPUT RELIGICN CODES                         */
912          GET FILE(CONST) EDIT((GRK(I) DO I=1 TO 11))(11(COL(1),A(30))); /* INPUT REGION CODES                          */
913          GET FILE(CONST) EDIT(ALPHA)(COL(1),A(26));                  /* INPUT ALPHABET                               */
914          GRC=SUBSTR(ALPHA,1,11);                                     /* GEOGRAPHICAL REGION KEYS                     */
915          GET FILE(CONST) EDIT(NALC)(COL(1),F(3));                    /* INPUT A'LEVEL NUMBER                         */
916          ALLOCATE ATAB(NALC),ATAB1(NALC);                            /* -> ALLOCATE STORAGE                          */
917          GET FILE(CONST) SKIP;                                       /*                                              */
918          GET FILE(CONST) EDIT((ATAB(I),ATAB1(I) DO I=1 TO NALC))     /* -> INPUT A'LEVEL CODES & COLUMNS             */
919          (X(2),A(3),X(1),F(2));                                      /*                                              */
920          DO I=1 TO 2;                                                /*                                              */
921             DO J=1 TO 16;                                            /*                                              */
922                GET FILE(CONST) LIST((NORM(I,J,K) DO K=1 TO 10));     /* INPUT NORM TABLES                            */
923                GET FILE(CONST) LIST((NORM2(I,J,K) DO K=1 TO 2));     /*                                              */
924             END;                                                     /*                                              */
925          END;                                                        /*                                              */
926          CLOSE FILE(CONST);                                          /* CLOSE CONST                                  */
927       END;                                                           /* END PROCEDURE CON                            */
```

Dataset Limited

3.3 COMPILER LISTING

THIS SECTION CONTAINS A LISTING OF THE OUTPUT PROVIDED BY THE PL/1 F COMPILER DURING THE COMPILATION OF PROGRAMME R.

THE JCL WAS AS FOLLOWS:

```
$RUN *PL1 SPUNCH=-R
  ----------------------------
 |                            |
 |                            |
 |  PROGRAMME R  SOURCE DECK  |
 |                            |
 |                            |
  ----------------------------
%PROCESS;
  ----------------------------
 |                            |
 |  PROCEDURE CON SOURCE DECK |
 |                            |
  ----------------------------
$ENDFILE
```

VERSION 5.0 03/09/70 MTS PL/I COMPILER (F) TIME 23:51.07 DATE 08-21-73

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRO
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST

*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
*OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
*OPTIONS IN EFFECT*     NEST,OPLIST
```

DURHAM UNIVERSITY SCIENCE 1 - 9 JAN 1974 SECTION LIBRARY

```
STMT LEVEL NEST
                                                                                R0000010
  1                 (SUBRG):(STRG):                                             R0000020
                    R:PROCEDURE OPTIONS(MAIN);                                  R0000030
  2     1                DCL 1 RECORD BASED(PTR),                               R0000040
                               2 (DESNO CHAR(8), NAME CHAR(40)),                R0000050
                               2 GENERAL,                                       R0000060
                                 3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),R0000070
                                 3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),      R0000080
                                 3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)), R0000090
                                 3 FPOS CHAR(4),                                R0000100
                               2 EDUCATION,                                     R0000110
                                 3 (SCHNO CHAR(2), SCH(10) CHAR(5)),            R0000120
                                 3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),         R0000130
                                 3 (UNCOL(2) CHAR(21), FPT(2) CHAR(1), FT(2,2) CHAR(6)),   R0000140
                                 3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),   R0000150
                                 3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),          R0000160
                                 3 (ADIP CHAR(30), ADIPC CHAR(1)),              R0000170
                                 3 DDIPC CHAR(1),                               R0000180
                                 3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),            R0000190
                                 3 ATO CHAR(1),                                 R0000200
                               2 COURSE,                                        R0000210
                                 3 (LEN CHAR(1), CODES(4) CHAR(3)),             R0000220
                                 3 ((LE,DC(4,24:64)) CHAR(1)),                  R0000230
                                 3 (DC1(4,26:30) CHAR(2)),                      R0000240
                                 3 ((DC46(4),DC49(4)) CHAR(3)),                 R0000250
                               2 FINALS,                                        R0000260
                                 3 ((TC,WS,BE,SUBCAR) CHAR(1)),                 R0000270
                               2 ANNPROG,                                       R0000280
                                 3 AP CHAR(802),                                R0000290
                                 3 ((LEAP,FK(100,2)) FIXED BINARY),             R0000300
                               2 INTEL,                                         R0000310
                                 3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY), R0000320
                                 3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),        R0000330
                                 3 STEN(2,16) FIXED BINARY;                     R0000340
                    /*-----------------------------------------------------------------*/R0000350
  3     1                DCL (NORM(2,16,13) FIXED BINARY, NORM2(2,16,2)         R0000360
                         FLOAT DECIMAL, (OPSOK(6) CHAR(40), PSOK(5) CHAR(65), GRC CHAR(11),R0000370
                         STK(6) CHAR(50),MK(3) CHAR(12),OK(10) CHAR(35),RK(5) CHAR(30),    R0000380
                         GRK(11) CHAR(30),OAK(4) CHAR(40),ORK(6) CHAR(30)) VARYING         R0000390
                         ) EXTERNAL;                                            R0000400
  4     1                DCL 1 RECL(500),                                       R0000410
                               2 NAML CHAR(40),                                 R0000420
                               2 DESL CHAR(8),                                  R0000430
                               2 (DOBL,START) CHAR(6),                          R0000440
                               2 FIN CHAR(7),                                   R0000450
                               2 TYPE CHAR(1),                                  R0000460
                               2 DHT CHAR(3),                                   R0000470
                               2 DHY CHAR(2),                                   R0000480
                               2 DHB CHAR(21);                                  R0000490
  5     1                DCL (FILN1,FILN2,DESNO2,DT,DES) CHAR(8),YE CHAR(4),TRANS CHAR(2), R0000500
                         (PI,WC,YT,NTT) CHAR(1), (B,DESB) BIT(1), (OPT,KT,YR,NT)  R0000510
                         FIXED BINARY, BCON BIT(1) INITIAL('1'B), BMAX BIT(1), (VP,CSC)   R0000520
                         CHAR(1), VPB BIT(1), (VPI,I) FIXED BINARY, DESNO3 CHAR(8);       R0000530
  6     1                DCL L(10) LABEL INITIAL(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10);         R0000540
  7     1                DCL CON ENTRY(CHAR(8)), DESOP BIT(1) INITIAL('0'B);              R0000550
                    /*-----------------------------------------------------------------*/R0000560
```

```
STMT LEVEL NEST

  8     1              DCL STUDENT FILE RECORD ENV(F(2416));                          R0000570
  9     1              DCL DIC FILE;                                                  R0000580
                    /*****************************************************************/R0000590
 10     1              RECIN:PROCEDURE;                                               R0000600
 11     2              DCL CONT CHAR(1), (S,Q,K) FIXED BINARY, X5 CHAR(5), X10 CHAR(10), R0000610
                       X2 CHAR(2);                                                    R0000620
 12     2              GET EDIT(NAME,TITLE,SEX,DOB,DOBC,TTFT(1,1),MART,REG,SET,FPOS,POCC,R0000630
                       SOCC,OV25,REL,CONF,YC,AUTHNO)(COL(9),A(40),X(1),A(2),          R0000640
                       2(A(1),A(6)),3(A(1)),A(4),5(A(1)),A(2),A(1));                  R0000650
 13     2              GET EDIT(PNAME,AUTH,LEN,CODES(1),CODES(2),DC(1,24),DC(1,25),   R0000660
                       DC1(1,26),DC1(1,27),DC1(1,29),DC1(1,30),DC(1,32),DC(1,34),DC(1,35)R0000670
                       ,DC(1,36),DC(1,37),DC(1,38),DC(1,41),DC(1,42),DC(1,43),DC(1,44), R0000680
                       DC(1,45),DC46(1),DC49(1),DC(1,59),DC(1,64))                    R0000690
                       (COL(9),A(17),A(19),A(1),2(A(3)),2(A(1)),4(A(2)),11(A(1)),2(A(3)),R0000700
                       2(A(1)));                                                      R0000710
 14     2         L2:  GET EDIT(CONT)(COL(9),A(1));                                   R0000720
 15     2              IF CONT¬='C' THEN                                              R0000730
 16     2              DO; GET EDIT(ATO,SCHNO,QUALNO)(A(1),X(1),2(A(2)));             R0000740
 18     2    1             S=SCHNO;                                   /* CONVERSION */R0000750
 19     2    1             DO K=1 TO MIN(S,10);                                       R0000760
 20     2    2                GET EDIT(SCH(K))(A(5));                                 R0000770
 21     2    2             END;                                                       R0000780
 22     2    1             IF S>10 THEN                                               R0000790
 23     2    1             DO; PUT EDIT(DESNO,'ERROR.....SCHOOL TYPE STORAGE EXCEEDED')  R0000800
                                (COL(1),A,X(1),A);                                    R0000810
 25     2    2                DO K=11 TO S;                                           R0000820
 26     2    3     L222:         GET EDIT(X5)(A(5));                                 R0000830
 27     2    3                   IF X5=SUBSTR(DESNO,1,5) THEN                        R0000840
 28     2    3                   DO; GET EDIT(X5)(A(5));                             R0000850
 30     2    4                       GOTO L222;                                      R0000860
 31     2    4                   END;                                                R0000870
 32     2    3                DO;                                                    R0000880
 33     2    2             END;                                                       R0000890
 34     2    1             IF MOD(S,2)=0 THEN GET EDIT(X5)(A(5));                     R0000900
 36     2    1             Q=QUALNO;                                  /* CONVERSION */R0000910
 37     2    1             DO K=1 TO MIN(Q,30);                                       R0000920
 38     2    2     L1:        GET EDIT(QUAL(K))(A(10));                               R0000930
 39     2    2                IF QUAL(K)=DESNO || '  ' THEN GOTO L1;                  R0000940
 41     2    2             END;                                                       R0000950
 42     2    1             IF Q>30 THEN                                               R0000960
 43     2    1             DO; PUT EDIT(DESNO,'ERROR.....QUALIFICATION STORAGE EXCEEDED')R0000970
                                (COL(1),A,X(1),A);                                    R0000980
 45     2    2                DO K=31 TO Q;                                           R0000990
 46     2    3     L223:         GET EDIT(X10)(A(10));                               R0001000
 47     2    3                   IF X10=DESNO || '  ' THEN GOTO L223;                R0001010
 49     2    3                DO;                                                    R0001020
 50     2    2             END;                                                       R0001030
 51     2    1          END; ELSE                                                     R0001040
 52     2               DO; GET EDIT(CONT)(A(1));                                    R0001050
 54     2    1             IF CONT='1' THEN                                           R0001060
 55     2    1             DO; DO K=1 TO 2;                                           R0001070
 57     2    3                   GET EDIT(UNCOL(K),FPT(K),FT(K,1),FT(K,2))          R0001080
                                (A(21),A(1),2(A(6)));                                 R0001090
 58     2    3                DO;                                                    R0001100
```

```
STMT LEVEL NEST

 59     2    2                GOTO L2;                                                R0001110
 60     2    2            END; ELSE                                                   R0001120
 61     2    1            IF CONT='2' | CONT='3' THEN                                 R0001130
 62     2    1            DO; K=CONT;                                   /* CONVERSION */R0001140
 64     2    2                K=K-1;                                                  R0001150
 65     2    2                GET EDIT(UNOTH(K),DEG(K),HPCD(K),SUB(K),DATT(K))        R0001160
                              (A(21),A(3),A(4),A(36),A(6));                           R0001170
 66     2    2                GOTO L2;                                                R0001180
 67     2    2            END; ELSE                                                   R0001190
 68     2    1            IF CONT='4' THEN                                            R0001200
 69     2    1            DO; GET EDIT(ADIP,ADIPC,DDIPC,OQ(1),OQC(1),OQ(2),OQC(2))    R0001210
                              (A(21),2(A(1)),2(A(19),A(4)));                          R0001220
 71     2    2                GOTO L2;                                                R0001230
 72     2    2            END;                                                        R0001240
 73     2    1        END;                                                            R0001250
 74     2            LE='1';                                                         R0001260
 75     2        END;                                                                R0001270
                  /*****************************************************************/R0001280
 76     1        CHECKREC:PROCEDURE;                                                 R0001290
 77     2            DCL X2 CHAR(2), (K,Q) FIXED BINARY, OKB BIT(1);                 R0001300
 78     2            IF SUBSTR(TTFT(1,1),5,2)<'72' THEN OKB='1'B; ELSE OKB='0'B;     R0001310
 81     2            X2=SUBSTR(TTFT(1,1),5,2);                                       R0001320
 82     2            K=X2;                                         /* CONVERSION */R0001330
 83     2            X2=SUBSTR(DOB,5,2);                                             R0001340
 84     2            Q=X2;                                         /* CONVERSION */R0001350
 85     2            IF K-Q<25 THEN OV25=' '; ELSE IF OV25=' ' THEN OV25='0';        R0001360
 89     2            IF Q<17 | Q>60 THEN                                             R0001370
 90     2            PUT EDIT(DESNO,'AGE CHECK, RECORDED AS',Q)                      R0001380
                      (COL(1),A,X(1),A,X(1),F(2));                                    R0001390
 91     2            IF SEX¬='M' & SEX¬='F' THEN                                     R0001400
 92     2            PUT EDIT(DESNO,'ERROR RETURNED.....SEX')(SKIP,A(8),X(1),A);     R0001410
 93     2            IF MART<'1' | MART>'3' THEN                                     R0001420
 94     2            PUT EDIT(DESNO,'ERROR RETURNED.....MARITAL STATUS')             R0001430
                      (SKIP,A(8),X(1),A);                                             R0001440
 95     2            IF REL<'1' | ((REL>'5' & ¬OKB) | (REL>'6' & OKB)) THEN          R0001450
 96     2            PUT EDIT(DESNO,'ERROR RETURNED.....RELIGION')(SKIP,A(8),X(1),A); R0001460
 97     2            IF SET<'1' | ((SET>'4' & OKB) | (SET>'6' & ¬OKB)) THEN          R0001470
 98     2            PUT EDIT(DESNO,'ERROR RETURNED.....SETTLEMENT')                 R0001480
                      (SKIP,A(8),X(1),A);                                             R0001490
 99     2            IF ¬OKB & (REG<'A' | REG>'K') THEN                              R0001500
100     2            PUT EDIT(DESNO,'ERROR RETURNED.....REGION')(SKIP,A(8),X(1),A);  R0001510
101     2            IF POCC<'1' | ((POCC>'5' & ¬OKB) | (POCC>'6' & OKB)) THEN       R0001520
102     2            PUT EDIT(DESNO,'ERROR RETURNED.....PARENT/GUARDIAN OCCUPATION') R0001530
                      (SKIP,A(8),X(1),A);                                             R0001540
103     2            IF SOCC¬=' ' & (SOCC<'1' | ((SOCC>'5' & ¬OKB) | (SOCC>'6' & OKB)))R0001550
104     2            THEN PUT EDIT(DESNO,'ERROR RETURNED.....STUDENT OCCUPATION')    R0001560
                      (SKIP,A(8),X(1),A);                                             R0001570
105     2            IF OV25¬=' ' & (OV25<'0' | OV25>'9') THEN                       R0001580
106     2            PUT EDIT(DESNO,'ERROR RETURNED.....OVER 25 OCCUPATION')         R0001590
                      (SKIP,A(8),X(1),A);                                             R0001600
107     2            IF SUBSTR(FPOS,1,2)>SUBSTR(FPOS,3,2) THEN                       R0001610
108     2            PUT EDIT(DESNO,'ERROR RETURNED.....FAMILY POSITION')            R0001620
                      (SKIP,A,X(1),A);                                                R0001630
109     2            IF SUBSTR(TTFT(1,1),3,4)¬=DC1(1,26) || DC1(1,27) THEN           R0001640
```

```
STMT LEVEL NEST

110     2              PUT EDIT(DESNO,'ERROR RETURNED.....START DATE CHECK')          R0001650
                       (COL(1),A,X(1),A);                                             R0001660
111     2              IF ¬OKB & (AUTHNO<'1' | AUTHNO>'5') THEN                       R0001670
112     2              PUT EDIT(DESNO,'ERROR RETURNED.....AUTHORITY/COUNTRY NUMBER')  R0001680
                       (COL(1),A,X(1),A);                                             R0001690
113     2          END;                                                               R0001700
                   /*****************************************************************/R0001710
114     1          ITRECIN:PROCEDURE;                                                 R0001720
115     2              DCL MNT(16) FIXED BINARY INITIAL(20,13,26,26,26,20,26,20,20,26,R0001730
                       20,26,20,20,20,26);                                            R0001740
116     2              DCL (I,J) FIXED BINARY, SW BIT(1) INITIAL('0'B);               R0001750
117     2              GET EDIT(VALUES(YR,*),DSCALE(YR),AH4(YR,*),PF16(YR,*))         R0001760
                       (6(F(2)),F(3),2(F(2)),F(3),16(F(2)));                          R0001770
118     2              DO K=1 TO 6;                                                   R0001780
119     2    1            IF VALUES(YR,K)<0 | VALUES(YR,K)>60 THEN                   R0001790
120     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR VALUES SCORE',K)        R0001800
                          (SKIP,COL(1),A,X(1),A,X(1),F(1));                          R0001810
121     2    1         END;                                                          R0001820
122     2              IF DSCALE(YR)<0 | DSCALE(YR)>300 THEN                          R0001830
123     2              PUT EDIT(DESNO,'ERROR RETURNED FOR D.SCALE SCORE')             R0001840
                       (SKIP,COL(1),A,X(1),A);                                        R0001850
124     2              DO K=1 TO 3;                                                   R0001860
125     2    1            IF AH4(YR,K)<0 | AH4(YR,K)>130 THEN                        R0001870
126     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR A.H.4 SCORE',K)         R0001880
                          (SKIP,COL(1),A,X(1),A,X(1),F(2));                          R0001890
127     2    1         END;                                                          R0001900
128     2              DO K=1 TO 16;                                                  R0001910
129     2    1            IF PF16(YR,K)<0 | PF16(YR,K)>MNT(K) THEN                   R0001920
130     2    1            DO; PUT EDIT(DESNO,'ERROR RETURNED FOR P.F.16 SCORE',K)    R0001930
                             (SKIP,COL(1),A,X(1),A,X(1),F(2));                       R0001940
132     2    2               SW='1'B;                                                R0001950
133     2    2            END;                                                       R0001960
134     2    1         END;                                                          R0001970
135     2              IF SW THEN GOTO L2;                                            R0001980
137     2              DO I=1 TO 16;                                                  R0001990
138     2    1            DO J=1 TO 9;                                               R0002000
139     2    2               IF PF16(YR,I)<=NORM(NT,I,1) THEN                        R0002010
140     2    2               DO; STEN(YR,I)=J;                                       R0002020
142     2    3                   GOTO L1;                                            R0002030
143     2    3               END; ELSE                                               R0002040
144     2    2               IF PF16(YR,I)>NORM(NT,I,J) & PF16(YR,I)<=NORM(NT,I,J+1) THENR0002050
145     2    2               DO; STEN(YR,I)=J+1;                                     R0002060
147     2    3                   GOTO L1;                                            R0002070
148     2    3               END;                                                    R0002080
149     2    2            END;                                                       R0002090
150     2    1     L1:  END;                                                         R0002100
151     2          L2:  SUBSTR(INTELB,YR,1)='*';                                     R0002110
152     2          END;                                                               R0002120
                   /*****************************************************************/R0002130
153     1          UPDATE:PROCEDURE;                                                  R0002140
154     2              DCL INST CHAR(70), X1 CHAR(1), XVAR CHAR(4) VARYING,           R0002150
                       (I,K,POSEND,POSEQ,J) FIXED BINARY;                             R0002160
155     2              GET EDIT(X1)(COL(9),A(1));                                     R0002170
156     2              IF X1>LEN THEN PUT EDIT(DESNO,'WARNING....YEAR OF UPDATE > COURSE'R0002180
```

```
STMT LEVEL NEST

                        ,' LENGTH')(COL(1),A,X(1),A,A);                          R0002190
158     2               GET EDIT(X1)(COL(10),A(1));                              R0002200
159     2               IF X1='C' THEN                                           R0002210
160     2               DO; PUT EDIT(DESNO,'PROCEEDING TO A FOURTH YEAR')(COL(1),A,  R0002220
                            X(1),A);                                             R0002230
162     2     1             LEN='4';                                             R0002240
163     2     1         END; ELSE IF X1='F' THEN GOTO L2;                        R0002250
166     2           L1: GET EDIT(INST)(COL(11),A(70));                           R0002260
167     2               DO I=1 BY 10 TO 70;                                      R0002270
168     2     1            IF SUBSTR(INST,I,2)='*A' THEN                         R0002280
169     2     1            DO; POSEQ=I+INDEX(SUBSTR(INST,I+1),'=');              R0002290
171     2     2                POSEND=I+INDEX(SUBSTR(INST,I+1),'*');             R0002300
172     2     2                IF POSEQ=I THEN GOTO L3;                          R0002310
174     2     2                IF POSEND=I THEN                                  R0002320
175     2     2                POSEND=I+INDEX(SUBSTR(INST,I+1),' ');             R0002330
176     2     2                XVAR=SUBSTR(INST,I+2,POSEQ-I-2);                  R0002340
177     2     2                K=XVAR;                                 /* CONVERSION */R0002350
178     2     2                IF FK(K,1)¬=0 THEN                                R0002360
179     2     2                DO; SUBSTR(AP,FK(K,1))=SUBSTR(AP,FK(K,1)+FK(K,2));  R0002370
181     2     3                    LEAP=LEAP-FK(K,2);                            R0002380
182     2     3                    DO J=1 TO 100;                                R0002390
183     2     4                       IF FK(J,1)¬=0 & FK(J,1)>FK(K,1) THEN       R0002400
184     2     4                       FK(J,1)=FK(J,1)-FK(K,2);                   R0002410
185     2     4                    END;                                          R0002420
186     2     3                    FK(K,*)=0;                                    R0002430
187     2     3                END;                                              R0002440
188     2     2                IF POSEND=POSEQ+1 THEN GOTO L3;                   R0002450
190     2     2                IF (LEAP+POSEND-POSEQ)>802 THEN                   R0002460
191     2     2                DO; PUT EDIT(DESNO,'ANNUAL PROGRESS VECTOR SIZE EXCEEDED', R0002470
                                   '...INFORMATION IGNORED')(COL(1),A,X(1),A,A);  R0002480
193     2     3                    DO J=1 BY 1;                                  R0002490
194     2     4                       IF SUBSTR(INST,70,1)='C' THEN GET SKIP; ELSE RETURN;R0002500
197     2     4                    END;                                          R0002510
198     2     3                END;                                              R0002520
199     2     2                SUBSTR(AP,LEAP,POSEND-POSEQ)=SUBSTR(INST,POSEQ+1,  R0002530
                               POSEND-POSEQ);                                    R0002540
200     2     2                FK(K,1)=LEAP;                                     R0002550
201     2     2                LEAP=LEAP+POSEND-POSEQ;                           R0002560
202     2     2                FK(K,2)=POSEND-POSEQ;                             R0002570
203     2     2            END;                                                  R0002580
204     2     1       L3: END;                                                   R0002590
205     2               IF SUBSTR(INST,70,1)='C' THEN GOTO L1; ELSE RETURN;      R0002600
208     2           L2: GET EDIT(TC,WS,BE,SUBCAR,TTFT(1,2))                      R0002610
                        (COL(20),A(1),COL(30),A(1),COL(40),A(1),COL(50),A(1),    R0002620
                        COL(60),A(6));                                           R0002630
209     2               TTFT(2,1)='LEFT  ';                                      R0002640
210     2           END;                                                         R0002650
                    /*****************************************************************/R0002660
211     1           SPNT:PROCEDURE;                                              R0002670
212     2               DCL (I,N) FIXED BINARY, PB BIT(1) INITIAL('0'B), SW BIT(1);  R0002680
213     2               IF PI='2' THEN                                           R0002690
214     2               DO; PUT SKIP(3);                                         R0002700
216     2     1             DO I=1 TO KT;                                      R0002710
217     2     2                IF MOD(I-1,10)=0 THEN PUT SKIP;                   R0002720
```

```
STMT LEVEL NEST

219     2    2                PUT EDIT(DESL(I))(X(2),A(8));                          R0002730
220     2    2                IF FIN(I)¬='        ' THEN IF SUBSTR(FIN(I),7,1)='*' THEN  R0002740
222     2    2                PUT EDIT('*')(A); ELSE PUT EDIT('+')(A); ELSE PUT EDIT(' ')R0002750
                              (A);                                                   R0002760
225     2    2             END;                                                       R0002770
226     2    1             RETURN;                                                    R0002780
227     2    1          END; ELSE                                                     R0002790
228     2               IF PI='3' THEN                                                R0002800
229     2               DO; ON ENDPAGE(SPRINT)                                        R0002810
231     2    1             BEGIN; IF PB THEN PUT PAGE;                                R0002820
234     3    1                    PUT EDIT('NAME','D.F.S.NO.','BIRTH','START','FINISH')  R0002830
                                  (SKIP(2),COL(1),A,COL(42),A,3(X(5),A));            R0002840
235     3    1                    PUT SKIP(2);                                       R0002850
236     3    1             END;                                                       R0002860
237     2    1             N=KT;                                                      R0002870
238     2    1     L1:     SW='0'B;                                                   R0002880
239     2    1             N=N-1;                                                     R0002890
240     2    1             DO I=1 TO N;                                               R0002900
241     2    2                IF NAML(I)>NAML(I+1) THEN                               R0002910
242     2    2                DO; RECL(KT+1)=RECL(I);                                 R0002920
244     2    3                    RECL(I)=RECL(I+1);                                  R0002930
245     2    3                    RECL(I+1)=RECL(KT+1);                               R0002940
246     2    3                    SW='1'B;                                            R0002950
247     2    3                END;                                                    R0002960
248     2    2             END;                                                       R0002970
249     2    1             IF SW THEN GOTO L1;                                        R0002980
251     2    1             SIGNAL ENDPAGE(SPRINT);                                    R0002990
252     2    1             PB='1'B;                                                   R0003000
253     2    1             DO I=1 TO KT;                                              R0003010
254     2    2                PUT EDIT(NAML(I),DESL(I),SUBSTR(DOBL(I),1,2),'/',SUBSTR(   R0003020
                              DOBL(I),3,2),'/',SUBSTR(DOBL(I),5,2),SUBSTR(START(I),1,2), R0003030
                              '/',SUBSTR(START(I),3,2),'/',SUBSTR(START(I),5,2))      R0003040
                              (SKIP,COL(1),A(40),X(2),A(8),2(X(2),5(A)));            R0003050
255     2    2                IF FIN(I)¬='        ' THEN PUT EDIT(SUBSTR(FIN(I),1,2),    R0003060
                              '/',SUBSTR(FIN(I),3,2),'/',SUBSTR(FIN(I),5,2))(X(3),5(A)); R0003070
257     2    2             END;                                                       R0003080
258     2    1             ON ENDPAGE(SPRINT) SYSTEM;                                 R0003090
259     2    1          END;                                                          R0003100
260     2            END;                                                             R0003110
                     /*****************************************************************/R0003120
261     1            PRINT:PROCEDURE;                                                 R0003130
262     2               DCL (I,K) FIXED BINARY;                                       R0003140
                        /*****************************************************************/R0003150
263     2            CDATE:PROCEDURE(X6) RETURNS(CHAR(8));                            R0003160
264     3               DCL X6 CHAR(6);                                               R0003170
265     3               IF X6¬='        ' THEN                                        R0003180
266     3               RETURN(SUBSTR(X6,1,2) || '/' || SUBSTR(X6,3,2) || '/' || SUBSTR(  R0003190
267     3               X6,5,2)); ELSE RETURN('        ');                            R0003200
268     3            END;                                                             R0003210
269     2            DCL CDATE ENTRY(CHAR(6)) RETURNS(CHAR(8));                       R0003220
                        /*****************************************************************/R0003230
270     2            GENPNT:PROCEDURE;                                                R0003240
271     3               DCL OKB BIT(1), X6 CHAR(6), X1 CHAR(1) DEFINED X6 POS(6);     R0003250
272     3               IF SUBSTR(TTFT(1,1),5,2)<'72' THEN OKB='1'B; ELSE OKB='0'B;   R0003260
```

```
STMT LEVEL NEST

275     3              IF SEX='M' THEN PUT EDIT('MALE,')(COL(10),A); ELSE                R0003270
277     3              PUT EDIT('FEMALE,')(COL(10),A);                                   R0003280
278     3              IF DOB¬='       ' THEN PUT EDIT('BORN: ',CDATE(DOB))(X(1),A,A);   R0003290
280     3              IF TITLE='M1' THEN PUT EDIT('TITLE: MR')(COL(10),A); ELSE         R0003300
282     3              IF TITLE='M2' THEN PUT EDIT('TITLE: MRS')(COL(10),A); ELSE        R0003310
284     3              IF TITLE='M3' THEN PUT EDIT('TITLE: MISS')(COL(10),A);            R0003320
286     3              PUT SKIP(2);                                                      R0003330
287     3              X6=TTFT(1,1);                                                     R0003340
288     3              PUT EDIT('TEACHERS'' TRAINING: FROM',CDATE(X6))(COL(10),A,X(1),A);R0003350
289     3              X6=TTFT(1,2);                                                     R0003360
290     3              IF X6¬='       ' THEN PUT EDIT('TEACHERS'' TRAINING: TO  ',       R0003370
                       CDATE(X6))(COL(10),A,X(1),A);                                     R0003380
292     3              X6=TTFT(2,1);                                                     R0003390
293     3              IF SUBSTR(X6,1,4)='WITH' THEN PUT EDIT('STUDENT WITHDREW, REASON',R0003400
                       'CODED AS',X1)(COL(10),A,2(X(1),A));                              R0003410
295     3              PUT SKIP(2);                                                      R0003420
296     3              IF PNAME¬='    ' THEN PUT EDIT('PREVIOUS SURNAME',PNAME)          R0003430
                       (COL(10),A,X(1),A);                                               R0003440
298     3              IF MART>='1' & MART<='3' THEN                                     R0003450
299     3              DO; I=MART;                                        /* CONVERSION */R0003460
301     3     1            PUT EDIT('MARITAL STATUS:',MK(I))(COL(10),A,X(1),A);          R0003470
302     3     1        END;                                                              R0003480
303     3              IF DOBC='1' THEN PUT EDIT('DATE OF BIRTH CERTIFICATE CONFIRMED')  R0003490
                       (COL(10),A);                                                      R0003500
305     3              IF FPOS¬='     ' THEN PUT EDIT('POSITION IN FAMILY: ',SUBSTR(FPOS, R0003510
                       1,2),'/',SUBSTR(FPOS,3,2))(COL(10),4(A));                         R0003520
307     3              IF SET>='1' & ((SET<='4' & OKB) | (SET<='6' & ¬OKB)) THEN         R0003530
308     3              DO; I=SET;                                         /* CONVERSION */R0003540
310     3     1            PUT EDIT('SETTLEMENT: ')(COL(10),A);                          R0003550
311     3     1            IF OKB THEN PUT EDIT(OAK(I))(A); ELSE PUT EDIT(STK(I))(A);    R0003560
314     3     1        END;                                                              R0003570
315     3              I=INDEX(GRC,REG);                                                 R0003580
316     3              IF REG>='1' & REG<='11' & ¬OKB THEN                               R0003590
317     3              PUT EDIT('REGION: ',GRK(I))(COL(10),A,A); ELSE                    R0003600
318     3              IF OKB THEN PUT EDIT('REGION: NOT CODED PRIOR TO 1972')           R0003610
                       (COL(10),A);                                                      R0003620
320     3              IF SOCC>='1' & ((SOCC<='6' & OKB) | (SOCC<='5' & ¬OKB)) THEN      R0003630
321     3              DO; I=SOCC;                                        /* CONVERSION */R0003640
323     3     1            PUT EDIT('STUDENT OCCUPATION: ')(COL(10),A);                  R0003650
324     3     1            IF OKB THEN PUT EDIT(OPSOK(I))(A); ELSE PUT EDIT(PSOK(I))(A); R0003660
327     3     1        END; ELSE                                                         R0003670
328     3              IF SOCC=' ' THEN PUT EDIT('STUDENT OCCUPATION: AT SCHOOL')(       R0003680
                       COL(10),A);                                                       R0003690
330     3              IF POCC>='1' & ((POCC<='6' & OKB) | (POCC<='5' & ¬OKB)) THEN      R0003700
331     3              DO; I=POCC;                                        /* CONVERSION */R0003710
333     3     1            PUT EDIT('PARENT OCCUPATION: ')(COL(10),A);                   R0003720
334     3     1            IF OKB THEN PUT EDIT(OPSOK(I))(A); ELSE PUT EDIT(PSOK(I))(A); R0003730
337     3     1        END;                                                              R0003740
338     3              IF REL>='1' & ((REL<='6' & OKB) | (REL<='5' & ¬OKB)) THEN         R0003750
339     3              DO; I=REL;                                         /* CONVERSION */R0003760
341     3     1            PUT EDIT('RELIGION: ')(COL(10),A);                            R0003770
342     3     1            IF OKB THEN PUT EDIT(ORK(I))(A); ELSE PUT EDIT(RK(I))(A);     R0003780
345     3     1        END;                                                              R0003790
346     3              IF CONF='Y' THEN PUT EDIT('CONFIRMED WHEN ADMITTED')(COL(10),A);  R0003800
```

```
STMT LEVEL NEST

348     3                ELSE PUT EDIT('NOT CONFIRMED WHEN ADMITTED')(COL(10),A);          R0003810
349     3                IF OV25=' ' THEN PUT EDIT('OVER 25 QUESTION: STUDENT UNDER 25 ',  R0003820
351     3                'WHEN ADMITTED')(COL(10),A,A); ELSE IF OV25>='0' & OV25<='9' THEN R0003830
352     3                DO; I=OV25;  IF I=0 THEN I=10;                    /* CONVERSION */R0003840
356     3    1               PUT EDIT('OVER 25 QUESTION: ',OK(I))(COL(10),2(A));           R0003850
357     3    1           END;                                                              R0003860
358     3                IF AUTH¬='  ' THEN PUT EDIT('AUTHORITY/COUNTRY: ',AUTH,'AUTHOR',  R0003870
                         'ITY CODE: ',AUTHNO)(COL(10),A,A,COL(10),A,A,A);                  R0003880
360     3           END;                                                                   R0003890
                    /*****************************************************************/R0003900
361     2           COUPNT:PROCEDURE;                                                      R0003910
362     3                PUT EDIT('LENGTH OF COURSE: ',LEN)(COL(10),A,A);                  R0003920
363     3                IF CODES(1)¬='   ' THEN PUT EDIT('MAIN SUBJECT CODE: ',CODES(1))  R0003930
                         (X(3),A,A);                                                       R0003940
365     3                IF CODES(2)¬='   ' THEN PUT EDIT('2ND. PROV CODE: ',CODES(2))     R0003950
                         (X(3),A,A);                                                       R0003960
367     3                PUT EDIT('DETAILS OF COURSE:  24 25   26 27 29 30   32   34 35 ', R0003970
                         '36 37 38 41 42   43 44 45   46-48 49-51   59   64')(COL(10),A,A);R0003980
368     3                K=LE;                                             /* CONVERSION */R0003990
369     3                DO I=1 TO K;                                                      R0004000
370     3    1              PUT EDIT(DC(I,24),DC(I,25),DC1(I,26),'/',DC1(I,27),DC1(I,29),  R0004010
                            '/',DC1(I,30),DC(I,32),DC(I,34),DC(I,35),DC(I,36),DC(I,37),    R0004020
                            DC(I,38),DC(I,41),DC(I,42),DC(I,43),DC(I,44),DC(I,45),DC46(I), R0004030
                            DC49(I),DC(I,59),DC(I,64))(COL(31),A,X(2),A,X(3),3(A),X(1),3(A)R0004040
                            ,X(4),A,X(2),7(X(2),A),X(2),3(X(2),A),X(5),A,X(3),A,2(X(4),A));R0004050
371     3    1           END;                                                              R0004060
372     3           END;                                                                   R0004070
                    /*****************************************************************/R0004080
373     2           SAQPNT:PROCEDURE;                                                      R0004090
374     3                PUT EDIT('NO. OF SCHOOLS ATTENDED: ',SCHNO,'  CODED: ')(COL(10),  R0004100
                         3(A));                                                            R0004110
375     3                K=SCHNO;                                          /* CONVERSION */R0004120
376     3                DO I=1 TO K;                                                      R0004130
377     3    1              PUT EDIT(SCH(I))(X(1),A);                                      R0004140
378     3    1           END;                                                              R0004150
379     3                PUT EDIT('NO. OF QUALIFICATIONS RECORDED: ',QUALNO)               R0004160
                         (SKIP(2),COL(10),A,A);                                            R0004170
380     3                K=QUALNO;                                         /* CONVERSION */R0004180
381     3                PUT EDIT('QUALIFICATIONS: ')(COL(10),A);                          R0004190
382     3                DO I=1 TO K;                                                      R0004200
383     3    1              PUT EDIT(QUAL(I))(X(4),A(10));                                 R0004210
384     3    1              IF MOD(I,6)=0 THEN PUT EDIT('  ')(COL(24),A);                  R0004220
386     3    1           END;                                                              R0004230
387     3                IF UNCOL(1)¬='   ' | UNCOL(2)¬='   ' THEN                         R0004240
388     3                DO; PUT EDIT('UNIV. OR COLL. ATTENDED FOR DEGREE STUDY','F/P TIME'R0004250
                             ,'FROM','TO')(SKIP(2),COL(10),A,X(4),A,X(8),A,X(10),A);       R0004260
390     3    1              PUT EDIT(UNCOL(1),FPT(1),CDATE(FT(1,1)),CDATE(FT(1,2)),        R0004270
                             UNCOL(2),FPT(2),CDATE(FT(2,1)),CDATE(FT(2,2)))(2(COL(10),     R0004280
                             A,COL(58),A,COL(66),A,COL(78),A));                            R0004290
391     3    1           END;                                                              R0004300
392     3                IF UNOTH(1)¬='   ' | UNOTH(2)¬='   ' THEN                         R0004310
393     3                DO; PUT EDIT('UNIV. OR OTHER AWARDING BODY','TITLE','H/P,C/D',    R0004320
                             'SUBJECTS','DATE')(SKIP(2),COL(10),A,COL(43),A,COL(50),A,     R0004330
                             COL(59),A,COL(103),A);                                        R0004340
```

```
STMT LEVEL NEST

395     3    1            PUT EDIT(SUBSTR(UNOTH(1),2),DEG(1),HPCD(1),SUB(1),          R0004350
                          CDATE(DATT(1)),SUBSTR(UNOTH(2),2),DEG(2),HPCD(2),          R0004360
                          SUB(2),CDATE(DATT(2)))(2(COL(10),A,COL(45),A,              R0004370
                          COL(53),A,COL(59),A,COL(99),A));                           R0004380
396     3    1        END;                                                            R0004390
397     3             IF ADIPC¬=' ' THEN PUT EDIT('DIPLOMA IN ART AND DESIGN',ADIP,   R0004400
                      ADIPC)(SKIP(2),COL(10),A,X(1),A,X(1),A);                        R0004410
399     3             IF DDIPC¬=' ' THEN PUT EDIT('NATIONAL DIPLOMA IN DESIGN',DDIPC) R0004420
                      (SKIP(2),COL(10),A,X(1),A);                                     R0004430
401     3             IF OQ(1)¬=' ' | OQ(2)¬=' ' THEN                                 R0004440
402     3             DO; PUT EDIT('OTHER ART QUALIFICATIONS: ')(SKIP(2),COL(10),A);  R0004450
404     3    1            PUT EDIT(OQ(1),OQC(1),OQ(2),OQC(2))(2(COL(36),A,X(1),A));   R0004460
405     3    1        END;                                                            R0004470
406     3             IF ATO¬=' ' THEN PUT EDIT('STUDENT EXCEPTED BY A.T.O. FROM',    R0004480
                      ' EXAMINATION REQUIREMENTS')(SKIP(2),COL(10),A,A);              R0004490
408     3         END;                                                                R0004500
                      /*****************************************************************/R0004510
409     2         APPNT:PROCEDURE;                                                    R0004520
410     3             PUT EDIT('ANNUAL PROGRESS VECTOR:  ')(COL(10),A);               R0004530
411     3             K=35;                                                           R0004540
412     3             DO I=1 TO 100;                                                  R0004550
413     3    1           IF FK(I,1)¬=0 THEN                                           R0004560
414     3    1           DO; PUT EDIT('*A',I,'=',SUBSTR(AP,FK(I,1),FK(I,2)))          R0004570
                             (COL(K),A,P'999',A,A);                                   R0004580
416     3    2               IF K+FK(I,2)+5>=95 | K=95 THEN K=35; ELSE                R0004590
418     3    2               IF FK(I,2)+5>=15 THEN K=K+30; ELSE K=K+15;               R0004600
421     3    2           END;                                                         R0004610
422     3    1        END;                                                            R0004620
423     3             IF TTFT(2,1)¬='      ' & SUBSTR(TTFT(2,1),1,4)¬='WITH' THEN     R0004630
424     3             DO; PUT EDIT('FINAL RESULT:','TEACHER''S CERTIFICATE',TC,       R0004640
                          'WING SCIENCE',WS,'B.ED.',BE)(COL(10),A,3(COL(25),A,        R0004650
                          COL(50),A));                                                R0004660
426     3    1            PUT EDIT('SUBSEQUENT CAREER:',SUBCAR)(COL(10),A,X(2),A);    R0004670
427     3    1        END;                                                            R0004680
428     3         END;                                                                R0004690
                      /*****************************************************************/R0004700
429     2             IF VPI=3 THEN PUT PAGE;                                         R0004710
431     2             PUT EDIT(DESNO,NAME)(COL(1),A,X(1),A);                          R0004720
432     2             IF VPI=0 THEN                                                   R0004730
433     2             DO; PUT EDIT('RECORDED AS HAVING WITHDRAWN ON',CDATE(TTFT(1,2)), R0004740
                          'REASON CODED AS',SUBSTR(TTFT(2,1),6,1))(COL(10),A,         R0004750
                          3(X(1),A));                                                 R0004760
435     2    1            RETURN;                                                     R0004770
436     2    1        END; ELSE                                                       R0004780
437     2             IF VPI=1 THEN                                                   R0004790
438     2             DO; PUT EDIT('RECORDED AS HAVING CHANGED ')(COL(10),A);         R0004800
440     2    1            IF CSC='C' THEN PUT EDIT('COURSE')(A); ELSE                 R0004810
442     2    1            PUT EDIT('SUBJECT')(A);                                     R0004820
443     2    1            CALL COUPNT;                                                R0004830
444     2    1            RETURN;                                                     R0004840
445     2    1        END; ELSE                                                       R0004850
446     2             IF VPI=2 THEN                                                   R0004860
447     2             DO; PUT EDIT('RECORD UPDATED, ANNUAL PROGRESS VECTOR OR ',      R0004870
                          'FINAL RESULT')(COL(10),A,A);                               R0004880
```

```
STMT LEVEL NEST

449     2    1              CALL APPNT;                                          R0004890
450     2    1              RETURN;                                              R0004900
451     2    1          END; ELSE                                                R0004910
452     2               IF VPI=3 THEN                                            R0004920
453     2               DO; PUT SKIP(2);                                         R0004930
455     2    1              CALL GENPNT;                                         R0004940
456     2    1              PUT SKIP(2);                                         R0004950
457     2    1              CALL SAQPNT;                                         R0004960
458     2    1              PUT SKIP(2);                                         R0004970
459     2    1              CALL COUPNT;                                         R0004980
460     2    1              PUT SKIP(2);                                         R0004990
461     2    1              CALL APPNT;                                          R0005000
462     2    1          END;                                                     R0005010
463     2           END;                                                         R0005020
                    /*****************************************************************/R0005030
464     1           SPNT1:PROCEDURE;                                             R0005040
465     2               DCL (I,N) FIXED BINARY, COD CHAR(3), B BIT(1);           R0005050
466     2               N=KT;                                                    R0005060
467     2           L1: B='0'B;                                                  R0005070
468     2               N=N-1;                                                   R0005080
469     2               DO I=1 TO N;                                             R0005090
470     2    1             IF DHT(I) || TYPE(I) || NAML(I)>DHT(I+1) || TYPE(I+1) ||  R0005100
                           NAML(I+1) THEN                                       R0005110
471     2    1             DO; RECL(KT+1)=RECL(I);                              R0005120
473     2    2                 RECL(I)=RECL(I+1);                               R0005130
474     2    2                 RECL(I+1)=RECL(KT+1);                            R0005140
475     2    2                 B='1'B;                                          R0005150
476     2    2             END;                                                 R0005160
477     2    1          END;                                                    R0005170
478     2               IF B THEN GOTO L1;                                       R0005180
480     2               COD=DHT(1); B='0'B;                                      R0005190
482     2               IF COD='   ' THEN                                        R0005200
483     2               DO; PUT EDIT('NO MAIN COURSE RECORDED','DATE OF LISTING',DT,  R0005210
                            'TYPE','D.E.S.NO','NAME')(PAGE,COL(1),A,SKIP(3),COL(1),A,X(1),R0005220
                            A,SKIP(3),COL(1),A,COL(7),A,COL(17),A);             R0005230
485     2    1              PUT SKIP;                                            R0005240
486     2    1              B='1'B;                                              R0005250
487     2    1          END;                                                     R0005260
488     2               DO I=1 TO KT;                                            R0005270
489     2    1      L2:     IF ¬B THEN                                          R0005280
490     2    1              DO; PUT EDIT('COURSE CODE',COD,'DATE OF LISTING',DT,'TYPE',  R0005290
                                'D.E.S.NO','NAME')(PAGE,COL(1),A,X(1),A,SKIP(3),COL(1),A,  R0005300
                                X(1),A,SKIP(3),COL(1),A,COL(7),A,COL(17),A);    R0005310
492     2    2                 PUT SKIP;                                        R0005320
493     2    2                 B='1'B;                                          R0005330
494     2    2              END;                                                 R0005340
495     2    1              IF DHT(I)¬=COD THEN                                  R0005350
496     2    1              DO; COD=DHT(I);                                      R0005360
498     2    2                 B='0'B;                                          R0005370
499     2    2                 GOTO L2;                                         R0005380
500     2    2              END;                                                 R0005390
501     2    1              IF TYPE(I)='P' THEN PUT EDIT('PGCE')(SKIP,COL(1),A); ELSE  R0005400
503     2    1              PUT EDIT(TYPE(I))(SKIP,COL(4),A);                    R0005410
504     2    1              PUT EDIT(DESL(I),NAML(I))(COL(7),A,COL(17),A);       R0005420
```

```
STMT LEVEL NEST

505     2    1          END;                                                    R0005430
506     2          END;                                                         R0005440
                   /*****************************************************************/R0005450
507     1          DESPNT:PROCEDURE;                                            R0005460
508     2               DCL X2 CHAR(2), (I,N) FIXED BINARY, SW BIT(1);          R0005470
509     2               ON ENDPAGE(SPRINT)                                      R0005480
510     2               BEGIN; PUT EDIT('.....BEDF COLLEGE  DURHAM',YE,'.....','TYPE',  R0005490
                              'D.E.S.NO','NAME','DATE OF BIRTH','COURSE___DATES',  R0005500
                              'DEGREES HELD')(PAGE,COL(38),A,X(2),A,A,SKIP(2),COL(1),A,  R0005510
                              COL(7),A,COL(17),A,COL(56),A,CCL(71),A,COL(88),A);  R0005520
512     3                     PUT SKIP(2);                                      R0005530
513     3               END;                                                    R0005540
514     2               IF KT=0 THEN GOTO L21;                                  R0005550
516     2               N=KT;                                                   R0005560
517     2          L20: SW='0'B;                                                R0005570
518     2               N=N-1;                                                  R0005580
519     2               DO I=1 TO N:                                            R0005590
520     2    1             IF TYPE(I) || NAML(I)>TYPE(I+1) || NAML(I+1) THEN   R0005600
521     2    1             DO; RECL(KT+1)=RECL(I);                             R0005610
523     2    2                 RECL(I)=RECL(I+1);                              R0005620
524     2    2                 RECL(I+1)=RECL(KT+1);                           R0005630
525     2    2                 SW='1'B;                                        R0005640
526     2    2             END;                                                R0005650
527     2    1          END;                                                    R0005660
528     2               IF SW THEN GOTO L20;                                    R0005670
530     2               N=0;                                                    R0005680
531     2               DO I=1 TO KT;                                           R0005690
532     2    1             IF TYPE(I)='P' THEN                                 R0005700
533     2    1             DO; N=N+1;                                          R0005710
535     2    2                 IF SUBSTR(DHB(I),1,1)='F' THEN SUBSTR(DHB(I),1,1)=' ';  R0005720
537     2    2             END; ELSE GOTO L2;                                  R0005730
539     2    1          END;                                                    R0005740
540     2          L2:  SW='0'B;                                                R0005750
541     2               N=N-1;                                                  R0005760
542     2               DO I=1 TO N;                                            R0005770
543     2    1             IF SUBSTR(DHB(I),1,1) || NAML(I) > SUBSTR(DHB(I+1),1,1) ||  R0005780
                           NAML(I+1) THEN                                      R0005790
544     2    1             DO; RECL(KT+1)=RECL(I);                             R0005800
546     2    2                 RECL(I)=RECL(I+1);                              R0005810
547     2    2                 RECL(I+1)=RECL(KT+1);                           R0005820
548     2    2                 SW='1'B;                                        R0005830
549     2    2             END;                                                R0005840
550     2    1          END;                                                    R0005850
551     2               IF SW THEN GOTO L2;                                     R0005860
553     2               X2=TYPE(1) || SUBSTR(DHB(1),1,1);                       R0005870
554     2               IF X2='P ' THEN                                         R0005880
555     2               DO; SIGNAL ENDPAGE(SPRINT);                             R0005890
557     2    1              PUT EDIT('PGCE STUDENT(S): FOREIGN DEGREES OR EQUIVALENT')  R0005900
                            (SKIP(2),COL(1),A);                                R0005910
558     2    1              PUT SKIP(2);                                       R0005920
559     2    1          END; ELSE SIGNAL ENDPAGE(SPRINT);                       R0005930
561     2               DO I=1 TO KT;                                           R0005940
562     2    1             IF TYPE(I) || SUBSTR(DHB(I),1,1)¬=X2 THEN           R0005950
563     2    1             DO; SIGNAL ENDPAGE(SPRINT);                         R0005960
```

```
STMT LEVEL NEST

565     2    2                X2=TYPE(I) || SUBSTR(DHB(I),1,1);                          R0005970
566     2    2            END;                                                            R0005980
567     2    1            IF TYPE(I)='P' THEN PUT EDIT('PGCE')(SKIP,COL(1),A); ELSE       R0005990
569     2    1            PUT EDIT(TYPE(I))(SKIP,COL(4),A);                               R0006000
570     2    1            PUT EDIT(DESL(I),NAML(I),SUBSTR(COBL(I),1,2),'/',SUBSTR(        R0006010
                          DOBL(I),3,2),'/',SUBSTR(DOBL(I),5,2),START(I),'TO',FIN(I))      R0006020
                          (COL(7),A,COL(17),A,COL(61),5(A),COL(71),A(5),X(1),A,X(1),A(5)  R0006030
                          );                                                              R0006040
571     2    1            IF DHB(I)¬='    ' THEN                                          R0006050
572     2    1            PUT EDIT(DHT(I),'19' || DHY(I),SUBSTR(DHB(I),2))                R0006060
                          (COL(88),A,2(X(2),A));                                          R0006070
573     2    1        END;                                                                R0006080
574     2        L21: PUT EDIT('NO. OF STUDENTS',KT)(SKIP(2),COL(1),A,X(2),F(3));        R0006090
575     2             ON ENDPAGE(SPRINT) SYSTEM;                                         R0006100
576     2        END;                                                                    R0006110
                 /*****************************************************************/R0006120
577     1             ON CONVERSION                                                      R0006130
578     1             BEGIN; PUT EDIT('********DATA CONVERSION ERROR********',            R0006140
                            'CURRENT D.E.S.NO.',DESNO)(SKIP(2),A,X(1),A,X(1),A);         R0006150
580     2             END;                                                               R0006160
581     1             ON ENDFILE(SCARDS) GOTO L09;                                       R0006170
583     1        L39: GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));         R0006180
584     1             GET LIST(OPT);                                                     R0006190
585     1             IF OPT<1 | OPT>10 THEN                                             R0006200
586     1             DO; PUT EDIT('OPTION INVALID.....PROGRAM TERMINATED')              R0006210
                          (SKIP,COL(1),A);                                               R0006220
588     1    1            GOTO L09;                                                      R0006230
589     1    1        END;                                                               R0006240
590     1             TRANS='CO'; B='0'B; PI='0'; YT='0'; NTT='0'; VP='0';               R0006250
596     1             DES='         '; BMAX='0'B;                                        R0006260
598     1             GET DATA;                                                          R0006270
599     1             IF DES¬='         ' THEN                                           R0006280
600     1             DO; DESB='1'B;                                                     R0006290
602     1    1            IF ¬DESOP THEN                                                 R0006300
603     1    1            DO; OPEN FILE(DIC) OUTPUT TITLE(DES);                          R0006310
605     1    2                DESOP='1'B;                                                R0006320
606     1    2            END;                                                           R0006330
607     1    1        END; ELSE DESB='0'B;                                               R0006340
609     1             IF VP='1' THEN VPB='1'B; ELSE VPB='0'B;                            R0006350
612     1             IF NTT='1' THEN NT=1; ELSE IF NTT='2' THEN NT=2; ELSE NT=1;        R0006360
617     1             IF YT='F' THEN YR=1; ELSE IF YT='L' THEN YR=2; ELSE YR=1;          R0006370
622     1             PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME R',       R0006380
                      'OPTION REQUESTED',OPT,'YEAR REQUESTED',YE)(PAGE,COL(1),A,         R0006390
                      SKIP(2),COL(1),A,COL(1),A,COL(32),F(2),COL(1),A,COL(30),A);        R0006400
623     1             PUT SKIP(2);                                                       R0006410
624     1             IF DES¬='         ' THEN PUT EDIT('DES O/P REQUIRED')(COL(1),A);   R0006420
626     1             IF PI¬='0' THEN PUT EDIT('PRINT INDICATOR',PI)                     R0006430
                      (COL(1),A,COL(33),A);                                              R0006440
628     1             IF YT¬='0' THEN PUT EDIT('YEAR OF TEST',YT)(COL(1),A,COL(33),A);   R0006450
630     1             IF NTT¬='0' THEN PUT EDIT('NORM TABLE REQUIRED',NTT)               R0006460
                      (COL(1),A,COL(33),A(1));                                           R0006470
632     1             IF VP¬='0' THEN PUT EDIT('VERIFICATION PRINT REQUIRED')(COL(1),A);R0006480
634     1             PUT SKIP(4);                                                       R0006490
635     1             IF BCON & (OPT=2 | (OPT=6 & PI='1') | (OPT=9 & VPB)) THEN          R0006500
```

```
STMT LEVEL NEST

636     1                DO; CALL CON(FILN2);                                          R0006510
638     1     1              BCON='0'B;                                                R0006520
639     1     1          END;                                                          R0006530
640     1                GOTO L(OPT);                                                  R0006540
                    /*-----------------------------------------------------------------*/R0006550
641     1           L1:L7:OPEN FILE(STUDENT) OUTPUT TITLE(FILN1);                      R0006560
642     1                ALLOCATE RECORD;                                              R0006570
643     1           L11: GENERAL=' '; EDUCATION=' '; COURSE=' '; AP=' '; INTEL=0;      R0006580
648     1                INTELB='  '; LEAP=1; FK=0;                                    R0006590
651     1                GET EDIT(DESNO)(COL(1),A(8));                                 R0006600
652     1                IF DESNO='********' THEN                                      R0006610
653     1                DO; CLOSE FILE(STUDENT);                                      R0006620
655     1     1              FREE RECORD;                                              R0006630
656     1     1              GOTO L39;                                                 R0006640
657     1     1          END;                                                          R0006650
658     1                CALL RECIN;                                                   R0006660
659     1                CALL CHECKREC;                                                R0006670
660     1                IF OPT=7 & DESB THEN PUT FILE(DIC) EDIT(DESNO || '15')(COL(1),A); R0006680
662     1                WRITE FILE(STUDENT) FROM(RECORD);                             R0006690
663     1                GOTO L11;                                                     R0006700
                    /*-----------------------------------------------------------------*/R0006710
664     1           L2:L3:L4:L8:L9:OPEN FILE(STUDENT) UPDATE TITLE(FILN1);             R0006720
665     1                ON ENDFILE(STUDENT)                                           R0006730
666     1                BEGIN; CLOSE FILE(STUDENT);                                   R0006740
668     2                       OPEN FILE(STUDENT) UPDATE TITLE(FILN1);                R0006750
669     2                       GOTO L41;                                              R0006760
670     2                END;                                                          R0006770
671     1           L42: GET EDIT(DESNO2)(COL(1),A(8));                                R0006780
672     1                IF DESNO2='********' THEN                                     R0006790
673     1                DO; CLOSE FILE(STUDENT);                                      R0006800
675     1     1              GOTO L39;                                                 R0006810
676     1     1          END;                                                          R0006820
677     1           L41: READ FILE(STUDENT) SET(PTR);                                  R0006830
678     1                IF DESNO=DESNO2 THEN                                          R0006840
679     1                DO; IF B THEN B='0'B;                                         R0006850
682     1     1              IF OPT=2 THEN CALL ITRECIN; ELSE                          R0006860
684     1     1              IF OPT=3 THEN                                             R0006870
685     1     1              DO; CALL UPDATE;                                          R0006880
687     1     2                  IF VPB THEN                                           R0006890
688     1     2                  DO; VPI=2;                                            R0006900
690     1     3                      CALL PRINT;                                       R0006910
691     1     3                  END;                                                  R0006920
692     1     2                  IF DESB & SUBSTR(TTFT(2,1),1,4)='LEFT' THEN           R0006930
693     1     2                  PUT FILE(DIC) EDIT(DESNO || '99')(COL(1),A);          R0006940
694     1     2              END; ELSE                                                 R0006950
695     1     1              IF OPT=4 THEN                                             R0006960
696     1     1              DO; GET EDIT(TTFT(1,2),WC)(COL(10),A(6),X(1),A(1));       R0006970
698     1     2                  TTFT(2,1)='WITH ' || WC;                              R0006980
699     1     2                  IF VPB THEN                                           R0006990
700     1     2                  DO; VPI=0;                                            R0007000
702     1     3                      CALL PRINT;                                       R0007010
703     1     3                  END;                                                  R0007020
704     1     2                  IF DESB THEN PUT FILE(DIC) EDIT(DESNO || '12')(COL(1),A); R0007030
706     1     2              END; ELSE                                                 R0007040
```

```
STMT LEVEL NEST

707     1    1            IF OPT=8 THEN                                                R0007050
708     1    1            DO; I=LE;                                   /* CONVERSION */R0007060
710     1    2                IF LE¬='4' THEN LE=SUBSTR('1234',I+1,1);                 R0007070
712     1    2                I=LE;                                   /* CONVERSION */R0007080
713     1    2                GET EDIT(CSC,DC(I,24),DC(I,25),DC1(I,26),DC1(I,27),      R0007090
                              DC1(I,29),DC1(I,30),DC(I,32),DC(I,34),DC(I,35),DC(I,36), R0007100
                              DC(I,37),DC(I,38),DC(I,41),DC(I,42),DC(I,43),DC(I,44),   R0007110
                              DC(I,45),DC46(I),DC49(I),DC(I,59),DC(I,64))(COL(10),A(1), R0007120
                              COL(52),2(A(1)),4(A(2)),11(A(1)),2(A(3)),2(A(1)));       R0007130
714     1    2                CODES(1)=DC46(I);                                        R0007140
715     1    2                CODES(2)=DC49(I);                                        R0007150
716     1    2                IF VPB THEN                                              R0007160
717     1    2                DO; VPI=1;                                               R0007170
719     1    3                    CALL PRINT;                                          R0007180
720     1    3                END;                                                     R0007190
721     1    2                IF DESB THEN                                             R0007200
722     1    2                PUT FILE(DIC) EDIT(DESNO || '11')(COL(1),A);             R0007210
723     1    2            END; ELSE                                                    R0007220
724     1    1            IF OPT=9 THEN                                                R0007230
725     1    1            DO; GET DATA;                                                R0007240
727     1    2                IF VPB THEN                                              R0007250
728     1    2                DO; VPI=3;                                               R0007260
730     1    3                    CALL PRINT;                                          R0007270
731     1    3                END;                                                     R0007280
732     1    2            END;                                                         R0007290
733     1    1                REWRITE FILE(STUDENT) FROM(RECORD);                      R0007300
734     1    1                GOTO L42;                                                R0007310
735     1    1            END; ELSE                                                    R0007320
736     1                 DO; IF ¬B THEN                                               R0007330
738     1    1                DO; DESNO3=DESNO;                                        R0007340
740     1    2                    B='1'B;                                              R0007350
741     1    2                END; ELSE                                                R0007360
742     1    1                DO; IF DESNO=DESNO3 THEN                                 R0007370
744     1    2                    DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')  R0007380
                                      (COL(1),A,X(1),A);                               R0007390
746     1    3                        B='0'B;                                          R0007400
747     1    3                        GOTO L42;                                        R0007410
748     1    3                    END;                                                 R0007420
749     1    2                END;                                                     R0007430
750     1    1                GOTO L41;                                                R0007440
751     1    1            END;                                                         R0007450
                     /*-----------------------------------------------------------------*/R0007460
752     1            L5:L6:L10:OPEN FILE(STUDENT) INPUT TITLE(FILN1);                 R0007470
753     1                 ON ENDFILE(STUDENT)                                          R0007480
754     1                 BEGIN; DT=DATE;                                              R0007490
756     2                        DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||  R0007500
                                 SUBSTR(DT,7,2);                                      R0007510
757     2                        IF OPT=5 THEN CALL DESPNT; ELSE                      R0007520
759     2                        IF OPT=6 THEN                                        R0007530
760     2                        DO; IF PI='2' THEN                                   R0007540
762     2    1                       DO; PUT EDIT(YE,'FILE.....',KT,'STUDENTS INCLUDED',  R0007550
                                         '.....NUMBERS FOLLOWED BY ''*'' SHOW THAT THE ',  R0007560
                                         'STUDENT HAS WITHDRAWN FROM THE COLLEGE','NUMBERS',R0007570
                                         ' FOLLOWED BY ''+'' SHOW THAT THE STUDENT HAS',  R0007580
```

```
STMT LEVEL NEST

                                    ' COMPLETED HIS COURSE AND LEFT')                R0007590
                                    (PAGE,COL(1),A,X(1),A,F(3),X(1),3(A),            R0007600
                                    SKIP,COL(41),3(A));                              R0007610
764     2     2                     IF KT¬=0 THEN CALL SPNT; ELSE GOTO L38;          R0007620
767     2     2                 END; ELSE                                            R0007630
768     2     1                 IF PI='3' THEN                                       R0007640
769     2     1                 DO; PUT EDIT(YE,'FILE.....',KT,'STUDENTS INCLUDED')  R0007650
                                    (PAGE,COL(1),A(4),X(1),A,F(3),X(1),A);           R0007660
771     2     2                     IF KT¬=0 THEN CALL SPNT; ELSE GOTO L38;          R0007670
774     2     2                 END; ELSE                                            R0007680
775     2     1                 IF PI='4' THEN                                       R0007690
776     2     1                 DO; PUT EDIT(YE,'FILE.....',KT,'STUDENTS INCLUDED')  R0007700
                                    (PAGE,COL(1),A(4),X(1),A,F(3),X(1),A);           R0007710
778     2     2                     IF KT¬=0 THEN CALL SPNT1;                        R0007720
780     2     2                 END;                                                 R0007730
781     2     1             END;                                                     R0007740
782     2         L38:      PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(2),A); R0007750
783     2                   IF BMAX THEN                                             R0007760
784     2                   DO; BMAX='0'B;                                           R0007770
786     2     1                 RECL=' '; KT=0;                                      R0007780
788     2     1                 GOTO L78;                                            R0007790
789     2     1             END;                                                     R0007800
790     2                   CLOSE FILE(STUDENT);                                     R0007810
791     2                   GOTO L39;                                                R0007820
792     2               END;                                                         R0007830
793     1               IF OPT=6 & PI='1' THEN VPI=3;                                R0007840
795     1               IF OPT=5 | OPT=6 THEN                                        R0007850
796     1               DO; RECL=' ';                                                R0007860
798     1     1             KT=0;                                                    R0007870
799     1     1         END;                                                         R0007880
800     1         L12:  READ FILE(STUDENT) SET(PTR);                                 R0007890
801     1               IF OPT=5 | OPT=10 THEN                                       R0007900
802     1               DO; IF SUBSTR(DESNO,1,2)¬=SUBSTR(TTFT(1,1),5,2) THEN         R0007910
804     1     1             DO; PUT EDIT(DESNO,'ADMITTED IN YEAR','19' ||            R0007920
                                SUBSTR(DESNO,1,2))(SKIP,COL(1),A,X(1),A,X(2),A);     R0007930
806     1     2                 GOTO L12;                                            R0007940
807     1     2             END;                                                     R0007950
808     1     1         END;                                                         R0007960
809     1         L78:  IF OPT=5 THEN                                                R0007970
810     1               DO; IF KT=499 THEN                                           R0007980
812     1     1             DO; PUT EDIT('********WARNING******** MAXIMUM SIZE REACHED', R0007990
                                '...FILE DIVIDED')(COL(1),A,A);                      R0008000
814     1     2                 BMAX='1'B;                                           R0008010
815     1     2                 SIGNAL ENDFILE(STUDENT);                             R0008020
816     1     2             END;                                                     R0008030
817     1     1             KT=KT+1;                                                 R0008040
818     1     1             DESL(KT)=DESNO;                                          R0008050
819     1     1             NAML(KT)=NAME;                                           R0008060
820     1     1             DOBL(KT)=DOB;                                            R0008070
821     1     1             START(KT)=DC1(1,26) || '/' || DC1(1,27);                 R0008080
822     1     1             FIN(KT)=DC1(1,29) || '/' || DC1(1,30);                   R0008090
823     1     1             IF DC(1,59)='1' THEN                                     R0008100
824     1     1             DO; TYPE(KT)='P';                                        R0008110
826     1     2                 IF UNOTH(1)¬=' ' THEN                                R0008120
```

```
STMT LEVEL NEST

827     1    2                DO; DHT(KT)=DEG(1);                                    R0008130
829     1    3                    DHY(KT)=SUBSTR(DATT(1),5,2);                       R0008140
830     1    3                    DHB(KT)=UNOTH(1);                                  R0008150
831     1    3                END; ELSE                                              R0008160
832     1    2                IF UNCOL(1)¬=' ' THEN                                  R0008170
833     1    2                DO; DHY(KT)=SUBSTR(FT(1,2),5,2);                       R0008180
835     1    3                    DHB(KT)=' ' || UNCOL(1);                           R0008190
836     1    3                END;                                                   R0008200
837     1    2            END; ELSE                                                  R0008210
838     1    1            IF LEN='1' | LEN='2' | LEN='3' THEN TYPE(KT)=LEN; ELSE     R0008220
840     1    1            DO; PUT EDIT(DESNO,'TYPE WARNING.....SET TO 3')            R0008230
                              (SKIP,COL(1),A,X(1),A);                                R0008240
842     1    2                TYPE(KT)='3';                                          R0008250
843     1    2            END;                                                       R0008260
844     1    1        END; ELSE                                                      R0008270
845     1             IF OPT=6 THEN                                                  R0008280
846     1             DO; IF PI='1' THEN CALL PRINT; ELSE                            R0008290
849     1    1            IF PI='2' | PI='3' | (PI='4' & TTFT(1,2)='        ') THEN  R0008300
850     1    1            DO; IF KT=499 THEN                                         R0008310
852     1    2                DO; PUT EDIT('********WARNING******** MAXIMUM SIZE REAC', R0008320
                                  'HED...FILE DIVIDED')(COL(1),A,A);                 R0008330
854     1    3                    BMAX='1'B;                                         R0008340
855     1    3                    SIGNAL ENDFILE(STUDENT);                           R0008350
856     1    3                END;                                                   R0008360
857     1    2                KT=KT+1;                                               R0008370
858     1    2                DESL(KT)=DESNO;                                        R0008380
859     1    2                IF TTFT(1,2)¬='       ' THEN                           R0008390
860     1    2                FIN(KT)=TTFT(1,2);                                     R0008400
861     1    2                IF TTFT(2,1)¬='       ' THEN                           R0008410
862     1    2                IF SUBSTR(TTFT(2,1),1,4)='WITH' THEN                   R0008420
863     1    2                SUBSTR(FIN(KT),7,1)='*'; ELSE SUBSTR(FIN(KT),7,1)='+'; R0008430
865     1    2                IF PI='3' THEN                                         R0008440
866     1    2                DO; NAML(KT)=NAME;                                     R0008450
868     1    3                    DOBL(KT)=DOB;                                      R0008460
869     1    3                    START(KT)=TTFT(1,1);                               R0008470
870     1    3                END; ELSE                                              R0008480
871     1    2                IF PI='4' THEN                                         R0008490
872     1    2                DO; NAML(KT)=NAME;                                     R0008500
874     1    3                    DHT(KT)=CODES(1);                                  R0008510
875     1    3                    I=LE;                                  /* CONVERSION */R0008520
876     1    3                    IF DC(I,59)='1' THEN TYPE(KT)='P'; ELSE            R0008530
878     1    3                    IF LEN>='1' & LEN<='4' THEN TYPE(KT)=LEN; ELSE     R0008540
880     1    3                    DO; PUT EDIT(DESNO,'TYPE WARNING.....SET TO 3')    R0008550
                                      (COL(1),A,X(1),A);                             R0008560
882     1    4                        TYPE(KT)='3';                                  R0008570
883     1    4                    END;                                               R0008580
884     1    3                END;                                                   R0008590
885     1    2            END;                                                       R0008600
886     1    1        END; ELSE                                                      R0008610
887     1             IF OPT=10 & DESB THEN PUT FILE(DIC) EDIT(DESNO || '01')(COL(1),A);R0008620
889     1             GOTO L12;                                                      R0008630
                 /*-----------------------------------------------------------------*/R0008640
890     1        L09: IF DESOP THEN CLOSE FILE(DIC);                                 R0008650
892     1             PUT PAGE;                                                      R0008660
```

```
STMT LEVEL NEST

893     1          END;                                                    R0008670
                                                                           R0008680
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
2	ADIP	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER 70,398
2	ADIPC	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 70,397,398
2 *********	AH4	(2,3)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0) 117,125,125
2	ANNPROG	IN RECORD,BASED(PTR),STRUCTURE
2	AP	IN ANNPROG IN RECORD,BASED(PTR),UNALIGNED,STRING(802),CHARACTER 180,180,199,415,646
409	APPNT	ENTRY,DECIMAL,FLOAT(SINGLE) 449,461
2	ATO	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 17,406
2	AUTH	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(19),CHARACTER 13,358,359
2	AUTHNO	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 12,111,111,359
465	B	AUTOMATIC,UNALIGNED,STRING(1),BIT 467,475,478,481,486,489,493,498
5	B	AUTOMATIC,UNALIGNED,STRING(1),BIT 591,680,681,737,740,746
5	BCON	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 635,638
2	BE	IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 208,425
5	BMAX	AUTOMATIC,UNALIGNED,STRING(1),BIT 597,783,785,814,854
263	CDATE	ENTRY,STRING(8),CHARACTER 279,288,291,390,390,390,390,395,395,434
76	CHECKREC	ENTRY,DECIMAL,FLOAT(SINGLE) 659
465	COD	AUTOMATIC,UNALIGNED,STRING(3),CHARACTER 480,482,491,495,497
2	CODES	(4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		13,13,363,364,365,366,714,715,874
7	CON	EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE) 637
2	CONF	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 12,346
11	CONT	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 14,15,53,54,61,61,63,68
361	COUPNT	ENTRY,DECIMAL,FLOAT(SINGLE) 443,459
2	COURSE	IN RECORD,BASED(PTR),STRUCTURE 645
5	CSC	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 440,713
	DATE	BUILT-IN FUNCTION 755
2	DATT	(2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER 65,395,395,829
2	DC	(4,24:64)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1), CHARACTER 13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,370,370,370,370,370,370 370,370,370,370,370,370,370,370,370,713,713,713,713,713,713,713,713 713,713,713,713,713,713,713,823,876
2	DC1	(4,26:30)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(2), CHARACTER 13,13,13,13,109,109,370,370,370,370,713,713,713,713,821,821,822,822
2	DC46	(4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER 13,370,713,714
2	DC49	(4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER 13,370,713,715
2	DDIPC	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 70,399,400
2	DEG	(2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER 65,395,395,828
5	DES	AUTOMATIC,UNALIGNED,STRING(8),CHARACTER 596,599,604,624
5	DESB	AUTOMATIC,UNALIGNED,STRING(1),BIT 601,608,660,692,704,721,887

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

4                DESL                          IN RECL(500),AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               219,254,504,570,818,858

2                DESNO                         IN RECORD,BASED(PTR),UNALIGNED,STRING(8),CHARACTER
                                               24,27,39,44,47,90,92,94,96,98,100,102,104,106,108,110,112,120,123
                                               126,131,157,161,192,431,579,651,652,661,678,693,705,722,739,743,803
                                               805,805,818,841,858,881,888

5                DESNO2                        AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               671,672,678,745

5                DESNO3                        AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               739,743

7                DESOP                         AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT
                                               602,605,890

507              DESPNT                        ENTRY,DECIMAL,FLOAT(SINGLE)
                                               758

4                DHB                           IN RECL(500),AUTOMATIC,UNALIGNED,STRING(21),CHARACTER
                                               535,536,543,543,553,562,565,571,572,830,835

4                DHT                           IN RECL(500),AUTOMATIC,UNALIGNED,STRING(3),CHARACTER
                                               470,470,480,495,497,572,828,874

4                DHY                           IN RECL(500),AUTOMATIC,UNALIGNED,STRING(2),CHARACTER
                                               572,829,834

9                DIC                           FILE,EXTERNAL
                                               604,661,693,705,722,888,891

2                DOB                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               12,83,278,279,820,868

2                DOBC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,303

4                DOBL                          IN RECL(500),AUTOMATIC,UNALIGNED,STRING(6),CHARACTER
                                               254,254,254,570,570,570,820,868

2       ********* DSCALE                       (2)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               117,122,122

5                DT                            AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               484,491,755,756,756,756,756,782

2                EDUCATION                     IN RECORD,BASED(PTR),STRUCTURE
                                               644

5                FILN1                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               583,641,664,668,752

5                FILN2                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               583,637

4                FIN                           IN RECL(500),AUTOMATIC,UNALIGNED,STRING(7),CHARACTER
                                               220,221,255,256,256,256,570,822,860,863,864

2                FINALS                        IN RECORD,BASED(PTR),STRUCTURE

2      ********* FK                            (100,2)IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED
                                               (15,0)
                                               178,180,180,180,181,182,183,183,184,184,184,186,200,202,413,415,415
                                               416,418,650

2                FPOS                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER
                                               12,107,107,305,306,306

2                FPT                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               57,390,390

2                FT                            (2,2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               57,57,390,390,390,390,834

2                GENERAL                       IN RECORD,BASED(PTR),STRUCTURE
                                               643

270              GENPNT                        ENTRY,DECIMAL,FLOAT(SINGLE)
                                               455

3                GRC                           STATIC,EXTERNAL,UNALIGNED,STRING(11),CHARACTER,VARYING
                                               315

3                GRK                           (11)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               317

2                HPCD                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER
                                               65,395,395

508    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               519,520,520,520,520,522,523,523,524,531,532,535,536,542,543,543,543
                                               543,545,546,546,547,561,562,562,565,565,567,569,570,570,570,570,570
                                               570,570,571,572,572,572

465    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               469,470,470,470,470,470,470,472,473,473,474,488,495,497,501,503,504
                                               504

262    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               300,301,309,312,313,315,317,322,325,326,332,335,336,340,343,344,353
                                               354,355,356,369,370,370,370,370,370,370,370,370,370,370,370,370,370
                                               370,370,370,370,370,370,370,370,376,377,382,383,384,412,413,415,415
                                               415,416,418

212    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               216,217,219,220,221,240,241,241,243,244,244,245,253,254,254,254,254
                                               254,254,254,254,255,256,256,256
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

154    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               167,168,170,170,171,171,172,174,175,175,176,176

116    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               137,139,139,141,144,144,144,144,146

5      ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               709,711,712,713,713,713,713,713,713,713,713,713,713,713,713,713,713
                                               713,713,713,713,713,713,713,714,715,875,876

                 INDEX                         GENERIC,BUILT-IN FUNCTION
                                               170,171,175,315

154              INST                          AUTOMATIC,UNALIGNED,STRING(70),CHARACTER
                                               166,168,170,171,175,176,194,199,205

2                INTEL                         IN RECORD,BASED(PTR),STRUCTURE
                                               647

2                INTELB                        IN INTEL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               151,648

114    ********* ITRECIN                       ENTRY,BINARY,FIXED(15,0)
                                               683

154    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               182,183,183,184,184,193

116    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               138,141,144,144,146

262    ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               368,369,375,376,380,382,411,415,416,416,417,419,419,420,420

154    ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               177,178,180,180,180,181,183,184,186,200,202

77     ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               82,85

11     ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               19,20,25,37,38,39,45,56,57,57,57,57,63,64,64,65,65,65,65,65

       ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               118,119,119,120,124,125,125,126,128,129,129,129,131

5      ********* KT                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               216,237,243,245,253,466,472,474,488,514,516,522,524,531,545,547,561
                                               574,763,764,770,771,777,778,787,798,811,817,817,818,819,820,821,822
                                               825,828,829,830,834,835,839,842,851,857,857,858,860,863,864,867,868
                                               869,873,874,877,879,882

6                L                             (10)AUTOMATIC,INITIAL,LABEL
```

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		640
890	L09	STATEMENT LABEL CONSTANT 582,588
38	L1	STATEMENT LABEL CONSTANT 40
150	L1	STATEMENT LABEL CONSTANT 142,147
166	L1	STATEMENT LABEL CONSTANT 206
238	L1	STATEMENT LABEL CONSTANT 250
467	L1	STATEMENT LABEL CONSTANT 479
641	L1	STATEMENT LABEL CONSTANT
752	L10	STATEMENT LABEL CONSTANT
643	L11	STATEMENT LABEL CONSTANT 663
800	L12	STATEMENT LABEL CONSTANT 806,889
14	L2	STATEMENT LABEL CONSTANT 59,66,71
151	L2	STATEMENT LABEL CONSTANT 136
208	L2	STATEMENT LABEL CONSTANT 165
489	L2	STATEMENT LABEL CONSTANT 499
540	L2	STATEMENT LABEL CONSTANT 538,552
664	L2	STATEMENT LABEL CONSTANT
517	L20	STATEMENT LABEL CONSTANT 529
574	L21	STATEMENT LABEL CONSTANT 515
26	L222	STATEMENT LABEL CONSTANT

```
DCL NO.          IDENTIFIER                ATTRIBUTES AND REFERENCES

                                           39

46               L223                      STATEMENT LABEL CONSTANT
                                           48

204              L3                        STATEMENT LABEL CONSTANT
                                           173,189

664              L3                        STATEMENT LABEL CONSTANT

782              L38                       STATEMENT LABEL CONSTANT
                                           766,773

583              L39                       STATEMENT LABEL CONSTANT
                                           656,675,791

664              L4                        STATEMENT LABEL CONSTANT

677              L41                       STATEMENT LABEL CONSTANT
                                           669,750

671              L42                       STATEMENT LABEL CONSTANT
                                           734,747

752              L5                        STATEMENT LABEL CONSTANT

752              L6                        STATEMENT LABEL CONSTANT

641              L7                        STATEMENT LABEL CONSTANT

809              L78                       STATEMENT LABEL CONSTANT
                                           788

664              L8                        STATEMENT LABEL CONSTANT

664              L9                        STATEMENT LABEL CONSTANT

2                LE                        IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                           74,368,709,710,711,712,875

2      ********* LEAP                      IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                           181,181,190,199,200,201,201,649

2                LEN                       IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                           13,156,162,362,838,838,838,839,878,878,879

2                MART                      IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                           12,93,93,298,298,300

                 MIN                       GENERIC,BUILT-IN FUNCTION
                                           19,37

3                MK                        (3)STATIC,EXTERNAL,UNALIGNED,STRING(12),CHARACTER,VARYING
                                           301
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

115    ********* MNT                           (16)AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0)
                                               129

                 MOD                           GENERIC,BUILT-IN FUNCTION
                                               34,217,384

508    ********* N                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               516,518,518,519,530,534,534,541,541,542

465    ********* N                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               466,468,468,469

212    ********* N                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               237,239,239,240

2                NAME                          IN RECORD,BASED(PTR),UNALIGNED,STRING(40),CHARACTER
                                               12,431,819,867,873

4                NAML                          IN RECL(500),AUTOMATIC,UNALIGNED,STRING(40),CHARACTER
                                               241,241,254,470,470,504,520,520,543,543,570,819,867,873

3      ********* NORM                          (2,16,13)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0)
                                               139,144,144

3                NORM2                         (2,16,2)STATIC,EXTERNAL,ALIGNED,DECIMAL,FLOAT(SINGLE)

5      ********* NT                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               139,144,144,613,615,616

5                NTT                           AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               594,612,614,630,631

3                OAK                           (4)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                               312

3                OK                            (10)STATIC,EXTERNAL,UNALIGNED,STRING(35),CHARACTER,VARYING
                                               356

271              OKB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               273,274,307,307,311,316,318,320,320,324,330,330,334,338,338,342

77               OKB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               79,80,95,95,97,97,99,101,101,103,103,111

3                OPSOK                         (6)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                               325,335

5      ********* OPT                           AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               584,585,585,622,635,635,635,640,660,682,684,695,707,724,757,759,793
                                               795,795,801,801,809,845,887

2                OQ                            (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER
                                               70,70,401,401,404,404
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

2                OQC                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER
                                               70,70,404,404

3                OPK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               343

2                OV25                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,86,87,88,105,105,105,349,351,351,353

212              PB                            AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT
                                               232,252

2      ********* PF16                          (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               117,129,129,139,144,144

5                PI                            AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               213,228,592,626,627,635,761,768,775,793,847,849,849,849,865,871

2                PNAME                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(17),CHARACTER
                                               13,296,297

2                POCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,101,101,101,330,330,330,332

154    ********* POSEND                        AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               171,174,175,188,190,199,199,201,202

154    ********* POSEQ                         AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               170,172,176,188,190,199,199,199,201,202

261              PRINT                         ENTRY,DECIMAL,FLOAT(SINGLE)
                                               690,702,719,730,848

3                PSOK                          (5)STATIC,EXTERNAL,UNALIGNED,STRING(65),CHARACTER,VARYING
                                               326,336

                 PTR                           AUTOMATIC,ALIGNED,POINTER
                                               12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,13,13,13,13,13,13
                                               13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,17,17,17
                                               18,20,24,27,36,38,39,39,44,47,57,57,57,57,65,65,65,65,65,70,70,70,70
                                               70,70,70,74,78,81,83,86,87,88,90,91,91,92,93,93,94,95,95,95,96,97,97
                                               97,98,99,99,100,101,101,101,102,103,103,103,103,104,105,105,105,106
                                               107,107,108,109,109,109,110,111,111,112,117,117,117,117,119,119,120
                                               122,122,123,125,125,126,129,129,131,139,141,144,144,146,151,156,157
                                               161,162,178,180,180,180,180,180,181,181,181,183,183,183,184,184,184
                                               186,190,192,199,199,200,200,201,201,202,208,208,208,208,208,209,272
                                               275,278,279,280,282,284,287,289,292,296,297,298,298,300,303,305,306
                                               306,307,307,307,309,315,316,316,320,320,320,322,328,330,330,330,332
                                               338,338,338,340,346,349,351,351,353,358,359,359,362,363,364,365,366
                                               368,370,370,370,370,370,370,370,370,370,370,370,370,370,370,370,370
                                               370,370,370,370,370,374,375,377,379,380,383,387,387,390,390,390,390
                                               390,390,390,390,392,392,395,395,395,395,395,395,395,395,395,395,397
                                               398,398,399,400,401,401,404,404,404,404,406,413,415,415,415,416,418
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               423,423,425,425,425,426,431,431,434,434,579,642,643,644,645,646,647
                                               648,649,650,651,652,655,661,662,677,678,692,693,697,698,705,709,710
                                               711,712,713,713,713,713,713,713,713,713,713,713,713,713,713,713,713
                                               713,713,713,713,713,713,714,714,715,715,722,733,739,743,800,803,803
                                               805,805,818,819,820,821,821,822,822,823,826,828,829,830,832,834,835
                                               838,838,838,839,841,849,858,859,860,861,862,867,868,869,873,874,875
                                               876,878,878,879,881,888

77     ********* Q                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               84,85,89,89,90

11     ********* Q                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               36,37,42,45

2                QUAL                          (30)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(10),CHARACTER
                                               38,39,383

2                QUALNO                        IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               17,36,379,380

1                R                             ENTRY,DECIMAL,FLOAT(SINGLE)

10               RECIN                         ENTRY,DECIMAL,FLOAT(SINGLE)
                                               658

4                RECL                          (500)AUTOMATIC,STRUCTURE
                                               243,243,244,244,245,245,472,472,473,473,474,474,522,522,523,523,524
                                               524,545,545,546,546,547,547,786,797

2                RECORD                        BASED(PTR),STRUCTURE
                                               642,655,662,733

2                REG                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,99,99,315,316,316

2                REL                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,95,95,95,338,338,338,340

3                RK                            (5)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               344

11     ********* S                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               18,19,22,25,34

373              SAQPNT                        ENTRY,DECIMAL,FLOAT(SINGLE)
                                               457

                 SCARDS                        FILE,EXTERNAL
                                               12,13,14,17,20,26,29,35,38,46,53,57,65,70,117,155,158,166,195,208
                                               581,583,584,598,651,671,697,713,726

2                SCH                           (10)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(5),CHARACTER
                                               20,377
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

2                SCHNO                         IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               17,18,374,375

2                SET                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,97,97,97,307,307,307,309

2                SEX                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,91,91,275

2                SOCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               12,103,103,103,103,320,320,320,322,328

211              SPNT                          ENTRY,DECIMAL,FLOAT(SINGLE)
                                               765,772

464              SPNT1                         ENTRY,DECIMAL,FLOAT(SINGLE)
                                               779

                 SPRINT                        FILE,EXTERNAL
                                               24,44,90,92,94,96,98,100,102,104,106,108,110,112,120,123,126,131,157
                                               161,192,215,218,219,222,223,224,230,233,234,235,251,254,256,258,276
                                               277,279,281,283,285,286,288,291,294,295,297,301,304,306,310,312,313
                                               317,319,323,325,326,329,333,335,336,341,343,344,347,348,350,356,359
                                               362,364,366,367,370,374,377,379,381,383,385,389,390,394,395,398,400
                                               403,404,407,410,415,425,426,430,431,434,439,441,442,448,454,456,458
                                               460,484,485,491,492,502,503,504,509,511,512,556,557,558,560,564,568
                                               569,570,572,574,575,579,587,622,623,625,627,629,631,633,634,745,763
                                               770,777,782,805,813,841,853,881,892

4                START                         IN RECL(500),AUTOMATIC,UNALIGNED,STRING(6),CHARACTER
                                               254,254,254,570,821,869

2        ********* STFN                        (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               141,146

3                STK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(50),CHARACTER,VARYING
                                               313

8                STUDENT                       FILE,EXTERNAL,RECORD,ENVIRONMENT(F(2416))
                                               641,654,662,664,665,667,668,674,677,733,752,753,790,800,815,855

2                SUB                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(36),CHARACTER
                                               65,395,395

2                SUBCAR                        IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               208,426

                 SUBSTR                        GENERIC,BUILT-IN FUNCTION
                                               27,78,81,83,107,107,109,151,168,170,171,175,176,180,180,194,199,199
                                               205,221,254,254,254,254,254,254,256,256,256,266,266,266,272,293,306
                                               306,395,395,415,423,434,535,536,543,543,553,562,565,570,570,570,572
                                               692,711,756,756,756,803,803,805,829,834,862,863,864

508              SW                            AUTOMATIC,UNALIGNED,STRING(1),BIT
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               517,525,528,540,548,551

212              SW                            AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               238,246,249

116              SW                            AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT
                                               132,135

2                TC                            IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               208,425

2                TITLE                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               12,280,282,284

5                TRANS                         AUTOMATIC,UNALIGNED,STRING(2),CHARACTER
                                               590

2                TTFT                          (2,2)IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               12,78,81,109,208,209,272,287,289,292,423,423,434,434,692,697,698,803
                                               849,859,860,861,862,869

4                TYPE                          IN RECL(500),AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               470,470,501,503,520,520,532,553,562,565,567,569,825,839,842,877,879
                                               882

2                UNCOL                         (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER
                                               57,387,387,390,390,832,835

2                UNOTH                         (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER
                                               65,392,392,395,395,826,830

153              UPDATE                        ENTRY,DECIMAL,FLOAT(SINGLE)
                                               686

2      ********* VALUES                        (2,6)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               117,119,119

5                VP                            AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               595,609,632

5                VPB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               610,611,635,687,699,716,727

5      ********* VPI                           AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               429,432,437,446,452,689,701,718,729,794

5                WC                            AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               697,698

2                WS                            IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               208,425

271              X1                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               294
```

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
154	X1	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 155,156,158,159,164
11	X10	AUTOMATIC,UNALIGNED,STRING(10),CHARACTER 46,47
508	X2	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER 553,554,562,565
77	X2	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER 81,82,83,84
11	X2	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER
11	X5	AUTOMATIC,UNALIGNED,STRING(5),CHARACTER 26,27,29,35
271	X6	AUTOMATIC,UNALIGNED,STRING(6),CHARACTER 287,288,289,290,291,292,293
264	X6	PARAMETER,UNALIGNED,STRING(6),CHARACTER 263,265,266,266,266
154	XVAR	AUTOMATIC,UNALIGNED,STRING(4),CHARACTER,VARYING 176,177
2	YC	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER 12
5	YE	AUTOMATIC,UNALIGNED,STRING(4),CHARACTER 511,583,622,763,770,777
5 *********	YR	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 117,117,117,117,119,119,122,122,125,125,129,129,139,141,144,144,146 151,618,620,621
5	YT	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 593,617,619,628,629

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	GRK	330
6	L	80
3	MK	36
115	MNT	32
3	NORM	832
3	NORM2	256
3	OAK	160
3	OK	350
3	OPSOK	240
3	ORK	180
3	PSOK	325
4	RECL	47000
2	RECORD	2416
3	RK	150
3	STK	300

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED R IS 49008 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED RECIN IS 1236 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED CHECKREC IS 1072 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED ITRECIN IS 1028 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED UPDATE IS 1160 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED SPNT IS 1228 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 231 IS 956 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED PRINT IS 1040 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED CDATE IS 1000 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED GENPNT IS 1200 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED COUPNT IS 1152 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED SAQPNT IS 1188 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED APPNT IS 1020 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED SPNT1 IS 1052 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED DESPNT IS 1184 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 510 IS 900 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 578 IS 908 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 581 IS 896 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 666 IS 928 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 754 IS 1192 BYTES LONG.

THE PROGRAM CSECT IS NAMED R AND IS 98056 BYTES LONG.

THE STATIC CSECT IS NAMED ******RA AND IS 11297 BYTES LONG.

STATISTICS SOURCE RECORDS = 868,PROG TEXT STMNTS = 893,OBJECT BYTES = 98056

COMPILER DIAGNOSTICS.

WARNINGS.

IEM0227I NO FILE/STRING OPTION SPECIFIED IN ONE OR MORE GET/PUT STATEMENTS. SYSIN/SYSPRINT HAS BEEN ASSUMED IN EACH CASE.

IEM0764I ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY ARE FLAGGED '*********' IN THE XREF/ATR LIST.

IEM1790I DATA CONVERSIONS WILL BE DONE BY SUBROUTINE CALL IN THE FOLLOWING STATEMENTS 18, 36, 63, 82, 84, 177, 300, 309, 322, 332, 340, 353, 368, 375, 380, 709, 712, 875.

END OF DIAGNOSTICS.

COMPILE TIME 97.41 SECS

ELAPSED TIME 243.33 SECS

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRO
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST

*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
*OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
*OPTIONS IN EFFECT*     NEST,OPLIST
```

```
STMT LEVEL NEST
                                                                        R0008700
 1                CON:PROC(FILN2);                                      R0008710
 2      1              DCL FILN2 CHAR(8);                              R0008720
 3      1              DCL (I,J,K) FIXED BINARY, AK(3,26) FIXED BINARY EXTERNAL,  R0008730
                       (AUTR(*) CHAR(19), AUTR1(*) CHAR(3), ATAB(*) CHAR(3), ATAB1(*)  R0008740
                       FIXED BINARY) CONTROLLED EXTERNAL, (ALPHA CHAR(26), NALC FIXED  R0008750
                       BINARY) EXTERNAL;                               R0008760
 4      1              DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),  R0008770
                       MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),  R0008780
                       GRK(11) CHAR(30),OAK(4) CHAR(40), ORK(6) CHAR(30), OPSOK(6)  R0008790
                       CHAR(40)) VARYING EXTERNAL;                     R0008800
 5      1              DCL NORM(2,16,10) FIXED BINARY EXTERNAL;        R0008810
 6      1              DCL NORM2(2,16,2) FLOAT DECIMAL EXTERNAL;       R0008820
                  /*-----------------------------------------------------------------*/R0008830
 7      1              DCL CONST FILE;                                 R0008840
                  /*-----------------------------------------------------------------*/R0008850
 8      1              OPEN FILE(CONST) INPUT TITLE(FILN2);            R0008860
 9      1              DO I=1 TO 26;                                   R0008870
10      1    1            GET FILE(CONST) LIST(AK(1,I));                R0008880
11      1    1         END;                                             R0008890
12      1              AK(2,1)=1;     AK(3,1)=AK(2,1)+AK(1,1)-1;        R0008900
14      1              DO I=2 TO 26;                                   R0008910
15      1    1            AK(2,I)=AK(3,I-1)+1;                          R0008920
16      1    1            AK(3,I)=AK(1,I)+AK(2,I)-1;                    R0008930
17      1    1         END;                                             R0008940
18      1              N=SUM(AK(1,*));                                 R0008950
19      1              ALLOCATE AUTR(N),AUTR1(N);                      R0008960
20      1              DO K=1 TO 26;                                   R0008970
21      1    1           DO I=AK(2,K) TO AK(3,K);                      R0008980
22      1    2            GET FILE(CONST) EDIT(AUTR(I),AUTR1(I))       R0008990
                          (COL(1),A(19),COL(40),A(3));                 R0009000
23      1    2            END;                                         R0009010
24      1    1         END;                                             R0009020
25      1              GET FILE(CONST) EDIT((OAK(K) DO K=1 TO 4))(4(COL(1),A(40)));  R0009030
26      1              GET FILE(CONST) EDIT((ORK(K) DO K=1 TO 6))(6(COL(1),A(30)));  R0009040
27      1              GET FILE(CONST) EDIT((OPSOK(K) DO K=1 TO 6))(6(COL(1),A(40)));  R0009050
28      1              GET FILE(CONST) EDIT((SCK(I) DO I=1 TO 9))(9(COL(1),A(60)));  R0009060
29      1              GET FILE(CONST) EDIT((PSOK(I) DO I=1 TO 5))(5(COL(1),A(65)));  R0009070
30      1              GET FILE(CONST) EDIT((STK(I) DO I=1 TO 6))(6(COL(1),A(50)));  R0009080
31      1              GET FILE(CONST) EDIT((MK(I) DO I=1 TO 3))(3(COL(1),A(12)));  R0009090
32      1              GET FILE(CONST) EDIT((OK(I) DO I=1 TO 10))(10(COL(1),A(35)));  R0009100
33      1              GET FILE(CONST) EDIT((RK(I) DO I=1 TO 5))(5(COL(1),A(30)));  R0009110
34      1              GET FILE(CONST) EDIT((GRK(I) DO I=1 TO 11))(11(COL(1),A(30)));  R0009120
35      1              GET FILE(CONST) EDIT(ALPHA)(COL(1),A(26));      R0009130
36      1              GRC=SUBSTR(ALPHA,1,11);                         R0009140
37      1              GET FILE(CONST) EDIT(NALC)(COL(1),F(3));        R0009150
38      1              ALLOCATE ATAB(NALC),ATAB1(NALC);                R0009160
39      1              GET FILE(CONST) SKIP;                           R0009170
40      1              GET FILE(CONST) EDIT((ATAB(I),ATAB1(I) DO I=1 TO NALC))  R0009180
                       (X(2),A(3),X(1),F(2));                          R0009190
41      1              DO I=1 TO 2;                                    R0009200
42      1    1           DO J=1 TO 16;                                 R0009210
43      1    2              GET FILE(CONST) LIST((NORM(I,J,K) DO K=1 TO 10));  R0009220
44      1    2              GET FILE(CONST) LIST((NORM2(I,J,K) DO K=1 TO 2));  R0009230
45      1    2           END;                                          R0009240
46      1    1         END;                                             R0009250
```

```
STMT LEVEL NEST

 47      1              CLOSE FILE(CONST);                                  R0009260
 48      1         END;                                                     R0009270
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
3 *********	AK	(3,26)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 10,12,13,13,13,15,15,16,16,16,18,21,21
3	ALPHA	STATIC,EXTERNAL,UNALIGNED,STRING(26),CHARACTER 35,36
3	ATAB	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 38,40
3 *********	ATAB1	(*)EXTERNAL,CONTROLLED,ALIGNED,BINARY,FIXED(15,0) 38,40
3	AUTR	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 19,22
3	AUTR1	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 19,22
1	CON	ENTRY,DECIMAL,FLOAT(SINGLE)
7	CONST	FILE,EXTERNAL 8,10,22,25,26,27,28,29,30,31,32,33,34,35,37,39,40,43,44,47
2	FILN2	PARAMETER,UNALIGNED,STRING(8),CHARACTER 1,8
4	GRC	STATIC,EXTERNAL,UNALIGNED,STRING(11),CHARACTER,VARYING 36
4	GRK	(11)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING 34
3 *********	I	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 9,10,14,15,15,16,16,16,21,22,22,28,28,29,29,30,30,31,31,32,32,33,33 34,34,40,40,40,41,43,44
3 *********	J	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 42,43,44
3 *********	K	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 20,21,21,25,25,26,26,27,27,43,43,44,44
4	MK	(3)STATIC,EXTERNAL,UNALIGNED,STRING(12),CHARACTER,VARYING 31
*********	N	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 18,19,19
3 *********	NALC	STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 37,38,38,40

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

5     ********* NORM                          (2,16,10)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0)
                                               43

6               NORM2                          (2,16,2)STATIC,EXTERNAL,ALIGNED,DECIMAL,FLOAT(SINGLE)
                                               44

4               OAK                            (4)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                               25

4               OK                             (10)STATIC,EXTERNAL,UNALIGNED,STRING(35),CHARACTER,VARYING
                                               32

4               OPSOK                          (6)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                               27

4               ORK                            (6)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               26

4               PSOK                           (5)STATIC,EXTERNAL,UNALIGNED,STRING(65),CHARACTER,VARYING
                                               29

4               RK                             (5)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               33

4               SCK                            (9)STATIC,EXTERNAL,UNALIGNED,STRING(60),CHARACTER,VARYING
                                               28

4               STK                            (6)STATIC,EXTERNAL,UNALIGNED,STRING(50),CHARACTER,VARYING
                                               30

                SUBSTR                         GENERIC,BUILT-IN FUNCTION
                                               36

                SUM                            GENERIC,BUILT-IN FUNCTION
                                               18
```

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	AK	156
38	ATAB	ADJUSTABLE
38	ATAB1	ADJUSTABLE
19	AUTR	ADJUSTABLE
19	AUTR1	ADJUSTABLE
4	GRK	330
4	MK	36
5	NORM	640
6	NORM2	256
4	OAK	160
4	OK	350
4	OPSOK	240
4	ORK	180
4	PSOK	325
4	RK	150
4	SCK	540
4	STK	300

STORAGE REQUIREMENTS.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED CON IS 388 BYTES LONG.

THE PROGRAM CSECT IS NAMED CON AND IS 4294 BYTES LONG.

THE STATIC CSECT IS NAMED ****CONA AND IS 1044 BYTES LONG.

STATISTICS SOURCE RECORDS = 58,PROG TEXT STMNTS = 48,OBJECT BYTES = 4294

```
COMPILER DIAGNOSTICS.


WARNINGS.

   IEM0526I      1   OPTION MAIN HAS NOT BEEN SPECIFIED FOR THE EXTERNAL PROCEDURE, STATEMENT NUMBER 1

   IEM0764I          ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY
                     ARE FLAGGED '*********' IN THE XREF/ATR LIST.


END OF DIAGNOSTICS.

COMPILE TIME        10.93 SECS

ELAPSED TIME        26.14 SECS
EXECUTION TERMINATED
```

4. PROGRAMME F.

4.2 PROGRAMME LISTING

THIS SECTION CONTAINS A LISTING OF THE SOURCE CODE OF PROGRAMME F.

```
 1                                                                  /*****************************************/
 2   (SUBRG):(STRG):                                                /*                                       */
 3   F:PROCEDURE OPTIONS(MAIN);                                     /*            BCRS....PROGRAMME F        */
 4        DCL 1 RECORD BASED(PTR),                                  /*                                       */
 5              2 (DESNO CHAR(8), NAME CHAR(40)),                   /*****************************************/
 6              2 GENERAL,                                          /*                                       */
 7                3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),  /* DECLARATION OF STRUCTURE 'RECORD'  */
 8                3 ((DOBC,MART,SET,POCC,SCCC,OV25,REL,CONF) CHAR(1)),  /* -> MINOR STRUCTURES OF 'RECORD'  */
 9                3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)),  /*    -> GENERAL (GENERAL INFORMATION)  */
10                3 FPOS CHAR(4),                                   /*    -> EDUCATION (PREVIOUS EDUCATION)  */
11              2 EDUCATION,                                        /*    -> COURSE (COURSE DETAILS)         */
12                3 (SCHNO CHAR(2), SCH(10) CHAR(5)),               /*    -> FINALS (FINAL RESULTS)          */
13                3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),            /*    -> ANNPROG (ANNUAL PROGRESS)       */
14                3 (UNCOL(2) CHAR(21), FPT(2) CHAR(1), FT(2,2) CHAR(6)),  /*    -> INTEL (INTELLIGENCE TEST SCORES)  */
15                3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),  /*                                  */
16                3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),             /*                                       */
17                3 (ADIP CHAR(30), ADIPC CHAR(1)),                 /*                                       */
18                3 DDIPC CHAR(1),                                  /*                                       */
19                3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),               /*                                       */
20                3 ATO CHAR(1),                                    /*                                       */
21              2 COURSE,                                           /*                                       */
22                3 (LEN CHAR(1), CODES(4) CHAR(3)),                /*                                       */
23                3 ((LE,DC(4,24:64)) CHAR(1)),                     /*                                       */
24                3 (DC1(4,26:30) CHAR(2)),                         /*                                       */
25                3 ((DC46(4),DC49(4)) CHAR(3)),                    /*                                       */
26              2 FINALS,                                           /*                                       */
27                3 ((TC,WS,BE,SUBCAR) CHAR(1)),                    /*                                       */
28              2 ANNPROG,                                          /*                                       */
29                3 AP CHAR(802),                                   /*                                       */
30                3 ((LEAP,FK(100,2)) FIXED BINARY),                /*                                       */
31              2 INTEL,                                            /*                                       */
32                3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY),  /*                              */
33                3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),           /*                                       */
34                3 STEN(2,16) FIXED BINARY;                        /*                                       */
35   /*-----------------------------------------------------------*/  /*-----------------------------------*/
36        DCL (AGF(3),SXF(2),(POF,RLF)(5),SOF(6),MAF(3),RGF(11),STF(6),  /* DECLARATION OF PROGRAMME VARIABLES  */
37        FPF(10,10),OVF(0:10),SCF(0:9)) FIXED BIN;                 /*                                       */
38        DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),  /*                                       */
39        MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),  /*                                 */
40        GRK(11) CHAR(30)) VARYING EXTERNAL;                       /*                                       */
41        DCL (OAK(4) CHAR(40), ORK(6) CHAR(30)) VARYING EXTERNAL;  /*                                       */
42        DCL (OAF(4),ORF(6)) FIXED BINARY;                         /*                                       */
43        DCL OPSOK(6) CHAR(40) VARYING EXTERNAL;                   /*                                       */
44        DCL (OPOCF(6),OSOCF(7)) FIXED BINARY;                     /*                                       */
45        DCL (CLAB,RLAB)(20) FIXED BINARY,RITEM(20) CHAR(20)       /*                                       */
46        VARYING, LOP(20) CHAR(1);                                 /*                                       */
47        DCL (FILN1,FILN2,WD,DT,DESL(1000)) CHAR(8), YE CHAR(4), WRL  /*                                   */
48        CHAR(3), (WCR,FREQ) CHAR(1), (WB,RB,LB,FB) BIT(1),(KT,OPT,K) FIXED  /*                            */
49        BINARY, REQUEST CHAR(400) VARYING, BCON BIT(1) INITIAL('1'B),  /*                                 */
50        (DESNO2,DESNO3) CHAR(8), (SUPP,WPB,LPB,RPB,B) BIT(1);     /*                                       */
51        DCL CON ENTRY(CHAR(8));                                   /*                                       */
52   /*-----------------------------------------------------------*/  /*-----------------------------------*/
53        DCL STUDENT FILE RECORD ENV(F(2416));                     /* DECLARATION OF FILE (STUDENT)       */
54   /*************************************************************/  /*************************************/
55   TRANS:PROC(REQUEST,LOP,CLAB,RLAB,RITEM,K);                     /* REQUEST TRANSLATION                 */
```

Dataset Limited

```
 56          DCL (CLAB,RLAB)(20) FIXED BINARY,RITEM(20) CHAR(20)                /*****************************************/
 57          VARYING, LOP(20) CHAR(1);                                          /* VARIABLE DECLARATIONS                 */
 58          DCL REQUEST CHAR(400) VARYING;                                     /*                                       */
 59          DCL PCOND(6) CHAR(2) VARYING INITIAL('=','¬=','>','¬>','<','¬<');  /* RELATIONAL OPERATOR LIST              */
 60          DCL X1 CHAR(1),LR CHAR(3),(I,J,K,LK,LL,LB,NB) FIXED BINARY;        /*                                       */
 61          LL=LENGTH(REQUEST);                                                /* CALCULATE LENGTH OF REQUEST           */
 62          K=0;  LK=1;                                                        /*                                       */
 63          DO I=1 TO LL;                                                      /*                                       */
 64             X1=SUBSTR(REQUEST,I,1);                                         /* REMOVE FIRST/NEXT CHARACTER           */
 65             IF X1=' ' THEN                                                  /* TEST FOR BLANK                        */
 66             DO; LB=I;                                                       /* -> RESET LOWER BOUND                  */
 67                 GOTO L61;                                                   /* -> BRANCH FOR NEXT CHARACTER          */
 68             END; ELSE                                                       /*                                       */
 69             IF X1='&' | X1='|' THEN                                         /* TEST FOR LOGICAL OPERATOR             */
 70             DO; LK=LK+1;                                                    /* -> UPDATE LOGICAL OPERATOR COUNTER    */
 71                 LOP(LK)=X1;                                                 /* -> ADD LOGICAL OPERATOR TO LOP VECTOR */
 72                 GOTO L61;                                                   /* -> BRANCH FOR NEXT CHARACTER          */
 73             END; ELSE                                                       /*                                       */
 74             DO J=1,3,5;                                                     /* TEST FOR RELATIONAL OPERATOR          */
 75                IF X1=PCOND(J) THEN GOTO L62;                                /* -> BRANCH IF FOUND                    */
 76             END;                                                            /*                                       */
 77             GOTO L61;                                                       /* BRANCH FOR NEXT CHARACTER             */
 78     L62:    NB=INDEX(SUBSTR(REQUEST,I),' ');                                /* LOCATE NEXT BLANK                     */
 79             IF SUBSTR(REQUEST,I-1,1)='¬' THEN                               /* TEST RELATIONAL OP. FOR 'NOT' CASE    */
 80             DO; K=K+1;                                                      /* -> UPDATE SIMPLE REQUEST COUNTER      */
 81                 CLAB(K)=J+1;                                                /* -> RECORD RELATIONAL OP. NUMBER       */
 82                 LR=SUBSTR(REQUEST,LB+1,I-LB-2);                             /* -> REMOVE ITEM NUMBER                 */
 83                 RLAB(K)=LR;                          /* CONVERSION */       /* -> RECORD ITEM NUMBER                 */
 84             END; ELSE                                                       /* RELATIONAL OP. NOT 'NOT' CASE         */
 85             DO; K=K+1;                                                      /* -> UPDATE SIMPLE REQUEST COUNTER      */
 86                 CLAB(K)=J;                                                  /* -> RECORD RELATIONAL OP. NUMBER       */
 87                 LR=SUBSTR(REQUEST,LB+1,I-LB-1);                             /* -> REMOVE ITEM NUMBER                 */
 88                 RLAB(K)=LR;                          /* CONVERSION */       /* -> RECORD ITEM NUMBER                 */
 89             END;                                                            /*                                       */
 90             RITEM(K)=SUBSTR(REQUEST,I+1,NB-2);                              /* RECORD ITEM VALUE                     */
 91             I=I+NB-1;                                                       /* RESET CONTROL VARIABLE TO NEXT BLANK  */
 92     L61: END;                                                               /* BRANCH FOR NEXT CHARACTER             */
 93          DO I=1 TO K;                                                       /*                                       */
 94             IF RLAB(I)<1 | RLAB(I)>40 THEN                                  /* TEST ITEM NUMBERS OF ALL SIMPLE REQUESTS */
 95             DO; PUT EDIT('ERROR.....ITEM',RLAB(I),'NOT INCLUDED')           /* -> CLASS G ERROR/WARNING              */
 96                 (SKIP,COL(1),A,X(1),F(2),X(1),A);                           /*                                       */
 97                 DO J=I TO K-1;                                              /* OVERWRITE SIMPLE REQUEST              */
 98                    LOP(J)=LOP(J+1);                                         /* -> LOGICAL OPERATOR                   */
 99                    RLAB(J)=RLAB(J+1);                                       /* -> ITEM NUMBER                        */
100                    CLAB(J)=CLAB(J+1);                                       /* -> RELATIONAL OPERATOR                */
101                    RITEM(J)=RITEM(J+1);                                     /* -> ITEM VALUE                         */
102                    K=K-1;                                                   /* -> RESET SIMPLE REQUEST COUNTER       */
103                 END;                                                        /*                                       */
104             END;                                                            /*                                       */
105          END;                                                               /*                                       */
106     END;                                                                    /*                                       */
107     /*****************************************************************/     /*****************************************/
108     LOCTST:PROC(LOP,CLAB,RLAB,RITEM,K) RETURNS(BIT(1));                     /* LOCATE AND TEST REQUEST ROUTINE       */
109          DCL LOP(20) CHAR(1),(CLAB,RLAB)(20) FIXED BINARY, RITEM(20)        /*****************************************/
110          CHAR(20) VARYING;                                                  /* VARIABLE DECLARATIONS                 */
111          DCL REQI CHAR(12), (REQM,ITEM) CHAR(20) VARYING, (K,I,J,QS)        /*                                       */
112          FIXED BINARY;                                                      /*                                       */
113          /*****************************************************************/ /*****************************************/
114     FIND:PROCEDURE(ITEM,LR,REQI);                                           /* LOCATE ITEM VALUE                     */
115          DCL (ITEM CHAR(20), XVAR CHAR(3)) VARYING, (LR,KC,I,J) FIXED       /*****************************************/
```

Dataset Limited

```
116          BINARY, X2 CHAR(2), R(40) LABEL, REQI CHAR(12), ST CHAR(1) DEFINED      /* VARIABLE DECLARATIONS                          */
117          REQI POS(2), SYF CHAR(2) DEFINED REQI POS(3), SYT CHAR(2) DEFINED       /*                                                */
118          REQI POS(5), QY CHAR(2) DEFINED REQI POS(2), QE CHAR(3) DEFINED         /*                                                */
119          REQI POS(4), QC CHAR(3) DEFINED REQI POS(7), QG CHAR(1) DEFINED         /*                                                */
120          REQI POS(10), QR CHAR(1) DEFINED REQI POS(11);                          /*                                                */
121          IF QE¬='***' THEN QE=TRANSLATE(QE,' ','*');                             /* REMOVE '*' FROM QUALIFICATION EXAM             */
122          IF LR>=26 & LR<=35 THEN I=LE;                          /* CONVERSION */ /* SET I=LATEST ENTRY                              */
123          GOTO R(LR);                                                             /* BRANCH TO REQUIRED FIELD                       */
124          R(1):X2=SUBSTR(TTFT(1,1),5,2); I=X2;                   /* CONVERSION */ /* -> AGE, CALCULATE AS:                           */
125          X2=SUBSTR(DOB,5,2); J=X2;                              /* CONVERSION */ /*    -> 'YEAR OF ADMISSION - YEAR OF BIRTH'      */
126          KC=I-J;                                                                 /*                                                */
127          GOTO L1;                                                                /*    -> BRANCH TO CONVERSION ROUTINE             */
128          R(2):ITEM=SEX; RETURN;  R(3):ITEM=MART; RETURN;                         /* -> SEX, MARITAL STATUS                         */
129          R(4):ITEM=SET; RETURN;  R(5):ITEM=REG; RETURN;                          /* -> SETTLEMENT, GEOGRAPHICAL REGION             */
130          R(6):ITEM=POCC; RETURN; R(7):ITEM=SOCC; RETURN;                         /* -> PARENT/GUARDIAN/STUDENT OCCUPATION          */
131          R(8):ITEM=OV25; RETURN; R(9):ITEM=REL; RETURN;                          /* -> OVER 25 OCCUPATION, RELIGION                */
132          R(10):ITEM=CONF; RETURN;  R(11):ITEM=AUTH; RETURN;                      /* -> CONFIRMATION, AUTHORITY/COUNTRY             */
133          R(12):ITEM=AUTHNO; RETURN;  R(13):ITEM=FPOS; RETURN;                    /* -> AUTHORITY NUMBER, FAMILY POSITION           */
134          R(14):ITEM=TTFT(1,1); RETURN;  R(15):ITEM=UNCOL(1); RETURN;             /* -> TEACHERS TRAINING, UNIV OR COLLEGE 1        */
135          R(16):ITEM=UNOTH(1); RETURN;  R(17):ITEM=DEG(1); RETURN;                /* -> UNIV OR OTHER 1, DEGREE AWARDED             */
136          R(18):ITEM=SCHNO; RETURN;                                               /* -> NO. OF SCHOOLS ATTENDED                     */
137          R(19):KC=0;  QS=SCHNO;                                 /* CONVERSION */ /* -> SCHOOLS                                      */
138          DO I=1 TO QS;                                                           /*                                                */
139             IF ST='*' | SUBSTR(SCH(I),1,1)=ST THEN                               /*    -> SCHOOL TYPE                              */
140             IF SYF='**' | SUBSTR(SCH(I),2,2)=SYF THEN                            /*    -> SCHOOL YEAR FROM                         */
141             IF SYT='**' | SUBSTR(SCH(I),4,2)=SYT THEN KC=KC+1;                   /*    -> SCHOOL YEAR TO                           */
142          END;                                                                    /*                                                */
143          GOTO L1;                                                                /*    -> BRANCH TO CONVERSION ROUTINE             */
144          R(20):ITEM=QUALNO; RETURN;                                              /* -> NO. OF QUALIFICATIONS RECORDED              */
145          R(21):KC=0; QS=QUALNO;                                 /* CONVERSION */ /* -> QUALIFICATIONS                               */
146          DO I=1 TO QS;                                                           /*                                                */
147             IF QY='**' | SUBSTR(QUAL(I),1,2)=QY THEN                             /*    -> QUALIFICATION YEAR                       */
148             IF QE='***' | SUBSTR(QUAL(I),3,3)=QE THEN                            /*    -> QUALIFICATION EXAM                       */
149             IF QC='***' | SUBSTR(QUAL(I),6,3)=QC THEN                            /*    -> QUALIFICATION CODE                       */
150             IF QG='*' | SUBSTR(QUAL(I),9,1)=QG THEN                              /*    -> QUALIFICATION GRADE                      */
151             IF QR='*' | SUBSTR(QUAL(I),10,1)=QR THEN KC=KC+1;                    /*    -> QUALIFICATION RESULT                     */
152          END;                                                                    /*                                                */
153          GOTO L1;                                                                /*    -> BRANCH TO CONVERSION ROUTINE             */
154          R(22):ITEM=ATO; RETURN;  R(23):ITEM=LEN; RETURN;                        /* -> A.T.O, COURSE LENGTH                        */
155          R(24):I=ST;                                            /* CONVERSION */ /* -> SET CODES REQUIRED                          */
156          ITEM=CODES(I); RETURN;  R(25):ITEM=DC46(1); RETURN;                     /*    -> COURSE CODES                             */
157          R(26):ITEM=DC(I,25); RETURN; R(27):ITEM=DC(I,34); RETURN;               /* -> DETAILS OF COURSE                           */
158          R(28):ITEM=DC(I,35); RETURN; R(29):ITEM=DC(I,36); RETURN;               /*                                                */
159          R(30):ITEM=DC(I,37); RETURN; R(31):ITEM=DC(I,38); RETURN;               /*                                                */
160          R(32):ITEM=DC(I,41); RETURN; R(33):ITEM=DC(I,42); RETURN;               /*                                                */
161          R(34):ITEM=DC(I,59); RETURN; R(35):ITEM=DC(I,64); RETURN;               /*                                                */
162          R(36):XVAR=SUBSTR(REQI,5,INDEX(REQI,')')-5);                            /* -> ANNUAL PROGRESS VECTOR                      */
163          I=XVAR;                                                /* CONVERSION */ /*                                                */
164          IF FK(I,1)¬=0 THEN ITEM=SUBSTR(AP,FK(I,1),FK(I,2)-1); ELSE              /*                                                */
165          ITEM=' '; RETURN;                                                       /*                                                */
166          R(37):ITEM=TC; RETURN;                                                  /* -> TEACHERS' CERTIFICATE                       */
167          R(38):ITEM=WS; RETURN;                                                  /* -> WING SCIENCE                                */
168          R(39):ITEM=BE; RETURN;                                                  /* -> B.ED.                                       */
169          R(40):ITEM=SUBCAR; RETURN;                                              /* -> SUBSEQUENT CAREER                           */
170    L1:   ITEM=CHAR(KC);                                                          /* CONVERSION ROUTINE                             */
171    END;                                                                          /*                                                */
172          /*****************************************************************/     /*************************************************/
173    TEST:PROC(ITEM,COND,REQU) RETURNS(BIT(1));                                   /* SIMPLE REQUEST TEST ROUTINE                    */
174          DCL (ITEM,REQU,X20) CHAR(20) VARYING, COND FIXED BINARY;                /*************************************************/
175          DCL C(6) LABEL INITIAL(C1,C2,C3,C4,C5,C6);                              /* VARIABLE DECLARATIONS
```

Dataset Limited

```
176          X20=ITEM;  ITEM=((20)' ');  SUBSTR(ITEM,21-LENGTH(X20))=X20;        /* RIGHT JUSTIFY RECORD RESPONSE                */
177          X20=REQU;  REQU=((20)' ');  SUBSTR(REQU,21-LENGTH(X20))=X20;        /* RIGHT JUSTIFY REQUIRED VALUE                 */
178          GOTO C(COND);                                                       /* BRANCH TO REQUIRED TEST                      */
179     C1:  IF ITEM=REQU THEN RETURN('1'B); ELSE RETURN('0'B);                  /* -> CASE '='                                  */
180     C2:  IF ITEM¬=REQU THEN RETURN('1'B); ELSE RETURN('0'B);                 /* -> CASE '¬='                                 */
181     C3:  IF ITEM>REQU THEN RETURN('1'B); ELSE RETURN('0'B);                  /* -> CASE '>'                                  */
182     C4:  IF ITEM¬>REQU THEN RETURN('1'B); ELSE RETURN('0'B);                 /* -> CASE '¬>'                                 */
183     C5:  IF ITEM<REQU THEN RETURN('1'B); ELSE RETURN('0'B);                  /* -> CASE '<'                                  */
184     C6:  IF ITEM¬<REQU THEN RETURN('1'B); ELSE RETURN('0'B);                 /* -> CASE '¬<'                                 */
185     END;                                                                     /*                                              */
186     DCL TEST ENTRY(,,) RETURNS(BIT(1));                                      /*                                              */
187          /*****************************************************************/  /*******************************************/
188          DO I=1 TO K;                                                        /* CONSIDER FIRST/NEXT SIMPLE REQUEST           */
189             IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19    /* -> LOCATE EXCEPTIONAL CASES 1.               */
190             | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN         /*                                              */
191             REQI=SUBSTR(RITEM(I),INDEX(RITEM(I),'('));                       /*    -> SEPARATE REQUIRED ITEM                 */
192             IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=36    /* -> LOCATE EXCEPTIONAL CASES 2.               */
193             | RLAB(I)=40 THEN IF SUBSTR(REQI,2,2)¬=SUBSTR(TTFT(1,1),5,2)     /*    -> TEST YEARS                             */
194             THEN IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);  /*    -> BRANCH OR REJECT RECORD                */
195             CALL FIND(ITEM,RLAB(I),REQI);                                    /* -> FIND REQUIRED FIELD VALUE                 */
196             IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19    /* -> LOCATE EXCEPTIONAL CASES 3.               */
197             | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN         /*                                              */
198             REQM=SUBSTR(RITEM(I),1,INDEX(RITEM(I),'(')-1); ELSE              /*    -> SEPARATE REQUIRED ITEM                 */
199             REQM=TRANSLATE(RITEM(I),' ',':');                                /*    -> CLEAR IMBEDDED BLANKS                  */
200             IF TEST(ITEM,CLAB(I),REQM) THEN                                  /* -> TEST SIMPLE REQUEST                       */
201             DO; DO J=I+1 TO K;                                               /*                                              */
202                    IF LOP(J)¬='|' THEN                                       /* -> TEST NEXT LOGICAL OPERATOR                */
203                    DO; I=J-1;                                                /*                                              */
204                        GOTO L66;                                             /* -> BRANCH FOR NEXT SIMPLE REQUEST            */
205                    END;                                                      /*                                              */
206                    END;                                                      /*                                              */
207                    RETURN('1'B);                                             /* -> ACCEPT THE RECORD                         */
208             END; ELSE                                                        /*                                              */
209             IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);       /* -> BRANCH OR REJECT RECORD                   */
210     L66: END;                                                                /* BRANCH FOR NEXT SIMPLE REQUEST               */
211     END;                                                                     /*                                              */
212     DCL LOCTST ENTRY(,,,,) RETURNS(BIT(1));                                  /*                                              */
213     /*****************************************************************/      /*******************************************/
214     SPNT:PROC;                                                               /* D.E.S.NO. OUTPUT ROUTINE                     */
215          DCL I FIXED BINARY;                                                 /*******************************************/
216          PUT SKIP(3);                                                        /*                                              */
217          DO I=1 TO KT;                                                       /*                                              */
218             IF MOD(I-1,10)=0 THEN PUT SKIP;                                  /* OUTPUT DES.NOS, TEN PER LINE                 */
219             PUT EDIT(DESL(I))(X(3),A(8));                                    /*                                              */
220          END;                                                                /*                                              */
221     END;                                                                     /*                                              */
222     /*****************************************************************/      /*******************************************/
223     RFREQK:PROCEDURE;                                                        /* RECORD FREQUENCY KEYS                        */
224          DCL K FIXED BINARY;                                                 /*******************************************/
225          PUT EDIT('MARITAL STATUS')(SKIP(3),A);                              /* MARITAL STATUS                               */
226          PUT SKIP;                                                           /*                                              */
227          DO K=1 TO 3;                                                        /*                                              */
228             PUT EDIT(K,MK(K))(SKIP,F(1),COL(8),A);                           /*                                              */
229          END;                                                                /*                                              */
230          PUT EDIT('GEOGRAPHICAL REGION','NOT CODED PRIOR TO 1972')           /* GEOGRAPHICAL REGION                          */
231          (SKIP(3),A,COL(80),A);                                              /*                                              */
232          PUT SKIP;                                                           /*                                              */
233          DO K=1 TO 11;                                                       /*                                              */
234             PUT EDIT(SUBSTR(GRC,K,1),GRK(K))(SKIP,A(1),COL(8),A);            /*                                              */
235          END;                                                                /*                                              */
```

Dataset Limited

```
236          PUT EDIT('SETTLEMENT','SETTLEMENT PRIOR TO 1972')                  /* SETTLEMENT                                  */
237          (SKIP(3),A,COL(80),A);                                             /*                                             */
238          PUT SKIP;                                                          /*                                             */
239          DO K=1 TO 4;                                                       /*                                             */
240             PUT EDIT(K,STK(K),K,OAK(K))(SKIP,F(1),COL(8),A,COL(80),F(1),    /*                                             */
241          COL(87),A);                                                        /*                                             */
242          END;                                                               /*                                             */
243          PUT EDIT('5',STK(5),'6',STK(6))(2(SKIP,A(1),COL(8),A));            /*                                             */
244          PUT EDIT('OCCUPATION IF OVER 25')(SKIP(3),A);                      /* OVER 25 OCCUPATION                          */
245          PUT EDIT('0',OK(10))(SKIP(2),A(1),COL(8),A);                       /*                                             */
246          DO K=1 TO 9;                                                       /*                                             */
247             PUT EDIT(K,OK(K))(SKIP,F(1),COL(8),A);                          /*                                             */
248          END;                                                               /*                                             */
249          PUT EDIT('STUDENT/PARENT/GUARDIAN OCCUPATION',                     /* STUDENT/PARENT/GUARDIAN OCCUPATION          */
250          'STUDENT/PARENT/GUARDIAN OCCUPATION PRIOR TO 1972')                /*                                             */
251          (PAGE,SKIP(3),A,COL(80),A);                                        /*                                             */
252          PUT SKIP;                                                          /*                                             */
253          DO K=1 TO 5;                                                       /*                                             */
254             PUT EDIT(K,PSOK(K),K,OPSOK(K))(SKIP,F(1),COL(8),A,COL(80),F(1), /*                                             */
255             COL(87),A);                                                     /*                                             */
256          END;                                                               /*                                             */
257          PUT EDIT('6',OPSOK(K))(SKIP,COL(80),A,COL(87),A);                  /*                                             */
258          PUT EDIT('RELIGION','RELIGION PRIOR TO 1972')                      /* RELIGION                                    */
259          (SKIP(3),A,COL(80),A);                                             /*                                             */
260          PUT SKIP;                                                          /*                                             */
261          DO K=1 TO 5;                                                       /*                                             */
262             PUT EDIT(K,RK(K),K,ORK(K))(SKIP,F(1),COL(8),A,COL(80),F(1),     /*                                             */
263             COL(87),A);                                                     /*                                             */
264          END;                                                               /*                                             */
265          PUT EDIT('6',ORK(6))(SKIP,COL(80),A,COL(87),A);                    /*                                             */
266          PUT EDIT('SCHOOL TYPE')(SKIP(3),A);                                /* SCHOOL TYPE                                 */
267          PUT SKIP;                                                          /*                                             */
268          DO K=1 TO 9;                                                       /*                                             */
269             PUT EDIT(K,SCK(K))(SKIP,F(1),COL(8),A);                         /*                                             */
270          END;                                                               /*                                             */
271       END;                                                                  /*                                             */
272       /*****************************************************************/    /*********************************************/
273       RECFREQ:PROC;                                                         /* FREQUENCY TABLE ACCUMULATION ROUTINE        */
274          DCL X2 CHAR(2), X1 CHAR(1), OKB BIT(1), (I,J,K) FIXED BINARY;      /*********************************************/
275          X2=SUBSTR(TTFT(1,1),5,2);  I=X2;                  /* CONVERSION */ /*                                             */
276          IF I<72 THEN OKB='1'B; ELSE OKB='0'B;                              /* SET OLD CODE BIT (PRE 1972)                 */
277          X2=SUBSTR(DOB,5,2);                                                /*                                             */
278          J=X2;                                             /* CONVERSION */ /*                                             */
279          IF I>J THEN                                                        /*                                             */
280          DO; IF I-J>=25 THEN AGF(1)=AGF(1)+1; ELSE AGF(2)=AGF(2)+1;         /* AGE                                         */
281              AGF(3)=AGF(3)+I-J;                                             /*                                             */
282          END; ELSE                                                          /*                                             */
283          PUT EDIT(DESNO,'ERROR RETURNED.....AGE')(SKIP,A(8),X(2),A);        /*                                             */
284          IF SEX='M' THEN SXF(1)=SXF(1)+1; ELSE                              /* SEX                                         */
285          IF SEX='F' THEN SXF(2)=SXF(2)+1; ELSE                              /*                                             */
286          PUT EDIT(DESNO,'ERROR RETURNED.....SEX')(SKIP,A(8),X(2),A);        /*                                             */
287          IF MART>='1' & MART<='3' THEN                                      /* MARITAL STATUS                              */
288          DO; I=MART;                                       /* CONVERSION */ /*                                             */
289              MAF(I)=MAF(I)+1;                                               /*                                             */
290          END; ELSE                                                          /*                                             */
291          PUT EDIT(DESNO,'ERROR RETURNED.....MARITAL STATUS')                /*                                             */
292          (SKIP,A(8),X(2),A);                                                /*                                             */
293          IF REL>='1'  & REL<='5' & ¬OKB THEN                                /* RELIGION                                    */
294          DO; I=REL;                                        /* CONVERSION */ /*                                             */
295              RLF(I)=RLF(I)+1;                                               /*                                             */
```

Dataset Limited

```
296          END; ELSE                                                          /*                                          */
297          IF REL>='1' & REL<='6' & CKB THEN                                  /* RELIGION PRE 1972                        */
298          DO; I=REL;                                          /* CONVERSION */  /*                                          */
299              ORF(I)=ORF(I)+1;                                               /*                                          */
300          END; ELSE                                                          /*                                          */
301          PUT EDIT(DESNO,'ERROR RETURNED.....RELIGION')(SKIP,A(8),X(2),A);   /*                                          */
302          IF SET>='1' & SET<='6' & ¬CKB THEN                                 /* SETTLEMENT                               */
303          DO; I=SET;                                          /* CONVERSION */  /*                                          */
304              STF(I)=STF(I)+1;                                               /*                                          */
305          END; ELSE                                                          /*                                          */
306          IF SET>='1' & SET<='4' & OKB THEN                                  /* SETTLEMENT PRE 1972                      */
307          DO; I=SET;                                          /* CONVERSION */  /*                                          */
308              OAF(I)=OAF(I)+1;                                               /*                                          */
309          END; ELSE                                                          /*                                          */
310          PUT EDIT(DESNO,'ERROR RETURNED.....SETTLEMENT')                    /*                                          */
311          (SKIP,A(8),X(2),A);                                                /*                                          */
312          IF POCC>='1' & POCC<='5' & ¬CKB THEN                               /* PARENT/GUARDIAN OCCUPATION               */
313          DO; I=POCC;                                         /* CONVERSION */  /*                                          */
314              POF(I)=POF(I)+1;                                               /*                                          */
315          END; ELSE                                                          /*                                          */
316          IF POCC>='1' & POCC<='6' & OKB THEN                                /* PARENT/GUARDIAN OCCUPATION PRE 1972      */
317          DO; I=POCC;                                         /* CONVERSION */  /*                                          */
318             OPOCF(I)=OPOCF(I)+1;                                            /*                                          */
319          END; ELSE                                                          /*                                          */
320          PUT EDIT(DESNO,'ERROR RETURNED.....PARENT/GUARDIAN OCCUPATION')    /*                                          */
321          (SKIP,A(8),X(2),A);                                                /*                                          */
322          IF SOCC=' ' THEN IF ¬OKB THEN SOF(6)=SOF(6)+1; ELSE                /* STUDENT OCCUPATION, AT SCHOOL            */
323          OSOCF(7)=OSOCF(7)+1; ELSE                                          /*                                          */
324          IF SOCC>='1' & SOCC<='5' & ¬OKB THEN                               /* STUDENT OCCUPATION                       */
325          DO; I=SOCC;                                         /* CONVERSION */  /*                                          */
326              SOF(I)=SOF(I)+1;                                               /*                                          */
327          END; ELSE                                                          /*                                          */
328          IF SOCC>='1' & SOCC<='6' & OKB THEN                                /* STUDENT OCCUPATION PRE 1972              */
329          DO; I=SOCC;                                         /* CONVERSION */  /*                                          */
330              OSOCF(I)=OSOCF(I)+1;                                           /*                                          */
331          END; ELSE                                                          /*                                          */
332          PUT EDIT(DESNO,'ERROR RETURNED.....STUDENT OCCUPATION')            /*                                          */
333          (SKIP,A(8),X(2),A);                                                /*                                          */
334          IF OV25=' ' THEN OVF(10)=OVF(10)+1; ELSE                           /* OVER 25 OCCUPATION, UNDER 25             */
335          IF OV25>='0' & OV25<='9' THEN                                      /* OVER 25 OCCUPATION                       */
336          DO; I=OV25;                                         /* CONVERSION */  /*                                          */
337              OVF(I)=OVF(I)+1;                                               /*                                          */
338          END; ELSE                                                          /*                                          */
339          PUT EDIT(DESNO,'ERROR RETURNED.....OVER 25 OCCUPATION')            /*                                          */
340          (SKIP,A(8),X(2),A);                                                /*                                          */
341          IF OKB THEN GOTO L3;                                               /* BRANCH IF OLD CODE (PRE 1972)            */
342          I=INDEX(GRC,REG);                                                  /* GEOGRAPHICAL REGION                      */
343          IF I>=1 & I<=11 THEN RGF(I)=RGF(I)+1; ELSE                         /*                                          */
344          PUT EDIT(DESNO,'ERROR RETURNED.....REGION')(SKIP,A(8),X(2),A);     /*                                          */
345     L3:  K=SCHNO;                                            /* CONVERSION */  /* SCHOOL TYPE                              */
346          DO J=1 TO K;                                                       /*                                          */
347             X1=SUBSTR(SCH(J),1,1);                                          /*                                          */
348             IF X1>='0' & X1<='9' THEN                                       /*                                          */
349             DO; I=X1;                                        /* CONVERSION */  /*                                          */
350                 SCF(I)=SCF(I)+1;                                            /*                                          */
351             END; ELSE                                                       /*                                          */
352             PUT EDIT(DESNO,'ERROR RETURNED.....SCHOOL TYPE')                /*                                          */
353             (SKIP,A(8),X(2),A);                                             /*                                          */
354          END;                                                               /*                                          */
355          X2=SUBSTR(FPOS,1,2);                                               /* FAMILY POSITION                          */
```

Dataset Limited

```
356          IF X2='  ' THEN I=100; ELSE I=X2;                    /* CONVERSION */      /* -> CALCULATE CHILD NUMBER                    */
357          X2=SUBSTR(FPOS,3,2);                                                        /*                                              */
358          IF X2='  ' THEN J=100; ELSE J=X2;                    /* CONVERSION */      /* -> OF TOTAL CHILDREN                         */
359          IF I>=1 & I<=10 & J>=1 & J<=10 THEN FPF(I,J)=FPF(I,J)+1; ELSE              /* -> UP TO 10 OF 10 ONLY.                      */
360          PUT EDIT(DESNO,'ERROR RETURNED.....FAMILY POSITION',                        /*                                              */
361          '(',I,'/',J,')')(SKIP,A(8),X(2),A,X(2),2(A,P'99'),A);                       /*                                              */
362       END;                                                                           /*                                              */
363       /*****************************************************************************/  /*****************************************************/
364       RECFPNT:PROC;                                                                  /* FREQUENCY TABLE PRINT ROUTINE                */
365          DCL (J,K) FIXED BINARY;                                                     /*****************************************************/
366          PUT EDIT('SEX FREQUENCY')(SKIP(3),A);                                       /* SEX                                          */
367          PUT EDIT('MALE',SXF(1),(SXF(1)/KT)*100)(SKIP(2),A,COL(10),F(4),             /* -> MALE                                      */
368          COL(16),F(6,2));                                                            /*                                              */
369          PUT EDIT('FEMALE',SXF(2),(SXF(2)/KT)*100)(SKIP,A,COL(10),F(4),              /* -> FEMALE                                    */
370          COL(16),F(6,2));                                                            /*                                              */
371          PUT EDIT('AGE FREQUENCY ON ADMISSION','AVERAGE AGE ON ADMISS',              /* AGE, AVE. AGE ON ADMISSION                   */
372          'ION')(SKIP(3),A,COL(80),A,A);                                              /*                                              */
373          PUT EDIT('UNDER 25',AGF(2),(AGF(2)/KT)*100,(AGF(3)/KT))                     /* -> UNDER 25 ON ADMISSION                     */
374          (SKIP(2),A,COL(10),F(4),COL(16),F(6,2),COL(80),F(2));                       /*                                              */
375          PUT EDIT('OVER 25',AGF(1),(AGF(1)/KT)*100)(SKIP,A,COL(10),F(4),             /* -> OVER 25 ON ADMISSION                      */
376          COL(16),F(6,2));                                                            /*                                              */
377          PUT EDIT('MARITAL STATUS')(SKIP(3),A);                                      /* MARITAL STATUS                               */
378          PUT SKIP;                                                                   /*                                              */
379          DO K=1 TO 3;                                                                /*                                              */
380             PUT EDIT(K,MAF(K),(MAF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),               /*                                              */
381             COL(16),F(6,2));                                                         /*                                              */
382          END;                                                                        /*                                              */
383          PUT EDIT('GEOGRAPHICAL REGION','NOT CODED PRIOR TO 1972')                   /* GEOGRAPHICAL REGION                          */
384          (SKIP(3),A,COL(80),A);                                                      /*                                              */
385          PUT SKIP;                                                                   /*                                              */
386          DO K=1 TO 11;                                                               /*                                              */
387             PUT EDIT(SUBSTR(GRC,K,1),RGF(K),(RGF(K)/KT)*100)(SKIP,A(1),              /*                                              */
388             COL(10),F(4),COL(16),F(6,2));                                            /*                                              */
389          END;                                                                        /*                                              */
390          PUT EDIT('SETTLEMENT','SETTLEMENT PRIOR TO 1972')                           /* SETTLEMENT                                   */
391          (SKIP(3),A,COL(80),A);                                                      /*                                              */
392          PUT SKIP;                                                                   /*                                              */
393          DO K=1 TO 4;                                                                /*                                              */
394             PUT EDIT(K,STF(K),(STF(K)/KT)*100,K,OAF(K),(OAF(K)/KT)*100)              /*                                              */
395          (SKIP,F(1),COL(10),F(4),COL(16),F(6,2),COL(80),F(1),COL(89),                /*                                              */
396          F(4),COL(94),F(6,2));                                                       /*                                              */
397          END;                                                                        /*                                              */
398          DO K=5,6;                                                                   /*                                              */
399             PUT EDIT(K,STF(K),(STF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),               /*                                              */
400             COL(16),F(6,2));                                                         /*                                              */
401          END;                                                                        /*                                              */
402          PUT EDIT('OCCUPATION IF OVER 25')(SKIP(3),A);                               /* OVER 25 OCCUPATION                           */
403          PUT SKIP;                                                                   /*                                              */
404          DO K=0 TO 9;                                                                /*                                              */
405             PUT EDIT(K,OVF(K),(OVF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),               /*                                              */
406             COL(16),F(6,2));                                                         /*                                              */
407          END;                                                                        /*                                              */
408          PUT EDIT('UNDER 25',OVF(10),(OVF(10)/KT)*100)(SKIP,A,COL(10),F(4),          /* -> UNDER 25                                  */
409          COL(16),F(6,2));                                                            /*                                              */
410          PUT EDIT('STUDENT OCCUPATION','PARENT/GUARDIAN OCCUPATION',                 /* STUDENT/PARENT/GUARDIAN OCCUPATION           */
411          'STUDENT OCCUPATION','PARENT/GUARDIAN OCCUPATION','PRIOR TO 1972',          /*                                              */
412          'PRIOR TO 1972')(PAGE,SKIP(2),A,COL(27),A,2(COL(65),A,COL(91),A));          /*                                              */
413          PUT SKIP;                                                                   /*                                              */
414          DO K=1 TO 5;                                                                /*                                              */
415             PUT EDIT(K,SOF(K),(SOF(K)/KT)*100,POF(K),(POF(K)/KT)*100,                /*                                              */
```

Dataset Limited

```
416            K,OSOCF(K),(OSOCF(K)/KT)*100,OPOCF(K),(OPOCF(K)/KT)*100)          /*                                          */
417            (SKIP,F(1),COL(10),F(4),COL(16),F(6,2),COL(27),F(4),COL(33),      /*                                          */
418            F(6,2),COL(65),F(1),COL(73),F(4),COL(79),F(6,2),COL(91),F(4),     /*                                          */
419            COL(97),F(6,2));                                                  /*                                          */
420         END;                                                                 /*                                          */
421         PUT EDIT('6',OSOCF(6),(OSOCF(6)/KT)*100,OPOCF(6),                    /*                                          */
422         (OPOCF(6)/KT)*100)(COL(65),A,COL(73),F(4),COL(79),F(6,2),            /*                                          */
423         COL(91),F(4),COL(97),F(6,2));                                        /*                                          */
424         PUT EDIT('@ SCHOOL',SOF(6),(SOF(6)/KT)*100,OSOCF(7),                 /* -> STUDENT OCCUPATION, AT SCHOOL         */
425         (OSOCF(7)/KT)*100)(SKIP,A,COL(10),F(4),COL(16),F(6,2),COL(73),       /*                                          */
426         F(4),COL(79),F(6,2));                                                /*                                          */
427         PUT EDIT('SCHOOL TYPE')(SKIP(3),A);                                  /* SCHOOL TYPE                              */
428         PUT SKIP;                                                            /*                                          */
429         DO K=1 TO 9;                                                         /*                                          */
430            PUT EDIT(K,SCF(K),(SCF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),        /*                                          */
431            COL(16),F(6,2));                                                  /*                                          */
432         END;                                                                 /*                                          */
433         PUT EDIT('UNKNOWN',SCF(0),(SCF(0)/KT)*100)(SKIP,A,COL(10),F(4),      /* -> SCHOOL TYPE, UNKNOWN                  */
434         COL(16),F(6,2));                                                     /*                                          */
435         PUT EDIT('RELIGION','RELIGION PRIOR TO 1972')                        /* RELIGION                                 */
436         (SKIP(3),A,COL(80),A);                                               /*                                          */
437         PUT SKIP;                                                            /*                                          */
438         DO K=1 TO 5;                                                         /*                                          */
439            PUT EDIT(K,RLF(K),(RLF(K)/KT)*100,K,ORF(K),(ORF(K)/KT)*100)       /*                                          */
440            (SKIP,F(1),COL(10),F(4),COL(16),F(6,2),COL(80),F(1),COL(89),      /*                                          */
441            F(4),COL(94),F(6,2));                                             /*                                          */
442         END;                                                                 /*                                          */
443         PUT EDIT('6',ORF(6),(ORF(6)/KT)*100)(SKIP,COL(80),A(1),COL(89),      /*                                          */
444         F(4),COL(94),F(6,2));                                                /*                                          */
445         PUT EDIT('POSITION IN FAMILY')(SKIP(3),A);                           /* FAMILY POSITION                          */
446         PUT SKIP;                                                            /* -> UP TO 10 OF 10 ONLY                   */
447         PUT EDIT('  ')(SKIP,A);                                              /*                                          */
448         DO K=1 TO 10;                                                        /*                                          */
449            PUT EDIT(K)(X(3),F(2));                                           /*                                          */
450         END;                                                                 /*                                          */
451         PUT SKIP;                                                            /*                                          */
452         DO K=1 TO 9;                                                         /*                                          */
453            PUT EDIT(K,' ')(SKIP,F(1),A(1));                                  /*                                          */
454            DO J=1 TO 10;                                                     /*                                          */
455               IF J<K THEN PUT EDIT('X')(X(4),A(1)); ELSE                     /*                                          */
456               PUT EDIT(FPF(K,J))(X(1),F(4));                                 /*                                          */
457            END;                                                              /*                                          */
458         END;                                                                 /*                                          */
459         PUT EDIT('10')(SKIP,A(2));                                           /*                                          */
460         DO K=1 TO 9;                                                         /*                                          */
461            PUT EDIT('X')(X(4),A);                                            /*                                          */
462         END;                                                                 /*                                          */
463         PUT EDIT(FPF(10,10))(X(1),F(4));                                     /*                                          */
464      END;                                                                    /*                                          */
465      /*****************************************************************/     /*****************************************/
466         ON ENDFILE(SCARDS) GOTO L49;                                         /* BRANCH ON ENDFILE (SCARDS)               */
467         OPEN FILE(SPRINT) LINESIZE(132);                                     /* OPEN SPRINT WITH MAXIMUM LINE SIZE       */
468      L39: GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));         /* INPUT OPTION PARAMETER CARD              */
469         GET LIST(OPT);                                                       /* INPUT OPTION                             */
470         IF OPT<1 | OPT>2 THEN                                                /* TEST OPTION                              */
471         DO; PUT EDIT('OPTION INVALID.....PROGRAM TERMINATED')                /* -> CLASS A ERROR/WARNING                 */
472             (SKIP,COL(1),A);                                                 /*                                          */
473             GOTO L49;                                                        /* -> BRANCH TO PROGRAMME TERMINATION       */
474         END;                                                                 /*                                          */
475         WRL='XXX'; WOR='X'; FREQ='X'; REQUEST=' '; B='0'B;                   /* INITIALISE OPTION PARAMETERS             */
```

Dataset Limited

```
476          GET DATA;                                                  /* INPUT OPTION PARAMETERS                    */
477          IF SUBSTR(WRL,1,1)='1' THEN WB='1'B; ELSE WB='0'B;         /* VERIFY OPTION PARAMETERS                   */
478          IF SUBSTR(WRL,1,1)='2' THEN WPB='1'B; ELSE WPB='0'B;       /* PARAMETER WRL                              */
479          IF SUBSTR(WRL,2,1)='1' THEN RB='1'B; ELSE RB='0'B;         /*                                            */
480          IF SUBSTR(WRL,2,1)='2' THEN RPB='1'B; ELSE RPB='0'B;       /*                                            */
481          IF SUBSTR(WRL,3,1)='1' THEN LB='1'B; ELSE LB='0'B;         /*                                            */
482          IF SUBSTR(WRL,3,1)='2' THEN LPB='1'B; ELSE LPB='0'B;       /*                                            */
483          IF FREQ='1' THEN FB='1'B; ELSE FB='0'B;                    /* PARAMETER FREQ                             */
484          PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME F',  /* OUTPUT TITLES                           */
485          'OPTION REQUESTED',OPT,'YEAR REQUESTED',YE)(PAGE,COL(1),A,  /*                                            */
486          SKIP(2),COL(1),A,COL(1),A,COL(32),F(2),COL(1),A,COL(30),A);  /*                                          */
487          PUT SKIP(2);                                               /*                                            */
488          IF WRL¬='XXX' THEN PUT EDIT('WRL REQUESTED',WRL)           /* OUTPUT OPTION PARAMETERS IN EFFECT         */
489          (COL(1),A,COL(31),A); ELSE WRL='000';                      /* -> WRL                                     */
490          IF WCR¬='X' THEN PUT EDIT('WCR REQUESTED',WCR)(COL(1),A,COL(33),  /* -> WCR                                 */
491          A); ELSE WCR='*';                                          /*                                            */
492          IF FB THEN PUT EDIT('FREQUENCY TABLES REQUIRED')           /* -> FREQ                                    */
493          (SKIP,COL(1),A);                                           /*                                            */
494          PUT SKIP(4);                                               /*                                            */
495          IF BCON & (OPT=1 | (OPT=2 & FB) | (OPT=2 & REQUEST='*')) THEN  /* TEST FOR PROCEDURE CON                 */
496          DO; CALL CON(FILN2);                                       /* -> CALL CON (INPUT OF DATA CONSTANTS)      */
497              BCON='0'B;                                             /* -> SET CON BIT                             */
498          END;                                                       /*                                            */
499          IF REQUEST='*' THEN GOTO L00;                              /* BRANCH IF SPECIAL CASE 'REQUEST='*''       */
500     /*-----------------------------------------------------------*/  /*-----------------------------------------*/
501     L12:OPEN FILE(STUDENT) INPUT TITLE(FILN1);                      /* OPTIONS 1 AND 2                            */
502          ON ENDFILE(STUDENT)                                        /* SET ENDFILE (STUDENT) ROUTINE              */
503          BEGIN; DT=DATE;                                            /* -> CALCULATE DATE                          */
504                 DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||  /*                                         */
505                 SUBSTR(DT,7,2);                                     /*                                            */
506                 IF OPT=1 | OPT=2 THEN                               /* -> OPTIONS 1 AND 2                         */
507                 DO; IF OPT=1 THEN PUT EDIT(YF,'FILE........BEDE COLLEGE',  /* -> OUTPUT TITLES FOR OPTION 1           */
508                     ' GENERAL FREQUENCIES')(SKIP(2),COL(1),A(4),X(1),A,A);  /*                                     */
509                     IF OPT=2 THEN PUT EDIT('FREQUENCY COUNT FOR REQUEST',  /* -> OUTPUT TITLES FOR OPTION 2           */
510                     REQUEST)(SKIP(2),COL(1),A,X(2),A);              /*                                            */
511                     PUT EDIT(KT,'STUDENTS INCLUDED')                /* -> OUTPUT SAMPLE SIZE 'KT'                 */
512                     (SKIP(3),COL(1),F(4),X(1),A);                   /*                                            */
513                     IF KT=0 THEN GOTO L29;                          /* -> BRANCH IF KT=0                          */
514                     IF ¬SUPP THEN CALL SPNT;                        /* -> TEST SUPPRESSION BIT, OUTPUT DES.NOS.   */
515                     IF OPT=2 & ¬FB THEN GOTO L29;                   /* -> BRANCH IF FREQUENCY TABLES NOT REQUIRED*/
516                     PUT EDIT('FREQUENCY TABLE KEYS')                /* -> OUTPUT FREQUENCY KEYS TITLE             */
517                     (PAGE,COL(1),A);                                /*                                            */
518                     CALL RFREQK;                                    /* -> OUTPUT FREQUENCY KEYS                   */
519                     PUT EDIT('FREQUENCIES','(TOTALS  PERCENTAGES)')  /* -> OUTPUT FREQUENCY TABLES TITLE           */
520                     (PAGE,COL(1),A,X(2),A);                         /*                                            */
521                     CALL RECFPNT;                                   /* -> OUTPUT FREQUENCY TABLES                 */
522     L29:            END;                                            /*                                            */
523                     PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);  /* -> OUTPUT DATE OF LISTING            */
524                     CLOSE FILE(STUDENT);                            /* -> CLOSE FILE                              */
525                     GOTO L39;                                       /* -> BRANCH TO PROGRAMME CONTROL SECTION     */
526          END;                                                       /* END OF ENDFILE ROUTINE                     */
527          DESL=' '; KT=0; SUPP='0'B;                                 /* INITIALISE VARIABLES ALL OPTIONS           */
528          IF OPT=1 THEN                                              /* INITIALISE VARIABLES OPTION 1              */
529          DO; AGF=0; SXF=0; POF=0; RLF=0; SOF=0; MAF=0; RGF=0; STF=0;  /*                                         */
530              FPF=0; OVF=0; SCF=0; OAF=0; ORF=0; OPOCF=0; CSOCF=0;    /*                                            */
531          END; ELSE                                                  /*                                            */
532          IF OPT=2 THEN                                              /* INITIALISE VARIABLES OPTION 2              */
533          DO; LOP=' '; CLAB=0; RLAB=0; RITEM=' '; K=0;               /*                                            */
534              IF SUBSTR(REQUEST,2,1)¬=' ' THEN                       /* TEST FOR REQUEST                           */
535              CALL TRANS(REQUEST,LOP,CLAB,RLAB,RITEM,K);             /* TRANSLATE REQUEST                          */
```

Dataset Limited

```
536             IF K=0 THEN                                                  /* TEST NO. OF SIMPLE REQUESTS, IF 0 THEN    */
537             DO; CLOSE FILE(STUDENT);                                     /* -> CLOSE FILE                             */
538                 GOTO L39;                                                /* -> BRANCH TO PROGRAMME CONTROL SECTION    */
539             END;                                                         /*                                           */
540             IF FB THEN                                                   /* TEST FOR FREQUENCY TABLES                 */
541             DO; AGF=0; SXF=0; POF=0; RLF=0; SOF=0; MAF=0; RGF=0; STF=0;  /* -> INITIALISE VARIABLES                   */
542                 FPF=0; OVF=0; SCF=0; OAF=0; ORF=0; OPOCF=0; OSOCF=0;     /*                                           */
543             END;                                                         /*                                           */
544         END;                                                             /*                                           */
545     L21: READ FILE(STUDENT) SET(PTR);                                    /* READ FIRST/NEXT RECORD                    */
546         IF ¬WB & SUBSTR(TTFT(2,1),1,4)='WITH' THEN                       /* -> TEST FOR WITHDRAWAL                    */
547         DO; IF WPB THEN                                                  /*                                           */
548             DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),     /*    -> CALCULATE WITHDRAWAL DATE           */
549                 3,2) || '/' || SUBSTR(TTFT(1,2),5,2);                    /*                                           */
550                 PUT EDIT(DESNO,'WITHDREW ON',WD)(COL(1),A,2(X(1),A));    /*    -> CLASS M ERROR/WARNING               */
551             END;                                                         /*                                           */
552             GOTO L21;                                                    /*    -> BRANCH FOR NEXT RECORD              */
553         END; ELSE                                                        /*                                           */
554         IF ¬LB & SUBSTR(TTFT(2,1),1,4)='LEFT' THEN                       /* -> TEST FOR LEFT                          */
555         DO; IF LPB THEN                                                  /*                                           */
556             DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),     /*    -> CALCULATE LEAVING DATE              */
557                 3,2) || '/' || SUBSTR(TTFT(1,2),5,2);                    /*                                           */
558                 PUT EDIT(DESNO,'LEFT ON',WD)(COL(1),A,2(X(1),A));        /*    -> CLASS M ERROR/WARNING               */
559             END;                                                         /*                                           */
560             GOTO L21;                                                    /*    -> BRANCH FOR NEXT RECORD              */
561         END; ELSE                                                        /*                                           */
562         IF ¬RB & SUBSTR(TTFT(2,1),1,4)='    ' THEN                       /* -> TEST FOR STUDENT AT COLLEGE            */
563         DO; IF RPB THEN PUT EDIT(DESNO,'AT COLLEGE')(COL(1),A,X(1),A);   /*    -> CLASS M ERROR/WARNING               */
564             GOTO L21;                                                    /*    -> BRANCH FOR NEXT RECORD              */
565         END; ELSE                                                        /*                                           */
566         IF WB & WCR¬='*' & WCR¬=SUBSTR(TTFT(2,1),6,1) THEN GOTO L21;     /* -> TEST WITHDRAWAL CODE                   */
567         IF OPT=1 THEN                                                    /* OPTION 1                                  */
568         DO; KT=KT+1;                                                     /* -> UPDATE COUNTER                         */
569             IF KT>1000 THEN SUPP='1'B;                                   /* -> TEST COUNTER AND SET SUPPRESSION BIT   */
570             IF ¬SUPP THEN DESL(KT)=DESNO;                                /* -> ASSIGN DES.NO TO LIST                  */
571             CALL RECFREQ;                                                /* -> ACCUMULATE FREQUENCIES                 */
572         END; ELSE                                                        /*                                           */
573         IF OPT=2 THEN                                                    /* OPTION 2                                  */
574         DO; IF LOCTST(LOP,CLAB,RLAB,RITEM,K) THEN                        /* -> TEST REQUEST                           */
575             DO; KT=KT+1;                                                 /* -> UPDATE COUNTER                         */
576             IF KT>1000 THEN SUPP='1'B;                                   /* -> TEST COUNTER AND SET SUPPRESSION BIT   */
577             IF ¬SUPP THEN DESL(KT)=DESNO;                                /* -> ASSIGN DES.NO. TO LIST                 */
578                 IF FB THEN                                               /* -> TEST FOR FREQUENCY TABLES              */
579                 CALL RECFREQ;                                            /* -> ACCUMULATE FREQUENCIES                 */
580             END;                                                         /*                                           */
581         END;                                                             /*                                           */
582         GOTO L21;                                                        /* BRANCH FOR NEXT RECORD                    */
583     /*-----------------------------------------------------------------*/ /*-----------------------------------------*/
584     L00: OPEN FILE(STUDENT) INPUT TITLE(FILN1);                          /* OPTION 2, SPECIAL CASE 'REQUEST='*''      */
585         ON ENDFILE(STUDENT)                                              /* SET ENDFILE (STUDENT) ROUTINE             */
586         BEGIN; CLOSE FILE(STUDENT);                                      /* -> CLOSE FILE                             */
587                OPEN FILE(STUDENT) INPUT TITLE(FILN1);                    /* -> OPEN FILE                              */
588                GOTO L01;                                                 /* -> BRANCH FOR FIRST RECORD                */
589         END;                                                             /* END ENDFILE ROUTINE                       */
590         AGF=0; SXF=0; POF=0; RLF=0; SOF=0; MAF=0; RGF=0; STF=0;          /* INITIALISE VARIABLES                      */
591         FPF=0; OVF=0; SCF=0; OAF=0; ORF=0; OPOCF=0; OSOCF=0;             /*                                           */
592         DESL=' '; KT=0; SUPP='0'B;                                       /*                                           */
593     L02: GET EDIT(DESNO2)(COL(1),A(8));                                  /* INPUT REQUIRED DES.NO.                    */
594         IF DESNO2='********' THEN                                        /* TEST FOR INPUT TERMINATOR                 */
595         DO; PUT EDIT(YE,'FILE........BEDE COLLEGE GENERAL FREQUENCIES',  /* OUTPUT ROUTINE, TITLES                    */
```

Dataset Limited

```
596            ' FOR REQUEST=''*''')(SKIP(2),COL(1),A,X(1),A,A);          /*                                          */
597            PUT EDIT(KT,'STUDENTS INCLUDED')(SKIP(3),COL(1),F(4),X(1),A);  /* -> OUTPUT SAMPLE SIZE 'KT'                  */
598            IF KT=0 THEN GOTO L03;                                     /* -> BRANCH IF KT=0                           */
599            IF ¬SUPP THEN CALL SPNT;                                   /* -> TEST SUPPRESSION BIT, OUTPUT DES.NOS.    */
600            PUT EDIT('FREQUENCY TABLE KEYS')(PAGE,COL(1),A);           /* -> OUTPUT FREQUENCY KEYS TITLE              */
601            CALL REFSQK;                                               /* -> OUTPUT FREQUENCY KEYS                    */
602            PUT EDIT('FREQUENCIES','(TOTALS  PERCENTAGES)')            /* -> OUTPUT FREQUENCY TABLES TITLE            */
603            (PAGE,COL(1),A,X(2),A);                                    /*                                             */
604            CALL RECFPNT;                                              /* -> OUTPUT FREQUENCY TABLES                  */
605    L03:    DT=DATE;                                                   /* -> CALCULATE DATE                           */
606            DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||       /*                                             */
607            SUBSTR(DT,7,2);                                            /*                                             */
608            PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);   /* -> OUTPUT DATE                              */
609            CLOSE FILE(STUDENT);                                       /* -> CLOSE FILE                               */
610            GOTO L39;                                                  /* -> BRANCH TO PROGRAMME CONTROL SECTION      */
611        END;                                                           /* END OUTPUT ROUTINE                          */
612    L01: READ FILE(STUDENT) SET(PTR);                                  /* READ FIRST/NEXT RECORD                      */
613        IF DESNO=DESNO2 THEN                                           /* TEST RECORD DES.NO. WITH REQUIRED DES.NO.   */
614        DO; IF B THEN B='0'B;                                          /* RECORD 'FOUND' ROUTINE, RESET BIT           */
615            KT=KT+1;                                                   /* -> UPDATE COUNTER                           */
616            IF KT>1000 THEN SUPP='1'B;                                 /* -> TEST COUNTER AND SET SUPPRESSION BIT     */
617            IF ¬SUPP THEN DESL(KT)=DESNO;                              /* -> ASSIGN DES.NO. TO LIST                   */
618            CALL RECFREQ;                                              /* -> ACCUMULATE FREQUENCIES                   */
619            GOTO L02;                                                  /* -> BRANCH FOR NEXT REQUIRED DES.NO.         */
620        END; ELSE                                                      /* END RECORD 'FOUND' ROUTINE                  */
621        DO; IF ¬B THEN                                                 /* RECORD 'NOT FOUND' ROUTINE                  */
622            DO; DESNO3=DESNO;                                          /* -> RECORD CURRENT DES.NO.                   */
623                B='1'B;                                                /* -> SET 'RECORD FOUND' BIT                   */
624            END; ELSE                                                  /*                                             */
625            DO; IF DESNO=DESNO3 THEN                                   /* -> TEST FOR COMPLETE FILE CYCLE             */
626                DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')  /* -> CLASS C ERROR/WARNING                   */
627                    (COL(1),A,X(1),A);                                 /*                                             */
628                    B='0'B;                                            /* -> RESET 'RECORD FOUND' BIT                 */
629                    GOTO L02;                                          /* -> BRANCH FOR NEXT REQUIRED DES.NO.         */
630                END;                                                   /*                                             */
631            END;                                                       /*                                             */
632            GOTO L01;                                                  /* -> BRANCH FOR NEXT RECORD                   */
633        END;                                                           /* END RECORD 'NOT FOUND' ROUTINE              */
634    /*-----------------------------------------------------------------*/  /*-----------------------------------------*/
635    L49: PUT PAGE;                                                     /*                                             */
636    END;                                                               /* END PROGRAMME F                             */
637                                                                       /*********************************************/
638   %PROCESS;                                                           /*                                             */
639                                                                       /*********************************************/
640    CON:PROC(FILN2);                                                   /*                                             */
641        DCL FILN2 CHAR(8);                                             /* PROCEDURE CON                               */
642        DCL (I,J,K) FIXED BINARY, AK(3,26) FIXED BINARY EXTERNAL,      /*                                             */
643        (AUTR(*) CHAR(19), AUTR1(*) CHAR(3), ATAB(*) CHAR(3), ATAB1(*)  /*********************************************/
644        FIXED BINARY) CONTROLLED EXTERNAL, (ALPHA CHAR(26), NALC FIXED  /*                                             */
645        BINARY) EXTERNAL;                                              /* VARIABLE DECLARATIONS                       */
646        DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),       /*                                             */
647        MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),  /*                                             */
648        GRK(11) CHAR(30),OAK(4) CHAR(40), ORK(6) CHAR(30), OPSOK(6)     /*                                             */
649        CHAR(40)) VARYING EXTERNAL;                                    /*                                             */
650        DCL NORM(2,16,10) FIXED BINARY EXTERNAL;                       /*                                             */
651        DCL NORM2(2,16,2) FLOAT DECIMAL EXTERNAL;                      /*                                             */
652    /*-----------------------------------------------------------------*/  /*-----------------------------------------*/
653        DCL CONST FILE;                                                /* DECLARATION OF FILE (CONST)                 */
654    /*-----------------------------------------------------------------*/  /*-----------------------------------------*/
655        OPEN FILE(CONST) INPUT TITLE(FILN2);                           /* OPEN CONST FOR INPUT                        */
```

Dataset Limited

```
656          DO I=1 TO 26;                                                  /*                                          */
657             GET FILE(CONST) LIST(AK(1,I));                              /* INPUT AUTHORITY/COUNTRY NUMBERS          */
658          END;                                                           /*                                          */
659          AK(2,1)=1;     AK(3,1)=AK(2,1)+AK(1,1)-1;                      /* -> CALCULATE AK KEYS                     */
660          DO I=2 TO 26;                                                  /*                                          */
661             AK(2,I)=AK(3,I-1)+1;                                        /* -> START POSITIONS                       */
662             AK(3,I)=AK(1,I)+AK(2,I)-1;                                  /* -> LAST POSITIONS                        */
663          END;                                                           /*                                          */
664          N=SUM(AK(1,*));                                                /* -> CALCULATE STORAGE REQUIRED            */
665          ALLOCATE AUTR(N),AUTR1(N);                                     /* -> ALLOCATE STORAGE                      */
666          DO K=1 TO 26;                                                  /*                                          */
667            DO I=AK(2,K) TO AK(3,K);                                     /*                                          */
668             GET FILE(CONST) EDIT(AUTR(I),AUTR1(I))                      /* -> INPUT AUTHORITY/COUNTRY & CODES       */
669             (COL(1),A(19),COL(40),A(3));                                /*                                          */
670             END;                                                        /*                                          */
671          END;                                                           /*                                          */
672          GET FILE(CONST) EDIT((OAK(K) DO K=1 TO 4))(4(COL(1),A(40)));   /* INPUT AREA CODES (PRE 1972)              */
673          GET FILE(CONST) EDIT((OPK(K) DO K=1 TO 6))(6(COL(1),A(30)));   /* INPUT RELIGION CODES (PRE 1972)          */
674          GET FILE(CONST) EDIT((OPSOK(K) DO K=1 TO 6))(6(COL(1),A(40)));  /* INPUT PARENT/STUDENT OCC.CODES (PRE 1972) */
675          GET FILE(CONST) EDIT((SCK(I) DO I=1 TO 9))(9(COL(1),A(60)));   /* INPUT SCHOOL CODES                       */
676          GET FILE(CONST) EDIT((PSOK(I) DO I=1 TO 5))(5(COL(1),A(65)));  /* INPUT PARENT/STUDENT OCC. CODES          */
677          GET FILE(CONST) EDIT((STK(I) DO I=1 TO 6))(6(COL(1),A(50)));   /* INPUT SETTLEMENT CODES                   */
678          GET FILE(CONST) EDIT((MK(I) DO I=1 TO 3))(3(COL(1),A(12)));    /* INPUT MARITAL STATUS CODES               */
679          GET FILE(CONST) EDIT((OK(I) DO I=1 TO 10))(10(COL(1),A(35)));  /* INPUT OVER 25 OCC. CODES                 */
680          GET FILE(CONST) EDIT((RK(I) DO I=1 TO 5))(5(COL(1),A(30)));    /* INPUT RELIGION CODES                     */
681          GET FILE(CONST) EDIT((GRK(I) DO I=1 TO 11))(11(COL(1),A(30)));  /* INPUT REGION CODES                       */
682          GET FILE(CONST) EDIT(ALPHA)(COL(1),A(26));                     /* INPUT ALPHABET                           */
683          GRC=SUBSTR(ALPHA,1,11);                                        /* GEOGRAPHICAL REGION KEYS                 */
684          GET FILE(CONST) EDIT(NALC)(COL(1),F(3));                       /* INPUT A'LEVEL NUMBER                     */
685          ALLOCATE ATAB(NALC),ATAB1(NALC);                               /* -> ALLOCATE STORAGE                      */
686          GET FILE(CONST) SKIP;                                          /*                                          */
687          GET FILE(CONST) EDIT((ATAB(I),ATAB1(I) DO I=1 TO NALC))        /* -> INPUT A'LEVEL CODES & COLUMNS         */
688          (X(2),A(3),X(1),F(2));                                         /*                                          */
689          DO I=1 TO 2;                                                   /*                                          */
690             DO J=1 TO 16;                                               /*                                          */
691                GET FILE(CONST) LIST((NORM(I,J,K) DO K=1 TO 10));        /* INPUT NORM TABLES                        */
692                GET FILE(CONST) LIST((NORM2(I,J,K) DO K=1 TO 2));        /*                                          */
693             END;                                                        /*                                          */
694          END;                                                           /*                                          */
695          CLOSE FILE(CONST);                                             /* CLOSE CONST                              */
696    END;                                                                 /* END PROCEDURE CON                        */
```

Dataset Limited

4.3 COMPILER LISTING

THIS SECTION CONTAINS A LISTING OF THE OUTPUT PROVIDED BY THE PL/1 F COMPILER DURING THE COMPILATION OF PROGRAMME F.

THE JCL WAS AS FOLLOWS:

```
$RUN *PL1 SPUNCH=-F
  ----------------------------
 |                            |
 |                            |
 |  PROGRAMME F  SOURCE DECK  |
 |                            |
 |                            |
  ----------------------------
%PROCESS;
  ----------------------------
 |                            |
 |  PROCEDURE CON SOURCE DECK |
 |                            |
  ----------------------------
$ENDFILE
```

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRO
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST
```

```
*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
*OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
*OPTIONS IN EFFECT*     NEST,OPLIST
```

Datesat Limited

```
STMT LEVEL NEST
                                                                                F0000010
  1                 (SUBRG):(STRG):                                             F0000020
                    F:PROCEDURE OPTIONS(MAIN);                                  F0000030
  2     1                DCL 1 RECORD BASED(PTR),                               F0000040
                               2 (DESNO CHAR(8), NAME CHAR(40)),                F0000050
                               2 GENERAL,                                       F0000060
                                 3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),F0000070
                                 3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),      F0000080
                                 3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)), F0000090
                                 3 FPOS CHAR(4),                                F0000100
                               2 EDUCATION,                                     F0000110
                                 3 (SCHNO CHAR(2), SCH(10) CHAR(5)),            F0000120
                                 3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),         F0000130
                                 3 (UNCOL(2) CHAR(21), FPT(2) CHAR(1), FT(2,2) CHAR(6)),   F0000140
                                 3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),   F0000150
                                 3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),          F0000160
                                 3 (ADIP CHAR(30), ADIPC CHAR(1)),              F0000170
                                 3 DDIPC CHAR(1),                               F0000180
                                 3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),            F0000190
                                 3 ATO CHAR(1),                                 F0000200
                               2 COURSE,                                        F0000210
                                 3 (LEN CHAR(1), CODES(4) CHAR(3)),             F0000220
                                 3 ((LE,DC(4,24:64)) CHAR(1)),                  F0000230
                                 3 (DC1(4,26:30) CHAR(2)),                      F0000240
                                 3 ((DC46(4),DC49(4)) CHAR(3)),                 F0000250
                               2 FINALS,                                        F0000260
                                 3 ((TC,WS,BE,SUBCAR) CHAR(1)),                 F0000270
                               2 ANNPROG,                                       F0000280
                                 3 AP CHAR(802),                                F0000290
                                 3 ((LEAP,FK(100,2)) FIXED BINARY),             F0000300
                               2 INTEL,                                         F0000310
                                 3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY), F0000320
                                 3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),        F0000330
                                 3 STEN(2,16) FIXED BINARY;                     F0000340
                    /*-----------------------------------------------------------------*/F0000350
  3     1                DCL (AGF(3),SXF(2),(POF,RLF)(5),SOF(6),MAF(3),RGF(11),STF(6),     F0000360
                         FPF(10,10),OVF(0:10),SCF(0:9)) FIXED BIN;              F0000370
  4     1                DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),          F0000380
                         MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),    F0000390
                         GRK(11) CHAR(30)) VARYING EXTERNAL;                    F0000400
  5     1                DCL (OAK(4) CHAR(40), ORK(6) CHAR(30)) VARYING EXTERNAL;          F0000410
  6     1                DCL (OAF(4),ORF(6)) FIXED BINARY;                      F0000420
  7     1                DCL OPSOK(6) CHAR(40) VARYING EXTERNAL;                F0000430
  8     1                DCL (OPOCF(6),OSOCF(7)) FIXED BINARY;                  F0000440
  9     1                DCL (CLAB,RLAB)(20) FIXED BINARY,RITEM(20) CHAR(20)    F0000450
                         VARYING, LOP(20) CHAR(1);                              F0000460
 10     1                DCL (FILN1,FILN2,WD,DT,DESL(1000)) CHAR(8), YE CHAR(4), WRL       F0000470
                         CHAR(3), (WCR,FREQ) CHAR(1), (WB,RB,LB,FB) BIT(1),(KT,OPT,K) FIXEDF0000480
                         BINARY, REQUEST CHAR(400) VARYING, BCON BIT(1) INITIAL('1'B),     F0000490
                         (DESNO2,DESNO3) CHAR(8), (SUPP,WPB,LPB,RPB,B) BIT(1);  F0000500
 11     1                DCL CON ENTRY(CHAR(8));                                F0000510
                    /*-----------------------------------------------------------------*/F0000520
 12     1                DCL STUDENT FILE RECORD ENV(F(2416));                  F0000530
                    /*****************************************************************/F0000540
 13     1           TRANS:PROC(REQUEST,LOP,CLAB,RLAB,RITEM,K);                  F0000550
 14     2                DCL (CLAB,RLAB)(20) FIXED BINARY,RITEM(20) CHAR(20)    F0000560
```

Dataset Limited

```
STMT LEVEL NEST

                        VARYING, LOP(20) CHAR(1);                                   F0000570
 15     2               DCL REQUEST CHAR(400) VARYING;                              F0000580
 16     2               DCL PCOND(6) CHAR(2) VARYING INITIAL('=','¬=','>','¬>','<','¬<'); F0000590
 17     2               DCL X1 CHAR(1),LB CHAR(3),(I,J,K,LK,LL,LB,NB) FIXED BINARY; F0000600
 18     2               LL=LENGTH(REQUEST);                                         F0000610
 19     2               K=0;  LK=1;                                                 F0000620
 21     2               DO I=1 TO LL;                                               F0000630
 22     2    1             X1=SUBSTR(REQUEST,I,1);                                 F0000640
 23     2    1             IF X1=' ' THEN                                          F0000650
 24     2    1             DO; LB=I;                                               F0000660
 26     2    2                 GOTO L61;                                           F0000670
 27     2    2             END; ELSE                                               F0000680
 28     2    1             IF X1='&' | X1='|' THEN                                 F0000690
 29     2    1             DO; LK=LK+1;                                            F0000700
 31     2    2                 LOP(LK)=X1;                                         F0000710
 32     2    2                 GOTO L61;                                           F0000720
 33     2    2             END; ELSE                                               F0000730
 34     2    1             DO J=1,3,5;                                             F0000740
 35     2    2                IF X1=PCOND(J) THEN GOTO L62;                        F0000750
 37     2    2             END;                                                    F0000760
 38     2    1             GOTO L61;                                               F0000770
 39     2    1     L62:    NB=INDEX(SUBSTR(REQUEST,I),' ');                        F0000780
 40     2    1             IF SUBSTR(REQUEST,I-1,1)='¬' THEN                       F0000790
 41     2    1             DO; K=K+1;                                              F0000800
 43     2    2                 CLAB(K)=J+1;                                        F0000810
 44     2    2                 LR=SUBSTR(REQUEST,LB+1,I-LB-2);                     F0000820
 45     2    2                 RLAB(K)=LR;                                 /* CONVERSION */F0000830
 46     2    2             END; ELSE                                               F0000840
 47     2    1             DO; K=K+1;                                              F0000850
 49     2    2                 CLAB(K)=J;                                          F0000860
 50     2    2                 LR=SUBSTR(REQUEST,LB+1,I-LB-1);                     F0000870
 51     2    2                 RLAB(K)=LR;                                 /* CONVERSION */F0000880
 52     2    2             END;                                                    F0000890
 53     2    1             RITEM(K)=SUBSTR(REQUEST,I+1,NB-2);                      F0000900
 54     2    1             I=I+NB-1;                                               F0000910
 55     2    1     L61: END;                                                       F0000920
 56     2               DO I=1 TO K;                                                F0000930
 57     2    1             IF RLAB(I)<1 | RLAB(I)>40 THEN                          F0000940
 58     2    1             DO; PUT EDIT('ERROR.....ITEM',RLAB(I),'NOT INCLUDED')   F0000950
                                (SKIP,COL(1),A,X(1),F(2),X(1),A);                  F0000960
 60     2    2                 DO J=I TO K-1;                                      F0000970
 61     2    3                    LOP(J)=LOP(J+1);                                 F0000980
 62     2    3                    RLAB(J)=RLAB(J+1);                               F0000990
 63     2    3                    CLAB(J)=CLAB(J+1);                               F0001000
 64     2    3                    RITEM(J)=RITEM(J+1);                             F0001010
 65     2    3                    K=K-1;                                           F0001020
 66     2    3                 END;                                                F0001030
 67     2    2             END;                                                    F0001040
 68     2    1          END;                                                       F0001050
 69     2          END;                                                            F0001060
                    /*****************************************************************/F0001070
 70     1          LOCTST:PROC(LOP,CLAB,RLAB,RITEM,K) RETURNS(BIT(1));             F0001080
 71     2               DCL LOP(20) CHAR(1),(CLAB,RLAB)(20) FIXED BINARY, RITEM(20) F0001090
                        CHAR(20) VARYING;                                          F0001100
```

Dataset Limited

```
STMT LEVEL NEST

 72     2              DCL REQI CHAR(12), (PEQM,ITEM) CHAR(20) VARYING, (K,I,J,QS)       F0001110
                       FIXED BINARY;                                                     F0001120
                       /*****************************************************************/F0001130
 73     2         FIND:PROCEDURE(ITEM,LR,REQI);                                          F0001140
 74     3              DCL (ITEM CHAR(20), XVAR CHAR(3)) VARYING, (LR,KC,I,J) FIXED      F0001150
                       BINARY, X2 CHAR(2), R(40) LABEL, REQI CHAR(12), ST CHAR(1) DEFINEDF0001160
                       REQI POS(2), SYF CHAR(2) DEFINED REQI POS(3), SYT CHAR(2) DEFINED F0001170
                       REQI POS(5), QY CHAR(2) DEFINED REQI POS(2), QE CHAR(3) DEFINED   F0001180
                       REQI POS(4), QC CHAR(3) DEFINED REQI POS(7), QG CHAR(1) DEFINED   F0001190
                       REQI POS(10), QR CHAR(1) DEFINED REQI POS(11);                    F0001200
 75     3              IF QE¬='***' THEN QE=TRANSLATE(QE,' ','*');                       F0001210
 77     3              IF LR>=26 & LR<=35 THEN I=LE;                        /* CONVERSION */F0001220
 79     3              GOTO R(LR);                                                       F0001230
 80     3              R(1):X2=SUBSTR(TTFT(1,1),5,2); I=X2;                 /* CONVERSION */F0001240
 82     3              X2=SUBSTR(DOB,5,2); J=X2;                            /* CONVERSION */F0001250
 84     3              KC=I-J;                                                           F0001260
 85     3              GOTO L1;                                                          F0001270
 86     3              R(2):ITEM=SEX; RETURN;  R(3):ITEM=MART; RETURN;                   F0001280
 90     3              R(4):ITEM=SET; RETURN;  R(5):ITEM=REG; RETURN;                    F0001290
 94     3              R(6):ITEM=POCC; RETURN; R(7):ITEM=SOCC; RETURN;                   F0001300
 98     3              R(8):ITEM=OV25; RETURN; R(9):ITEM=REL; RETURN;                    F0001310
102     3              R(10):ITEM=CONF; RETURN;  R(11):ITEM=AUTH; RETURN;                F0001320
106     3              R(12):ITEM=AUTHNO; RETURN;  R(13):ITEM=FPOS; RETURN;              F0001330
110     3              R(14):ITEM=TTFT(1,1); RETURN;  R(15):ITEM=UNCOL(1); RETURN;       F0001340
114     3              R(16):ITEM=UNOTH(1); RETURN;  R(17):ITEM=DEG(1); RETURN;          F0001350
118     3              R(18):ITEM=SCHNO; RETURN;                                         F0001360
120     3              R(19):KC=0;  QS=SCHNO;                               /* CONVERSION */F0001370
122     3              DO I=1 TO QS;                                                     F0001380
123     3     1           IF ST='*' | SUBSTR(SCH(I),1,1)=ST THEN                         F0001390
124     3     1           IF SYF='**' | SUBSTR(SCH(I),2,2)=SYF THEN                      F0001400
125     3     1           IF SYT='**' | SUBSTR(SCH(I),4,2)=SYT THEN KC=KC+1;             F0001410
127     3     1        END;                                                              F0001420
128     3              GOTO L1;                                                          F0001430
129     3              R(20):ITEM=QUALNO; RETURN;                                        F0001440
131     3              R(21):KC=0; QS=QUALNO;                               /* CONVERSION */F0001450
133     3              DO I=1 TO QS;                                                     F0001460
134     3     1           IF QY='**' | SUBSTR(QUAL(I),1,2)=QY THEN                       F0001470
135     3     1           IF QE='***' | SUBSTR(QUAL(I),3,3)=QE THEN                      F0001480
136     3     1           IF QC='***' | SUBSTR(QUAL(I),6,3)=QC THEN                      F0001490
137     3     1           IF QG='*' | SUBSTR(QUAL(I),9,1)=QG THEN                        F0001500
138     3     1           IF QR='*' | SUBSTR(QUAL(I),10,1)=QR THEN KC=KC+1;              F0001510
140     3     1        END;                                                              F0001520
141     3              GOTO L1;                                                          F0001530
142     3              R(22):ITEM=ATO; RETURN;  R(23):ITEM=LEN; RETURN;                  F0001540
146     3              R(24):I=ST;                                          /* CONVERSION */F0001550
147     3              ITEM=CODES(I); RETURN;  R(25):ITEM=DC46(1); RETURN;               F0001560
151     3              R(26):ITEM=DC(I,25); RETURN; R(27):ITEM=DC(I,34); RETURN;         F0001570
155     3              R(28):ITEM=DC(I,35); RETURN; R(29):ITEM=DC(I,36); RETURN;         F0001580
159     3              R(30):ITEM=DC(I,37); RETURN; R(31):ITEM=DC(I,38); RETURN;         F0001590
163     3              R(32):ITEM=DC(I,41); RETURN; R(33):ITEM=DC(I,42); RETURN;         F0001600
167     3              R(34):ITEM=DC(I,59); RETURN; R(35):ITEM=DC(I,64); RETURN;         F0001610
171     3              R(36):XVAR=SUBSTR(REQI,5,INDEX(REQI,')')-5);                      F0001620
172     3              I=XVAR;                                              /* CONVERSION */F0001630
173     3              IF FK(I,1)¬=0 THEN ITEM=SUBSTR(AP,FK(I,1),FK(I,2)-1); ELSE        F0001640
```

Dataset Limited

```
STMT LEVEL NEST

175     3                ITEM=' '; RETURN;                                          F0001650
177     3                R(37):ITEM=TC; RETURN;                                     F0001660
179     3                R(38):ITEM=WS; RETURN;                                     F0001670
181     3                R(39):ITEM=BE; RETURN;                                     F0001680
183     3                R(40):ITEM=SUBCAR; RETURN;                                 F0001690
185     3           L1:  ITEM=CHAR(KC);                                             F0001700
186     3           END;                                                            F0001710
                         /*****************************************************/F0001720
187     2           TEST:PROC(ITEM,COND,REQU) RETURNS(BIT(1));                      F0001730
188     3                DCL (ITEM,REQU,X20) CHAR(20) VARYING, COND FIXED BINARY;   F0001740
189     3                DCL C(6) LABEL INITIAL(C1,C2,C3,C4,C5,C6);                 F0001750
190     3                X20=ITEM;  ITEM=((20)' ');  SUBSTR(ITEM,21-LENGTH(X20))=X20;  F0001760
193     3                X20=REQU;  REQU=((20)' ');  SUBSTR(REQU,21-LENGTH(X20))=X20;  F0001770
196     3                GOTO C(COND);                                              F0001780
197     3           C1:  IF ITEM=REQU THEN RETURN('1'B); ELSE RETURN('0'B);         F0001790
200     3           C2:  IF ITEM¬=REQU THEN RETURN('1'B); ELSE RETURN('0'B);        F0001800
203     3           C3:  IF ITEM>REQU THEN RETURN('1'B); ELSE RETURN('0'B);         F0001810
206     3           C4:  IF ITEM¬>REQU THEN RETURN('1'B); ELSE RETURN('0'B);        F0001820
209     3           C5:  IF ITEM<REQU THEN RETURN('1'B); ELSE RETURN('0'B);         F0001830
212     3           C6:  IF ITEM¬<REQU THEN RETURN('1'B); ELSE RETURN('0'B);        F0001840
215     3           END;                                                            F0001850
215     2           DCL TEST ENTRY(,,) RETURNS(BIT(1));                             F0001860
                         /*****************************************************/F0001870
217     2                DO I=1 TO K;                                               F0001880
218     2    1              IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19  F0001890
                            | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN  F0001900
219     2    1              REQI=SUBSTR(RITEM(I),INDEX(RITEM(I),'('));              F0001910
220     2    1              IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=36  F0001920
221     2    1              | RLAB(I)=40 THEN IF SUBSTR(REQI,2,2)¬=SUBSTR(TTFT(1,1),5,2)  F0001930
222     2    1              THEN IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);F0001940
225     2    1              CALL FIND(ITEM,RLAB(I),REQI);                           F0001950
226     2    1              IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19  F0001960
                            | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN  F0001970
227     2    1              REQM=SUBSTR(RITEM(I),1,INDEX(RITEM(I),'(')-1); ELSE     F0001980
228     2    1              REQM=TRANSLATE(RITEM(I),' ',':');                       F0001990
229     2    1              IF TEST(ITEM,CLAB(I),REQM) THEN                         F0002000
230     2    1              DO; DO J=I+1 TO K;                                      F0002010
232     2    3                     IF LOP(J)¬='|' THEN                              F0002020
233     2    3                     DO; I=J-1;                                       F0002030
235     2    4                         GOTO L66;                                    F0002040
236     2    4                     END;                                             F0002050
237     2    3                 END;                                                 F0002060
238     2    2                 RETURN('1'B);                                        F0002070
239     2    2              END; ELSE                                               F0002080
240     2    1              IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);  F0002090
243     2    1       L66: END;                                                      F0002100
244     2            END;                                                           F0002110
245     1            DCL LOCTST ENTRY(,,,,) RETURNS(BIT(1));                        F0002120
                     /*****************************************************/F0002130
246     1            SPNT:PROC;                                                     F0002140
247     2                 DCL I FIXED BINARY;                                       F0002150
248     2                 PUT SKIP(3);                                              F0002160
249     2                 DO I=1 TO KT;                                             F0002170
250     2    1              IF MOD(I-1,10)=0 THEN PUT SKIP;                         F0002180
```

Dataset Limited

```
STMT LEVEL NEST

252     2    1            PUT EDIT(DESL(I))(X(3),A(8));                                F0002190
253     2    1         END;                                                            F0002200
254     2           END;                                                               F0002210
                    /*****************************************************************/F0002220
255     1           RFREQK:PROCEDURE;                                                  F0002230
256     2              DCL K FIXED BINARY;                                             F0002240
257     2              PUT EDIT('MARITAL STATUS')(SKIP(3),A);                          F0002250
258     2              PUT SKIP;                                                       F0002260
259     2              DO K=1 TO 3;                                                    F0002270
260     2    1            PUT EDIT(K,MK(K))(SKIP,F(1),COL(8),A);                       F0002280
261     2    1         END;                                                            F0002290
262     2              PUT EDIT('GEOGRAPHICAL REGION','NOT CODED PRIOR TO 1972')       F0002300
                       (SKIP(3),A,COL(80),A);                                          F0002310
263     2              PUT SKIP;                                                       F0002320
264     2              DO K=1 TO 11;                                                   F0002330
265     2    1            PUT EDIT(SUBSTR(GRC,K,1),GRK(K))(SKIP,A(1),COL(8),A);        F0002340
266     2    1         END;                                                            F0002350
267     2              PUT EDIT('SETTLEMENT','SETTLEMENT PRIOR TO 1972')               F0002360
                       (SKIP(3),A,COL(80),A);                                          F0002370
268     2              PUT SKIP;                                                       F0002380
269     2              DO K=1 TO 4;                                                    F0002390
270     2    1            PUT EDIT(K,STK(K),K,OAK(K))(SKIP,F(1),COL(8),A,COL(80),F(1),F0002400
                       COL(87),A);                                                     F0002410
271     2    1         END;                                                            F0002420
272     2              PUT EDIT('5',STK(5),'6',STK(6))(2(SKIP,A(1),COL(8),A));         F0002430
273     2              PUT EDIT('OCCUPATION IF OVER 25')(SKIP(3),A);                   F0002440
274     2              PUT EDIT('0',OK(10))(SKIP(2),A(1),COL(8),A);                    F0002450
275     2              DO K=1 TO 9;                                                    F0002460
276     2    1            PUT EDIT(K,OK(K))(SKIP,F(1),COL(8),A);                       F0002470
277     2    1         END;                                                            F0002480
278     2              PUT EDIT('STUDENT/PARENT/GUARDIAN OCCUPATION',                  F0002490
                       'STUDENT/PARENT/GUARDIAN OCCUPATION PRIOR TO 1972')             F0002500
                       (PAGE,SKIP(3),A,COL(80),A);                                     F0002510
279     2              PUT SKIP;                                                       F0002520
280     2              DO K=1 TO 5;                                                    F0002530
281     2    1            PUT EDIT(K,PSOK(K),K,OPSOK(K))(SKIP,F(1),COL(8),A,COL(80),F(1),F0002540
                       COL(87),A);                                                     F0002550
282     2    1         END;                                                            F0002560
283     2              PUT EDIT('6',OPSOK(K))(SKIP,COL(80),A,COL(87),A);               F0002570
284     2              PUT EDIT('RELIGION','RELIGION PRIOR TO 1972')                   F0002580
                       (SKIP(3),A,COL(80),A);                                          F0002590
285     2              PUT SKIP;                                                       F0002600
286     2              DO K=1 TO 5;                                                    F0002610
287     2    1            PUT EDIT(K,RK(K),K,ORK(K))(SKIP,F(1),COL(8),A,COL(80),F(1),F0002620
                       COL(87),A);                                                     F0002630
288     2    1         END;                                                            F0002640
289     2              PUT EDIT('6',ORK(6))(SKIP,COL(80),A,COL(87),A);                 F0002650
290     2              PUT EDIT('SCHOOL TYPE')(SKIP(3),A);                             F0002660
291     2              PUT SKIP;                                                       F0002670
292     2              DO K=1 TO 9;                                                    F0002680
293     2    1            PUT EDIT(K,SCK(K))(SKIP,F(1),COL(8),A);                      F0002690
294     2    1         END;                                                            F0002700
295     2           END;                                                               F0002710
                    /*****************************************************************/F0002720
```

Dataset Limited

```
STMT LEVEL NEST

296     1          RECFREQ:PROC;                                                    F0002730
297     2               DCL X2 CHAR(2), X1 CHAR(1), OKB BIT(1), (I,J,K) FIXED BINARY;  F0002740
298     2               X2=SUBSTR(TTFT(1,1),5,2);  I=X2;                 /* CONVERSION */F0002750
300     2               IF I<72 THEN OKB='1'B; ELSE OKB='0'B;                          F0002760
303     2               X2=SUBSTR(DOB,5,2);                                            F0002770
304     2               J=X2;                                            /* CONVERSION */F0002780
305     2               IF I>J THEN                                                    F0002790
306     2               DO; IF I-J>=25 THEN AGF(1)=AGF(1)+1; ELSE AGF(2)=AGF(2)+1;     F0002800
310     2    1              AGF(3)=AGF(3)+I-J;                                         F0002810
311     2    1          END; ELSE                                                      F0002820
312     2               PUT EDIT(DESNO,'ERROR RETURNED.....AGE')(SKIP,A(8),X(2),A);    F0002830
313     2               IF SEX='M' THEN SXF(1)=SXF(1)+1; ELSE                          F0002840
315     2               IF SEX='F' THEN SXF(2)=SXF(2)+1; ELSE                          F0002850
317     2               PUT EDIT(DESNO,'ERROR RETURNED.....SEX')(SKIP,A(8),X(2),A);    F0002860
318     2               IF MART>='1' & MART<='3' THEN                                  F0002870
319     2               DO; I=MART;                                      /* CONVERSION */F0002880
321     2    1              MAF(I)=MAF(I)+1;                                           F0002890
322     2    1          END; ELSE                                                      F0002900
323     2               PUT EDIT(DESNO,'ERROR RETURNED.....MARITAL STATUS')            F0002910
                        (SKIP,A(8),X(2),A);                                            F0002920
324     2               IF REL>='1'  & REL<='5' & ¬OKB THEN                            F0002930
325     2               DO; I=REL;                                       /* CONVERSION */F0002940
327     2    1              RLF(I)=RLF(I)+1;                                           F0002950
328     2    1          END; ELSE                                                      F0002960
329     2               IF REL>='1' & REL<='6' & OKB THEN                              F0002970
330     2               DO; I=REL;                                       /* CONVERSION */F0002980
332     2    1              ORF(I)=ORF(I)+1;                                           F0002990
333     2    1          END; ELSE                                                      F0003000
334     2               PUT EDIT(DESNO,'ERROR RETURNED.....RELIGION')(SKIP,A(8),X(2),A);  F0003010
335     2               IF SET>='1' & SET<='6' & ¬OKB THEN                             F0003020
336     2               DO; I=SET;                                       /* CONVERSION */F0003030
338     2    1              STF(I)=STF(I)+1;                                           F0003040
339     2    1          END; ELSE                                                      F0003050
340     2               IF SET>='1' & SET<='4' & OKB THEN                              F0003060
341     2               DO; I=SET;                                       /* CONVERSION */F0003070
343     2    1              OAF(I)=OAF(I)+1;                                           F0003080
344     2    1          END; ELSE                                                      F0003090
345     2               PUT EDIT(DESNO,'ERROR RETURNED.....SETTLEMENT')                F0003100
                        (SKIP,A(8),X(2),A);                                            F0003110
346     2               IF POCC>='1' & POCC<='5' & ¬OKB THEN                           F0003120
347     2               DO; I=POCC;                                      /* CONVERSION */F0003130
349     2    1              POF(I)=POF(I)+1;                                           F0003140
350     2    1          END; ELSE                                                      F0003150
351     2               IF POCC>='1' & POCC<='6' & OKB THEN                            F0003160
352     2               DO; I=POCC;                                      /* CONVERSION */F0003170
354     2    1             OPOCF(I)=OPOCF(I)+1;                                        F0003180
355     2    1          END; ELSE                                                      F0003190
356     2               PUT EDIT(DESNO,'ERROR RETURNED.....PARENT/GUARDIAN OCCUPATION')  F0003200
                        (SKIP,A(8),X(2),A);                                            F0003210
357     2               IF SOCC=' ' THEN IF ¬OKB THEN SOF(6)=SOF(6)+1; ELSE            F0003220
360     2               OSOCF(7)=OSOCF(7)+1; ELSE                                      F0003230
361     2               IF SOCC>='1' & SOCC<='5' & ¬OKB THEN                           F0003240
362     2               DO; I=SOCC;                                      /* CONVERSION */F0003250
364     2    1              SOF(I)=SOF(I)+1;                                           F0003260
```

Dataset Limited

```
STMT LEVEL NEST

365     2    1          END; ELSE                                                    F0003270
366     2               IF SOCC>='1' & SOCC<='6' & OKB THEN                          F0003280
367     2               DO; I=SOCC;                                  /* CONVERSION */F0003290
369     2    1              OSOCF(I)=OSOCF(I)+1;                                     F0003300
370     2    1          END; ELSE                                                    F0003310
371     2               PUT EDIT(DESNO,'ERROR RETURNED.....STUDENT OCCUPATION')      F0003320
                        (SKIP,A(8),X(2),A);                                          F0003330
372     2               IF OV25=' ' THEN OVF(10)=OVF(10)+1; ELSE                     F0003340
374     2               IF OV25>='0' & OV25<='9' THEN                                F0003350
375     2               DO; I=OV25;                                  /* CONVERSION */F0003360
377     2    1              OVF(I)=OVF(I)+1;                                         F0003370
378     2    1          END; ELSE                                                    F0003380
379     2               PUT EDIT(DESNO,'ERROR RETURNED.....OVER 25 OCCUPATION')      F0003390
                        (SKIP,A(8),X(2),A);                                          F0003400
380     2               IF OKB THEN GOTO L3;                                         F0003410
382     2               I=INDEX(GRC,REG);                                            F0003420
383     2               IF I>=1 & I<=11 THEN RGF(I)=RGF(I)+1; ELSE                   F0003430
385     2               PUT EDIT(DESNO,'ERROR RETURNED.....REGION')(SKIP,A(8),X(2),A);  F0003440
386     2          L3:  K=SCHNO;                                     /* CONVERSION */F0003450
387     2               DO J=1 TO K;                                                 F0003460
388     2    1             X1=SUBSTR(SCH(J),1,1);                                   F0003470
389     2    1             IF X1>='0' & X1<='9' THEN                                F0003480
390     2    1             DO; I=X1;                                   /* CONVERSION */F0003490
392     2    2                 SCF(I)=SCF(I)+1;                                     F0003500
393     2    2             END; ELSE                                                F0003510
394     2    1             PUT EDIT(DESNO,'ERROR RETURNED.....SCHOOL TYPE')         F0003520
                           (SKIP,A(8),X(2),A);                                      F0003530
395     2    1          END;                                                         F0003540
396     2               X2=SUBSTR(FPOS,1,2);                                         F0003550
397     2               IF X2='  ' THEN I=100; ELSE I=X2;                /* CONVERSION */F0003560
400     2               X2=SUBSTR(FPOS,3,2);                                         F0003570
401     2               IF X2='  ' THEN J=100; ELSE J=X2;                /* CONVERSION */F0003580
404     2               IF I>=1 & I<=10 & J>=1 & J<=10 THEN FPF(I,J)=FPF(I,J)+1; ELSE F0003590
406     2               PUT EDIT(DESNO,'ERROR RETURNED.....FAMILY POSITION',         F0003600
                        '(',I,'/',J,')')(SKIP,A(8),X(2),A,X(2),2(A,P'99'),A);        F0003610
407     2          END;                                                              F0003620
                   /*****************************************************************/F0003630
408     1          RECFPNT:PROC;                                                     F0003640
409     2               DCL (J,K) FIXED BINARY;                                      F0003650
410     2               PUT EDIT('SEX FREQUENCY')(SKIP(3),A);                        F0003660
411     2               PUT EDIT('MALE',SXF(1),(SXF(1)/KT)*100)(SKIP(2),A,COL(10),F(4),  F0003670
                        COL(16),F(6,2));                                             F0003680
412     2               PUT EDIT('FEMALE',SXF(2),(SXF(2)/KT)*100)(SKIP,A,COL(10),F(4),  F0003690
                        COL(16),F(6,2));                                             F0003700
413     2               PUT EDIT('AGE FREQUENCY ON ADMISSION','AVERAGE AGE ON ADMISS',  F0003710
                        'ION')(SKIP(3),A,COL(80),A,A);                               F0003720
414     2               PUT EDIT('UNDER 25',AGF(2),(AGF(2)/KT)*100,(AGF(3)/KT))      F0003730
                        (SKIP(2),A,COL(10),F(4),COL(16),F(6,2),COL(80),F(2));        F0003740
415     2               PUT EDIT('OVER 25',AGF(1),(AGF(1)/KT)*100)(SKIP,A,COL(10),F(4),  F0003750
                        COL(16),F(6,2));                                             F0003760
416     2               PUT EDIT('MARITAL STATUS')(SKIP(3),A);                       F0003770
417     2               PUT SKIP;                                                    F0003780
418     2               DO K=1 TO 3;                                                 F0003790
419     2    1             PUT EDIT(K,MAF(K),(MAF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),  F0003800
```

Dataset Limited

```
STMT LEVEL NEST

                          COL(16),F(6,2));                                    F0003810
420     2     1         END;                                                   F0003820
421     2               PUT EDIT('GEOGRAPHICAL REGION','NOT CODED PRIOR TO 1972')  F0003830
                        (SKIP(3),A,COL(80),A);                                F0003840
422     2               PUT SKIP;                                              F0003850
423     2               DO K=1 TO 11;                                          F0003860
424     2     1            PUT EDIT(SUBSTR(GRC,K,1),RGF(K),(RGF(K)/KT)*100)(SKIP,A(1),  F0003870
                          COL(10),F(4),COL(16),F(6,2));                       F0003880
425     2     1         END;                                                   F0003890
426     2               PUT EDIT('SETTLEMENT','SETTLEMENT PRIOR TO 1972')      F0003900
                        (SKIP(3),A,COL(80),A);                                F0003910
427     2               PUT SKIP;                                              F0003920
428     2               DO K=1 TO 4;                                           F0003930
429     2     1            PUT EDIT(K,STF(K),(STF(K)/KT)*100,K,OAF(K),(OAF(K)/KT)*100)  F0003940
                        (SKIP,F(1),COL(10),F(4),COL(16),F(6,2),COL(80),F(1),COL(89),  F0003950
                        F(4),COL(94),F(6,2));                                 F0003960
430     2     1         END;                                                   F0003970
431     2               DO K=5,6;                                              F0003980
432     2     1            PUT EDIT(K,STF(K),(STF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),  F0003990
                          COL(16),F(6,2));                                    F0004000
433     2     1         END;                                                   F0004010
434     2               PUT EDIT('OCCUPATION IF OVER 25')(SKIP(3),A);          F0004020
435     2               PUT SKIP;                                              F0004030
436     2               DO K=0 TO 9;                                           F0004040
437     2     1            PUT EDIT(K,OVF(K),(OVF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),  F0004050
                          COL(16),F(6,2));                                    F0004060
438     2     1         END;                                                   F0004070
439     2               PUT EDIT('UNDER 25',OVF(10),(OVF(10)/KT)*100)(SKIP,A,COL(10),F(4),F0004080
                        COL(16),F(6,2));                                      F0004090
440     2               PUT EDIT('STUDENT OCCUPATION','PARENT/GUARDIAN OCCUPATION',  F0004100
                        'STUDENT OCCUPATION','PARENT/GUARDIAN OCCUPATION','PRIOR TO 1972',F0004110
                        'PRIOR TO 1972')(PAGE,SKIP(2),A,COL(27),A,2(COL(65),A,COL(91),A));F0004120
441     2               PUT SKIP;                                              F0004130
442     2               DO K=1 TO 5;                                           F0004140
443     2     1            PUT EDIT(K,SOF(K),(SOF(K)/KT)*100,POF(K),(POF(K)/KT)*100,  F0004150
                          K,OSOCF(K),(OSOCF(K)/KT)*100,OPOCF(K),(OPOCF(K)/KT)*100)  F0004160
                          (SKIP,F(1),COL(10),F(4),COL(16),F(6,2),COL(27),F(4),COL(33),  F0004170
                          F(6,2),COL(65),F(1),COL(73),F(4),COL(79),F(6,2),COL(91),F(4),  F0004180
                          COL(97),F(6,2));                                    F0004190
444     2     1         END;                                                   F0004200
445     2               PUT EDIT('6',OSOCF(6),(OSOCF(6)/KT)*100,OPOCF(6),      F0004210
                        (OPOCF(6)/KT)*100)(COL(65),A,COL(73),F(4),COL(79),F(6,2),  F0004220
                        COL(91),F(4),COL(97),F(6,2));                         F0004230
446     2               PUT EDIT('@ SCHOOL',SOF(6),(SOF(6)/KT)*100,OSOCF(7),   F0004240
                        (OSOCF(7)/KT)*100)(SKIP,A,COL(10),F(4),COL(16),F(6,2),COL(73),  F0004250
                        F(4),COL(79),F(6,2));                                 F0004260
447     2               PUT EDIT('SCHOOL TYPE')(SKIP(3),A);                    F0004270
448     2               PUT SKIP;                                              F0004280
449     2               DO K=1 TO 9;                                           F0004290
450     2     1            PUT EDIT(K,SCF(K),(SCF(K)/KT)*100)(SKIP,F(1),COL(10),F(4),  F0004300
                          COL(16),F(6,2));                                    F0004310
451     2     1         END;                                                   F0004320
452     2               PUT EDIT('UNKNOWN',SCF(0),(SCF(0)/KT)*100)(SKIP,A,COL(10),F(4),  F0004330
                        COL(16),F(6,2));                                      F0004340
```

Dataset Limited

```
STMT LEVEL NEST

453     2              PUT EDIT('RELIGION','RELIGION PRIOR TO 1972')                    F0004350
                       (SKIP(3),A,COL(80),A);                                           F0004360
454     2              PUT SKIP;                                                        F0004370
455     2              DO K=1 TO 5;                                                     F0004380
456     2    1            PUT EDIT(K,RLF(K),(RLF(K)/KT)*100,K,ORF(K),(ORF(K)/KT)*100)   F0004390
                         (SKIP,F(1),COL(10),F(4),COL(16),F(6,2),COL(80),F(1),COL(89),  F0004400
                         F(4),COL(94),F(6,2));                                          F0004410
457     2    1         END;                                                             F0004420
458     2              PUT EDIT('6',ORF(6),(ORF(6)/KT)*100)(SKIP,COL(80),A(1),COL(89),  F0004430
                       F(4),COL(94),F(6,2));                                            F0004440
459     2              PUT EDIT('POSITION IN FAMILY')(SKIP(3),A);                       F0004450
460     2              PUT SKIP;                                                        F0004460
461     2              PUT EDIT('  ')(SKIP,A);                                          F0004470
462     2              DO K=1 TO 10;                                                    F0004480
463     2    1            PUT EDIT(K)(X(3),F(2));                                       F0004490
464     2    1         END;                                                             F0004500
465     2              PUT SKIP;                                                        F0004510
466     2              DO K=1 TO 9;                                                     F0004520
467     2    1            PUT EDIT(K,' ')(SKIP,F(1),A(1));                              F0004530
468     2    1            DO J=1 TO 10;                                                 F0004540
469     2    2               IF J<K THEN PUT EDIT('X')(X(4),A(1)); ELSE                 F0004550
471     2    2               PUT EDIT(FPF(K,J))(X(1),F(4));                             F0004560
472     2    2            END;                                                          F0004570
473     2    1         END;                                                             F0004580
474     2              PUT EDIT('10')(SKIP,A(2));                                       F0004590
475     2              DO K=1 TO 9;                                                     F0004600
476     2    1            PUT EDIT('X')(X(4),A);                                        F0004610
477     2    1         END;                                                             F0004620
478     2              PUT EDIT(FPF(10,10))(X(1),F(4));                                 F0004630
479     2         END;                                                                  F0004640
                  /*****************************************************************/F0004650
480     1              ON ENDFILE(SCARDS) GOTO L49;                                     F0004660
482     1              OPEN FILE(SPRINT) LINESIZE(132);                                 F0004670
483     1         L39: GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));       F0004680
484     1              GET LIST(OPT);                                                   F0004690
485     1              IF OPT<1 | OPT>2 THEN                                            F0004700
486     1              DO; PUT EDIT('OPTION INVALID.....PROGRAM TERMINATED')            F0004710
                           (SKIP,COL(1),A);                                             F0004720
488     1    1             GOTO L49;                                                    F0004730
489     1    1         END;                                                             F0004740
490     1              WRL='XXX'; WCR='X'; FREQ='X'; REQUEST=' '; B='0'B;               F0004750
495     1              GET DATA;                                                        F0004760
496     1              IF SUBSTR(WRL,1,1)='1' THEN WB='1'B; ELSE WB='0'B;               F0004770
499     1              IF SUBSTR(WRL,1,1)='2' THEN WPB='1'B; ELSE WPB='0'B;             F0004780
502     1              IF SUBSTR(WRL,2,1)='1' THEN RB='1'B; ELSE RB='0'B;               F0004790
505     1              IF SUBSTR(WRL,2,1)='2' THEN RPB='1'B; ELSE RPB='0'B;             F0004800
508     1              IF SUBSTR(WRL,3,1)='1' THEN LB='1'B; ELSE LB='0'B;               F0004810
511     1              IF SUBSTR(WRL,3,1)='2' THEN LPB='1'B; ELSE LPB='0'B;             F0004820
514     1              IF FREQ='1' THEN FB='1'B; ELSE FB='0'B;                          F0004830
517     1              PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME F',     F0004840
                       'OPTION REQUESTED',OPT,'YEAR REQUESTED',YE)(PAGE,COL(1),A,       F0004850
                       SKIP(2),COL(1),A,COL(1),A,COL(32),F(2),COL(1),A,COL(30),A);      F0004860
518     1              PUT SKIP(2);                                                     F0004870
519     1              IF WRL¬='XXX' THEN PUT EDIT('WRL REQUESTED',WRL)                 F0004880
```

Dataset Limited

```
STMT LEVEL NEST

521     1              (COL(1),A,COL(31),A); ELSE WRL='GOO';                          F0004890
522     1              IF WCR¬='X' THEN PUT EDIT('WCR REQUESTED',WCR)(COL(1),A,COL(33),  F0004900
524     1              A); ELSE WCR='*';                                              F0004910
525     1              IF FB THEN PUT EDIT('FREQUENCY TABLES REQUIRED')               F0004920
                       (SKIP,COL(1),A);                                               F0004930
527     1              PUT SKIP(4);                                                   F0004940
528     1              IF BCON & (OPT=1 | (OPT=2 & FB) | (OPT=2 & REQUEST='*')) THEN  F0004950
529     1              DO; CALL CON(FILN2);                                           F0004960
531     1    1             BCON='0'B;                                                F0004970
532     1    1         END;                                                           F0004980
533     1              IF REQUEST='*' THEN GOTO L00;                                  F0004990
                 /*-----------------------------------------------------------------*/F0005000
535     1        L12:OPEN FILE(STUDENT) INPUT TITLE(FILN1);                          F0005010
536     1              ON ENDFILE(STUDENT)                                            F0005020
537     1              BEGIN; DT=DATE;                                                F0005030
539     2                     DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||    F0005040
                              SUBSTR(DT,7,2);                                        F0005050
540     2                     IF OPT=1 | OPT=2 THEN                                  F0005060
541     2                     DO; IF OPT=1 THEN PUT EDIT(YE,'FILE........BEDE COLLEGE',  F0005070
                                  ' GENERAL FREQUENCIES')(SKIP(2),COL(1),A(4),X(1),A,A); F0005080
544     2    1                    IF OPT=2 THEN PUT EDIT('FREQUENCY COUNT FOR REQUEST',  F0005090
                                  REQUEST)(SKIP(2),COL(1),A,X(2),A);                 F0005100
546     2    1                    PUT EDIT(KT,'STUDENTS INCLUDED')                   F0005110
                                  (SKIP(3),COL(1),F(4),X(1),A);                      F0005120
547     2    1                    IF KT=0 THEN GOTO L29;                             F0005130
549     2    1                    IF ¬SUPP THEN CALL SPNT;                           F0005140
551     2    1                    IF OPT=2 & ¬FB THEN GOTO L29;                      F0005150
553     2    1                    PUT EDIT('FREQUENCY TABLE KEYS')                   F0005160
                                  (PAGE,COL(1),A);                                   F0005170
554     2    1                    CALL RFREQK;                                       F0005180
555     2    1                    PUT EDIT('FREQUENCIES','(TOTALS  PERCENTAGES)')    F0005190
                                  (PAGE,COL(1),A,X(2),A);                            F0005200
556     2    1                    CALL RECFPNT;                                      F0005210
557     2    1   L29:          END;                                                  F0005220
558     2                     PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);  F0005230
559     2                     CLOSE FILE(STUDENT);                                   F0005240
560     2                     GOTO L39;                                              F0005250
561     2              END;                                                           F0005260
562     1              DESL=' '; KT=0; SUPP='0'B;                                     F0005270
565     1              IF OPT=1 THEN                                                  F0005280
566     1              DO; AGF=0; SXF=0; POF=0; RLF=0; SOF=0; MAF=0; RGF=0; STF=0;    F0005290
575     1    1             FPF=0; OVF=0; SCF=0; OAF=0; ORF=0; OPOCF=0; OSOCF=0;      F0005300
582     1    1         END; ELSE                                                     F0005310
583     1              IF OPT=2 THEN                                                  F0005320
584     1              DO; LOP=' '; CLAB=0; RLAB=0; RITEM=' '; K=0;                   F0005330
590     1    1             IF SUBSTR(REQUEST,2,1)¬=' ' THEN                          F0005340
591     1    1             CALL TRANS(REQUEST,LOP,CLAB,RLAB,RITEM,K);                F0005350
592     1    1             IF K=0 THEN                                               F0005360
593     1    1             DO; CLOSE FILE(STUDENT);                                  F0005370
595     1    2                 GOTO L39;                                             F0005380
596     1    2             END;                                                      F0005390
597     1    1             IF FB THEN                                                F0005400
598     1    1             DO; AGF=0; SXF=0; POF=0; RLF=0; SOF=0; MAF=0; RGF=0; STF=0;  F0005410
607     1    2                 FPF=0; OVF=0; SCF=0; OAF=0; ORF=0; OPOCF=0; OSOCF=0;  F0005420
```

Dataset Limited

```
STMT LEVEL NEST

614     1    2            END;                                                    F0005430
615     1    1        END;                                                        F0005440
616     1         L21: READ FILE(STUDENT) SET(PTR);                              F0005450
617     1             IF ¬WB & SUBSTR(TTFT(2,1),1,4)='WITH' THEN                 F0005460
618     1             DO; IF WPB THEN                                            F0005470
620     1    1            DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),  F0005480
                              3,2) || '/' || SUBSTR(TTFT(1,2),5,2);              F0005490
622     1    2                PUT EDIT(DESNO,'WITHDREW ON',WD)(COL(1),A,2(X(1),A));  F0005500
623     1    2            END;                                                    F0005510
624     1    1            GOTO L21;                                               F0005520
625     1    1        END; ELSE                                                   F0005530
626     1             IF ¬LB & SUBSTR(TTFT(2,1),1,4)='LEFT' THEN                 F0005540
627     1             DO; IF LPB THEN                                            F0005550
629     1    1            DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),  F0005560
                              3,2) || '/' || SUBSTR(TTFT(1,2),5,2);              F0005570
631     1    2                PUT EDIT(DESNO,'LEFT ON',WD)(COL(1),A,2(X(1),A));  F0005580
632     1    2            END;                                                    F0005590
633     1    1            GOTO L21;                                               F0005600
634     1    1        END; ELSE                                                   F0005610
635     1             IF ¬RB & SUBSTR(TTFT(2,1),1,4)='    ' THEN                 F0005620
636     1             DO; IF RPB THEN PUT EDIT(DESNO,'AT COLLEGE')(COL(1),A,X(1),A);  F0005630
639     1    1            GOTO L21;                                               F0005640
640     1    1        END; ELSE                                                   F0005650
641     1             IF WB & WCR¬='*' & WCR¬=SUBSTR(TTFT(2,1),6,1) THEN GOTO L21;  F0005660
643     1             IF OPT=1 THEN                                              F0005670
644     1             DO; KT=KT+1;                                               F0005680
646     1    1            IF KT>1000 THEN SUPP='1'B;                              F0005690
648     1    1            IF ¬SUPP THEN DESL(KT)=DESNO;                           F0005700
650     1    1            CALL RECFREQ;                                           F0005710
651     1    1        END; ELSE                                                   F0005720
652     1             IF OPT=2 THEN                                              F0005730
653     1             DO; IF LOCTST(LOP,CLAB,RLAB,RITEM,K) THEN                  F0005740
655     1    1            DO; KT=KT+1;                                            F0005750
657     1    2            IF KT>1000 THEN SUPP='1'B;                              F0005760
659     1    2            IF ¬SUPP THEN DESL(KT)=DESNO;                           F0005770
661     1    2                IF FB THEN                                          F0005780
662     1    2                CALL RECFREQ;                                       F0005790
663     1    2            END;                                                    F0005800
664     1    1        END;                                                        F0005810
665     1             GOTO L21;                                                  F0005820
                   /*-------------------------------------------------------------*/F0005830
666     1         L00: OPEN FILE(STUDENT) INPUT TITLE(FILN1);                    F0005840
667     1             ON ENDFILE(STUDENT)                                        F0005850
668     1             BEGIN; CLOSE FILE(STUDENT);                                F0005860
670     2                    OPEN FILE(STUDENT) INPUT TITLE(FILN1);              F0005870
671     2                    GOTO L01;                                           F0005880
672     2             END;                                                       F0005890
673     1             AGF=0; SXF=0; POF=0; RLF=0; SOF=0; MAF=0; RGF=0; STF=0;    F0005900
681     1             FPF=0; OVF=0; SCF=0; OAF=0; ORF=0; OPOCF=0; OSOCF=0;       F0005910
688     1             DESL=' '; KT=0; SUPP='0'B;                                 F0005920
691     1         L02: GET EDIT(DESNO2)(COL(1),A(8));                            F0005930
692     1             IF DESNO2='********' THEN                                  F0005940
693     1             DO; PUT EDIT(YE,'FILE........BEDE COLLEGE GENERAL FREQUENCIES',  F0005950
                          ' FOR REQUEST=''*''')(SKIP(2),COL(1),A,X(1),A,A);      F0005960
```

Dataset Limited

```
STMT LEVEL NEST

695     1    1              PUT EDIT(KT,'STUDENTS INCLUDED')(SKIP(3),COL(1),F(4),X(1),A); F0005970
696     1    1              IF KT=0 THEN GOTO L03;                                        F0005980
698     1    1              IF ¬SUPP THEN CALL SPNT;                                      F0005990
700     1    1              PUT EDIT('FREQUENCY TABLE KEYS')(PAGE,COL(1),A);              F0006000
701     1    1              CALL RFREQK;                                                  F0006010
702     1    1              PUT EDIT('FREQUENCIES','(TOTALS  PERCENTAGES)')               F0006020
                             (PAGE,COL(1),A,X(2),A);                                      F0006030
703     1    1              CALL RECFPNT;                                                 F0006040
704     1    1       L03:   DT=DATE;                                                      F0006050
705     1    1              DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||          F0006060
                             SUBSTR(DT,7,2);                                              F0006070
706     1    1              PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);      F0006080
707     1    1              CLOSE FILE(STUDENT);                                          F0006090
708     1    1              GOTO L39;                                                     F0006100
709     1    1          END;                                                              F0006110
710     1           L01: READ FILE(STUDENT) SET(PTR);                                     F0006120
711     1               IF DESNO=DESNO2 THEN                                              F0006130
712     1               DO; IF B THEN B='0'B;                                             F0006140
715     1    1              KT=KT+1;                                                      F0006150
716     1    1              IF KT>1000 THEN SUPP='1'B;                                    F0006160
718     1    1              IF ¬SUPP THEN DESL(KT)=DESNO;                                 F0006170
720     1    1              CALL RECFREQ;                                                 F0006180
721     1    1              GOTO L02;                                                     F0006190
722     1    1          END; ELSE                                                         F0006200
723     1               DO; IF ¬B THEN                                                    F0006210
725     1    1              DO; DESNO3=DESNO;                                             F0006220
727     1    2                  B='1'B;                                                   F0006230
728     1    2              END; ELSE                                                     F0006240
729     1    1              DO; IF DESNO=DESNO3 THEN                                      F0006250
731     1    2                  DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')    F0006260
                                     (COL(1),A,X(1),A);                                   F0006270
733     1    3                      B='0'B;                                               F0006280
734     1    3                      GOTO L02;                                             F0006290
735     1    3                  END;                                                      F0006300
736     1    2              END;                                                          F0006310
737     1    1              GOTO L01;                                                     F0006320
738     1    1          END;                                                              F0006330
                    /*-------------------------------------------------------------------*/F0006340
739     1           L49: PUT PAGE;                                                        F0006350
740     1           END;                                                                  F0006360
                                                                                          F0006370
```

Dataset Limited

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
2	ADIP	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER
2	ADIPC	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
3 *********	AGF	(3)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 308,308,309,309,310,310,414,414,414,415,415,567,599,673
2 *********	AH4	(2,3)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
2	ANNPROG	IN RECORD,BASED(PTR),STRUCTURE
2	AP	IN ANNPROG IN RECORD,BASED(PTR),UNALIGNED,STRING(802),CHARACTER 174
2	ATO	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 142
2	AUTH	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(19),CHARACTER 104
2	AUTHNO	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 106
10	B	AUTOMATIC,UNALIGNED,STRING(1),BIT 494,713,714,724,727,733
10	BCON	AUTOMATIC,UNALIGNED,INITIAL,STRING(1),BIT 528,531
2	BE	IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 181
189	C	(6)AUTOMATIC,INITIAL,LABEL 196
197	C1	STATEMENT LABEL CONSTANT
200	C2	STATEMENT LABEL CONSTANT
203	C3	STATEMENT LABEL CONSTANT
206	C4	STATEMENT LABEL CONSTANT
209	C5	STATEMENT LABEL CONSTANT
212	C6	STATEMENT LABEL CONSTANT
	CHAR	GENERIC,BUILT-IN FUNCTION 185
71 *********	CLAB	(20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               70,229

14     ********* CLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               13,43,49,63,63

9      ********* CLAB                          (20)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               586,591,654

2                CODES                         (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                               147

11               CON                           EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE)
                                               530

188    ********* COND                          PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               187,196

2                CONF                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               102

2                COURSE                        IN RECORD,BASED(PTR),STRUCTURE

                 DATE                          BUILT-IN FUNCTION
                                               538,704

2                DATT                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER

2                DC                            (4,24:64)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),
                                               CHARACTER
                                               151,153,155,157,159,161,163,165,167,169

2                DC1                           (4,26:30)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(2),
                                               CHARACTER

2                DC46                          (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                               149

2                DC49                          (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER

2                DDIPC                         IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

2                DEG                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                               116

10               DESL                          (1000)AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               252,562,649,660,688,719

2                DESNO                         IN RECORD,BASED(PTR),UNALIGNED,STRING(8),CHARACTER
                                               312,317,323,334,345,356,371,379,385,394,406,622,631,638,649,660,711
                                               719,726,730

10               DESNO2                        AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               691,692,711,732
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

10               DESNO3                        AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               726,730

2                DOB                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               82,303

2                DOBC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

2      ********* DSCALE                        (2)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

10               DT                            AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               538,539,539,539,539,558,704,705,705,705,705,706

2                EDUCATION                     IN RECORD,BASED(PTR),STRUCTURE

1                F                             ENTRY,DECIMAL,FLOAT(SINGLE)

10               FB                            AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               515,516,525,528,551,597,661

10               FILN1                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               483,535,666,670

10               FILN2                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               483,530

2                FINALS                        IN RECORD,BASED(PTR),STRUCTURE

73               FIND                          ENTRY,DECIMAL,FLOAT(SINGLE)
                                               225

2      ********* FK                            (100,2)IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED
                                               (15,0)
                                               173,174,174

3      ********* FPF                           (10,10)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               405,405,471,478,575,607,681

2                FPOS                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER
                                               108,396,400

2                FPT                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

10               FREQ                          AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               492,514

2                FT                            (2,2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER

2                GENERAL                       IN RECORD,BASED(PTR),STRUCTURE

4                GRC                           STATIC,EXTERNAL,UNALIGNED,STRING(11),CHARACTER,VARYING
                                               265,382,424

4                GRK                           (11)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               265

2                HPCD                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER

297    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               299,300,305,307,310,320,321,321,326,327,327,331,332,332,337,338,338
                                               342,343,343,348,349,349,353,354,354,363,364,364,368,369,369,376,377
                                               377,382,383,383,384,384,391,392,392,398,399,404,404,405,405,406

247    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               249,250,252

74     ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               78,81,84,122,123,124,125,133,134,135,136,137,138,146,147,151,153,155
                                               157,159,161,163,165,167,169,172,173,174,174

72     ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               217,218,218,218,218,218,218,218,218,218,219,219,220,220,220,220,220
                                               220,222,222,225,226,226,226,226,226,226,226,226,226,227,227,228,229
                                               231,234,240,240

17     ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               21,22,25,39,40,44,50,53,54,54,56,57,57,59,60

                 INDEX                         GENERIC,BUILT-IN FUNCTION
                                               39,171,219,227,382

2                INTEL                         IN RECORD,BASED(PTR),STRUCTURE

2                INTELB                        IN INTEL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER

188              ITEM                          PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               187,190,191,192,197,200,203,206,209,212

74               ITEM                          PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               73,86,88,90,92,94,96,98,100,102,104,106,108,110,112,114,116,118,129
                                               142,144,147,149,151,153,155,157,159,161,163,165,167,169,174,175,177
                                               179,181,183,185

72               ITEM                          AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               225,229

409    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               468,469,471

297    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               304,305,307,310,387,388,402,403,404,404,405,405,406

74     ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               83,84

72     ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               231,232,234
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

17     ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               34,35,43,49,60,61,61,62,62,63,63,64,64

409    ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               418,419,419,419,423,424,424,424,428,429,429,429,429,429,429,431,432
                                               432,432,436,437,437,437,442,443,443,443,443,443,443,443,443,443,443
                                               449,450,450,450,455,456,456,456,456,456,456,462,463,466,467,469,471
                                               475

297    ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               386,387

256    ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               259,260,260,264,265,265,269,270,270,270,270,275,276,276,280,281,281
                                               281,281,283,286,287,287,287,287,292,293,293

72     ********* K                             PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               70,217,222,231,240

17     ********* K                             PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               13,19,42,42,43,45,48,48,49,51,53,56,60,65,65

10     ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               589,591,592,654

74     ********* KC                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               84,120,126,126,131,139,139,185

10     ********* KT                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               249,411,412,414,414,415,419,424,429,429,432,437,439,443,443,443,443
                                               445,445,446,446,450,452,456,456,458,546,547,563,645,645,646,649,656
                                               656,657,660,689,695,696,715,715,716,719

666            L00                             STATEMENT LABEL CONSTANT
                                               534

710            L01                             STATEMENT LABEL CONSTANT
                                               671,737

691            L02                             STATEMENT LABEL CONSTANT
                                               721,734

704            L03                             STATEMENT LABEL CONSTANT
                                               697

185            L1                              STATEMENT LABEL CONSTANT
                                               85,128,141

535            L12                             STATEMENT LABEL CONSTANT

616            L21                             STATEMENT LABEL CONSTANT
                                               624,633,639,642,665

557            L29                             STATEMENT LABEL CONSTANT
```

Dataset Limited

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			548,552
386		L3	STATEMENT LABEL CONSTANT 381
483		L39	STATEMENT LABEL CONSTANT 560,595,708
739		L49	STATEMENT LABEL CONSTANT 481,488
55		L61	STATEMENT LABEL CONSTANT 26,32,38
39		L62	STATEMENT LABEL CONSTANT 36
243		L66	STATEMENT LABEL CONSTANT 223,235,241
17	*********	LB	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 25,44,44,50,50
10		LB	AUTOMATIC,UNALIGNED,STRING(1),BIT 509,510,626
2		LE	IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 78
2	*********	LEAP	IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
2		LEN	IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 144
		LENGTH	GENERIC,BUILT-IN FUNCTION 18,192,195
17	*********	LK	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 20,30,30,31
17	*********	LL	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 18,21
70		LOCTST	ENTRY,STRING(1),BIT 654
71		LOP	(20)PARAMETER,UNALIGNED,STRING(1),CHARACTER 70,222,232,240
14		LOP	(20)PARAMETER,UNALIGNED,STRING(1),CHARACTER 13,31,61,61
9		LOP	(20)AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 585,591,654

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

10               LPB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               512,513,628

74     ********* LR                            PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               73,77,77,79

17               LR                            AUTOMATIC,UNALIGNED,STRING(3),CHARACTER
                                               44,45,50,51

3      ********* MAF                           (3)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               321,321,419,419,572,604,678

2                MART                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               88,318,318,320

4                MK                            (3)STATIC,EXTERNAL,UNALIGNED,STRING(12),CHARACTER,VARYING
                                               260

                 MOD                           GENERIC,BUILT-IN FUNCTION
                                               250

2                NAME                          IN RECORD,BASED(PTR),UNALIGNED,STRING(40),CHARACTER

17     ********* NB                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               39,53,54

6      ********* OAF                           (4)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               343,343,429,429,578,610,684

5                OAK                           (4)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                               270

4                OK                            (10)STATIC,EXTERNAL,UNALIGNED,STRING(35),CHARACTER,VARYING
                                               274,276

297              OKB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               301,302,324,329,335,340,346,351,358,361,366,380

8      ********* OPOCF                         (6)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               354,354,443,443,445,445,580,612,686

7                OPSOK                         (6)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                               281,283

10     ********* OPT                           AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               484,485,485,517,528,528,528,540,540,542,544,551,565,583,643,652

2                OQ                            (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER

2                OQC                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER

6      ********* ORF                           (6)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               332,332,456,456,458,458,579,611,685
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

5                ORK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               287,289

8      ********* OSOCF                         (7)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               360,360,369,369,443,443,445,445,446,446,581,613,687

2                OV25                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               98,372,374,374,376

3      ********* OVF                           (0:10)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               373,373,377,377,437,437,439,439,576,608,682

16               PCOND                         (6)AUTOMATIC,UNALIGNED,INITIAL,STRING(2),CHARACTER,VARYING
                                               35

2      ********* PF16                          (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

2                PNAME                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(17),CHARACTER

2                POCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               94,346,346,348,351,351,353

3      ********* POF                           (5)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               349,349,443,443,569,601,675

4                PSOK                          (5)STATIC,EXTERNAL,UNALIGNED,STRING(65),CHARACTER,VARYING
                                               281

                 PTR                           AUTOMATIC,ALIGNED,POINTER
                                               78,80,82,86,88,90,92,94,96,98,100,102,104,106,108,110,112,114,116
                                               118,121,123,124,125,129,132,134,135,136,137,138,142,144,147,149,151
                                               153,155,157,159,161,163,165,167,169,173,174,174,174,177,179,181,183
                                               221,298,303,312,313,315,317,318,318,320,323,324,324,326,329,329,331
                                               334,335,335,337,340,340,342,345,346,346,348,351,351,353,356,357,361
                                               361,363,366,366,368,371,372,374,374,376,379,382,385,386,388,394,396
                                               400,406,616,617,621,621,621,622,626,630,630,630,631,635,638,641,649
                                               660,710,711,719,726,730

74               QC                            AUTOMATIC,DEFINED,UNALIGNED,STRING(3),CHARACTER
                                               136,136

74               QE                            AUTOMATIC,DEFINED,UNALIGNED,STRING(3),CHARACTER
                                               75,76,76,135,135

74               QG                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               137,137

74               QR                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               138,138

72     ********* QS                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               121,122,132,133
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

2                QUAL                          (30)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(10),CHARACTER
                                               134,135,136,137,138

2                QUALNO                        IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               129,132

74               QY                            AUTOMATIC,DEFINED,UNALIGNED,STRING(2),CHARACTER
                                               134,134

74               R                             (40)AUTOMATIC,INITIAL,LABEL
                                               80,86,88,90,92,94,96,98,100,102,104,106,108,110,112,114,116,118,120
                                               129,131,142,144,146,149,151,153,155,157,159,161,163,165,167,169,171
                                               177,179,181,183,79

10               RB                            AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               503,504,635

408              RECFPNT                       ENTRY,DECIMAL,FLOAT(SINGLE)
                                               556,703

296              RECFREQ                       ENTRY,DECIMAL,FLOAT(SINGLE)
                                               650,662,720

2                RECORD                        BASED(PTR),STRUCTURE

2                REG                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               92,382

2                REL                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               100,324,324,326,329,329,331

74               REQI                          PARAMETER,UNALIGNED,STRING(12),CHARACTER
                                               73,171,171

72               REQI                          AUTOMATIC,UNALIGNED,STRING(12),CHARACTER
                                               219,221,225

72               REQM                          AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               227,228,229

188              REQU                          PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               187,193,194,195,197,200,203,206,209,212

15               REQUEST                       PARAMETER,UNALIGNED,STRING(400),CHARACTER,VARYING
                                               13,18,22,39,40,44,50,53

10               REQUEST                       AUTOMATIC,UNALIGNED,STRING(400),CHARACTER,VARYING
                                               493,528,533,545,590,591

255              RFREQK                        ENTRY,DECIMAL,FLOAT(SINGLE)
                                               554,701

3      ********* RGF                           (11)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               384,384,424,424,573,605,679
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

71               RITEM                         (20)PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               70,219,219,227,227,228

14               RITEM                         (20)PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               13,53,64,64

9                RITEM                         (20)AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               588,591,654

4                RK                            (5)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                               287

71     ********* RLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               70,218,218,218,218,218,218,218,218,218,220,220,220,220,220,220,225
                                               226,226,226,226,226,226,226,226,226

14     ********* RLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               13,45,51,57,57,59,62,62

9      ********* RLAB                          (20)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               587,591,654

3      ********* RLF                           (5)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               327,327,456,456,570,602,676

10               RPB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               506,507,637

                 SCARDS                        FILE,EXTERNAL
                                               480,483,484,495,691

3      ********* SCF                           (0:9)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               392,392,450,450,452,452,577,609,683

2                SCH                           (10)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(5),CHARACTER
                                               123,124,125,388

2                SCHNO                         IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               118,121,386

4                SCK                           (9)STATIC,EXTERNAL,UNALIGNED,STRING(60),CHARACTER,VARYING
                                               293

2                SET                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               90,335,335,337,340,340,342

2                SEX                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               86,313,315

2                SOCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               96,357,361,361,363,366,366,368

3      ********* SOF                           (6)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
```

Dataset Limited

```
DCL NO.          IDENTIFIER                ATTRIBUTES AND REFERENCES

                                           359,359,364,364,443,443,446,446,571,603,677

246              SPNT                      ENTRY,DECIMAL,FLOAT(SINGLE)
                                           550,699

                 SPRINT                    FILE,EXTERNAL
                                           59,248,251,252,257,258,260,262,263,265,267,268,270,272,273,274,276
                                           278,279,281,283,284,285,287,289,290,291,293,312,317,323,334,345,356
                                           371,379,385,394,406,410,411,412,413,414,415,416,417,419,421,422,424
                                           426,427,429,432,434,435,437,439,440,441,443,445,446,447,448,450,452
                                           453,454,456,458,459,460,461,463,465,467,470,471,474,476,478,482,487
                                           517,518,520,523,526,527,543,545,546,553,555,558,622,631,638,694,695
                                           700,702,706,732,739

74               ST                        AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                           123,123,146

2      ********* STEN                      (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

3      ********* STF                       (6)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                           338,338,429,429,432,432,574,606,680

4                STK                       (6)STATIC,EXTERNAL,UNALIGNED,STRING(50),CHARACTER,VARYING
                                           270,272,272

12               STUDENT                   FILE,EXTERNAL,RECORD,ENVIRONMENT(F(2416))
                                           535,536,559,594,616,666,667,669,670,707,710

2                SUB                       (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(36),CHARACTER

2                SUBCAR                    IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                           183

                 SUBSTR                    GENERIC,BUILT-IN FUNCTION
                                           22,39,40,44,50,53,80,82,123,124,125,134,135,136,137,138,171,174,192
                                           195,219,221,221,227,265,298,303,388,396,400,424,496,499,502,505,508
                                           511,539,539,539,590,617,621,621,621,626,630,630,630,635,641,705,705
                                           705

10               SUPP                      AUTOMATIC,UNALIGNED,STRING(1),BIT
                                           549,564,647,648,658,659,690,698,717,718

3      ********* SXF                       (2)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                           314,314,316,316,411,411,412,412,568,600,674

74               SYF                       AUTOMATIC,DEFINED,UNALIGNED,STRING(2),CHARACTER
                                           124,124

74               SYT                       AUTOMATIC,DEFINED,UNALIGNED,STRING(2),CHARACTER
                                           125,125

2                TC                        IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                           177
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

187              TEST                          ENTRY,STRING(1),BIT
                                               229

2                TITLE                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER

13               TRANS                         ENTRY,DECIMAL,FLOAT(SINGLE)
                                               591

                 TRANSLATE                     GENERIC,BUILT-IN FUNCTION
                                               76,228

2                TTFT                          (2,2)IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               80,110,221,298,617,621,621,621,626,630,630,630,635,641

2                UNCOL                         (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER
                                               112

2                UNOTH                         (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER
                                               114

2      ********* VALUES                        (2,6)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

10               WB                            AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               497,498,617,641

10               WCR                           AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               491,522,523,524,641,641

10               WD                            AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               621,622,630,631

10               WPB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               500,501,619

10               WRL                           AUTOMATIC,UNALIGNED,STRING(3),CHARACTER
                                               490,496,499,502,505,508,511,519,520,521

2                WS                            IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               179

297              X1                            AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               388,389,389,391

17               X1                            AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               22,23,28,28,31,35

297              X2                            AUTOMATIC,UNALIGNED,STRING(2),CHARACTER
                                               298,299,303,304,396,397,399,400,401,403

74               X2                            AUTOMATIC,UNALIGNED,STRING(2),CHARACTER
                                               80,81,82,83

188              X20                           AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               190,192,192,193,195,195
```

Dataset Limited

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
74	XVAR	AUTOMATIC,UNALIGNED,STRING(3),CHARACTER,VARYING 171,172
2	YC	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
10	YE	AUTOMATIC,UNALIGNED,STRING(4),CHARACTER 483,517,543,694

Dataset Limited

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	AGF	6
189	C	48
9	CLAB	40
10	DESL	8000
3	FPF	200
4	GRK	330
9	LOP	20
3	MAF	6
4	MK	36
6	OAF	8
5	OAK	160
4	OK	350
8	OPOCF	12
7	OPSOK	240
6	ORF	12
5	ORK	180
8	OSOCF	14
3	OVF	22
16	PCOND	12
3	POF	10
4	PSOK	325
74	R	320
2	RECORD	2416
3	RGF	22
9	RITEM	400
4	RK	150

Dataset Limited

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
9	RLAB	40
3	RLF	10
3	SCF	20
4	SCK	540
3	SDF	12
3	STF	12
4	STK	300
3	SXF	4

Dataset Limited

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED F IS 11100 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED TRANS IS 1140 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED LOCTST IS 1120 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED FIND IS 1436 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED TEST IS 1076 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED SPNT IS 924 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED RFREQK IS 1056 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED RECFREQ IS 1072 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED RECFPNT IS 1180 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 480 IS 896 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 537 IS 1024 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 668 IS 928 BYTES LONG.

THE PROGRAM CSECT IS NAMED F AND IS 61448 BYTES LONG.

THE STATIC CSECT IS NAMED ******FA AND IS 9240 BYTES LONG.

STATISTICS SOURCE RECORDS = 637,PROG TEXT STMNTS = 740,OBJECT BYTES = 61448

Dataset Limited

COMPILER DIAGNOSTICS.

WARNINGS.

IEM0227I NO FILE/STRING OPTION SPECIFIED IN ONE OR MORE GET/PUT STATEMENTS. SYSIN/SYSPRINT HAS BEEN ASSUMED IN EACH CASE.

IEM0764I ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY ARE FLAGGED '*********' IN THE XREF/ATR LIST.

IEM1790I DATA CONVERSIONS WILL BE DONE BY SUBROUTINE CALL IN THE FOLLOWING STATEMENTS 45, 51, 78, 81, 83, 121, 132, 146, 172, 299, 304, 320, 326, 331, 337, 342, 348, 353, 363, 368, 376, 386, 391, 399, 403.

END OF DIAGNOSTICS.

COMPILE TIME 68.86 SECS

ELAPSED TIME 219.04 SECS

Dataset Limited

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRO
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST

*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
*OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
*OPTIONS IN EFFECT*     NEST,OPLIST
```

Dataset Limited

```
STMT LEVEL NEST
                                                                        F0006390
 1                CON:PROC(FILN2);                                      F0006400
 2      1              DCL FILN2 CHAR(8);                              F0006410
 3      1              DCL (I,J,K) FIXED BINARY, AK(3,26) FIXED BINARY EXTERNAL,  F0006420
                       (AUTR(*) CHAR(19), AUTR1(*) CHAR(3), ATAB(*) CHAR(3), ATAB1(*)  F0006430
                       FIXED BINARY) CONTROLLED EXTERNAL, (ALPHA CHAR(26), NALC FIXED  F0006440
                       BINARY) EXTERNAL;                                F0006450
 4      1              DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),  F0006460
                       MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),  F0006470
                       GRK(11) CHAR(30),OAK(4) CHAR(40), ORK(6) CHAR(30), OPSOK(6)  F0006480
                       CHAR(40)) VARYING EXTERNAL;                      F0006490
 5      1              DCL NORM(2,16,10) FIXED BINARY EXTERNAL;         F0006500
 6      1              DCL NORM2(2,16,2) FLOAT DECIMAL EXTERNAL;        F0006510
                  /*-----------------------------------------------------------*/F0006520
 7      1              DCL CONST FILE;                                  F0006530
                  /*-----------------------------------------------------------*/F0006540
 8      1              OPEN FILE(CONST) INPUT TITLE(FILN2);             F0006550
 9      1              DO I=1 TO 26;                                    F0006560
10      1     1           GET FILE(CONST) LIST(AK(1,I));                F0006570
11      1     1        END;                                             F0006580
12      1              AK(2,1)=1;     AK(3,1)=AK(2,1)+AK(1,1)-1;        F0006590
14      1              DO I=2 TO 26;                                    F0006600
15      1     1           AK(2,I)=AK(3,I-1)+1;                          F0006610
16      1     1           AK(3,I)=AK(1,I)+AK(2,I)-1;                    F0006620
17      1     1        END;                                             F0006630
18      1              N=SUM(AK(1,*));                                  F0006640
19      1              ALLOCATE AUTR(N),AUTR1(N);                       F0006650
20      1              DO K=1 TO 26;                                    F0006660
21      1     1          DO I=AK(2,K) TO AK(3,K);                       F0006670
22      1     2           GET FILE(CONST) EDIT(AUTR(I),AUTR1(I))        F0006680
                          (COL(1),A(19),COL(40),A(3));                  F0006690
23      1     2           END;                                          F0006700
24      1     1        END;                                             F0006710
25      1              GET FILE(CONST) EDIT((OAK(K) DO K=1 TO 4))(4(COL(1),A(40)));  F0006720
26      1              GET FILE(CONST) EDIT((ORK(K) DO K=1 TO 6))(6(COL(1),A(30)));  F0006730
27      1              GET FILE(CONST) EDIT((OPSOK(K) DO K=1 TO 6))(6(COL(1),A(40)));  F0006740
28      1              GET FILE(CONST) EDIT((SCK(I) DO I=1 TO 9))(9(COL(1),A(60)));  F0006750
29      1              GET FILE(CONST) EDIT((PSOK(I) DO I=1 TO 5))(5(COL(1),A(65)));  F0006760
30      1              GET FILE(CONST) EDIT((STK(I) DO I=1 TO 6))(6(COL(1),A(50)));  F0006770
31      1              GET FILE(CONST) EDIT((MK(I) DO I=1 TO 3))(3(COL(1),A(12)));  F0006780
32      1              GET FILE(CONST) EDIT((OK(I) DO I=1 TO 10))(10(COL(1),A(35)));  F0006790
33      1              GET FILE(CONST) EDIT((RK(I) DO I=1 TO 5))(5(COL(1),A(30)));  F0006800
34      1              GET FILE(CONST) EDIT((GRK(I) DO I=1 TO 11))(11(COL(1),A(30)));  F0006810
35      1              GET FILE(CONST) EDIT(ALPHA)(COL(1),A(26));       F0006820
36      1              GRC=SUBSTR(ALPHA,1,11);                          F0006830
37      1              GET FILE(CONST) EDIT(NALC)(COL(1),F(3));         F0006840
38      1              ALLOCATE ATAB(NALC),ATAB1(NALC);                 F0006850
39      1              GET FILE(CONST) SKIP;                            F0006860
40      1              GET FILE(CONST) EDIT((ATAB(I),ATAB1(I) DO I=1 TO NALC))  F0006870
                       (X(2),A(3),X(1),F(2));                           F0006880
41      1              DO I=1 TO 2;                                     F0006890
42      1     1           DO J=1 TO 16;                                 F0006900
43      1     2              GET FILE(CONST) LIST((NORM(I,J,K) DO K=1 TO 10));  F0006910
44      1     2              GET FILE(CONST) LIST((NORM2(I,J,K) DO K=1 TO 2));  F0006920
45      1     2           END;                                          F0006930
46      1     1        END;                                             F0006940
```

Dataset Limited

```
STMT LEVEL NEST

 47     1              CLOSE FILE(CONST);                                  F0006950
 48     1         END;                                                     F0006960
```

Dataset Limited

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
3	*********	AK	(3,26)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 10,12,13,13,13,15,15,16,16,16,18,21,21
3		ALPHA	STATIC,EXTERNAL,UNALIGNED,STRING(26),CHARACTER 35,36
3		ATAB	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 38,40
3	*********	ATAB1	(*)EXTERNAL,CONTROLLED,ALIGNED,BINARY,FIXED(15,0) 38,40
3		AUTR	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 19,22
3		AUTR1	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 19,22
1		CON	ENTRY,DECIMAL,FLOAT(SINGLE)
7		CONST	FILE,EXTERNAL 8,10,22,25,26,27,28,29,30,31,32,33,34,35,37,39,40,43,44,47
2		FILN2	PARAMETER,UNALIGNED,STRING(8),CHARACTER 1,8
4		GRC	STATIC,EXTERNAL,UNALIGNED,STRING(11),CHARACTER,VARYING 36
4		GRK	(11)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING 34
3	*********	I	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 9,10,14,15,15,16,16,16,21,22,22,28,28,29,29,30,30,31,31,32,32,33,33 34,34,40,40,40,41,43,44
3	*********	J	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 42,43,44
3	*********	K	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 20,21,21,25,25,26,26,27,27,43,43,44,44
4		MK	(3)STATIC,EXTERNAL,UNALIGNED,STRING(12),CHARACTER,VARYING 31
	*********	N	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 18,19,19
3	*********	NALC	STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 37,38,38,40

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

5     ********   NORM                          (2,16,10)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0)
                                                43

6                NOPM2                         (2,16,2)STATIC,EXTERNAL,ALIGNED,DECIMAL,FLOAT(SINGLE)
                                                44

4                OAK                           (4)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                                25

4                OK                            (10)STATIC,EXTERNAL,UNALIGNED,STRING(35),CHARACTER,VARYING
                                                32

4                OPSOK                         (6)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                                27

4                ORK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                                26

4                PSOK                          (5)STATIC,EXTERNAL,UNALIGNED,STRING(65),CHARACTER,VARYING
                                                29

4                RK                            (5)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                                33

4                SCK                           (9)STATIC,EXTERNAL,UNALIGNED,STRING(60),CHARACTER,VARYING
                                                28

4                STK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(50),CHARACTER,VARYING
                                                30

                 SUBSTR                        GENERIC,BUILT-IN FUNCTION
                                                36

                 SUM                           GENERIC,BUILT-IN FUNCTION
                                                18
```

Dataset Limited

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	AK	156
38	ATAB	ADJUSTABLE
38	ATAB1	ADJUSTABLE
19	AUTR	ADJUSTABLE
19	AUTR1	ADJUSTABLE
4	GRK	330
4	MK	36
5	NORM	640
6	NORM2	256
4	OAK	160
4	OK	350
4	OPSOK	240
4	ORK	180
4	PSOK	325
4	RK	150
4	SCK	540
4	STK	300

Dasset Limited

STORAGE REQUIREMENTS.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED CON IS 388 BYTES LONG.

THE PROGRAM CSECT IS NAMED CON AND IS 4294 BYTES LONG.

THE STATIC CSECT IS NAMED ****CONA AND IS 1044 BYTES LONG.

STATISTICS SOURCE RECORDS = 58,PROG TEXT STMNTS = 48,OBJECT BYTES = 4294

Dataset Limited

```
COMPILER DIAGNOSTICS.


WARNINGS.

   IEM0526I      1   OPTION MAIN HAS NOT BEEN SPECIFIED FOR THE EXTERNAL PROCEDURE, STATEMENT NUMBER 1

   IEM0764I          ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY
                     ARE FLAGGED '*********' IN THE XREF/ATR LIST.


END OF DIAGNOSTICS.

COMPILE TIME       11.00 SECS

ELAPSED TIME       38.15 SECS
EXECUTION TERMINATED
```

Dataset Limited

5. PROGRAMME I.

5.2 PROGRAMME LISTING

THIS SECTION CONTAINS A LISTING OF THE SOURCE CODE OF PROGRAMME I.

```
 1                                                                  /*****************************************/
 2    (SUBRG):(STRG):                                               /*                                       */
 3    I:PROCEDURE OPTIONS(MAIN);                                    /*          BCPS....PROGRAMME I          */
 4         DCL 1 RECORD BASED(PTR),                                 /*                                       */
 5               2 (DESNO CHAR(8), NAME CHAR(40)),                  /*****************************************/
 6               2 GENERAL,                                         /*                                       */
 7                 3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),   /* DECLARATION OF STRUCTURE 'RECORD'  */
 8                 3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),         /* -> MINOR STRUCTURES OF 'RECORD'    */
 9                 3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)),    /*    -> GENERAL (GENERAL INFORMATION)*/
10                 3 FPOS CHAR(4),                                  /*    -> EDUCATION (PREVIOUS EDUCATION)*/
11               2 EDUCATION,                                       /*    -> COURSE (COURSE DETAILS)       */
12                 3 (SCHNO CHAR(2), SCH(10) CHAR(5)),              /*    -> FINALS (FINAL RESULTS)        */
13                 3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),           /*    -> ANNPROG (ANNUAL PROGRESS)     */
14                 3 (UNCOL(2) CHAR(21), FPT(1) CHAR(2), FT(2,2) CHAR(6)),      /*    -> INTEL (INTELLIGENCE TEST SCORES)*/
15                 3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),      /*                                       */
16                 3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),            /*                                       */
17                 3 (ADIP CHAR(30), ADIPC CHAR(1)),                /*                                       */
18                 3 DDIPC CHAR(1),                                 /*                                       */
19                 3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),              /*                                       */
20                 3 ATO CHAR(1),                                   /*                                       */
21               2 COURSE,                                          /*                                       */
22                 3 (LEN CHAR(1), CODES(4) CHAR(3)),               /*                                       */
23                 3 ((LE,DC(4,24:64)) CHAR(1)),                    /*                                       */
24                 3 (DC1(4,26:30) CHAR(2)),                        /*                                       */
25                 3 ((DC46(4),DC49(4)) CHAR(3)),                   /*                                       */
26               2 FINALS,                                          /*                                       */
27                 3 ((TC,WS,BE,SUBCAR) CHAR(1)),                   /*                                       */
28               2 ANNPROG,                                         /*                                       */
29                 3 AP CHAR(802),                                  /*                                       */
30                 3 ((LEAP,FK(100,2)) FIXED BINARY),               /*                                       */
31               2 INTEL,                                           /*                                       */
32                 3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY),    /*                                       */
33                 3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),          /*                                       */
34                 3 STEN(2,16) FIXED BINARY;                       /*                                       */
35    /*-----------------------------------------------------------*/ /*-----------------------------------------*/
36         DCL (VF(0:60,6),DSF(0:300),AHF(0:130,3),PFF(0:26,16),SF(0:10,16))   /* DECLARATION OF PROGRAMME VARIABLES*/
37         FIXED BINARY;                                            /*                                       */
38         DCL ((VT,VST)(6),(DST,DSST),(AHT,AHST)(2),(PFT,PFST)(16),/*                                       */
39         (SST,SSST)(16)) FLOAT DECIMAL(15);                       /*                                       */
40         DCL LOP(20) CHAR(1), (RLAB,CLAB)(20) FIXED BINARY,       /*                                       */
41         RITEM(20) CHAR(20) VARYING;                              /*                                       */
42         DCL MNT(16) FIXED BINARY INITIAL(20,13,26,26,26,20,26,   /*                                       */
43         20,20,26,20,26,20,20,20,26);                             /*                                       */
44         DCL (FILN1,FILN2,WD,DT,DESL(1000)) CHAR(8), YE CHAR(4), WRL  /*                                       */
45         CHAR(3), (YT,WCR,PI) CHAR(1), (KT,OPT,YR,K) FIXED BINARY, (WB,LB,   /*                                       */
46         RB,SUPP,WPB,LPB,RPB,B) BIT(1), REQUEST CHAR(400) VARYING,   /*                                       */
47         (DESNO3,DESNO2) CHAR(8);                                 /*                                       */
48    /*-----------------------------------------------------------*/ /*-----------------------------------------*/
49         DCL STUDENT FILE RECORD ENV(F(2416));                    /* DECLARATION OF FILE (STUDENT)         */
50    /*************************************************************/ /*****************************************/
51    ITTMS:PROC;                                                   /* ACCUMULATE TOTALS ROUTINE             */
52         DCL K FIXED BINARY;                                      /*****************************************/
53         DO K=1 TO 6;                                             /* TEST VALUES SCORES                    */
54            IF VALUES(YR,K)>=0 & VALUES(YR,K)<=60 THEN            /* -> ACCUMULATE TOTALS, AND             */
55            DO; VT(K)=VT(K)+VALUES(YR,K);                         /* -> ACCUMULATE SQUARES TOTALS, OR      */
```

Dataset Limited

```
56                 VST(K)=VST(K)+VALUES(YR,K)**2;                    /*                                          */
57              END; ELSE                                            /*                                          */
58              PUT EDIT(DESNO,'ERROR RETURNED FOR VALUES SCORE',K)  /* -> CLASS F ERROR/WARNING                 */
59              (SKIP,COL(1),A,X(1),A,X(1),F(1));                    /*                                          */
60           END;                                                    /*                                          */
61           IF DSCALE(YR)>=0 & DSCALE(YR)<=300 THEN                 /* TEST D.SCALE SCORES                      */
62           DO; DST=DST+DSCALE(YR);                                 /* -> ACCUMULATE TOTALS, AND                */
63               DSST=DSST+DSCALE(YR)**2;                            /* -> ACCUMULATE SQUARES TOTALS, OR         */
64           END; ELSE                                               /*                                          */
65           PUT EDIT(DESNO,'ERROR RETURNED FOR D.SCALE SCORE')      /* -> CLASS F ERROR/WARNING                 */
66           (SKIP,COL(1),A,X(1),A);                                 /*                                          */
67           DO K=1,2;                                               /*                                          */
68              IF AH4(YR,K)>=0 & AH4(YR,K)<=130 THEN                /* TEST A.H.4 SCORES                        */
69              DO; AHT(K)=AHT(K)+AH4(YR,K);                         /* -> ACCUMULATE TOTALS, AND                */
70                  AHST(K)=AHST(K)+AH4(YR,K)**2;                    /* -> ACCUMULATE SQUARES TOTALS, OR         */
71              END; ELSE                                            /*                                          */
72              PUT EDIT(DESNO,'ERROR RETURNED FOR A.H.4 SCORE',K)   /* -> CLASS F ERROR WARNING                 */
73              (SKIP,COL(1),A,X(1),A,X(1),F(2));                    /*                                          */
74           END;                                                    /*                                          */
75           DO K=1 TO 16;                                           /*                                          */
76              IF PF16(YR,K)>=0 & PF16(YR,K)<=MNT(K) THEN           /* TEST P.F.16 SCORES                       */
77              DO; PFT(K)=PFT(K)+PF16(YR,K);                        /* -> ACCUMULATE TOTALS,                    */
78                  PFST(K)=PFST(K)+PF16(YR,K)**2;                   /* -> ACCUMULATE SQUARES TOTALS,            */
79                  SST(K)=SST(K)+STEN(YR,K);                        /* -> ACCUMULATE STEN TOTALS, AND           */
80                  SSST(K)=SSST(K)+STEN(YR,K)**2;                   /* -> ACCUMULATE STEN SQUARES TOTALS, OR    */
81              END; ELSE                                            /*                                          */
82              PUT EDIT(DESNO,'ERROR RETURNED FOR P.F.16 SCORE',K)  /* -> CLASS F ERROR/WARNING                 */
83              (SKIP,COL(1),A,X(1),A,X(1),F(2));                    /*                                          */
84           END;                                                    /*                                          */
85     END;                                                          /*                                          */
86     /*****************************************************************/  /*****************************************/
87     ITFREQ:PROC;                                                  /* ACCUMULATE FREQUENCIES ROUTINE           */
88           DCL K FIXED BINARY;                                     /*****************************************/
89           DO K=1 TO 6;                                            /*                                          */
90              IF VALUES(YR,K)>=0 & VALUES(YR,K)<=60 THEN           /* TEST VALUES SCORES                       */
91              VF(VALUES(YR,K),K)=VF(VALUES(YR,K),K)+1; ELSE        /* -> ACCUMULATE FREQUENCY, OR              */
92              PUT EDIT(DESNO,'ERROR RETURNED FOR VALUES SCORE',K)  /* -> CLASS F ERROR/WARNING                 */
93              (SKIP,COL(1),A,X(1),A,X(1),F(1));                    /*                                          */
94           END;                                                    /*                                          */
95           IF DSCALE(YR)>=0 & DSCALE(YR)<=300 THEN                 /* TEST D.SCALE SCORES                      */
96           DSF(DSCALE(YR))=DSF(DSCALE(YR))+1; ELSE                 /* -> ACCUMULATE FREQUENCY, OR              */
97           PUT EDIT(DESNO,'ERROR RETURNED FOR D.SCALE SCORE')      /* -> CLASS F ERROR/WARNING                 */
98           (SKIP,COL(1),A,X(1),A);                                 /*                                          */
99           DO K=1,2;                                               /*                                          */
100             IF AH4(YR,K)>=0 & AH4(YR,K)<=130 THEN                /* TEST A.H.4 SCORES                        */
101             AHF(AH4(YR,K),K)=AHF(AH4(YR,K),K)+1; ELSE            /* -> ACCUMULATE FREQUENCY, OR              */
102             PUT EDIT(DESNO,'ERROR RETURNED FOR A.H.4 SCORE',K)   /* -> CLASS F ERROR/WARNING                 */
103             (SKIP,COL(1),A,X(1),A,X(1),F(2));                    /*                                          */
104          END;                                                    /*                                          */
105          DO K=1 TO 16;                                           /*                                          */
106             IF PF16(YR,K)>=0 & PF16(YR,K)<=MNT(K) THEN           /* TEST P.F.16 SCORES                       */
107             PFF(PF16(YR,K),K)=PFF(PF16(YR,K),K)+1; ELSE          /* -> ACCUMULATE FREQUENCY, OR              */
108             DO; PUT EDIT(DESNO,'ERROR RETURNED FOR P.F.16 SCORE',K)  /* -> CLASS F ERROR/WARNING             */
109                 (SKIP,COL(1),A,X(1),A,X(1),F(2));                /*                                          */
110                 RETURN;                                          /*                                          */
111             END;                                                 /*                                          */
112             SF(STEN(YR,K),K)=SF(STEN(YR,K),K)+1;                 /* -> ACCUMULATE STEN FREQUENCY             */
113          END;                                                    /*                                          */
114    END;                                                          /*                                          */
115    /*****************************************************************/  /*****************************************/
```

Dataset Limited

```
SPNT:PROC;                                                          /* D.E.S.NO. OUTPUT ROUTINE                    */
     DCL I FIXED BINARY;                                            /*******************************************/
     PUT SKIP(3);                                                   /*                                           */
     DO I=1 TO KT;                                                  /*                                           */
        IF MOD(I-1,10)=0 THEN PUT EDIT(DESL(I))(SKIP,X(3),A(8)); ELSE /* OUTPUT DES.NOS, TEN PER LINE              */
        PUT EDIT(DESL(I))(X(3),A(8));                               /*                                           */
     END;                                                           /*                                           */
END;                                                                /*                                           */
/*******************************************************************/ /*******************************************/
PRINT:PROC;                                                         /* RECORD OUTPUT ROUTINE                     */
     IF MOD(KT-1,25)=0 THEN                                         /*******************************************/
     DO; PUT EDIT('D.E.S.NO','(-----VALUES----)','(D.SCALE)',       /* OUTPUT TITLES                             */
         '(--AH.4--)','(--------------------PF.16--------------------)' /*                                       */
         )(PAGE,COL(1),A(8),4(X(4),A));                             /*                                           */
         PUT SKIP;                                                  /*                                           */
     END;                                                           /*                                           */
     PUT EDIT(DESNO,VALUES(YR,*),DSCALE(YR),AH4(YR,*),PF16(YR,*))   /* OUTPUT RECORDS 25 PER PAGE                */
     (SKIP,COL(1),A,X(3),6(X(1),F(2)),X(7),F(3),X(6),2(X(1),F(2)),X(2), /*                                      */
     F(3),X(3),16(X(1),F(2)));                                      /*                                           */
     PUT EDIT(STEN(YR,*))(SKIP,COL(60),16(X(1),F(2)));              /*                                           */
END;                                                                /*                                           */
/*******************************************************************/ /*******************************************/
ITPTMS:PROC;                                                        /* TOTALS PRINT ROUTINES                     */
     DCL K FIXED BINARY;                                            /*******************************************/
     DCL PFK(16) CHAR(2) INITIAL(' A',' B',' C',' E',' F',' G',' H', /*                                           */
     ' I',' L',' M',' N',' O','Q1','Q2','Q3','Q4');                 /*                                           */
     VST=SQRT((VST-(VT**2/KT))/KT);                                 /* CALCULATE VALUES ST.DEVIATIONS            */
     DSST=SQRT((DSST-(DST**2/KT))/KT);                              /* CALCULATE D.SCALE ST.DEVIATIONS           */
     AHST=SQRT((AHST-(AHT**2/KT))/KT);                              /* CALCULATE A.H.4 ST.DEVIATIONS             */
     PFST=SQRT((PFST-(PFT**2/KT))/KT);                              /* CALCULATE P.F.16 ST.DEVIATIONS            */
     SSST=SQRT((SSST-(SST**2/KT))/KT);                              /* CALCULATE STEN ST.DEVIATIONS              */
     PUT EDIT('TEST','TOTAL( MEAN )ST.DEV','TOTAL( MEAN )ST.DEV')   /* OUTPUT TITLES                             */
     (SKIP(2),COL(7),A,COL(20),A,X(5),A);                           /*                                           */
     PUT EDIT('VALUES')(SKIP(2),COL(5),A);                          /* OUTPUT VALUES                             */
     DO K=1 TO 6;                                                   /*                                           */
        PUT EDIT('(',K,')')(COL(11),A,F(2),A);                      /*                                           */
        PUT EDIT(VT(K),'(',(VT(K)/KT),')',VST(K))                   /*                                           */
        (X(2),F(8),A,F(6,2),A,F(6,2));                              /*                                           */
     END;                                                           /*                                           */
     PUT EDIT('D.SCALE     ')(SKIP(2),COL(4),A);                    /* OUTPUT D.SCALE                            */
     PUT EDIT(DST,'(',(DST/KT),')',DSST)                            /*                                           */
     (X(2),F(8),A,F(6,2),A,F(6,2));                                 /*                                           */
     PUT EDIT('A.H.4')(SKIP(2),COL(6),A);                           /* OUTPUT A.H.4                              */
     DO K=1,2;                                                      /*                                           */
        PUT EDIT('(',K,')')(COL(11),A,F(2),A);                      /*                                           */
        PUT EDIT(AHT(K),'(',(AHT(K)/KT),')',AHST(K))                /*                                           */
        (X(2),F(8),A,F(6,2),A,F(6,2));                              /*                                           */
     END;                                                           /*                                           */
     PUT EDIT('STEN SCORES')(X(13),A);                              /* OUTPUT P.F.16 AND STENS                   */
     PUT EDIT('P.F.16')(SKIP(2),COL(5),A);                          /*                                           */
     DO K=1 TO 16;                                                  /*                                           */
        PUT EDIT('(',PFK(K),')')(COL(11),A,A(2),A);                 /*                                           */
        PUT EDIT(PFT(K),'(',(PFT(K)/KT),')',PFST(K))                /*                                           */
        (X(2),F(8),A,F(6,2),A,F(6,2));                              /*                                           */
        PUT EDIT(SST(K),'(',(SST(K)/KT),')',SSST(K))                /*                                           */
        (X(2),F(8),A,F(6,2),A,F(6,2));                              /*                                           */
     END;                                                           /*                                           */
END;                                                                /*                                           */
/*******************************************************************/ /*******************************************/
ITPFREQ:PROC;                                                       /* FREQUENCY TABLE PRINT ROUTINES            */
```

Dataset Limited

```
176          DCL (I,J,K) FIXED BINARY;                                  /*****************************************/
177          PUT EDIT('VALUES','SCORE','FREQUENCY')(SKIP(2),COL(1),A,SKIP,  /* OUTPUT VALUES                          */
178          COL(2),A,COL(25),A);                                       /*                                        */
179          PUT SKIP;                                                  /*                                        */
180          DO I=0 TO 60;                                              /*                                        */
181             DO J=1 TO 6;                                            /*                                        */
182                IF VF(I,J)>0 THEN                                    /*                                        */
183                DO; PUT EDIT(I,(VF(I,K) DO K=1 TO 6))(SKIP,COL(5),F(2),  /*                                    */
184                    6(X(4),F(4)));                                   /*                                        */
185                    GOTO L1;                                         /*                                        */
186                END;                                                 /*                                        */
187             END;                                                    /*                                        */
188    L1:   END;                                                       /*                                        */
189          PUT EDIT('D.SCALE','SCORE','FREQUENCY')(PAGE,COL(1),A,SKIP,COL(3),  /* OUTPUT D.SCALE                 */
190          A,COL(20),A);                                              /*                                        */
191          PUT SKIP;                                                  /*                                        */
192          DO I=0 TO 300;                                             /*                                        */
193             IF DSF(I)>0 THEN PUT EDIT(I,DSF(I))(SKIP,COL(5),F(3),COL(21),  /*                                 */
194             F(4));                                                  /*                                        */
195          END;                                                       /*                                        */
196          PUT EDIT('A.H.4','SCORE','FREQUENCY')(PAGE,COL(1),A,SKIP,COL(1),A,  /* OUTPUT A.H.4                   */
197          COL(20),A);                                                /*                                        */
198          PUT SKIP;                                                  /*                                        */
199          DO I=1 TO 130;                                             /*                                        */
200             IF AHF(I,1)>0 | AHF(I,2)>0 THEN                         /*                                        */
201             PUT EDIT(I,AHF(I,1),AHF(I,2))(SKIP,COL(3),F(3),         /*                                        */
202             X(7),2(X(4),F(4)));                                     /*                                        */
203          END;                                                       /*                                        */
204          PUT EDIT('P.F.16','SCORE','FREQUENCY')(PAGE,COL(1),A,SKIP,COL(2),  /* OUTPUT P.F.16                  */
205          A,COL(50),A);                                              /*                                        */
206          PUT SKIP;                                                  /*                                        */
207          DO I=0 TO 26;                                              /*                                        */
208             PUT EDIT(I)(SKIP,COL(5),F(2));                          /*                                        */
209             DO J=1 TO 16;                                           /*                                        */
210                IF I<=MNT(J) THEN                                    /*                                        */
211                PUT EDIT(PFF(I,J))(X(3),F(3)); ELSE                  /*                                        */
212                PUT EDIT('      ')(A);                               /*                                        */
213             END;                                                    /*                                        */
214          END;                                                       /*                                        */
215          PUT EDIT('STEN','SCORE','FREQUENCY')(SKIP(4),COL(3),A,COL(2),A,  /* OUTPUT STENS                      */
216          COL(50),A);                                                /*                                        */
217          PUT SKIP;                                                  /*                                        */
218          DO I=0 TO 10;                                              /*                                        */
219             PUT EDIT(I)(SKIP,COL(5),F(2));                          /*                                        */
220              DO J=1 TO 16;                                          /*                                        */
221                PUT EDIT(SF(I,J))(X(3),F(3));                        /*                                        */
222             END;                                                    /*                                        */
223          END;                                                       /*                                        */
224    END;                                                             /*                                        */
225    /*****************************************************************/  /*****************************************/
226    TRANS:PROC(REQUEST,LOP,CLAB,RLAB,RITEM,K);                       /* REQUEST TRANSLATION                    */
227          DCL (CLAB,RLAB)(20) FIXED BINARY, LOP(20) CHAR(1),         /*****************************************/
228          RITEM(20) CHAR(20) VARYING;                                /* VARIABLE DECLARATIONS                  */
229          DCL REQUEST CHAR(400) VARYING;                             /*                                        */
230          DCL PCOND(6) CHAR(2) VARYING INITIAL('=','¬=','>','¬>','<','¬<');  /* RELATIONAL OPERATOR LIST          */
231          DCL X1 CHAR(1),LR CHAR(3),(I,J,K,LK,LL,LB,NB) FIXED BINARY;  /*                                    */
232          LL=LENGTH(REQUEST);                                        /* CALCULATE LENGTH OF REQUEST            */
233          K=0;  LK=1;                                                /*                                        */
234          DO I=1 TO LL;                                              /*                                        */
235             X1=SUBSTR(REQUEST,I,1);                                 /* REMOVE FIRST/NEXT CHARACTER            */
```

Dataset Limited

```
236            IF X1=' ' THEN                                          /* TEST FOR BLANK                                  */
237            DO; LB=I;                                               /* -> RESET LOWER BOUND                            */
238                GOTO L61;                                           /* -> BRANCH FOR NEXT CHARACTER                    */
239            END; ELSE                                               /*                                                 */
240            IF X1='&' | X1='|' THEN                                 /* TEST FOR LOGICAL OPERATOR                       */
241            DO; LK=LK+1;                                            /* -> UPDATE LOGICAL OPERATOR COUNTER              */
242                LOP(LK)=X1;                                         /* -> ADD LOGICAL OPERATOR TO LOP VECTOR           */
243                GOTO L61;                                           /* -> BRANCH FOR NEXT CHARACTER                    */
244            END; ELSE                                               /*                                                 */
245            DO J=1,3,5;                                             /* TEST FOR RELATIONAL OPERATOR                    */
246               IF X1=PCOND(J) THEN GOTO L62;                        /* -> BRANCH IF FOUND                              */
247            END;                                                    /*                                                 */
248            GOTO L61;                                               /* BRANCH FOR NEXT CHARACTER                       */
249    L62:    NB=INDEX(SUBSTR(REQUEST,I),' ');                        /* LOCATE NEXT BLANK                               */
250            IF SUBSTR(REQUEST,I-1,1)='¬' THEN                       /* TEST RELATIONAL OP. FOR 'NOT' CASE              */
251            DO; K=K+1;                                              /* -> UPDATE SIMPLE REQUEST COUNTER                */
252                CLAB(K)=J+1;                                        /* -> RECORD RELATIONAL OP. NUMBER                 */
253                LR=SUBSTR(REQUEST,LB+1,I-LB-2);                     /* -> REMOVE ITEM NUMBER                           */
254                RLAB(K)=LR;                          /* CONVERSION */ /* -> RECORD ITEM NUMBER                           */
255            END; ELSE                                               /* RELATIONAL OP. NOT 'NOT' CASE                   */
256            DO; K=K+1;                                              /* -> UPDATE SIMPLE REQUEST COUNTER                */
257                CLAB(K)=J;                                          /* -> RECORD RELATIONAL OP. NUMBER                 */
258                LR=SUBSTR(REQUEST,LB+1,I-LB-1);                     /* -> REMOVE ITEM NUMBER                           */
259                RLAB(K)=LR;                          /* CONVERSION */ /* -> RECORD ITEM NUMBER                           */
260            END;                                                    /*                                                 */
261            RITEM(K)=SUBSTR(REQUEST,I+1,NB-2);                      /* RECORD ITEM VALUE                               */
262            I=I+NB-1;                                               /* RESET CONTROL VARIABLE TO NEXT BLANK            */
263    L61: END;                                                       /* BRANCH FOR NEXT CHARACTER                       */
264         DO I=1 TO K;                                               /*                                                 */
265            IF RLAB(I)<1 | RLAB(I)>66 THEN                          /* TEST ITEM NUMBERS OF ALL SIMPLE REQUESTS        */
266            DO; PUT EDIT('ERROR.....ITEM',RLAB(I),'NOT INCLUDED')   /* -> CLASS G ERROR/WARNING                        */
267                (SKIP,COL(1),A,X(1),F(2),X(1),A);                   /*                                                 */
268                DO J=I TO K-1;                                      /* OVERWRITE SIMPLE REQUEST                        */
269                   LOP(J)=LOP(J+1);                                 /* -> LOGICAL OPERATOR                             */
270                   RLAB(J)=RLAB(J+1);                               /* -> ITEM NUMBER                                  */
271                   CLAB(J)=CLAB(J+1);                               /* -> RELATIONAL OPERATOR                          */
272                   RITEM(J)=RITEM(J+1);                             /* -> ITEM VALUE                                   */
273                   K=K-1;                                           /* -> RESET SIMPLE REQUEST COUNTER                 */
274                END;                                                /*                                                 */
275            END;                                                    /*                                                 */
276         END;                                                       /*                                                 */
277    END;                                                            /*                                                 */
278    /*****************************************************************/ /*****************************************************/
279    LOCTST:PROC(YR,LOP,CLAB,RLAB,RITEM,K) RETURNS(BIT(1));          /* LOCATE AND TEST REQUEST ROUTINE                 */
280         DCL LOP(20) CHAR(1),(CLAB,RLAB)(20) FIXED BINARY, RITEM(20) /*****************************************************/
281         CHAR(20) VARYING;                                          /* VARIABLE DECLARATIONS                           */
282         DCL REQI CHAR(12),REQM CHAR(20) VARYING;                   /*                                                 */
283         DCL ITEM CHAR(20) VARYING, YR FIXED BINARY;                /*                                                 */
284         DCL (K,I,J,QS) FIXED BINARY;                               /*                                                 */
285         /*****************************************************/    /*****************************************************/
286    FIND:PROC(YR,ITEM,LR,REQI);                                     /* LOCATE ITEM VALUE                               */
287         DCL (ITEM CHAR(20), XVAR CHAR(3)) VARYING, (KC,LR,I,J,YR) FIXED /*****************************************************/
288         BINARY, X2 CHAR(2), R(66) LABEL, REQI CHAR(12), ST CHAR(1) DEFINED /* VARIABLE DECLARATIONS                    */
289         REQI POS(2), SYF CHAR(2) DEFINED REQI POS(3), SYT CHAR(2) DEFINED /*                                          */
290         REQI POS(5), QY CHAR(2) DEFINED REQI POS(2), QE CHAR(3) DEFINED /*                                            */
291         REQI POS(4), QC CHAR(3) DEFINED REQI POS(7), QG CHAR(1) DEFINED /*                                            */
292         REQI POS(10), QR CHAR(1) DEFINED REQI POS(11);             /*                                                 */
293         IF QE¬='***' THEN QE=TRANSLATE(QE,' ','*');                /* REMOVE '*' FROM QUALIFICATION EXAM              */
294         IF LR>=26 & LR<=35 THEN I=LE;                 /* CONVERSION */ /* SET I = LATEST ENTRY                          */
295         GOTO R(LR);                                                /* BRANCH TO REQUIRED FIELD                        */
```

Dataset Limited

```
296          R(1):X2=SUBSTR(TTFT(1,1),5,2); I=X2;                    /* CONVERSION */     /* -> AGE, CALCULATE AS:                          */
297          X2=SUBSTR(DOB,5,2); J=X2;                               /* CONVERSION */     /*    -> 'YEAR OF ADMISSION - YEAR OF BIRTH'  */
298          KC=I-J;                                                                      /*                                                */
299          GOTO L1;                                                                     /*    -> BRANCH TO CONVERSION ROUTINE             */
300          R(2):ITEM=SEX; RETURN;  R(3):ITEM=MART; RETURN;                              /* -> SEX, MARITAL STATUS                         */
301          R(4):ITEM=SET; RETURN;  R(5):ITEM=REG; RETURN;                               /* -> SETTLEMENT, GEOGRAPHICAL REGION             */
302          R(6):ITEM=PCCC; RETURN; R(7):ITEM=SCCC; RETURN;                              /* -> PARENT/GUARDIAN/STUDENT OCCUPATION          */
303          R(8):ITEM=OV25; RETURN; R(9):ITEM=REL; RETURN;                               /* -> OVER 25 OCCUPATION, RELIGION                */
304          R(10):ITEM=CONF; RETURN;  R(11):ITEM=AUTH; RETURN;                           /* -> CONFIRMATION, AUTHORITY/COUNTRY             */
305          R(12):ITEM=AUTHNO; RETURN;  R(13):ITEM=FPCS; RETURN;                         /* -> AUTHORITY NUMBER, FAMILY POSITION           */
306          R(14):ITEM=TTFT(1,1); RETURN;  R(15):ITEM=UNCOL(1); RETURN;                  /* -> TEACHERS TRAINING, UNIV OR COLLEGE 1        */
307          R(16):ITEM=UNOTH(1); RETURN;  R(17):ITEM=DEG(1); RETURN;                     /* -> UNIV OR OTHER 1, DEGREE AWARDED             */
308          R(18):ITEM=SCHNO; RETURN;                                                    /* -> NO. OF SCHOOLS ATTENDED                     */
309          R(19):KC=0;  QS=SCHNO;                                  /* CONVERSION */     /* -> SCHOOLS                                     */
310          DO I=1 TO QS;                                                                /*                                                */
311             IF ST='*' | SUBSTR(SCH(I),1,1)=ST THEN                                    /*    -> SCHOOL TYPE                              */
312             IF SYF='**' | SUBSTR(SCH(I),2,2)=SYF THEN                                 /*    -> SCHOOL YEAR FROM                         */
313             IF SYT='**' | SUBSTR(SCH(I),4,2)=SYT THEN KC=KC+1;                        /*    -> SCHOOL YEAR TO                           */
314          END;                                                                         /*                                                */
315          GOTO L1;                                                                     /*    -> BRANCH TO CONVERSION ROUTINE             */
316          R(20):ITEM=QUALNO; RETURN;                                                   /* -> NO. OF QUALIFICATIONS RECORDED              */
317          R(21):KC=0; QS=QUALNO;                                  /* CONVERSION */     /* -> QUALIFICATIONS                              */
318          DO I=1 TO QS;                                                                /*                                                */
319             IF QY='**' | SUBSTR(QUAL(I),1,2)=QY THEN                                  /*    -> QUALIFICATION YEAR                       */
320             IF QE='***' | SUBSTR(QUAL(I),3,3)=QE THEN                                 /*    -> QUALIFICATION EXAM                       */
321             IF QC='***' | SUBSTR(QUAL(I),6,3)=QC THEN                                 /*    -> QUALIFICATION CODE                       */
322             IF QG='*' | SUBSTR(QUAL(I),9,1)=QG THEN                                   /*    -> QUALIFICATION GRADE                      */
323             IF QR='*' | SUBSTR(QUAL(I),10,1)=QR THEN KC=KC+1;                         /*    -> QUALIFICATION RESULT                     */
324          END;                                                                         /*                                                */
325          GOTO L1;                                                                     /*    -> BRANCH TO CONVERSION ROUTINE             */
326          R(22):ITEM=ATO; RETURN;  R(23):ITEM=LEN; RETURN;                             /* -> A.T.C, COURSE LENGTH                        */
327          R(24):I=ST;                                             /* CONVERSION */     /* -> SET CODES REQUIRED                          */
328          ITEM=CODES(I); RETURN;  R(25):ITEM=DC46(1); RETURN;                          /*    -> COURSE CODES                             */
329          R(26):ITEM=DC(I,25); RETURN; R(27):ITEM=DC(I,34); RETURN;                    /* -> DETAILS OF COURSE                           */
330          R(28):ITEM=DC(I,35); RETURN; R(29):ITEM=DC(I,36); RETURN;                    /*                                                */
331          R(30):ITEM=DC(I,37); RETURN; R(31):ITEM=DC(I,38); RETURN;                    /*                                                */
332          R(32):ITEM=DC(I,41); RETURN; R(33):ITEM=DC(I,42); RETURN;                    /*                                                */
333          R(34):ITEM=DC(I,59); RETURN; R(35):ITEM=DC(I,64); RETURN;                    /*                                                */
334          R(36):XVAR=SUBSTR(REQI,5,INDEX(REQI,')')-5);                                 /* -> ANNUAL PROGRESS VECTOR                      */
335          I=XVAR;                                                 /* CONVERSION */     /*                                                */
336          IF FK(I,1)¬=0 THEN ITEM=SUBSTR(AP,FK(I,1),FK(I,2)-1); ELSE                   /*                                                */
337          ITEM=' '; RETURN;                                                            /*                                                */
338          R(37):ITEM=TC; RETURN;                                                       /* -> TEACHERS' CERTIFICATE                       */
339          R(38):ITEM=WS; RETURN;                                                       /* -> WING SCIENCE                                */
340          R(39):ITEM=BE; RETURN;                                                       /* -> B.ED.                                       */
341          R(40):ITEM=SUBCAR; RETURN;                                                   /* -> SUBSEQUENT CAREER                           */
342          R(41):R(42):R(43):R(44):R(45):R(46):KC=VALUES(YR,LR-40); GOTO L1;            /* -> VALUES SCORES                               */
343          R(47):KC=DSCALE(YR); GOTO L1;                                                /* -> D.SCALE SCORE                               */
344          R(48):R(49):R(50):KC=AH4(YR,LR-47); GOTO L1;                                 /* -> A.H.4 SCORES                                */
345          R(51):R(52):R(53):R(54):R(55):R(56):KC=PF16(YR,LR-50); GOTO L1;              /* -> P.F.16 SCORES                               */
346          R(57):R(58):R(59):R(60):R(61):R(62):KC=PF16(YR,LR-50); GOTO L1;              /*                                                */
347          R(63):R(64):R(65):R(66):KC=PF16(YR,LR-50);                                   /*                                                */
348    L1:   ITEM=CHAR(KC);                                                                /* CONVERSION ROUTINE                             */
349    END;                                                                                /*                                                */
350          /*****************************************************************/          /*********************************************/
351    TEST:PROC(ITEM,COND,REQU) RETURNS(BIT(1));                                          /* SIMPLE REQUEST TEST ROUTINE                    */
352          DCL (ITEM,REQU,X20) CHAR(20) VARYING, COND FIXED BINARY;                     /*********************************************/
353          DCL C(6) LABEL INITIAL(C1,C2,C3,C4,C5,C6);                                   /* VARIABLE DECLARATIONS                          */
354          X20=ITEM;  ITEM=((20)' ');  SUBSTR(ITEM,21-LENGTH(X20))=X20;                 /* RIGHT JUSTIFY RECORD RESPONSE                  */
355          X20=REQU;  REQU=((20)' ');  SUBSTR(REQU,21-LENGTH(X20))=X20;                 /* RIGHT JUSTIFY REQUIRED VALUE                   */
```

Dataset Limited

```
356          GOTO C(COND);                                          /* BRANCH TO REQUIRED TEST                    */
357     C1:  IF ITEM=REQU THEN RETURN('1'B); ELSE RETURN('0'B);     /* -> CASE '='                                */
358     C2:  IF ITEM¬=REQU THEN RETURN('1'B); ELSE RETURN('0'B);    /* -> CASE '¬='                               */
359     C3:  IF ITEM>REQU THEN RETURN('1'B); ELSE RETURN('0'B);     /* -> CASE '>'                                */
360     C4:  IF ITEM¬>REQU THEN RETURN('1'B); ELSE RETURN('0'B);    /* -> CASE '¬>'                               */
361     C5:  IF ITEM<REQU THEN RETURN('1'B); ELSE RETURN('0'B);     /* -> CASE '<'                                */
362     C6:  IF ITEM¬<REQU THEN RETURN('1'B); ELSE RETURN('0'B);    /* -> CASE '¬<'                               */
363     END;                                                        /*                                            */
364     DCL TEST ENTRY(,,) RETURNS(BIT(1));                         /*                                            */
365          /*****************************************************/ /*******************************************/
366          DO I=1 TO K;                                           /* CONSIDER FIRST/NEXT SIMPLE REQUEST         */
367             IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19  /* -> LOCATE EXCEPTIONAL CASES 1.     */
368             | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN  /*                                    */
369             REQI=SUBSTR(RITEM(I),INDEX(RITEM(I),'('));             /*    -> SEPARATE REQUIRED ITEM               */
370             IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=36  /* -> LOCATE EXCEPTIONAL CASES 2.     */
371             | RLAB(I)=40 THEN IF SUBSTR(REQI,2,2)¬=SUBSTR(TTFT(1,1),5,2)  /*    -> TEST YEARS                   */
372             THEN IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);  /*    -> BRANCH OR REJECT RECORD      */
373             CALL FIND(YR,ITEM,RLAB(I),REQI);                       /* -> FIND REQUIRED FIELD VALUE               */
374             IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19  /* -> LOCATE EXCEPTIONAL CASES 3.     */
375             | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN  /*                                    */
376             REQM=SUBSTR(RITEM(I),1,INDEX(RITEM(I),'(')-1); ELSE    /*    -> SEPARATE REQUIRED ITEM               */
377             REQM=TRANSLATE(RITEM(I),' ',':');                      /* -> CLEAR IMBEDDED BLANKS                   */
378             IF TEST(ITEM,CLAB(I),REQM) THEN                        /* -> TEST SIMPLE REQUEST                     */
379             DO; DO J=I+1 TO K;                                     /*                                            */
380                    IF LOP(J)¬='|' THEN                             /* -> TEST NEXT LOGICAL OPERATOR              */
381                    DO; I=J-1;                                      /*                                            */
382                        GOTO L66;                                   /*    -> BRANCH  FOR NEXT SIMPLE REQUEST      */
383                    END;                                            /*                                            */
384                 END;                                               /*                                            */
385                 RETURN('1'B);                                      /*    -> ACCEPT RECORD                        */
386             END; ELSE                                              /*                                            */
387             IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);  /* -> BRANCH OR REJECT RECORD            */
388     L66: END;                                                      /* BRANCH FOR NEXT SIMPLE REQUEST             */
389     END;                                                           /*                                            */
390     DCL LOCTST ENTRY(,,,,,) RETURNS(BIT(1));                       /*                                            */
391     /*****************************************************************/ /*******************************************/
392          ON ENDFILE(SCARDS) GOTO L09;                              /* BRANCH ON ENDFILE (SCARDS)                 */
393     L39: GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));  /* INPUT OPTION PARAMETER CARD               */
394          GET LIST(OPT);                                            /* INPUT OPTION                               */
395          IF OPT<1 | OPT>4 THEN                                     /* TEST OPTION                                */
396          DO; PUT EDIT('OPTION INVALID.....PROGRAM TERMINATED')     /* -> CLASS A ERROR/WARNING                   */
397              (SKIP,COL(1),A);                                      /*                                            */
398              GOTO L09;                                             /* -> BRANCH TO PROGRAMME TERMINATION         */
399          END;                                                      /*                                            */
400          WRL='XXX'; WCR='X'; YT='F'; PI='0'; REQUEST=' '; B='0'B;  /* INITIALISE OPTION PARAMETERS               */
401          GET DATA;                                                 /* INPUT OPTION PARAMETERS                    */
402          IF SUBSTR(WRL,1,1)='1' THEN WB='1'B; ELSE WB='0'B;        /* VERIFY OPTION PARAMETERS                   */
403          IF SUBSTR(WRL,1,1)='2' THEN WPB='1'B; ELSE WPB='0'B;      /* -> PARAMETER WRL                           */
404          IF SUBSTR(WRL,2,1)='1' THEN RB='1'B; ELSE RB='0'B;        /*                                            */
405          IF SUBSTR(WRL,2,1)='2' THEN RPB='1'B; ELSE RPB='0'B;      /*                                            */
406          IF SUBSTR(WRL,3,1)='1' THEN LB='1'B; ELSE LB='0'B;        /*                                            */
407          IF SUBSTR(WRL,3,1)='2' THEN LPB='1'B; ELSE LPB='0'B;      /*                                            */
408          IF YT='F' THEN YR=1; ELSE IF YT='L' THEN YR=2; ELSE YR=1; /* -> PARAMETER YT                            */
409          PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME I',  /* OUTPUT TITLES                          */
410          'OPTION REQUESTED',OPT,'YEAR REQUESTED',YE)(PAGE,COL(1),A,  /*                                          */
411          SKIP(2),COL(1),A,COL(1),A,COL(32),F(2),COL(1),A,COL(30),A);  /*                                        */
412          PUT SKIP(2);                                              /*                                            */
413          IF WRL¬='XXX' THEN PUT EDIT('WRL REQUESTED',WRL)          /* OUTPUT OPTION PARAMETERS IN EFFECT         */
414          (COL(1),A,COL(31),A); ELSE WRL='000';                     /* -> WRL                                     */
415          IF WCR¬='X' THEN PUT EDIT('WCR REQUESTED',WCR)(COL(1),A,COL(33),  /* -> WCR                             */
```

Dataset Limited

```
416          A); ELSE WCR='*';                                          /*                                          */
417          PUT EDIT('YEAR OF TEST',YT)(COL(1),A,COL(33),A);           /* -> YT                                    */
418          IF PI¬='0' THEN PUT EDIT('PRINT INDICATOR',PI)             /* -> PI                                    */
419          (COL(1),A,COL(33),A);                                      /*                                          */
420          PUT SKIP(4);                                               /*                                          */
421          IF REQUEST='*' THEN GOTO LOO;                              /* BRANCH IF SPECIAL CASE 'REQUEST ='*''    */
422     /*-----------------------------------------------------------*/ /*-----------------------------------------*/
423     L1234:OPEN FILE(STUDENT) INPUT TITLE(FILN1);                    /* OPTIONS 1,2,3 AND 4                      */
424          ON ENDFILE(STUDENT)                                        /* SET ENDFILE (STUDENT) ROUTINE            */
425          BEGIN; DT=DATE;                                            /* -> CALCULATE DATE                        */
426                 DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||  /*                                      */
427                 SUBSTR(DT,7,2);                                     /*                                          */
428                 IF OPT=1 | OPT=2 THEN                               /* -> OPTIONS 1 AND 2                       */
429                 DO; IF OPT=1 THEN PUT EDIT(YE,'FILE........INTELLIGENCE ',  /* -> OUTPUT TITLES FOR OPTION 1        */
430                     'TEST TOTALS (MEANS) ST. DEV')(SKIP(2),A(4),X(1),A,A);  /*                                  */
431                     IF OPT=2 THEN PUT EDIT(YE,'FILE........INTELLIGENCE ',  /* -> OUTPUT TITLES FOR OPTION 2        */
432                     'TEST FREQUENCY COUNTS')(SKIP(2),A(4),X(1),A,A);  /*                                       */
433                     PUT EDIT(KT,'STUDENTS INCLUDED')                  /* -> OUTPUT SAMPLE SIZE 'KT'            */
434                     (SKIP(3),COL(1),F(4),X(1),A);                     /*                                       */
435                     IF KT=0 THEN GOTO L21;                            /* -> BRANCH IF KT=0                     */
436                     IF ¬SUPP THEN CALL SPNT;                          /* -> TEST SUPPRESSION BIT, OUTPUT DES.NOS. */
437                     PUT PAGE;                                         /*                                       */
438                     IF OPT=1 THEN CALL ITPTMS; ELSE CALL ITPFREQ;     /* -> CALL REQUIRED OUTPUT ROUTINE       */
439     L21:            END; ELSE                                         /*                                       */
440                 IF OPT=3 THEN                                         /* -> OPTION 3                           */
441                 DO; IF PI¬='1' THEN                                   /* -> TEST PRINT INDICATOR, CASE PI ¬=1  */
442                     DO; PUT EDIT(YE,'FILE.....INTELLIGENCE TESTS',KT,  /* -> OUTPUT TITLES                     */
443                         'STUDENTS INCLUDED')(PAGE,COL(1),A,X(1),A,X(1),  /*                                  */
444                         F(4),X(1),A);                                 /*                                       */
445                         IF ¬SUPP THEN CALL SPNT;                      /* -> TEST SUPPRESSION BIT, OUTPUT DES.NOS. */
446                     END;                                              /*                                       */
447                 END; ELSE                                             /*                                       */
448                 IF OPT=4 THEN                                         /* -> OPTION 4                           */
449                 DO; PUT EDIT('TOTALS,MEANS,STANDARD DEVIATIONS AND ',  /* -> OUTPUT TITLES                     */
450                     'FREQUENCY COUNTS FOR REQUEST',REQUEST)           /*                                       */
451                     (SKIP(2),COL(1),A,A,X(2),A);                      /*                                       */
452                     PUT EDIT(KT,'STUDENTS INCLUDED')                  /* -> OUTPUT SAMPLE SIZE 'KT'            */
453                     (SKIP(3),COL(1),F(4),X(1),A);                     /*                                       */
454                     IF KT=0 THEN GOTO L51;                            /* -> BRANCH IF KT=0                     */
455                     IF ¬SUPP THEN CALL SPNT;                          /* -> TEST SUPPRESSION BIT, OUTPUT DES.NOS. */
456                     PUT PAGE;                                         /*                                       */
457                     CALL ITPTMS;                                      /* -> OUTPUT TOTALS, MEANS AND ST.DEVIATIONS */
458                     PUT PAGE;                                         /*                                       */
459                     CALL ITPFREQ;                                     /* -> OUTPUT FREQUENCIES                 */
460     L51:            END;                                              /*                                       */
461                 PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);  /* -> OUTPUT DATE OF LISTING        */
462                 CLOSE FILE(STUDENT);                                  /* -> CLOSE FILE                         */
463                 GOTO L39;                                             /* -> BRANCH TO PROGRAMME CONTROL SECTION */
464          END;                                                         /* END OF ENDFILE ROUTINE                */
465          KT=0; DESL=' '; SUPP='0'B;                                   /* INITIALISE VARIABLES ALL OPTIONS      */
466          IF OPT=1 THEN                                                /* INITIALISE VARIABLE OPTION 1          */
467          DO; VT=0; VST=0; DST=0; DSST=0; AHT=0; AHST=0; PFT=0; PFST=0;  /*                                  */
468              SST=0; SSST=0;                                           /*                                       */
469          END; ELSE                                                    /*                                       */
470          IF OPT=2 THEN                                                /* INITIALISE VARIABLES OPTION 2         */
471          DO; VF=0; DSF=0; AHF=0;                                      /*                                       */
472              PFF=0; SF=0;                                             /*                                       */
473          END; ELSE                                                    /*                                       */
474          IF OPT=4 THEN                                                /* INITIALISE VARIABLES OPTION 4         */
475          DO; LOP=' '; CLAB=0; RLAB=0; RITEM=' '; K=0;                 /*                                       */
```

Dataset Limited

```
476             IF SUBSTR(REQUEST,2,1)¬=' ' THEN                        /* TEST FOR REQUEST                          */
477             CALL TRANS(REQUEST,LOP,CLAB,RLAB,RITEM,K);              /* TRANSLATE REQUEST                         */
478             IF K=0 THEN                                             /* TEST NO. OF SIMPLE REQUESTS, IF 0 THEN    */
479             DO; CLOSE FILE(STUDENT);                                /* -> CLOSE FILE                             */
480                 GOTO L39;                                           /* -> BRANCH TO PROGRAMME CONTROL SECTION    */
481             END;                                                    /*                                           */
482             VT,VST=0; DST,DSST=0; AHT,AHST=0; PFT,PFST=0; SST,SSST=0;/* INITIALISE VARIABLES OPTION 4             */
483             VF=0; DSF=0; AHF=0; PFF=0; SF=0;                        /*                                           */
484         END;                                                        /*                                           */
485     L31: READ FILE(STUDENT) SET(PTR);                               /* READ FIRST/NEXT RECORD                    */
486         IF SUBSTR(INTELB,YR,1)¬='*' THEN                            /* TEST FOR INTELLIGENCE DATA                */
487         DO; IF OPT¬=3 THEN PUT EDIT(DESNO,'HAS NO CORRESPONDING DATA')/* -> CLASS L ERROR/WARNING                */
488             (COL(1),A,X(1),A);                                      /*                                           */
489             GOTO L31;                                               /* -> BRANCH FOR NEXT RECORD                 */
490         END;                                                        /*                                           */
491         IF ¬WB & SUBSTR(TTFT(2,1),1,4)='WITH' THEN                  /* -> TEST FOR WITHDRAWAL                    */
492         DO; IF WPB THEN                                             /*                                           */
493             DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),/*    -> CALCULATE WITHDRAWAL DATE           */
494                 3,2) || '/' || SUBSTR(TTFT(1,2),5,2);               /*                                           */
495                 PUT EDIT(DESNO,'WITHDREW ON',WD)(COL(1),A,2(X(1),A));/*    -> CLASS M ERROR/WARNING               */
496             END;                                                    /*                                           */
497             GOTO L31;                                               /*    -> BRANCH FOR NEXT RECORD              */
498         END; ELSE                                                   /*                                           */
499         IF ¬LB & SUBSTR(TTFT(2,1),1,4)='LEFT' THEN                  /* -> TEST FOR LEFT                          */
500         DO; IF LPB THEN                                             /*                                           */
501             DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),/*    -> CALCULATE LEAVING DATE              */
502                 3,2) || '/' || SUBSTR(TTFT(1,2),5,2);               /*                                           */
503                 PUT EDIT(DESNO,'LEFT ON',WD)(COL(1),A,2(X(1),A));   /*    -> CLASS M ERROR/WARNING               */
504             END;                                                    /*                                           */
505             GOTO L31;                                               /*    -> BRANCH FOR NEXT RECORD              */
506         END; ELSE                                                   /*                                           */
507         IF ¬RB & SUBSTR(TTFT(2,1),1,4)='    ' THEN                  /* -> TEST FOR STUDENT AT COLLEGE            */
508         DO; IF RPB THEN PUT EDIT(DESNO,'AT COLLEGE')(COL(1),A,X(1),A);/*    -> CLASS M ERROR/WARNING               */
509             GOTO L31;                                               /*    -> BRANCH FOR NEXT RECORD               *
510         END; ELSE                                                   /*                                           */
511         IF WB & WCR¬='*' & WCR¬=SUBSTR(TTFT(2,1),6,1) THEN GOTO L31;/* -> TEST WITHDRAWAL CODE                   */
512         IF OPT=1 | OPT=2 | OPT=3 |                                  /* OPTIONS 1,2,3 OR                          */
513         (OPT=4 & LOCTST(YR,LOP,CLAB,RLAB,RITEM,K)) THEN             /* OPTION 4 WITH REQUEST 'TRUE'              */
514         DO; KT=KT+1;                                                /* -> UPDATE COUNTER                         */
515             IF KT>1000 THEN SUPP='1'B;                              /* -> TEST COUNTER AND SET SUPPRESSION BIT   */
516             IF ¬SUPP THEN DESL(KT)=DESNO;                           /* -> ASSIGN DES.NO. TO LIST                 */
517             IF OPT=1 | OPT=4 THEN                                   /* -> OPTIONS 1 AND 4                        */
518             CALL ITTMS;                                             /*    -> ACCUMULATE TOTALS                   */
519             IF OPT=2 | OPT=4 THEN                                   /* -> OPTIONS 2 AND 4                        */
520             CALL ITFREQ;                                            /*    -> ACCUMULATE FREQUENCIES              */
521             IF OPT=3 & PI='1' THEN CALL PRINT;                      /* -> OPTION 3, PRINT SCORES IF PI=1         */
522         END;                                                        /*                                           */
523         GOTO L31;                                                   /* BRANCH FOR NEXT RECORD.                   */
524 /*-----------------------------------------------------------------*/ /*-----------------------------------------*/
525     L00: OPEN FILE(STUDENT) INPUT TITLE(FILN1);                     /* OPTION 4, SPECIAL CASE 'REQUEST='*''      */
526         ON ENDFILE(STUDENT)                                         /* SET ENDFILE (STUDENT) ROUTINE             */
527         BEGIN; CLOSE FILE(STUDENT);                                 /* -> CLOSE FILE                             */
528                OPEN FILE(STUDENT) INPUT TITLE(FILN1);               /* -> OPEN FILE                              */
529                GOTO L01;                                            /* -> BRANCH FOR FIRST RECORD                */
530         END;                                                        /* END ENDFILE ROUTINE                       */
531         VT=0; VST=0; DST=0; DSST=0; AHT=0; AHST=0; PFT=0; PFST=0;   /* INITIALISE VARIABLES                      */
532         VF=0; DSF=0; AHF=0; PFF=0; SF=0; SST=0; SSST=0;             /*                                           */
533         DESL=' '; KT=0; SUPP='0'B;                                  /*                                           */
534     L02: GET EDIT(DESNO2)(COL(1),A(8));                             /* INPUT REQUIRED DES.NO.                    */
535         IF DESNO2='********' THEN                                   /* TEST FOR INPUT TERMINATOR                 */
```

Dataset Limited

```
536          DO; PUT EDIT('TOTALS,MEANS,STANDARD DEVIATIONS AND FREQUENCY',      /* OUTPUT ROUTINE, TITLES                    */
537              ' COUNTS FOR REQUEST=''*''')(SKIP(2),COL(1),A,A);               /*                                           */
538              PUT EDIT(KT,'STUDENTS INCLUDED')(SKIP(3),COL(1),F(4),X(1),A);   /* -> OUTPUT SAMPLE SIZE 'KT'                */
539              IF KT=0 THEN GOTO L03;                                          /* -> BRANCH IF KT=0                         */
540              IF ¬SUPP THEN CALL SPNT;                                        /* -> TEST SUPPRESSION BIT, OUTPUT DES.NO.   */
541              PUT PAGE;                                                       /*                                           */
542              CALL ITPTMS;                                                    /* -> OUTPUT TOTALS, MEANS AND ST.DEVIATIONS.*/
543              PUT PAGE;                                                       /*                                           */
544              CALL ITPFREQ;                                                   /* -> OUTPUT FREQUENCIES                     */
545     L03:     DT=DATE;                                                        /* -> CALCULATE DATE                         */
546              DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||            /*                                           */
547              SUBSTR(DT,7,2);                                                 /*                                           */
548              PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);        /* -> OUTPUT DATE                            */
549              CLOSE FILE(STUDENT);                                            /* -> CLOSE FILE                             */
550              GOTO L39;                                                       /* -> BRANCH TO PROGRAMME CONTROL SECTION    */
551          END;                                                                /* END OUTPUT ROUTINE                        */
552     L01: READ FILE(STUDENT) SET(PTR);                                        /* READ FIRST/NEXT RECORD                    */
553          IF DESNO=DESNO2 THEN                                                /* TEST RECORD DES.NO. WITH REQUIRED DES.NO. */
554          DO; IF B THEN B='0'B;                                               /* -> RECORD 'FOUND' ROUTINE, RESET BIT      */
555              IF SUBSTR(INTELB,YP,1)¬='*' THEN                                /* -> TEST FOR INTELLIGENCE DATA             */
556              DO; PUT EDIT(DESNO,'HAS NO CORRESPONDING DATA')                 /*    -> CLASS L ERROR/WARNING               */
557                  (COL(1),A,X(1),A);                                          /*                                           */
558                  GOTO L02;                                                   /*    -> BRANCH FOR NEXT REQUIRED DES.NO.    */
559              END;                                                            /*                                           */
560              KT=KT+1;                                                        /* -> UPDATE COUNTER                         */
561              IF KT>1000 THEN SUPP='1'B;                                      /* -> TEST COUNTER AND SET SUPPRESSION BIT   */
562              IF ¬SUPP THEN DESL(KT)=DESNO;                                   /* -> ASSIGN DES.NO. TO LIST                 */
563              CALL ITTMS;                                                     /* -> ACCUMULATE TOTALS                      */
564              CALL ITFREQ;                                                    /* -> ACCUMULATE FREQUENCIES                 */
565              GOTO L02;                                                       /* -> BRANCH FOR NEXT REQUIRED DES.NO.       */
566          END; ELSE                                                           /* END 'RECORD FOUND' ROUTINE                */
567          DO; IF ¬B THEN                                                      /* RECORD NOT FOUND ROUTINE                  */
568              DO; DESNO3=DESNO;                                               /* -> RECORD CURRENT DES.NO.                 */
569                  B='1'B;                                                     /* -> SET 'RECORD FOUND' BIT                 */
570              END; ELSE                                                       /*                                           */
571              DO; IF DESNO=DESNO3 THEN                                        /* -> TEST FOR COMPLETE FILE CYCLE           */
572                  DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')      /* -> CLASS C ERROR/WARNING                  */
573                      (COL(1),A,X(1),A);                                      /*                                           */
574                      B='0'B;                                                 /* -> RESET 'RECORD FOUND' BIT               */
575                      GOTO L02;                                               /* -> BRANCH FOR NEXT REQUIRED DES.NO.       */
576                  END;                                                        /*                                           */
577              END;                                                            /*                                           */
578              GOTO L01;                                                       /* -> BRANCH FOR NEXT RECORD                 */
579          END;                                                                /* END RECORD 'NOT FOUND' ROUTINE            */
580     /*-----------------------------------------------------------------*/    /*-----------------------------------------*/
581     L09: PUT PAGE;                                                           /*                                           */
582     END;                                                                     /* END PROGRAMME I.                          */
```

Dataset Limited

5.3 COMPILER LISTING

THIS SECTION CONTAINS A LISTING OF THE OUTPUT PROVIDED BY THE PL/1 F COMPILER DURING THE COMPILATION OF PROGRAMME I.

THE JCL WAS AS FOLLOWS:

```
$RUN *PL1 SPUNCH=-I
  ----------------------------
 |                            |
 |                            |
 |  PROGRAMME I  SOURCE DECK  |
 |                            |
 |                            |
  ----------------------------
$ENDFILE
```

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRC
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST
```

```
*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRC,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
*OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
*OPTIONS IN EFFECT*     NEST,OPLIST
```

```
STMT LEVEL NEST
                                                                        I0000010
  1                 (SUBRG):(STRG):                                     I0000020
                    I:PROCEDURE OPTIONS(MAIN);                          I0000030
  2     1                DCL 1 RECORD BASED(PTR),                       I0000040
                               2 (DESNO CHAR(8), NAME CHAR(40)),        I0000050
                               2 GENERAL,                               I0000060
                                 3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),I0000070
                                 3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),      I0000080
                                 3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)), I0000090
                                 3 FPOS CHAR(4),                        I0000100
                               2 EDUCATION,                             I0000110
                                 3 (SCHNO CHAR(2), SCH(10) CHAR(5)),    I0000120
                                 3 (QUALNO CHAR(2), QUAL(30) CHAR(10)), I0000130
                                 3 (UNCOL(2) CHAR(21), FPT(1) CHAR(2), FT(2,2) CHAR(6)),   I0000140
                                 3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),   I0000150
                                 3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),  I0000160
                                 3 (ADIP CHAR(30), ADIPC CHAR(1)),      I0000170
                                 3 DDIPC CHAR(1),                       I0000180
                                 3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),    I0000190
                                 3 ATO CHAR(1),                         I0000200
                               2 COURSE,                                I0000210
                                 3 (LEN CHAR(1), CODES(4) CHAR(3)),     I0000220
                                 3 ((LE,DC(4,24:64)) CHAR(1)),          I0000230
                                 3 (DC1(4,26:30) CHAR(2)),              I0000240
                                 3 ((DC46(4),DC49(4)) CHAR(3)),         I0000250
                               2 FINALS,                                I0000260
                                 3 ((TC,WS,BE,SUBCAR) CHAR(1)),         I0000270
                               2 ANNPROG,                               I0000280
                                 3 AP CHAR(802),                        I0000290
                                 3 ((LEAP,FK(100,2)) FIXED BINARY),     I0000300
                               2 INTEL,                                 I0000310
                                 3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY), I0000320
                                 3 ((AH4(2,3),PF16(2,16)) FIXED BINARY), I0000330
                                 3 STEN(2,16) FIXED BINARY;             I0000340
                    /*-----------------------------------------------------------------*/I0000350
  3     1                DCL (VF(0:60,6),DSF(0:300),AHF(0:130,3),PFF(0:26,16),SF(0:10,16)) I0000360
                         FIXED BINARY;                                  I0000370
  4     1                DCL ((VT,VST)(6),(DST,DSST),(AHT,AHST)(2),(PFT,PFST)(16),         I0000380
                         (SST,SSST)(16)) FLOAT DECIMAL(15);             I0000390
  5     1                DCL LOP(20) CHAR(1), (RLAB,CLAB)(20) FIXED BINARY,                I0000400
                         RITEM(20) CHAR(20) VARYING;                    I0000410
  6     1                DCL MNT(16) FIXED BINARY INITIAL(20,13,26,26,26,20,26,            I0000420
                         20,20,26,20,26,20,20,20,26);                   I0000430
  7     1                DCL (FILN1,FILN2,WD,DT,DESL(1000)) CHAR(8), YE CHAR(4), WRL       I0000440
                         CHAR(3), (YT,WCP,PI) CHAR(1), (KT,OPT,YR,K) FIXED BINARY, (WB,LB, I0000450
                         RB,SUPP,WPB,LPB,RPB,B) BIT(1), REQUEST CHAR(400) VARYING,         I0000460
                         (DESNO3,DESNO2) CHAR(8);                       I0000470
                    /*-----------------------------------------------------------------*/I0000480
  8     1                DCL STUDENT FILE RECORD ENV(F(2416));          I0000490
                    /*****************************************************************/I0000500
  9     1           ITTMS:PROC;                                         I0000510
 10     2                DCL K FIXED BINARY;                            I0000520
 11     2                DO K=1 TO 6;                                   I0000530
 12     2     1             IF VALUES(YR,K)>=0 & VALUES(YR,K)<=60 THEN  I0000540
 13     2     1             DO; VT(K)=VT(K)+VALUES(YR,K);               I0000550
 15     2     2                 VST(K)=VST(K)+VALUES(YR,K)**2;          I0000560
```

Dateset Limited

```
STMT LEVEL NEST

 16     2    2            END; ELSE                                          I0000570
 17     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR VALUES SCORE',K)  I0000580
                          (SKIP,COL(1),A,X(1),A,X(1),F(1));                  I0000590
 18     2    1         END;                                                  I0000600
 19     2              IF DSCALE(YR)>=0 & DSCALE(YR)<=300 THEN               I0000610
 20     2              DO; DST=DST+DSCALE(YR);                               I0000620
 22     2    1             DSST=DSST+DSCALE(YR)**2;                         I0000630
 23     2    1         END; ELSE                                             I0000640
 24     2              PUT EDIT(DESNO,'ERROR RETURNED FOR D.SCALE SCORE')    I0000650
                       (SKIP,COL(1),A,X(1),A);                               I0000660
 25     2              DO K=1,2;                                             I0000670
 26     2    1            IF AH4(YR,K)>=0 & AH4(YR,K)<=130 THEN              I0000680
 27     2    1            DO; AHT(K)=AHT(K)+AH4(YR,K);                       I0000690
 29     2    2                AHST(K)=AHST(K)+AH4(YR,K)**2;                  I0000700
 30     2    2            END; ELSE                                          I0000710
 31     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR A.F.4 SCORE',K) I0000720
                          (SKIP,COL(1),A,X(1),A,X(1),F(2));                  I0000730
 32     2    1         END;                                                  I0000740
 33     2              DO K=1 TO 16;                                         I0000750
 34     2    1            IF PF16(YR,K)>=0 & PF16(YR,K)<=MNT(K) THEN         I0000760
 35     2    1            DO; PFT(K)=PFT(K)+PF16(YR,K);                      I0000770
 37     2    2                PFST(K)=PFST(K)+PF16(YR,K)**2;                 I0000780
 38     2    2                SST(K)=SST(K)+STEN(YR,K);                      I0000790
 39     2    2                SSST(K)=SSST(K)+STEN(YR,K)**2;                 I0000800
 40     2    2            END; ELSE                                          I0000810
 41     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR P.F.16 SCORE',K) I0000820
                          (SKIP,COL(1),A,X(1),A,X(1),F(2));                  I0000830
 42     2    1         END;                                                  I0000840
 43     2           END;                                                     I0000850
                     /*****************************************************/I0000860
 44     1           ITFREQ:PROC;                                             I0000870
 45     2              DCL K FIXED BINARY;                                   I0000880
 46     2              DO K=1 TO 6;                                          I0000890
 47     2    1            IF VALUES(YR,K)>=0 & VALUES(YR,K)<=60 THEN         I0000900
 48     2    1            VF(VALUES(YR,K),K)=VF(VALUES(YR,K),K)+1; ELSE      I0000910
 49     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR VALUES SCORE',K) I0000920
                          (SKIP,COL(1),A,X(1),A,X(1),F(1));                  I0000930
 50     2    1         END;                                                  I0000940
 51     2              IF DSCALE(YR)>=0 & DSCALE(YR)<=300 THEN               I0000950
 52     2              DSF(DSCALE(YR))=DSF(DSCALE(YR))+1; ELSE               I0000960
 53     2              PUT EDIT(DESNO,'ERROR RETURNED FOR D.SCALE SCORE')    I0000970
                       (SKIP,COL(1),A,X(1),A);                               I0000980
 54     2              DO K=1,2;                                             I0000990
 55     2    1            IF AH4(YR,K)>=0 & AH4(YR,K)<=130 THEN              I0001000
 56     2    1            AHF(AH4(YR,K),K)=AHF(AH4(YR,K),K)+1; ELSE          I0001010
 57     2    1            PUT EDIT(DESNO,'ERROR RETURNED FOR A.F.4 SCORE',K) I0001020
                          (SKIP,COL(1),A,X(1),A,X(1),F(2));                  I0001030
 58     2    1         END;                                                  I0001040
 59     2              DO K=1 TO 16;                                         I0001050
 60     2    1            IF PF16(YR,K)>=0 & PF16(YR,K)<=MNT(K) THEN         I0001060
 61     2    1            PFF(PF16(YR,K),K)=PFF(PF16(YR,K),K)+1; ELSE        I0001070
 62     2    1            DO; PUT EDIT(DESNO,'ERROR RETURNED FOR P.F.16 SCORE',K) I0001080
                              (SKIP,COL(1),A,X(1),A,X(1),F(2));              I0001090
 64     2    2                RETURN;                                        I0001100
```

Dataset Limited

```
STMT LEVEL NEST

 65     2    2            END;                                                    I0001110
 66     2    1            SF(STEN(YR,K),K)=SF(STEN(YR,K),K)+1;                    I0001120
 67     2    1         END;                                                       I0001130
 68     2           END;                                                          I0001140
                     /*****************************************************************/I0001150
 69     1           SPNT:PROC;                                                    I0001160
 70     2              DCL I FIXED BINARY;                                        I0001170
 71     2              PUT SKIP(3);                                               I0001180
 72     2              DO I=1 TO KT;                                              I0001190
 73     2    1            IF MOD(I-1,10)=0 THEN PUT EDIT(DESL(I))(SKIP,X(3),A(8)); ELSE  I0001200
 75     2    1            PUT EDIT(DESL(I))(X(3),A(8));                          I0001210
 76     2    1         END;                                                       I0001220
 77     2           END;                                                          I0001230
                     /*****************************************************************/I0001240
 78     1           PRINT:PROC;                                                   I0001250
 79     2              IF MOD(KT-1,25)=0 THEN                                     I0001260
 80     2              DO; PUT EDIT('D.E.S.NO','(-----VALUES----)','(D.SCALE)',   I0001270
                           '(--AH.4--)','(--------------------PF.16--------------------)'I0001280
                           )(PAGE,COL(1),A(8),4(X(4),A));                         I0001290
 82     2    1            PUT SKIP;                                              I0001300
 83     2    1         END;                                                       I0001310
 84     2              PUT EDIT(DESNO,VALUES(YR,*),DSCALE(YR),AH4(YR,*),PF16(YR,*))  I0001320
                        (SKIP,COL(1),A,X(3),6(X(1),F(2)),X(7),F(3),X(6),2(X(1),F(2)),X(2),I0001330
                        F(3),X(3),16(X(1),F(2)));                                 I0001340
 85     2              PUT EDIT(STEN(YR,*))(SKIP,COL(60),16(X(1),F(2)));          I0001350
 86     2           END;                                                          I0001360
                     /*****************************************************************/I0001370
 87     1           ITPTMS:PROC;                                                  I0001380
 88     2              DCL K FIXED BINARY;                                        I0001390
 89     2              DCL PFK(16) CHAR(2) INITIAL(' A',' B',' C',' E',' F',' G',' H',  I0001400
                        ' I',' L',' M',' N',' O','Q1','Q2','Q3','Q4');            I0001410
 90     2              VST=SQRT((VST-(VT**2/KT))/KT);                             I0001420
 91     2              DSST=SQRT((DSST-(DST**2/KT))/KT);                          I0001430
 92     2              AHST=SQRT((AHST-(AHT**2/KT))/KT);                          I0001440
 93     2              PFST=SQRT((PFST-(PFT**2/KT))/KT);                          I0001450
 94     2              SSST=SQRT((SSST-(SST**2/KT))/KT);                          I0001460
 95     2              PUT EDIT('TEST','TOTAL( MEAN )ST.DEV','TOTAL( MEAN )ST.DEV')  I0001470
                        (SKIP(2),COL(7),A,COL(20),A,X(5),A);                      I0001480
 96     2              PUT EDIT('VALUES')(SKIP(2),COL(5),A);                      I0001490
 97     2              DO K=1 TO 6;                                               I0001500
 98     2    1            PUT EDIT('(',K,')')(COL(11),A,F(2),A);                 I0001510
 99     2    1            PUT EDIT(VT(K),'(',(VT(K)/KT),')',VST(K))              I0001520
                           (X(2),F(8),A,F(6,2),A,F(6,2));                         I0001530
100     2    1         END;                                                       I0001540
101     2              PUT EDIT('D.SCALE     ')(SKIP(2),COL(4),A);                I0001550
102     2              PUT EDIT(DST,'(',(DST/KT),')',DSST)                        I0001560
                        (X(2),F(8),A,F(6,2),A,F(6,2));                            I0001570
103     2              PUT EDIT('A.H.4')(SKIP(2),COL(6),A);                       I0001580
104     2              DO K=1,2;                                                  I0001590
105     2    1            PUT EDIT('(',K,')')(COL(11),A,F(2),A);                 I0001600
106     2    1            PUT EDIT(AHT(K),'(',(AHT(K)/KT),')',AHST(K))           I0001610
                           (X(2),F(8),A,F(6,2),A,F(6,2));                         I0001620
107     2    1         END;                                                       I0001630
108     2              PUT EDIT('STEN SCORES')(X(13),A);                          I0001640
```

Dataset Limited

```
STMT LEVEL NEST

109     2              PUT EDIT('P.F.16')(SKIP(2),COL(5),A);                          I0001650
110     2              DO K=1 TO 16;                                                  I0001660
111     2    1            PUT EDIT('(',PFK(K),')')(COL(11),A,A(2),A);                I0001670
112     2    1            PUT EDIT(PFT(K),'(',(PFT(K)/KT),')',PFST(K))               I0001680
                          (X(2),F(8),A,F(6,2),A,F(6,2));                             I0001690
113     2    1            PUT EDIT(SST(K),'(',(SST(K)/KT),')',SSST(K))               I0001700
                          (X(2),F(8),A,F(6,2),A,F(6,2));                             I0001710
114     2    1         END;                                                          I0001720
115     2          END;                                                              I0001730
                   /*****************************************************************/I0001740
116     1          ITPFREQ:PROC;                                                     I0001750
117     2              DCL (I,J,K) FIXED BINARY;                                     I0001760
118     2              PUT EDIT('VALUES','SCORE','FREQUENCY')(SKIP(2),COL(1),A,SKIP, I0001770
                       COL(2),A,COL(25),A);                                          I0001780
119     2              PUT SKIP;                                                     I0001790
120     2              DO I=0 TO 60;                                                 I0001800
121     2    1            DO J=1 TO 6;                                              I0001810
122     2    2               IF VF(I,J)>0 THEN                                      I0001820
123     2    2               DO; PUT EDIT(I,(VF(I,K) DO K=1 TO 6))(SKIP,COL(5),F(2), I0001830
                                 6(X(4),F(4)));                                     I0001840
125     2    3                   GOTO L1;                                           I0001850
126     2    3               END;                                                   I0001860
127     2    2            END;                                                      I0001870
128     2    1     L1:  END;                                                        I0001880
129     2              PUT EDIT('D.SCALE','SCORE','FREQUENCY')(PAGE,COL(1),A,SKIP,COL(3),I0001890
                       A,COL(20),A);                                                 I0001900
130     2              PUT SKIP;                                                     I0001910
131     2              DO I=0 TO 300;                                                I0001920
132     2    1            IF DSF(I)>0 THEN PUT EDIT(I,DSF(I))(SKIP,COL(5),F(3),COL(21), I0001930
                          F(4));                                                    I0001940
134     2    1         END;                                                          I0001950
135     2              PUT EDIT('A.H.4','SCORE','FREQUENCY')(PAGE,COL(1),A,SKIP,COL(1),A,I0001960
                       COL(20),A);                                                   I0001970
136     2              PUT SKIP;                                                     I0001980
137     2              DO I=1 TO 130;                                                I0001990
138     2    1            IF AHF(I,1)>0 | AHF(I,2)>0 THEN                           I0002000
139     2    1            PUT EDIT(I,AHF(I,1),AHF(I,2))(SKIP,COL(3),F(3),           I0002010
                          X(7),2(X(4),F(4)));                                       I0002020
140     2    1         END;                                                          I0002030
141     2              PUT EDIT('P.F.16','SCORE','FREQUENCY')(PAGE,COL(1),A,SKIP,COL(2), I0002040
                       A,COL(50),A);                                                 I0002050
142     2              PUT SKIP;                                                     I0002060
143     2              DO I=0 TO 26;                                                 I0002070
144     2    1            PUT EDIT(I)(SKIP,COL(5),F(2));                            I0002080
145     2    1            DO J=1 TO 16;                                             I0002090
146     2    2               IF I<=MNT(J) THEN                                      I0002100
147     2    2               PUT EDIT(PFF(I,J))(X(3),F(3)); ELSE                    I0002110
148     2    2               PUT EDIT('       ')(A);                                I0002120
149     2    2            END;                                                      I0002130
150     2    1         END;                                                          I0002140
151     2              PUT EDIT('STEN','SCORE','FREQUENCY')(SKIP(4),COL(3),A,COL(2),A, I0002150
                       COL(50),A);                                                   I0002160
152     2              PUT SKIP;                                                     I0002170
153     2              DO I=0 TO 10;                                                 I0002180
```

Dataset Limited

```
STMT LEVEL NEST

154     2    1            PUT EDIT(I)(SKIP,COL(5),F(2));                                  I0002190
155     2    1             DO J=1 TO 16;                                                  I0002200
156     2    2               PUT EDIT(SF(I,J))(X(3),F(3));                                I0002210
157     2    2            END;                                                            I0002220
158     2    1         END;                                                               I0002230
159     2           END;                                                                   I0002240
                    /*****************************************************************/I0002250
160     1           TRANS:PROC(REQUEST,LOP,CLAB,RLAB,RITEM,K);                             I0002260
161     2                 DCL (CLAB,RLAB)(20) FIXED BINARY, LOP(20) CHAR(1),               I0002270
                          RITEM(20) CHAR(20) VARYING;                                      I0002280
162     2                 DCL REQUEST CHAR(400) VARYING;                                   I0002290
163     2                 DCL PCOND(6) CHAR(2) VARYING INITIAL('=','¬=','>','¬>','<','¬<'); I0002300
164     2                 DCL X1 CHAR(1),LR CHAR(3),(I,J,K,LK,LL,LB,NB) FIXED BINARY;      I0002310
165     2                 LL=LENGTH(REQUEST);                                              I0002320
166     2                 K=0;  LK=1;                                                      I0002330
168     2                 DO I=1 TO LL;                                                    I0002340
169     2    1               X1=SUBSTR(REQUEST,I,1);                                        I0002350
170     2    1               IF X1=' ' THEN                                                 I0002360
171     2    1               DO; LB=I;                                                      I0002370
173     2    2                   GOTO L61;                                                  I0002380
174     2    2               END; ELSE                                                      I0002390
175     2    1               IF X1='&' | X1='|' THEN                                        I0002400
176     2    1               DO; LK=LK+1;                                                   I0002410
178     2    2                   LOP(LK)=X1;                                                I0002420
179     2    2                   GOTO L61;                                                  I0002430
180     2    2               END; ELSE                                                      I0002440
181     2    1               DO J=1,3,5;                                                    I0002450
182     2    2                  IF X1=PCOND(J) THEN GOTO L62;                               I0002460
184     2    2               END;                                                           I0002470
185     2    1               GOTO L61;                                                      I0002480
186     2    1     L62:      NB=INDEX(SUBSTR(REQUEST,I),' ');                               I0002490
187     2    1               IF SUBSTR(REQUEST,I-1,1)='¬' THEN                              I0002500
188     2    1               DO; K=K+1;                                                     I0002510
190     2    2                   CLAB(K)=J+1;                                               I0002520
191     2    2                   LR=SUBSTR(REQUEST,LB+1,I-LB-2);                            I0002530
192     2    2                   RLAB(K)=LR;                                 /* CONVERSION */I0002540
193     2    2               END; ELSE                                                      I0002550
194     2    1               DO; K=K+1;                                                     I0002560
196     2    2                   CLAB(K)=J;                                                 I0002570
197     2    2                   LR=SUBSTR(REQUEST,LB+1,I-LB-1);                            I0002580
198     2    2                   RLAB(K)=LR;                                 /* CONVERSION */I0002590
199     2    2               END;                                                           I0002600
200     2    1               RITEM(K)=SUBSTR(REQUEST,I+1,NB-2);                             I0002610
201     2    1               I=I+NB-1;                                                      I0002620
202     2    1     L61: END;                                                                I0002630
203     2                 DO I=1 TO K;                                                     I0002640
204     2    1               IF RLAB(I)<1 | RLAB(I)>66 THEN                                 I0002650
205     2    1               DO; PUT EDIT('ERROR.....ITEM',RLAB(I),'NOT INCLUDED')          I0002660
                                 (SKIP,COL(1),A,X(1),F(2),X(1),A);                          I0002670
207     2    2                   DO J=I TO K-1;                                             I0002680
208     2    3                      LOP(J)=LOP(J+1);                                        I0002690
209     2    3                      RLAB(J)=RLAB(J+1);                                      I0002700
210     2    3                      CLAB(J)=CLAB(J+1);                                      I0002710
211     2    3                      RITEM(J)=RITEM(J+1);                                    I0002720
```

Dataset Limited

```
STMT LEVEL NEST

212     2    3                K=K-1;                                          I0002730
213     2    3              END;                                              I0002740
214     2    2           END;                                                 I0002750
215     2    1        END;                                                    I0002760
216     2          END;                                                       I0002770
                   /*****************************************************/I0002780
217     1          LOCTST:PROC(YR,LOP,CLAB,RLAB,RITEM,K) RETURNS(BIT(1));   I0002790
218     2               DCL LOP(20) CHAR(1),(CLAB,RLAB)(20) FIXED BINARY, RITEM(20)  I0002800
                        CHAR(20) VARYING;                                   I0002810
219     2               DCL REQI CHAR(12),REQM CHAR(20) VARYING;            I0002820
220     2               DCL ITEM CHAR(20) VARYING, YR FIXED BINARY;         I0002830
221     2               DCL (K,I,J,QS) FIXED BINARY;                        I0002840
                        /*****************************************************/I0002850
222     2          FIND:PROC(YR,ITEM,LR,REQI);                              I0002860
223     3               DCL (ITEM CHAR(20), XVAR CHAR(3)) VARYING, (KC,LR,I,J,YR) FIXED   I0002870
                        BINARY, X2 CHAR(2), R(66) LABEL, REQI CHAR(12), ST CHAR(1) DEFINEDI0002880
                        REQI POS(2), SYF CHAR(2) DEFINED REQI POS(3), SYT CHAR(2) DEFINED I0002890
                        REQI POS(5), QY CHAR(2) DEFINED REQI POS(2), QE CHAR(3) DEFINED   I0002900
                        REQI POS(4), QC CHAR(3) DEFINED REQI POS(7), QG CHAR(1) DEFINED   I0002910
                        REQI POS(10), QR CHAR(1) DEFINED REQI POS(11);      I0002920
224     3               IF QE¬='***' THEN QE=TRANSLATE(QE,' ','*');         I0002930
226     3               IF LR>=26 & LR<=35 THEN I=LE;              /* CONVERSION */I0002940
228     3               GOTO R(LR);                                         I0002950
229     3               R(1):X2=SUBSTR(TTFT(1,1),5,2); I=X2;       /* CONVERSION */I0002960
231     3               X2=SUBSTR(DOB,5,2); J=X2;                  /* CONVERSION */I0002970
233     3               KC=I-J;                                             I0002980
234     3               GOTO L1;                                            I0002990
235     3               R(2):ITEM=SEX; RETURN;  R(3):ITEM=MART; RETURN;     I0003000
239     3               R(4):ITEM=SET; RETURN;  R(5):ITEM=REG; RETURN;      I0003010
243     3               R(6):ITEM=POCC; RETURN; R(7):ITEM=SOCC; RETURN;     I0003020
247     3               R(8):ITEM=OV25; RETURN; R(9):ITEM=REL; RETURN;      I0003030
251     3               R(10):ITEM=CONF; RETURN;  R(11):ITEM=AUTH; RETURN;  I0003040
255     3               R(12):ITEM=AUTHNO; RETURN;  R(13):ITEM=FPOS; RETURN; I0003050
259     3               R(14):ITEM=TTFT(1,1); RETURN;  R(15):ITEM=UNCOL(1); RETURN;  I0003060
263     3               R(16):ITEM=UNOTH(1); RETURN;  R(17):ITEM=DEG(1); RETURN;  I0003070
267     3               R(18):ITEM=SCHNO; RETURN;                           I0003080
269     3               R(19):KC=0;  QS=SCHNO;                     /* CONVERSION */I0003090
271     3               DO I=1 TO QS;                                       I0003100
272     3    1             IF ST='*' | SUBSTR(SCH(I),1,1)=ST THEN          I0003110
273     3    1             IF SYF='**' | SUBSTR(SCH(I),2,2)=SYF THEN       I0003120
274     3    1             IF SYT='**' | SUBSTR(SCH(I),4,2)=SYT THEN KC=KC+1;  I0003130
276     3    1          END;                                               I0003140
277     3               GOTO L1;                                            I0003150
278     3               R(20):ITEM=QUALNO; RETURN;                          I0003160
280     3               R(21):KC=0; QS=QUALNO;                     /* CONVERSION */I0003170
282     3               DO I=1 TO QS;                                       I0003180
283     3    1             IF QY='**' | SUBSTR(QUAL(I),1,2)=QY THEN        I0003190
284     3    1             IF QE='***' | SUBSTR(QUAL(I),3,3)=QE THEN       I0003200
285     3    1             IF QC='***' | SUBSTR(QUAL(I),6,3)=QC THEN       I0003210
286     3    1             IF QG='*' | SUBSTR(QUAL(I),9,1)=QG THEN         I0003220
287     3    1             IF QR='*' | SUBSTR(QUAL(I),10,1)=QR THEN KC=KC+1;  I0003230
289     3    1          END;                                               I0003240
290     3               GOTO L1;                                            I0003250
291     3               R(22):ITEM=ATO; RETURN;  R(23):ITEM=LEN; RETURN;    I0003260
```

Data Facet Limited

```
STMT LEVEL NEST

295     3              R(24):I=ST;                                          /* CONVERSION */I0003270
296     3              ITEM=CODES(I); RETURN;  R(25):ITEM=DC46(1); RETURN;                 I0003280
300     3              R(26):ITEM=DC(I,25); RETURN; R(27):ITEM=DC(I,34); RETURN;           I0003290
304     3              R(28):ITEM=DC(I,35); RETURN; R(29):ITEM=DC(I,36); RETURN;           I0003300
308     3              R(30):ITEM=DC(I,37); RETURN; R(31):ITEM=DC(I,38); RETURN;           I0003310
312     3              R(32):ITEM=DC(I,41); RETURN; R(33):ITEM=DC(I,42); RETURN;           I0003320
316     3              R(34):ITEM=DC(I,59); RETURN; R(35):ITEM=DC(I,64); RETURN;           I0003330
320     3              R(36):XVAR=SUBSTR(REQI,5,INDEX(REQI,')')-5);                        I0003340
321     3              I=XVAR;                                              /* CONVERSION */I0003350
322     3              IF FK(I,1)¬=0 THEN ITEM=SUBSTR(AP,FK(I,1),FK(I,2)-1); ELSE          I0003360
324     3              ITEM=' '; RETURN;                                                   I0003370
326     3              R(37):ITEM=TC; RETURN;                                              I0003380
328     3              R(38):ITEM=WS; RETURN;                                              I0003390
330     3              R(39):ITEM=BE; RETURN;                                              I0003400
332     3              R(40):ITEM=SUBCAR; RETURN;                                          I0003410
334     3              R(41):R(42):R(43):R(44):R(45):R(46):KC=VALUES(YR,LR-40); GOTO L1;   I0003420
336     3              R(47):KC=DSCALE(YR); GOTO L1;                                       I0003430
338     3              R(48):R(49):R(50):KC=AH4(YR,LR-47); GOTO L1;                        I0003440
340     3              R(51):R(52):R(53):R(54):R(55):R(56):KC=PF16(YR,LR-50); GOTO L1;     I0003450
342     3              R(57):R(58):R(59):R(60):R(61):R(62):KC=PF16(YR,LR-50); GOTO L1;     I0003460
344     3              R(63):R(64):R(65):R(66):KC=PF16(YR,LR-50);                          I0003470
345     3         L1:  ITEM=CHAR(KC);                                                      I0003480
346     3         END;                                                                     I0003490
                       /*****************************************************************/I0003500
347     2         TEST:PROC(ITEM,COND,REQU) RETURNS(BIT(1));                               I0003510
348     3              DCL (ITEM,REQU,X20) CHAR(20) VARYING, COND FIXED BINARY;            I0003520
349     3              DCL C(6) LABEL INITIAL(C1,C2,C3,C4,C5,C6);                          I0003530
350     3              X20=ITEM;  ITEM=((20)' ');  SUBSTR(ITEM,21-LENGTH(X20))=X20;        I0003540
353     3              X20=REQU;  REQU=((20)' ');  SUBSTR(REQU,21-LENGTH(X20))=X20;        I0003550
356     3              GOTO C(COND);                                                       I0003560
357     3         C1:  IF ITEM=REQU THEN RETURN('1'B); ELSE RETURN('0'B);                  I0003570
360     3         C2:  IF ITEM¬=REQU THEN RETURN('1'B); ELSE RETURN('0'B);                 I0003580
363     3         C3:  IF ITEM>REQU THEN RETURN('1'B); ELSE RETURN('0'B);                  I0003590
366     3         C4:  IF ITEM¬>REQU THEN RETURN('1'B); ELSE RETURN('0'B);                 I0003600
369     3         C5:  IF ITEM<REQU THEN RETURN('1'B); ELSE RETURN('0'B);                  I0003610
372     3         C6:  IF ITEM¬<REQU THEN RETURN('1'B); ELSE RETURN('0'B);                 I0003620
375     3         END;                                                                     I0003630
376     2         DCL TEST ENTRY(,,) RETURNS(BIT(1));                                      I0003640
                       /*****************************************************************/I0003650
377     2              DO I=1 TO K;                                                        I0003660
378     2    1            IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19    I0003670
                          | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN         I0003680
379     2    1            REQI=SUBSTR(RITEM(I),INDEX(RITEM(I),'('));                        I0003690
380     2    1            IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=36    I0003700
381     2    1            | RLAB(I)=40 THEN IF SUBSTR(REQI,2,2)¬=SUBSTR(TTFT(1,1),5,2)     I0003710
382     2    1            THEN IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);I0003720
385     2    1            CALL FIND(YR,ITEM,RLAB(I),REQI);                                  I0003730
386     2    1            IF RLAB(I)=4 | RLAB(I)=6 | RLAB(I)=7 | RLAB(I)=9 | RLAB(I)=19    I0003740
                          | RLAB(I)=21 | RLAB(I)=24 | RLAB(I)=36 | RLAB(I)=40 THEN         I0003750
387     2    1            REQM=SUBSTR(RITEM(I),1,INDEX(RITEM(I),'(')-1); ELSE               I0003760
388     2    1            REQM=TRANSLATE(RITEM(I),' ',':');                                 I0003770
389     2    1            IF TEST(ITEM,CLAB(I),REQM) THEN                                   I0003780
390     2    1            DO; DO J=I+1 TO K;                                                I0003790
392     2    3                 IF LOP(J)¬='|' THEN                                          I0003800
```

Dataset Limited

```
STMT LEVEL NEST

393     2    3                  DO; I=J-1;                                          I0003810
395     2    4                      GOTO L66;                                       I0003820
396     2    4                  END;                                                I0003830
397     2    3                END;                                                  I0003840
398     2    2                RETURN('1'B);                                         I0003850
399     2    2            END; ELSE                                                 I0003860
400     2    1            IF I+1<=K & LOP(I+1)='|' THEN GOTO L66; ELSE RETURN('0'B);  I0003870
403     2    1       L66: END;                                                      I0003880
404     2            END;                                                           I0003890
405     1            DCL LOCTST ENTRY(,,,,,) RETURNS(BIT(1));                       I0003900
                     /*****************************************************************/I0003910
406     1                 ON ENDFILE(SCARDS) GOTO L09;                              I0003920
408     1            L39: GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));  I0003930
409     1                 GET LIST(OPT);                                            I0003940
410     1                 IF OPT<1 | OPT>4 THEN                                     I0003950
411     1                 DO; PUT EDIT('OPTION INVALID.....PROGRAM TERMINATED')     I0003960
                              (SKIP,COL(1),A);                                      I0003970
413     1    1                GOTO L09;                                             I0003980
414     1    1            END;                                                      I0003990
415     1                 WRL='XXX'; WCR='X'; YT='F'; PI='0'; REQUEST=' '; B='0'B;  I0004000
421     1                 GET DATA;                                                 I0004010
422     1                 IF SUBSTR(WRL,1,1)='1' THEN WB='1'B; ELSE WB='0'B;        I0004020
425     1                 IF SUBSTR(WRL,1,1)='2' THEN WPB='1'B; ELSE WPB='0'B;      I0004030
428     1                 IF SUBSTR(WRL,2,1)='1' THEN RB='1'B; ELSE RB='0'B;        I0004040
431     1                 IF SUBSTR(WRL,2,1)='2' THEN RPB='1'B; ELSE RPB='0'B;      I0004050
434     1                 IF SUBSTR(WRL,3,1)='1' THEN LB='1'B; ELSE LB='0'B;        I0004060
437     1                 IF SUBSTR(WRL,3,1)='2' THEN LPB='1'B; ELSE LPB='0'B;      I0004070
440     1                 IF YT='F' THEN YR=1; ELSE IF YT='L' THEN YR=2; ELSE YR=1; I0004080
445     1                 PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME I',  I0004090
                          'OPTION REQUESTED',OPT,'YEAR REQUESTED',YE)(PAGE,COL(1),A,  I0004100
                          SKIP(2),COL(1),A,COL(1),A,COL(32),F(2),COL(1),A,COL(30),A);  I0004110
446     1                 PUT SKIP(2);                                              I0004120
447     1                 IF WRL¬='XXX' THEN PUT EDIT('WRL REQUESTED',WRL)          I0004130
449     1                 (COL(1),A,COL(31),A); ELSE WRL='000';                     I0004140
450     1                 IF WCR¬='X' THEN PUT EDIT('WCR REQUESTED',WCR)(COL(1),A,COL(33),  I0004150
452     1                 A); ELSE WCR='*';                                         I0004160
453     1                 PUT EDIT('YEAR OF TEST',YT)(COL(1),A,COL(33),A);          I0004170
454     1                 IF PI¬='0' THEN PUT EDIT('PRINT INDICATOR',PI)            I0004180
                          (COL(1),A,COL(33),A);                                     I0004190
456     1                 PUT SKIP(4);                                              I0004200
457     1                 IF REQUEST='*' THEN GOTO L00;                             I0004210
                     /*-----------------------------------------------------------------*/I0004220
459     1            L1234:OPEN FILE(STUDENT) INPUT TITLE(FILN1);                   I0004230
460     1                 ON ENDFILE(STUDENT)                                       I0004240
461     1                 BEGIN; DT=DATE;                                           I0004250
463     2                        DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||  I0004260
                                 SUBSTR(DT,7,2);                                    I0004270
464     2                        IF OPT=1 | OPT=2 THEN                              I0004280
465     2                        DO; IF OPT=1 THEN PUT EDIT(YE,'FILE........INTELLIGENCE ',  I0004290
                                     'TEST TOTALS (MEANS) ST. DEV')(SKIP(2),A(4),X(1),A,A);  I0004300
468     2    1                       IF OPT=2 THEN PUT EDIT(YE,'FILE........INTELLIGENCE ',  I0004310
                                     'TEST FREQUENCY COUNTS')(SKIP(2),A(4),X(1),A,A);  I0004320
470     2    1                       PUT EDIT(KT,'STUDENTS INCLUDED')               I0004330
                                     (SKIP(3),COL(1),F(4),X(1),A);                  I0004340
```

Dataset Limited

```
STMT LEVEL NEST

471     2     1                    IF KT=0 THEN GOTO L21;                                  I0004350
473     2     1                    IF ¬SUPP THEN CALL SPNT;                                I0004360
475     2     1                    PUT PAGE;                                               I0004370
476     2     1                    IF OPT=1 THEN CALL ITPTMS; ELSE CALL ITPFREQ;           I0004380
479     2     1     L21:       END; ELSE                                                   I0004390
480     2                      IF OPT=3 THEN                                               I0004400
481     2                      DO; IF PI¬='1' THEN                                         I0004410
483     2     1                    DO; PUT EDIT(YE,'FILE.....INTELLIGENCE TESTS',KT,      I0004420
                                       'STUDENTS INCLUDED')(PAGE,COL(1),A,X(1),A,X(1),    I0004430
                                       F(4),X(1),A);                                      I0004440
485     2     2                        IF ¬SUPP THEN CALL SPNT;                           I0004450
487     2     2                    END;                                                    I0004460
488     2     1              END; ELSE                                                     I0004470
489     2                    IF OPT=4 THEN                                                 I0004480
490     2                    DO; PUT EDIT('TOTALS,MEANS,STANDARD DEVIATIONS AND ',         I0004490
                                 'FREQUENCY COUNTS FOR REQUEST',REQUEST)                   I0004500
                                 (SKIP(2),COL(1),A,A,X(2),A);                              I0004510
492     2     1                    PUT EDIT(KT,'STUDENTS INCLUDED')                        I0004520
                                   (SKIP(3),COL(1),F(4),X(1),A);                           I0004530
493     2     1                    IF KT=0 THEN GOTO L51;                                  I0004540
495     2     1                    IF ¬SUPP THEN CALL SPNT;                                I0004550
497     2     1                    PUT PAGE;                                               I0004560
498     2     1                    CALL ITPTMS;                                            I0004570
499     2     1                    PUT PAGE;                                               I0004580
500     2     1                    CALL ITPFREQ;                                           I0004590
501     2     1     L51:           END;                                                    I0004600
502     2                          PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A); I0004610
503     2                          CLOSE FILE(STUDENT);                                    I0004620
504     2                          GOTO L39;                                               I0004630
505     2                END;                                                              I0004640
506     1                KT=0; DESL=' '; SUPP='0'B;                                        I0004650
509     1                IF OPT=1 THEN                                                     I0004660
510     1                DO; VT=0; VST=0; DST=0; DSST=0; AHT=0; AHST=0; PFT=0; PFST=0;     I0004670
519     1     1              SST=0; SSST=0;                                              I0004680
521     1     1          END; ELSE                                                         I0004690
522     1                IF OPT=2 THEN                                                     I0004700
523     1                DO; VF=0; DSF=0; AHF=0;                                           I0004710
527     1     1              PFF=0; SF=0;                                                I0004720
529     1     1          END; ELSE                                                         I0004730
530     1                IF OPT=4 THEN                                                     I0004740
531     1                DO; LOP=' '; CLAB=0; RLAB=0; RITEM=' '; K=0;                      I0004750
537     1     1              IF SUBSTR(REQUEST,2,1)¬=' ' THEN                            I0004760
538     1     1              CALL TRANS(REQUEST,LOP,CLAB,RLAB,RITEM,K);                  I0004770
539     1     1              IF K=0 THEN                                                 I0004780
540     1     1              DO; CLOSE FILE(STUDENT);                                    I0004790
542     1     2                  GOTO L39;                                               I0004800
543     1     2              END;                                                        I0004810
544     1     1              VT,VST=0; DST,DSST=0; AHT,AHST=0; PFT,PFST=0; SST,SSST=0;   I0004820
549     1     1              VF=0; DSF=0; AHF=0; PFF=0; SF=0;                            I0004830
554     1     1          END;                                                              I0004840
555     1          L31: READ FILE(STUDENT) SET(PTR);                                      I0004850
556     1               IF SUBSTR(INTELB,YR,1)¬='*' THEN                                  I0004860
557     1               DO; IF OPT¬=3 THEN PUT EDIT(DESNO,'HAS NO CORRESPONDING DATA')    I0004870
                            (COL(1),A,X(1),A);                                            I0004880
```

Dataset Limited

```
STMT LEVEL NEST

560     1    1            GOTO L31;                                                  IC004890
561     1    1        END;                                                           IO004900
562     1             IF ¬WB & SUBSTR(TTFT(2,1),1,4)='WITH' THEN                     IO004910
563     1             DO; IF WPB THEN                                                IO004920
565     1    1            DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),   IO004930
                              3,2) || '/' || SUBSTR(TTFT(1,2),5,2);                  IO004940
567     1    2                PUT EDIT(DESNO,'WITHDREW ON',WD)(COL(1),A,2(X(1),A));  IO004950
568     1    2            END;                                                       IO004960
569     1    1            GOTO L31;                                                  IO004970
570     1    1        END; ELSE                                                      IO004980
571     1             IF ¬LB & SUBSTR(TTFT(2,1),1,4)='LEFT' THEN                     IO004990
572     1             DO; IF LPB THEN                                                IO005000
574     1    1            DO; WD=SUBSTR(TTFT(1,2),1,2) || '/' || SUBSTR(TTFT(1,2),   IO005010
                              3,2) || '/' || SUBSTR(TTFT(1,2),5,2);                  IO005020
576     1    2                PUT EDIT(DESNO,'LEFT ON',WD)(COL(1),A,2(X(1),A));      IO005030
577     1    2            END;                                                       IO005040
578     1    1            GOTO L31;                                                  IO005050
579     1    1        END; ELSE                                                      IO005060
580     1             IF ¬RB & SUBSTR(TTFT(2,1),1,4)='    ' THEN                     IO005070
581     1             DO; IF RPB THEN PUT EDIT(DESNO,'AT COLLEGE')(COL(1),A,X(1),A); IO005080
584     1    1            GOTO L31;                                                  IO005090
585     1    1        END; ELSE                                                      IO005100
586     1             IF WB & WCR¬='*' & WCR¬=SUBSTR(TTFT(2,1),6,1) THEN GOTO L31;   IO005110
588     1             IF OPT=1 | OPT=2 | OPT=3 |                                     IO005120
                      (OPT=4 & LOCTST(YR,LOP,CLAB,RLAB,RITEM,K)) THEN                IO005130
589     1             DO; KT=KT+1;                                                   IO005140
591     1    1            IF KT>1000 THEN SUPP='1'B;                                 IO005150
593     1    1            IF ¬SUPP THEN DESL(KT)=DESNO;                              IO005160
595     1    1            IF OPT=1 | OPT=4 THEN                                      IO005170
596     1    1            CALL ITTMS;                                                IO005180
597     1    1            IF OPT=2 | OPT=4 THEN                                      IO005190
598     1    1            CALL ITFREQ;                                               IO005200
599     1    1            IF OPT=3 & PI='1' THEN CALL PRINT;                         IO005210
601     1    1        END;                                                           IO005220
602     1             GOTO L31;                                                      IO005230
                   /*-------------------------------------------------------------*/IO005240
603     1          L00: OPEN FILE(STUDENT) INPUT TITLE(FILN1);                       IO005250
604     1             ON ENDFILE(STUDENT)                                            IO005260
605     1             BEGIN; CLOSE FILE(STUDENT);                                    IO005270
607     2                    OPEN FILE(STUDENT) INPUT TITLE(FILN1);                  IO005280
608     2                    GOTO L01;                                               IO005290
609     2             END;                                                           IO005300
610     1             VT=0; VST=0; DST=0; DSST=0; AHT=0; AHST=0; PFT=0; PFST=0;      IO005310
618     1             VF=0; DSF=0; AHF=0; PFF=0; SF=0; SST=0; SSST=0;                IO005320
625     1             DESL=' '; KT=0; SUPP='0'B;                                     IO005330
628     1          L02: GET EDIT(DESNO2)(COL(1),A(8));                               IO005340
629     1             IF DESNO2='********' THEN                                      IO005350
630     1             DO; PUT EDIT('TOTALS,MEANS,STANDARD DEVIATIONS AND FREQUENCY', IO005360
                          ' COUNTS FOR REQUEST=''*''')(SKIP(2),COL(1),A,A);          IO005370
632     1    1            PUT EDIT(KT,'STUDENTS INCLUDED')(SKIP(3),COL(1),F(4),X(1),A); IO005380
633     1    1            IF KT=0 THEN GOTO L03;                                     IO005390
635     1    1            IF ¬SUPP THEN CALL SPNT;                                   IO005400
637     1    1            PUT PAGE;                                                  IO005410
638     1    1            CALL ITPTMS;                                               IO005420
```

Dataset Limited

```
STMT LEVEL NEST

639     1    1              PUT PAGE;                                          I0005430
640     1    1              CALL ITPFREQ;                                      I0005440
641     1    1    L03:      DT=DATE;                                           I0005450
642     1    1              DT=SUBSTR(DT,4,2) || '/' || SUBSTR(DT,1,2) || '/' ||  I0005460
                            SUBSTR(DT,7,2);                                    I0005470
643     1    1              PUT EDIT('DATE OF LISTING',DT)(SKIP(2),COL(1),A,X(1),A);  I0005480
644     1    1              CLOSE FILE(STUDENT);                               I0005490
645     1    1              GOTO L39;                                          I0005500
646     1    1         END;                                                    I0005510
647     1         L01: READ FILE(STUDENT) SET(PTR);                           I0005520
648     1              IF DESNO=DESNO2 THEN                                   I0005530
649     1              DO; IF B THEN B='0'B;                                  I0005540
652     1    1              IF SUBSTR(INTELB,YR,1)¬='*' THEN                   I0005550
653     1    1              DO; PUT EDIT(DESNO,'HAS NO CORRESPONDING DATA')    I0005560
                                (COL(1),A,X(1),A);                             I0005570
655     1    2                  GOTO L02;                                      I0005580
656     1    2              END;                                               I0005590
657     1    1              KT=KT+1;                                           I0005600
658     1    1              IF KT>1000 THEN SUPP='1'B;                         I0005610
660     1    1              IF ¬SUPP THEN DESL(KT)=DESNO;                      I0005620
662     1    1              CALL ITTMS;                                        I0005630
663     1    1              CALL ITFREQ;                                       I0005640
664     1    1              GOTO L02;                                          I0005650
665     1    1         END; ELSE                                               I0005660
666     1              DO; IF ¬B THEN                                         I0005670
668     1    1              DO; DESNO3=DESNO;                                  I0005680
670     1    2                  B='1'B;                                        I0005690
671     1    2              END; ELSE                                          I0005700
672     1    1              DO; IF DESNO=DESNO3 THEN                           I0005710
674     1    2                  DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')  I0005720
                                    (COL(1),A,X(1),A);                         I0005730
676     1    3                      B='0'B;                                    I0005740
677     1    3                      GOTO L02;                                  I0005750
678     1    3                  END;                                           I0005760
679     1    2              END;                                               I0005770
680     1    1              GOTO L01;                                          I0005780
681     1    1         END;                                                    I0005790
                    /*-----------------------------------------------------------*/I0005800
682     1         L09: PUT PAGE;                                              I0005810
683     1         END;                                                        I0005820
```

Dataset Limited

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
2	ADIP	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER
2	ADIPC	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
2 *********	AH4	(2,3)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0) 26,26,28,29,55,55,56,56,84,338
3 *********	AHF	(0:130,3)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 56,56,138,138,139,139,526,551,620
4	AHST	(2)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE) 29,29,92,92,106,516,546,615
4	AHT	(2)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE) 28,28,92,106,106,515,546,614
2	ANNPROG	IN RECORD,BASED(PTR),STRUCTURE
2	AP	IN ANNPROG IN RECORD,BASED(PTR),UNALIGNED,STRING(802),CHARACTER 323
2	ATO	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 291
2	AUTH	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(19),CHARACTER 253
2	AUTHNO	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 255
7	B	AUTOMATIC,UNALIGNED,STRING(1),BIT 420,650,651,667,670,676
2	BE	IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 330
349	C	(6)AUTOMATIC,INITIAL,LABEL 356
357	C1	STATEMENT LABEL CONSTANT
360	C2	STATEMENT LABEL CONSTANT
363	C3	STATEMENT LABEL CONSTANT
366	C4	STATEMENT LABEL CONSTANT
369	C5	STATEMENT LABEL CONSTANT
372	C6	STATEMENT LABEL CONSTANT

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                 CHAR                          GENERIC,BUILT-IN FUNCTION
                                               345

218    ********* CLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               217,389

161    ********* CLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               160,190,196,210,210

5      ********* CLAB                          (20)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               533,538,588

2                CODES                         (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                               296

348    ********* COND                          PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               347,356

2                CONF                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               251

2                COURSE                        IN RECORD,BASED(PTR),STRUCTURE

                 DATE                          BUILT-IN FUNCTION
                                               462,641

2                DATT                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER

2                DC                            (4,24:64)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),
                                               CHARACTER
                                               300,302,304,306,308,310,312,314,316,318

2                DC1                           (4,26:30)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(2),
                                               CHARACTER

2                DC46                          (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                               298

2                DC49                          (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER

2                DDIPC                         IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

2                DEG                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                               265

7                DESL                          (1000)AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               74,75,507,594,625,661

2                DESNO                         IN RECORD,BASED(PTR),UNALIGNED,STRING(8),CHARACTER
                                               17,24,31,41,49,53,57,63,84,559,567,576,583,594,648,654,661,669,673

7                DESNO2                        AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               628,629,648,675
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

7                DESNO3                        AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               669,673

2                DOB                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               231

2                DOBC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

2      ********* DSCALE                        (2)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               19,19,21,22,51,51,52,52,84,336

3      ********* DSF                           (0:300)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               52,52,132,133,525,550,619

4                DSST                          AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)
                                               22,22,91,91,102,514,545,613

4                DST                           AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)
                                               21,21,91,102,102,513,545,612

7                DT                            AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               462,463,463,463,463,502,641,642,642,642,642,643

2                EDUCATION                     IN RECORD,BASED(PTR),STRUCTURE

7                FILN1                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               408,459,603,607

7                FILN2                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                               408

2                FINALS                        IN RECORD,BASED(PTR),STRUCTURE

222              FIND                          ENTRY,DECIMAL,FLOAT(SINGLE)
                                               385

2      ********* FK                            (100,2)IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED
                                               (15,0)
                                               322,323,323

2                FPOS                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER
                                               257

2                FPT                           (1)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER

2                FT                            (2,2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER

2                GENERAL                       IN RECORD,BASED(PTR),STRUCTURE

2                HPCD                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER

223    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               227,230,233,271,272,273,274,282,283,284,285,286,287,295,296,300,302
                                               304,306,308,310,312,314,316,318,321,322,323,323
```

Data East Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

221    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               377,378,378,378,378,378,378,378,378,378,379,379,380,380,380,380,380
                                               380,382,382,385,386,386,386,386,386,386,386,386,386,387,387,388,389
                                               391,394,400,400

164    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               168,169,172,186,187,191,197,200,201,201,203,204,204,206,207

117    ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               120,122,124,124,131,132,133,133,137,138,138,139,139,139,143,144,146
                                               147,153,154,156

70     ********* I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               72,73,74,75

1      ********* I                             ENTRY,BINARY,FIXED(15,0)

                 INDEX                         GENERIC,BUILT-IN FUNCTION
                                               186,320,379,387

2                INTEL                         IN RECORD,BASED(PTR),STRUCTURE

2                INTELB                        IN INTEL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               556,652

348              ITEM                          PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               347,350,351,352,357,360,363,366,369,372

223              ITEM                          PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               222,235,237,239,241,243,245,247,249,251,253,255,257,259,261,263,265
                                               267,278,291,293,296,298,300,302,304,306,308,310,312,314,316,318,323
                                               324,326,328,330,332,345

220              ITEM                          AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               385,389

44     ********* ITFREQ                        ENTRY,BINARY,FIXED(15,0)
                                               598,663

116    ********* ITPFREQ                       ENTRY,BINARY,FIXED(15,0)
                                               478,500,640

87     ********* ITPTMS                        ENTRY,BINARY,FIXED(15,0)
                                               477,498,638

9      ********* ITTMS                         ENTRY,BINARY,FIXED(15,0)
                                               596,662

223    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               232,233

221    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               391,392,394
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

164    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               181,182,190,196,207,208,208,209,209,210,210,211,211

117    ********* J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               121,122,145,146,147,155,156

221    ********* K                             PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               217,377,382,391,400

164    ********* K                             PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               160,166,189,189,190,192,195,195,196,198,200,203,207,212,212

117    ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               124,124

88     ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               97,98,99,99,99,104,105,106,106,106,110,111,112,112,112,113,113,113

45     ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               46,47,47,48,48,48,48,49,54,55,55,56,56,56,56,57,59,60,60,60,61,61,61
                                               61,63,66,66,66,66

10     ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               11,12,12,14,14,14,15,15,15,17,25,26,26,28,28,28,29,29,29,31,33,34,34
                                               34,36,36,36,37,37,37,38,38,38,39,39,39,41

7      ********* K                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               536,538,539,588

223    ********* KC                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               233,269,275,275,280,288,288,334,336,338,340,342,344,345

7      ********* KT                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               72,79,90,90,91,91,92,92,93,93,94,94,99,102,106,112,113,470,471,484
                                               492,493,506,590,590,591,594,626,632,633,657,657,658,661

603              L00                           STATEMENT LABEL CONSTANT
                                               458

647              L01                           STATEMENT LABEL CONSTANT
                                               608,680

628              L02                           STATEMENT LABEL CONSTANT
                                               655,664,677

641              L03                           STATEMENT LABEL CONSTANT
                                               634

682              L09                           STATEMENT LABEL CONSTANT
                                               407,413

128              L1                            STATEMENT LABEL CONSTANT
                                               125
```

Dataset Limited

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
345	L1	STATEMENT LABEL CONSTANT 234,277,290,335,337,339,341,343
459	L1234	STATEMENT LABEL CONSTANT
479	L21	STATEMENT LABEL CONSTANT 472
555	L31	STATEMENT LABEL CONSTANT 560,569,578,584,587,602
408	L39	STATEMENT LABEL CONSTANT 504,542,645
501	L51	STATEMENT LABEL CONSTANT 494
202	L61	STATEMENT LABEL CONSTANT 173,179,185
186	L62	STATEMENT LABEL CONSTANT 183
403	L66	STATEMENT LABEL CONSTANT 383,395,401
164	********* LB	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 172,191,191,197,197
7	LB	AUTOMATIC,UNALIGNED,STRING(1),BIT 435,436,571
2	LE	IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 227
2	********* LEAP	IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
2	LEN	IN CCURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 293
	LENGTH	GENERIC,BUILT-IN FUNCTION 165,352,355
164	********* LK	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 167,177,177,178
164	********* LL	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 165,168
217	LOCTST	ENTRY,STRING(1),BIT 588
218	LOP	(20)PARAMETER,UNALIGNED,STRING(1),CHARACTER

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               217,382,392,400

161              LOP                           (20)PARAMETER,UNALIGNED,STRING(1),CHARACTER
                                               160,178,208,208

5                LOP                           (20)AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               532,538,588

7                LPB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               438,439,573

223    ********* LR                            PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               222,226,226,228,334,338,340,342,344

164              LR                            AUTOMATIC,UNALIGNED,STRING(3),CHARACTER
                                               191,192,197,198

2                MART                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               237

6      ********* MNT                           (16)AUTOMATIC,ALIGNED,INITIAL,BINARY,FIXED(15,0)
                                               34,60,146

                 MOD                           GENERIC,BUILT-IN FUNCTION
                                               73,79

2                NAME                          IN RECORD,BASED(PTR),UNALIGNED,STRING(40),CHARACTER

164    ********* NB                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               186,200,201

7      ********* OPT                           AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               409,410,410,445,464,464,466,468,476,480,489,509,522,530,558,588,588
                                               588,588,595,595,597,597,599

2                OQ                            (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER

2                OQC                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER

2                OV25                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               247

163              PCOND                         (6)AUTOMATIC,UNALIGNED,INITIAL,STRING(2),CHARACTER,VARYING
                                               182

2      ********* PF16                          (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               34,34,36,37,60,60,61,61,84,340,342,344

3      ********* PFF                           (0:26,16)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               61,61,147,527,552,621

89               PFK                           (16)AUTOMATIC,UNALIGNED,INITIAL,STRING(2),CHARACTER
                                               111
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

4                PFST                          (16)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)
                                               37,37,93,93,112,518,547,617

4                PFT                           (16)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)
                                               36,36,93,112,112,517,547,616

7                PI                            AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                               418,454,455,482,599

2                PNAME                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(17),CHARACTER

2                POCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               243

78               PRINT                         ENTRY,DECIMAL,FLOAT(SINGLE)
                                               600

                 PTR                           AUTOMATIC,ALIGNED,POINTER
                                               12,12,14,15,17,19,19,21,22,24,26,26,28,29,31,34,34,36,37,38,39,41,47
                                               47,48,48,49,51,51,52,52,53,55,55,56,56,57,60,60,61,61,63,66,66,84,84
                                               84,84,84,85,227,229,231,235,237,239,241,243,245,247,249,251,253,255
                                               257,259,261,263,265,267,270,272,273,274,278,281,283,284,285,286,287
                                               291,293,296,298,300,302,304,306,308,310,312,314,316,318,322,323,323
                                               323,326,328,330,332,334,336,338,340,342,344,381,555,556,559,562,566
                                               566,566,567,571,575,575,575,576,580,583,586,594,647,648,652,654,661
                                               669,673

223              QC                            AUTOMATIC,DEFINED,UNALIGNED,STRING(3),CHARACTER
                                               285,285

223              QE                            AUTOMATIC,DEFINED,UNALIGNED,STRING(3),CHARACTER
                                               224,225,225,284,284

223              QG                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               286,286

223              QR                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               287,287

221    ********* QS                            AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               270,271,281,282

2                QUAL                          (30)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(10),CHARACTER
                                               283,284,285,286,287

2                QUALNO                        IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               278,281

223              QY                            AUTOMATIC,DEFINED,UNALIGNED,STRING(2),CHARACTER
                                               283,283

223              R                             (66)AUTOMATIC,INITIAL,LABEL
                                               229,235,237,239,241,243,245,247,249,251,253,255,257,259,261,263,265
                                               267,269,278,280,291,293,295,298,300,302,304,306,308,310,312,314,316
```

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               318,320,326,328,330,332,334,334,334,334,334,334,336,338,338,338,340
                                               340,340,340,340,340,342,342,342,342,342,342,344,344,344,344,228

7                RB                            AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               429,430,580

2                RECORD                        BASED(PTR),STRUCTURE

2                REG                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               241

2                REL                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               249

223              REQI                          PARAMETER,UNALIGNED,STRING(12),CHARACTER
                                               222,320,320

219              REQI                          AUTOMATIC,UNALIGNED,STRING(12),CHARACTER
                                               379,381,385

219              REQM                          AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               387,388,389

348              REQU                          PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               347,353,354,355,357,360,363,366,369,372

162              REQUEST                       PARAMETER,UNALIGNED,STRING(400),CHARACTER,VARYING
                                               160,165,169,186,187,191,197,200

7                REQUEST                       AUTOMATIC,UNALIGNED,STRING(400),CHARACTER,VARYING
                                               419,457,491,537,538

218              RITEM                         (20)PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               217,379,379,387,387,388

161              RITEM                         (20)PARAMETER,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               160,200,211,211

5                RITEM                         (20)AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING
                                               535,538,588

218    ********* RLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               217,378,378,378,378,378,378,378,378,378,380,380,380,380,380,380,385
                                               386,386,386,386,386,386,386,386,386

161    ********* RLAB                          (20)PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                               160,192,198,204,204,206,209,209

5      ********* RLAB                          (20)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               534,538,588

7                RPB                           AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               432,433,582
```

Release Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                 SCARDS                        FILE,EXTERNAL
                                               406,408,409,421,628

2                SCH                           (10)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(5),CHARACTER
                                               272,273,274

2                SCHNO                         IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
                                               267,270

2                SET                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               239

2                SEX                           IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               235

3      ********* SF                            (0:10,16)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               66,66,156,528,553,622

2                SOCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               245

69               SPNT                          ENTRY,DECIMAL,FLOAT(SINGLE)
                                               474,486,496,636

                 SPRINT                        FILE,EXTERNAL
                                               17,24,31,41,49,53,57,63,71,74,75,81,82,84,85,95,96,98,99,101,102,103
                                               105,106,108,109,111,112,113,118,119,124,129,130,133,135,136,139,141
                                               142,144,147,148,151,152,154,156,206,412,445,446,448,451,453,455,456
                                               467,469,470,475,484,491,492,497,499,502,559,567,576,583,631,632,637
                                               639,643,654,675,682

                 SQRT                          GENERIC,BUILT-IN FUNCTION
                                               90,91,92,93,94

4                SSST                          (16)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)
                                               39,39,94,94,113,520,548,624

4                SST                           (16)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)
                                               38,38,94,113,113,519,548,623

223              ST                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               272,272,295

2      ********* STEN                          (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
                                               38,39,66,66,85

8                STUDENT                       FILE,EXTERNAL,RECORD,ENVIRONMENT(F(2416))
                                               459,460,503,541,555,603,604,606,607,644,647

2                SUB                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(36),CHARACTER

2                SUBCAR                        IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               332
```

Dataset Limited

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
	SUBSTR	GENERIC,BUILT-IN FUNCTION 169,186,187,191,197,200,229,231,272,273,274,283,284,285,286,287,320 323,352,355,379,381,381,387,422,425,428,431,434,437,463,463,463,537 556,562,566,566,566,571,575,575,575,580,586,642,642,642,652
7	SUPP	AUTOMATIC,UNALIGNED,STRING(1),BIT 473,485,495,508,592,593,627,635,659,660
223	SYF	AUTOMATIC,DEFINED,UNALIGNED,STRING(2),CHARACTER 273,273
223	SYT	AUTOMATIC,DEFINED,UNALIGNED,STRING(2),CHARACTER 274,274
2	TC	IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 326
347	TEST	ENTRY,STRING(1),BIT 389
2	TITLE	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
160	TRANS	ENTRY,DECIMAL,FLOAT(SINGLE) 538
	TRANSLATE	GENERIC,BUILT-IN FUNCTION 225,388
2	TTFT	(2,2)IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER 229,259,381,562,566,566,566,571,575,575,575,580,586
2	UNCOL	(2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER 261
2	UNOTH	(2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER 263
2	********* VALUES	(2,6)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0) 12,12,14,15,47,47,48,48,84,334
3	********* VF	(0:60,6)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 48,48,122,124,524,549,618
4	VST	(6)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE) 15,15,90,90,99,512,544,611
4	VT	(6)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE) 14,14,90,99,99,511,544,610
7	WB	AUTOMATIC,UNALIGNED,STRING(1),BIT 423,424,562,586
7	WCR	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 416,450,451,452,586,586

Dataset Limited

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
7	WD	AUTOMATIC,UNALIGNED,STRING(8),CHARACTER 566,567,575,576
7	WPB	AUTOMATIC,UNALIGNED,STRING(1),BIT 426,427,564
7	WRL	AUTOMATIC,UNALIGNED,STRING(3),CHARACTER 415,422,425,428,431,434,437,447,448,449
2	WS	IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 328
164	X1	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 169,170,175,175,178,182
223	X2	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER 229,230,231,232
348	X20	AUTOMATIC,UNALIGNED,STRING(20),CHARACTER,VARYING 350,352,352,353,355,355
223	XVAR	AUTOMATIC,UNALIGNED,STRING(3),CHARACTER,VARYING 320,321
2	YC	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
7	YE	AUTOMATIC,UNALIGNED,STRING(4),CHARACTER 408,445,467,469,484
223	********* YR	PARAMETER,ALIGNED,BINARY,FIXED(15,0) 222,334,336,338,340,342,344
220	********* YR	PARAMETER,ALIGNED,BINARY,FIXED(15,0) 217,385
7	********* YR	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 12,12,14,15,19,19,21,22,26,26,28,29,34,34,36,37,38,39,47,47,48,48,51 51,52,52,55,55,56,56,60,60,61,61,66,66,84,84,84,84,85,441,443,444 556,588,652
7	YT	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 417,440,442,453

Dataset Limited

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	AHF	786
4	AHST	16
4	AHT	16
349	C	48
5	CLAB	40
7	DESL	8000
3	DSF	602
5	LOP	20
6	MNT	32
163	PCOND	12
3	PFF	864
89	PFK	32
4	PFST	128
4	PFT	128
223	R	528
2	RECORD	2416
5	RITEM	400
5	RLAB	40
3	SF	352
4	SSST	128
4	SST	128
3	VF	732
4	VST	48
4	VT	48

Released Unlimited

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED I IS 14768 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED ITTMS IS 964 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED ITFREQ IS 944 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED SPNT IS 932 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED PRINT IS 1044 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED ITPTMS IS 1080 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED ITPFREQ IS 984 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED TRANS IS 1140 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED LOCTST IS 1128 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED FIND IS 1644 BYTES LONG.

THE STORAGE AREA FOR THE PROCEDURE LABELLED TEST IS 1076 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 406 IS 896 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 461 IS 1028 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 605 IS 928 BYTES LONG.

THE PROGRAM CSECT IS NAMED I AND IS 61812 BYTES LONG.

THE STATIC CSECT IS NAMED ******IA AND IS 8141 BYTES LONG.

STATISTICS SOURCE RECORDS = 582,PROG TEXT STMNTS = 683,OBJECT BYTES = 61812

Dataset Limited

COMPILER DIAGNOSTICS.

WARNINGS.

IEM0227I NO FILE/STRING OPTION SPECIFIED IN ONE OR MORE GET/PUT STATEMENTS. SYSIN/SYSPRINT HAS BEEN ASSUMED IN EACH CASE.

IEM0764I ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY ARE FLAGGED '*********' IN THE XREF/ATR LIST.

IEM1790I DATA CONVERSIONS WILL BE DONE BY SUBROUTINE CALL IN THE FOLLOWING STATEMENTS 192, 198, 227, 230, 232, 270, 281, 295, 321.

END OF DIAGNOSTICS.

COMPILE TIME 68.95 SECS

ELAPSED TIME 210.22 SECS
EXECUTION TERMINATED

Datase Limited

6. PROGRAMME D.

6.2 PROGRAMME LISTING

THIS SECTION CONTAINS A LISTING OF THE SOURCE CODE OF PROGRAMME D.

```
 1                                                                  /*****************************************/
 2    (SUBRG):(STRG):                                               /*                                       */
 3    D:PROC OPTIONS(MAIN);                                         /*          BCRS....PROGRAMME D          */
 4         DCL 1 RECORD BASED(PTR),                                 /*                                       */
 5               2 (DESNO CHAR(8), NAME CHAR(40)),                  /*****************************************/
 6               2 GENERAL,                                         /*                                       */
 7                 3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),   /* DECLARATION OF STRUCTURE 'RECORD'  */
 8                 3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),         /* -> MINOR STRUCTURES OF 'RECORD'    */
 9                 3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)),    /*    -> GENERAL (GENERAL INFORMATION)*/
10                 3 FPOS CHAR(4),                                  /*    -> EDUCATION (PREVIOUS EDUCATION)*/
11               2 EDUCATION,                                       /*    -> COURSE (COURSE DETAILS)       */
12                 3 (SCHNO CHAR(2), SCH(10) CHAR(5)),              /*    -> FINALS (FINAL RESULT)         */
13                 3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),           /*    -> ANNPROG (ANNUAL PROGRESS)     */
14                 3 (UNCOL(2) CHAR(21), FPT(2) CHAR(1), FT(2,2) CHAR(6)),      /*    -> INTEL (INTELLIGENCE TEST SCORES)*/
15                 3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),      /*                                       */
16                 3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),            /*                                       */
17                 3 (ADIP CHAR(30), ADIPC CHAR(1)),                /*                                       */
18                 3 DDIPC CHAR(1),                                 /*                                       */
19                 3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),              /*                                       */
20                 3 ATO CHAR(1),                                   /*                                       */
21               2 COURSE,                                          /*                                       */
22                 3 (LEN CHAR(1), CODES(4) CHAR(3)),               /*                                       */
23                 3 ((LE,DC(4,24:64)) CHAR(1)),                    /*                                       */
24                 3 (DC1(4,26:30) CHAR(2)),                        /*                                       */
25                 3 ((DC46(4),DC49(4)) CHAR(3)),                   /*                                       */
26               2 FINALS,                                          /*                                       */
27                 3 ((TC,WS,BE,SUBCAR) CHAR(1)),                   /*                                       */
28               2 ANNPROG,                                         /*                                       */
29                 3 AP CHAR(802),                                  /*                                       */
30                 3 ((LEAP,FK(100,2)) FIXED BINARY),               /*                                       */
31               2 INTEL,                                           /*                                       */
32                 3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY),    /*                                       */
33                 3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),          /*                                       */
34                 3 STEN(2,16) FIXED BINARY;                       /*                                       */
35    /*-----------------------------------------------------------*/ /*-----------------------------------------*/
36         DCL ((AUTR(*) CHAR(19), (AUTR1(*),ATAB(*)) CHAR(3), ATAB1(*)   /* DECLARATION OF PROGRAMME VARIABLES  */
37         FIXED BINARY) CONTROLLED, (NALC,AK(3,26)) FIXED BINARY, ALPHA  /*                                       */
38         CHAR(26)) EXTERNAL;                                    /*                                       */
39         DCL (FILN1,FILN2,DESNO2,DESNO3,DES) CHAR(8), YE CHAR(4), B BIT(1),   /*                                       */
40         TRANS CHAR(2), C4FLAG BIT(1);                          /*                                       */
41         DCL CON ENTRY(CHAR(8));                                /*                                       */
42    /*-----------------------------------------------------------*/ /*-----------------------------------------*/
43         DCL STUDENT FILE RECORD ENV(F(2416));                  /* DECLARATION OF FILE STUDENT           */
44         DCL DIC FILE;                                          /* DECLARATION OF FILE DIC               */
45    /*************************************************************/ /*****************************************/
46    NMON:PROCEDURE(AMON) RETURNS(CHAR(1));                      /* MONTH CALCULATION ROUTINE             */
47         DCL AMON CHAR(2);                                      /*****************************************/
48         IF AMON='10' THEN RETURN('&'); ELSE                    /* OCTOBER CODE '&'                      */
49         IF AMON='11' THEN RETURN('-'); ELSE                    /* NOVEMBER CODE '-'                     */
50         IF AMON='12' THEN RETURN('0'); ELSE                    /* DECEMBER CODE '0'                     */
51         RETURN(SUBSTR(AMON,2,1));                              /* OTHERWISE CODE '1' TO '9'             */
52    END;                                                        /*                                       */
53    DCL NMON ENTRY(CHAR(2)) RETURNS(CHAR(1));                   /*                                       */
54    /*************************************************************/ /*****************************************/
55    DKEY:PROCEDURE(RC);                                         /* CARD KEY ROUTINE                      */
```

Dataset Limited

```
56          DCL RC CHAR(80), MON CHAR(1);                                  /*****************************************/
57          SUBSTR(RC,1,7)=SUBSTR(DESNO,1,2) || SUBSTR(DESNO,4,5);         /* D.E.S.NO.                             */
58          IF SEX='M' THEN SUBSTR(RC,8,1)='1'; ELSE SUBSTR(RC,8,1)='2';   /* SEX                                   */
59          MON=NMON(SUBSTR(DOB,3,2));                                     /* DATE OF BIRTH                         */
60          SUBSTR(RC,9,5)=SUBSTR(DOB,1,2) || MON || SUBSTR(DOB,5,2);      /*                                       */
61          SUBSTR(RC,14,6)=SUBSTR(NAME,1,INDEX(NAME,',')-1);              /* SURNAME (1'ST 6 CHARACTERS ONLY)      */
62     END;                                                                /*                                       */
63     /*****************************************************************/  /*****************************************/
64     DCOUR:PROCEDURE(RC,I);                                              /* DETAILS OF COURSE                     */
65          DCL RC CHAR(80), MON CHAR(1), I FIXED BINARY;                  /*****************************************/
66          SUBSTR(RC,21,1)='F';                                /* DC21  */ /* FULL TIME STUDENT                     */
67          IF DC(I,24)¬=' ' THEN SUBSTR(RC,24,1)='1';          /* DC24  */ /* YEAR OF COURSE                        */
68          IF DC(I,25)¬=' ' THEN SUBSTR(RC,25,1)='1';          /* DC25  */ /* REPETITION                            */
69          MON=NMON(DC1(I,26));                                /* DC26  */ /* START DATE                            */
70          SUBSTR(RC,26,3)=MON || DC1(I,27);                   /* 27,28 */ /*                                       */
71          MON=NMON(DC1(I,29));                                /* DC29  */ /* EXPECTED END DATE                     */
72          SUBSTR(RC,29,3)=MON || DC1(I,30);                   /* 30,31 */ /*                                       */
73          IF LEN¬='1' THEN SUBSTR(RC,32,1)=DC(I,32);         /* DC 32 */ /* A.T.O. SHORTENED COURSE               */
74          SUBSTR(RC,34,5)=DC(I,34) || DC(I,35) || DC(I,36) || DC(I,37)   /* SCOPE OF TRAINING                     */
75          || DC(I,38);                                        /* 34-38 */ /*                                       */
76          SUBSTR(RC,41,2)=DC(I,41) || DC(I,42);               /* 41-42 */ /*                                       */
77          IF DC(I,59)=' ' | (DC(I,59)¬=' ' & SUBSTR(UNOTH(1),1,1)='F') THEN  /* NON PGCE STUDENTS                     */
78          DO; SUBSTR(RC,43,1)=DC(I,43);                       /* DC43 */ /* -> NO. OF MAIN SUBJECTS               */
79              IF DC(I,44)¬=' ' THEN SUBSTR(RC,44,1)='1';      /* DC44 */ /* -> NO. OF SUBSIDUARY SUBJECTS         */
80              SUBSTR(RC,45,1)=DC(I,45);                       /* DC45  */ /* -> UNDECIDED                          */
81              SUBSTR(RC,46,6)=DC46(I) || DC49(I);             /* 46,49 */ /* -> SUBJECT CODES.                     */
82          END;                                                            /*                                       */
83          IF DC(I,59)¬=' ' & SUBSTR(UNOTH(1),1,1)='B' THEN                /* PGCE STUDENTS                         */
84          SUBSTR(RC,59,1)=DC(I,59);                           /* DC59  */ /* -> RECORD PGCE                        */
85          SUBSTR(RC,64,1)=DC(I,64);                           /* DC64  */ /* B.ED COURSE                           */
86     END;                                                                /*                                       */
87     /*****************************************************************/  /*****************************************/
88     QUALC:PROCEDURE(C);                                                 /* QUALIFICATION CALCULATION ROUTINE     */
89          DCL C(5) CHAR(80), (B11,B61) BIT(1), (CAQ,COQ,CSEQ) PICTURE'99',  /*****************************************/
90          (Q,I,J) FIXED BINARY;                                          /* VARIABLE DECLARATIONS                 */
91          DCL QUAL2 CHAR(10), QT CHAR(3) DEFINED QUAL2 POS(3), QC CHAR(3) /*                                      */
92          DEFINED QUAL2 POS(6), QG CHAR(1) DEFINED QUAL2 POS(9), QP CHAR(1) /*                                    */
93          DEFINED QUAL2 POS(10),AQT(20) CHAR(4),OQT(30) CHAR(3);         /*                                       */
94          COQ=0; CAQ=0; CSEQ=0; AQT='    '; OQT='   ';                   /* INITIALISE VARIABLES                  */
95          B61='1'B;                                                      /* INITIALISE ENGLISH OBTAINED BIT       */
96          B11='1'B;                                                      /* INITIALISE MATHEMATICS OBTAINED BIT   */
97          Q=QUALNO;                                           /* CONVERSION */ /* CALCULATE NO. OF QUALIFICATIONS      */
98          DO I=1 TO Q;                                                   /*                                       */
99             QUAL2=QUAL(I);                                              /* CONSIDER FIRST/NEXT QUALIFICATION     */
100            IF QT='  A' & QP='P' THEN                        /* GCE A */ /* LOCATE A'LEVEL PASSES                 */
101            DO; DO J=1 TO CAQ;                                          /*                                       */
102                   IF QC=SUBSTR(AQT(J),1,3) THEN                        /* -> A'LEVEL ALREADY RECORD             */
103                   DO; IF QG>SUBSTR(AQT(J),4,1) THEN AQT(J)=QC || QG;   /*    -> RE-RECORD IF BETTER GRADE       */
104                       GOTO L2;                                         /*    -> BRANCH FOR NEXT QUALIFICATION   */
105                   END;                                                 /*                                       */
106               END;                                                     /*                                       */
107               CAQ=CAQ+1;                                               /* -> UPDATE A'LEVEL COUNTER             */
108               AQT(CAQ)=QC || QG;                                       /* -> ADD TO A'LEVEL LIST                */
109            END; ELSE                                                   /*                                       */
110            IF (QT='  O' & QP='P') |                         /* GCE O */ /* LOCATE O'LEVEL PASSES, OR             */
111            (QT='  A' & QP='F' & QG='O') THEN                           /* A'LEVEL FAILS WITH O'LEVEL AWARD      */
112            DO; DO J=1 TO COQ;                                          /*                                       */
113                   IF QC=OQT(J) THEN GOTO L2;                           /* -> O'LEVEL ALREADY RECORDED, BRANCH   */
114               END;                                                     /*                                       */
115               COQ=COQ+1;  OQT(COQ)=QC;                                 /* -> UPDATE COUNTER AND ADD TO LIST     */
```

Dataset Limited

```
116                 GOTO L1;                                                  /* -> BRANCH TO ENG/MATHS ROUTINE              */
117             END; ELSE                                                     /*                                             */
118             IF QT='CSE' & QP='P' & QG='1' THEN                  /* CSE 1 */  /* LOCATE CSE GRADE 1 PASSES                   */
119             DO; DO J=I+1 TO Q;                                            /*                                             */
120                     IF SUBSTR(QUAL(J),3,3)='  O' & SUBSTR(QUAL(J),10,1)='P'  /* -> TEST FOR O'LEVEL IN SAME SUBJECT         */
121                     & SUBSTR(QUAL(J),6,3)=SUBSTR(QUAL(I),6,3) THEN GOTO L1;  /*    -> BRANCH TO ENG/MATHS ROUTINE           */
122                 END;                                                      /*                                             */
123                 CSEQ=CSEQ+1;                                              /*                                             */
124                 GOTO L1;                                                  /*                                             */
125             END; ELSE                                                     /*                                             */
126             IF QT='SCC' & QP='P' & (QC='061' | QC='046') THEN   /* SCC   */  /* LOCATE SCHOOL CERTIFICATE PASSES            */
127             DO; SUBSTR(C(3),20,1)='1';                                    /* -> RECORD PASS                              */
128                 GOTO L1;                                                  /* -> BRANCH TO ENG/MATHS ROUTINE              */
129             END; ELSE                                                     /*                                             */
130             IF QG='S' & QP='P' THEN                             /* ORELQ */  /* LOCATE OTHER RELEVANT QUALIFICATIONS        */
131             DO; IF SUBSTR(C(4),32,4)='    ' THEN                          /* -> RECORD CODE                              */
132                 SUBSTR(C(4),32,4)='0' || QC; ELSE                         /*                                             */
133                 IF SUBSTR(C(4),36,4)='    ' THEN                          /* -> RECORD CODE                              */
134                 SUBSTR(C(4),36,4)='0' || QC;                              /*                                             */
135                 C4FLAG='1'B;                                              /* -> SET OPTIONAL CARD 4 BIT                  */
136             END;                                                          /*                                             */
137             GOTO L2;                                                      /* -> BRANCH FOR NEXT QUALIFICATION            */
138     L1:     IF QC='061' & B61 THEN                              /* 61&11 */  /* ENGLISH AWARD ROUTINE                       */
139             DO; B61='0'B;                                                 /* -> SET ENGLISH OBTAINED BIT                 */
140                 SUBSTR(C(3),28,1)='1';                                    /* -> RECORD QUALIFICATION                     */
141                 GOTO L2;                                                  /* -> BRANCH FOR NEXT QUALIFICATION            */
142             END;                                                          /*                                             */
143             IF QC='011' & B11 THEN                                        /* MATHEMATICS AWARD ROUTINE                   */
144             DO; B11='0'B;                                                 /* -> SET MATHEMATICS OBTAINED BIT             */
145                 SUBSTR(C(3),29,1)='1';                                    /* -> RECORD QUALIFICATION                     */
146             END;                                                          /*                                             */
147     L2:     END;                                                          /* BRANCH FOR NEXT QUALIFICATION               */
148             SUBSTR(C(3),21,2)=CSEQ;             /* CSE 1 */   /* CONVERSION */  /* RECORD CSE QUALIFICATIONS                   */
149             SUBSTR(C(3),23,2)=COQ;              /* GCE O */   /* CONVERSION */  /* RECORD O'LEVEL QUALIFICATIONS               */
150             DO I=1 TO CAQ;                                                /* RECORD A'LEVEL GRADES                       */
151                 DO J=1 TO NALC;                                           /*                                             */
152                     IF SUBSTR(AQT(I),1,3)=ATAB(J) THEN                    /* -> LOCATE QUALIFICATION CODE IN ATAB        */
153                     DO; IF SUBSTR(AQT(I),4,1)='X' THEN SUBSTR(C(3),ATAB1(J),1)  /* -> RECORD GRADE OR 'P' IF PASS AND          */
154                         ='P'; ELSE SUBSTR(C(3),ATAB1(J),1)=SUBSTR(AQT(I),4,1);  /* -> GRADE UNKNOWN                            */
155                         GOTO L3;                                          /* -> BRANCH FOR NEXT A'LEVEL                  */
156                     END;                                                  /*                                             */
157                 END;                                                      /*                                             */
158                 PUT EDIT(DESNO,'A.LEVEL CODE',SUBSTR(AQT(I),1,3),'NOT FOUND')  /* -> CLASS K ERROR/WARNING                    */
159                 (SKIP,COL(1),A,3(X(1),A));                                /*                                             */
160     L3:     END;                                                          /*                                             */
161             SUBSTR(C(3),25,2)=CAQ;              /* GCE A */   /* CONVERSION */  /* RECORD A'LEVEL QUALIFICATIONS               */
162     END;                                                                  /*                                             */
163     /*****************************************************************/   /*********************************************/
164     DESADM:PROCEDURE;                                                     /* RECORD ADMISSIONS                           */
165             DCL C(5) CHAR(80), CNAME CHAR(18), CFNAME CHAR(31), MON CHAR(1),  /*********************************************/
166             FIRST CHAR(1),(I,ESNAME,POS) FIXED BINARY;                    /* VARIABLE DECLARATIONS                       */
167             C=' '; CNAME=' '; CFNAME=' ';                                 /* INITIALISE VARIABLES                        */
168             ESNAME=INDEX(NAME,',');                             /* NAME  */  /* CALCULATE POSITION 'END OF SURNAME'         */
169             CNAME=SUBSTR(NAME,1,ESNAME-1);                                /* SURNAME                                     */
170             CFNAME=SUBSTR(NAME,ESNAME+2);                       /* FNAME */  /* CHRISTIAN NAME(S)                           */
171             CALL DKEY(C(5));                                              /* CALL CARD KEY ROUTINE, CARD(5)              */
172             DO I=2 TO 4;                                                  /*                                             */
173                 SUBSTR(C(I),1,19)=SUBSTR(C(5),1,19);                      /* SET KEYS CARDS(2->4)                        */
174             END;                                                          /*                                             */
175             SUBSTR(C(1),1,13)=SUBSTR(C(5),1,13);                          /* SET KEYS CARD(1)                            */
```

Dataset Limited

```
176          IF DOBC¬=' ' THEN SUBSTR(C(1),14,1)='1';                          /* DATE OF BIRTH                                  */
177          SUBSTR(C(1),15,18)=CNAME;                                 /* NAME   */  /* SURNAME                                        */
178          SUBSTR(C(1),33,31)=CFNAME;                                /* FNAME */  /* CHRISTIAN NAME(S)                              */
179          SUBSTR(C(*),74,3)='999';                                            /* BATCH NUMBER                                   */
180          SUBSTR(C(*),77,2)='01';                                             /* TRANSACTION NUMBER                             */
181          SUBSTR(C(1),79,2)='01';                                   /* C.N01 */  /* CARD NUMBER '01'                               */
182          SUBSTR(C(2),22,17)=PNAME;                                 /* PNAME */  /* PREVICUS SURNAME                               */
183          MON=NMON(SUBSTR(TTFT(1,1),3,2));                          /* START */  /* START DATE                                     */
184          SUBSTR(C(2),42,5)=SUBSTR(TTFT(1,1),1,2) || MON ||                     /*                                                */
185          SUBSTR(TTFT(1,1),5,2);                                              /*                                                */
186          SUBSTR(C(2),47,4)='9625';                                 /* ESTNO */  /* ESTABLISHMENT NUMBER                           */
187          SUBSTR(C(2),70,1)=AUTHNO;                                 /* AUTH  */  /* AUTHORITY/CCUNTRY NUMBER                       */
188          FIRST=SUBSTR(AUTH,1,1);                                   /* AUTHC */  /* CALCULATION OF AUTHORITY CODE                  */
189          POS=INDEX(ALPHA,FIRST);                                             /* -> LOCATE GROUP                                */
190          IF POS=0 THEN GOTO L6;                                              /* -> BRANCH IF GROUP ZERO                        */
191          DO I=AK(2,POS) TO AK(3,POS);                                        /*                                                */
192             IF AUTH =AUTR(I) THEN                                            /* -> LOCATE AUTHORITY/COUNTRY                    */
193             DO; SUBSTR(C(2),39,3)=AUTR1(I);                                  /* -> RECORD CODE                                 */
194                 GOTO L6;                                                     /* -> BRANCH OUT OF LOOP                          */
195             END;                                                             /*                                                */
196          END;                                                                /*                                                */
197          PUT EDIT(DESNO,'AUTHORITY CODE',AUTH,'NOT FOUND')                   /* -> CLASS K ERROR/WARNING                       */
198          (SKIP,COL(1),A,3(X(1),A));                                          /*                                                */
199     L6:  SUBSTR(C(2),71,1)=OV25;                                   /* OV25  */  /* OVER 25 OCCUPATION                             */
200          SUBSTR(C(2),79,2)='02';                                   /* C.N02 */  /* CARD NUMBER '02'                               */
201          CALL QUALC(C);                                                      /* CALL QUALIFICATION CALCULATION ROUTINE         */
202          SUBSTR(C(3),79,2)='03';                                   /* C.N03 */  /* CARD NUMBER '03'                               */
203          IF ADIPC¬=' ' | DDIPC¬=' ' | OQC(1)¬='    ' | OQC(2)¬='    '          /* LOCATE OTHER ART QUALIFICATIONS                */
204          | ATO¬=' ' THEN                                                     /*                                                */
205          DO; C4FLAG='1'B;                                                    /* -> SET OPTIONAL CARD 4 BIT                     */
206             SUBSTR(C(4),22,1)=ADIPC;                                 /* ADIPC */  /* -> ART DIPLOMA                                 */
207             SUBSTR(C(4),23,1)=DDIPC;                                 /* DDIPC */  /* -> NATIONAL DIPLOMA                            */
208             SUBSTR(C(4),24,4)=OQC(1);                                /* OADV1 */  /* -> OTHER ART QUALIFICATIONS                    */
209             SUBSTR(C(4),28,4)=OQC(2);                                /* OADV2 */  /* -> OTHER ART QUALIFICATIONS                    */
210             SUBSTR(C(4),44,1)=ATO;                                   /* ATO   */  /* -> EXCEPTED BY A.T.O.                          */
211          END;                                                                /*                                                */
212          SUBSTR(C(4),79,2)='04';                                   /* C.N04 */  /* CARD NUMBER '04'                               */
213          I=1;                                                                /* SET I=1'ST ENTRY                               */
214          CALL DCOUR(C(5),I);                                                 /* CALL DETAILS CF COURSE                         */
215          SUBSTR(C(5),79,2)='05';                                   /* C.N05 */  /* CARD NUMBER '05'                               */
216          IF C4FLAG THEN PUT FILE(SPUNCH) EDIT(C(*))(5(A(80))); ELSE          /* OUTPUT TO SPUNCH CARDS (1->5), OR              */
217          PUT FILE(SPUNCH) EDIT(C(1),C(2),C(3),C(5))(4(A(80)));               /* OUTPUT TO SPUNCH CARDS (1->3 & 5)              */
218     END;                                                                     /*                                                */
219     /*****************************************************************/      /*************************************************/
220     DESWOL:PROCEDURE;                                                        /* WITHDRAWAL/END OF COURSE                       */
221          DCL C CHAR(80), MON CHAR(1), I FIXED BINARY;                        /*************************************************/
222          C=' ';                                                              /*                                                */
223          CALL DKEY(C);                                                       /* CALL KEY ROUTINE                               */
224          IF TRANS='99' THEN GOTO L1;                                         /* BRANCH IF END OF COURSE                        */
225          MON=NMON(SUBSTR(TTFT(1,2),3,2));                                    /* WITHDRAWAL DATE                                */
226          SUBSTR(C,20,5)=SUBSTR(TTFT(1,2),1,2) || MON || SUBSTR(TTFT(1,2),    /*                                                */
227          5,2);                                                               /*                                                */
228          SUBSTR(C,25,4)='9625';                                    /* ESTNO */  /* ESTABLISHMENT NUMBER                           */
229          SUBSTR(C,29,1)=SUBSTR(TTFT(2,1),6,1);                     /* WRDCO */  /* WITHDRAWAL REASON                              */
230          MON=NMON(SUBSTR(TTFT(1,2),3,2));                          /* WRDDT */  /* WITHDRAWAL DATE                                */
231          SUBSTR(C,30,3)=MON || SUBSTR(TTFT(1,2),5,2);                        /*                                                */
232          GOTO L2;                                                            /* BRANCH TO END OF CARD                          */
233     L1:  IF TC¬=' ' THEN SUBSTR(C,40,1)=TC; ELSE                             /* TEACHER'S CERTIFICATE                          */
234          IF WS¬=' ' THEN SUBSTR(C,40,1)=WS; ELSE                             /* WING SCIENCE                                   */
235          IF BE¬=' ' THEN SUBSTR(C,40,1)=BE;                        /* FINRS */  /* B.ED. RESULT                                   */
```

Dataset Limited

```
236          SUBSTR(C,41,1)='1';                                  /* DEGCF */     /* DEGREE VERIFIED                                  */
237          SUBSTR(C,42,1)=SUBCAR;                               /* SUBCR */     /* SUBSEQUENT CAREER                                */
238          MON=NMON(SUBSTR(TTFT(1,2),3,2));                     /* TODAT */     /* TO DATE
239          SUBSTR(C,43,5)=SUBSTR(TTFT(1,2),1,2) || MON || SUBSTR(TTFT(1,2),        /*                                                  */
240          5,2);                                                                   /*                                                  */
241          SUBSTR(C,48,4)='9625';                               /* ESTNO */     /* ESTABLISHMENT NUMBER                             */
242     L2:  SUBSTR(C,77,2)='12';                                 /* TRANS */     /* TRANSACTION NUMBER                               */
243          SUBSTR(C,74,3)='999';                                                   /* BATCH NUMBER                                     */
244          PUT FILE(SPUNCH) EDIT(C)(A(80));                                        /* OUTPUT CARD TO SPUNCH                            */
245     END;                                                                         /*                                                  */
246     /*****************************************************************/          /*****************************************************/
247     DESCOCH:PROCEDURE;                                                           /* CHANGE OF COURSE/SUBJECT/COLLEGE                 */
248          DCL C(2) CHAR(80), MON CHAR(1), I FIXED BINARY;                         /*****************************************************/
249          C=' ';                                                                  /*                                                  */
250          CALL DKEY(C(1));                                                        /* CALL KEY ROUTINE                                 */
251          SUBSTR(C(2),1,19)=SUBSTR(C(1),1,19);                                    /* SET KEYS CARD(2)                                 */
252          SUBSTR(C(*),74,3)='999';                                                /* BATCH NUMBER                                     */
253          SUBSTR(C(*),77,2)=TRANS;                                                /* TRANSACTION NUMBER                               */
254          SUBSTR(C(1),47,4)='9625';                            /* ESTNO */     /* ESTABLISHMENT NUMBER                             */
255          SUBSTR(C(1),79,2)='02';                              /* C.N02 */     /* CARD NUMBER '02'                                 */
256          SUBSTR(C(2),79,2)='05';                              /* C.N05 */     /* CARD NUMBER '05'                                 */
257          IF TRANS='15' THEN I=1; ELSE I=LE;                /* CONVERSION */   /* SET I=LATEST ENTRY                               */
258          CALL DCOUR(C(2),I);                                                     /* CALL DETAILS OF COURSE                           */
259          IF TRANS='15' THEN                                                      /*                                                  */
260          DO; MON=NMON(SUBSTR(TTFT(1,1),3,2));                 /* START */     /* START DATE                                       */
261              SUBSTR(C(1),42,5)=SUBSTR(TTFT(1,1),1,2) || MON ||                   /*                                                  */
262              SUBSTR(TTFT(1,1),5,2);                                              /*                                                  */
263          END;                                                                    /*                                                  */
264          PUT FILE(SPUNCH) EDIT(C(*))(2(A(80)));                                  /* OUTPUT TO SPUNCH CARDS(1 & 2)                    */
265     END;                                                                         /*                                                  */
266     /*****************************************************************/          /*****************************************************/
267          ON ENDFILE(DIC) GOTO L49;                                               /* SET ENDFILE (DIC) ROUTINE                        */
268          GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));              /* INPUT OPTION PARAMETER CARD                      */
269          GET DATA;                                                               /* INPUT OPTION PARAMETERS                          */
270          IF DES='        ' THEN                                                  /* TEST FOR TRANSACTION FILE                        */
271          DO; PUT EDIT('TRANSACTION FILE NOT SPECIFIED....EXECUTION',             /* -> CLASS B ERROR/WARNING MESSAGE                 */
272              ' TERMINATED')(PAGE,COL(1),A,A);                                    /*                                                  */
273              GOTO L49;                                                           /* -> BRANCH TO END OF PROGRAMME                    */
274          END;                                                                    /*                                                  */
275          PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME D',            /* OUTPUT TITLES                                    */
276          'YEAR REQUESTED',YE)(PAGE,COL(1),A,SKIP(2),COL(1),A,COL(1),A,           /*                                                  */
277          COL(30),A);                                                             /*                                                  */
278          PUT SKIP(2);                                                            /*                                                  */
279          B='0'B;                                                                 /*                                                  */
280          OPEN FILE(STUDENT) INPUT TITLE(FILN1);                                  /* OPEN FILE STUDENT                                */
281          OPEN FILE(DIC) INPUT TITLE(DES);                                        /* OPEN FILE DIC                                    */
282          CALL CON(FILN2);                                                        /* CALL CON (INPUT OF DATA CONSTANTS)               */
283     /*-----------------------------------------------------------------*/        /*-----------------------------------------------------*/
284          ON ENDFILE(STUDENT)                                                     /* SET ENDFILE (STUDENT) ROUTINE                    */
285          BEGIN; CLOSE FILE(STUDENT);                                             /* -> CLOSE FILE                                    */
286                 OPEN FILE(STUDENT) INPUT TITLE(FILN1);                           /* -> OPEN FILE                                     */
287                 GOTO L41;                                                        /* -> BRANCH FOR FIRST RECORD                       */
288          END;                                                                    /* END ENDFILE ROUTINE                              */
289     L42: GET FILE(DIC) EDIT(DESNO2,TRANS)(COL(1),A(8),A(2));                     /* INPUT FIRST/NEXT DES.NO. & TRANS. NO.            */
290          IF TRANS¬='01' & TRANS¬='11' & TRANS¬='12' & TRANS¬='15' &              /* TEST TRANS. NO.                                  */
291          TRANS¬='99' THEN                                                        /*                                                  */
292          DO; PUT EDIT(DESNO2,'TRANSACTION IN ERROR....TRANSACTION IGNORED')      /* -> CLASS J ERROR/WARNING                         */
293              (COL(1),A,X(1),A);                                                  /*                                                  */
294              GOTO L42;                                                           /* BRANCH FOR NEXT NUMBER                           */
295          END;                                                                    /*                                                  */
```

Dataset Limited

```
296    L41: READ FILE(STUDENT) SET(PTR);                                  /* READ FIRST/NEXT RECORD                          */
297         IF DESNO=DESNO2 THEN                                          /* TEST RECORD DES.NO. WITH REQUIRED DES.NO.       */
298         DO; IF B THEN B='0'B;                                         /* RECORD 'FOUND' ROUTINE, RESET BIT               */
299             C4FLAG='0'B;                                              /* -> INITIALISE CARD 4 BIT                        */
300             PUT EDIT(DESNO2,'TRANSACTION',TRANS)(COL(1),A,2(X(1),A));  /* -> OUTPUT TEXT                                  */
301             IF TRANS='01' THEN CALL DESADM; ELSE                      /* -> CALL REQUIRED PROCEDURE                      */
302             IF TRANS='11' | TRANS='15' THEN CALL DESCCCH; ELSE        /*                                                 */
303              IF TRANS='12' | TRANS='99' THEN CALL DESWOL;             /*                                                 */
304             PUT EDIT('COMPLETED')(X(1),A);                            /* -> COMPLETE TEXT                                */
305             GOTO L42;                                                 /* -> BRANCH FOR NEXT NUMBER                       */
306         END; ELSE                                                     /* END 'RECORD FOUND' ROUTINE                      */
307         DO; IF ¬B THEN                                                /* RECORD 'NOT FOUND' ROUTINE                      */
308             DO; DESNO3=DESNO;                                         /* -> RECORD CURRENT DES.NO.                       */
309                 B='1'B;                                               /* -> SET 'RECORD FOUND' BIT                       */
310             END; ELSE                                                 /*                                                 */
311             DO; IF DESNO=DESNO3 THEN                                  /* -> TEST FOR COMPLETE FILE CYCLE                 */
312                 DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')  /* -> CLASS C ERROR/WARNING                      */
313                    (COL(1),A,X(1),A);                                 /*                                                 */
314                    B='0'B;                                            /* -> RESET 'RECORD FOUND' BIT                     */
315                    GOTO L42;                                          /* -> BRANCH FOR NEXT NUMBER                       */
316                 END;                                                  /*                                                 */
317             END;                                                      /*                                                 */
318             GOTO L41;                                                 /* -> BRANCH FOR NEXT RECORD                       */
319         END;                                                          /* END RECORD 'NOT FOUND' ROUTINE                  */
320    /*-----------------------------------------------------------------*/  /*-----------------------------------------------*/
321    L49: PUT PAGE;                                                     /*                                                 */
322         CLOSE FILE(STUDENT);                                          /* CLOSE STUDENT                                   */
323         CLOSE FILE(DIC);                                              /* CLOSE DIC                                       */
324    END;                                                               /* END PROGRAMME D                                 */
325                                                                       /*************************************************/
326   %PROCESS;                                                           /* %PROCESS;                                       */
327                                                                       /*************************************************/
328    CON:PROC(FILN2);                                                   /*                                                 */
329         DCL FILN2 CHAR(8);                                            /* PROCEDURE CON                                   */
330         DCL (I,J,K) FIXED BINARY, AK(3,26) FIXED BINARY EXTERNAL,     /*                                                 */
331         (AUTR(*) CHAR(19), AUTR1(*) CHAR(3), ATAB(*) CHAR(3), ATAB1(*)  /*************************************************/
332         FIXED BINARY) CONTROLLED EXTERNAL, (ALPHA CHAR(26), NALC FIXED  /*                                             */
333         BINARY) EXTERNAL;                                             /* VARIABLE DECLARATIONS                           */
334         DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),      /*                                                 */
335         MK(3) CHAR(12), OK(10) CHAR(35), PK(5) CHAR(30), GRC CHAR(11),  /*                                             */
336         GRK(11) CHAR(30),OAK(4) CHAR(40), ORK(6) CHAR(30), OPSOK(6)   /*                                                 */
337         CHAR(40)) VARYING EXTERNAL;                                   /*                                                 */
338         DCL NORM(2,16,10) FIXED BINARY EXTERNAL;                      /*                                                 */
339         DCL NORM2(2,16,2) FLOAT DECIMAL EXTERNAL;                     /*                                                 */
340    /*-----------------------------------------------------------------*/  /*-----------------------------------------------*/
341         DCL CONST FILE;                                               /* DECLARATION OF FILE (CONST)                     */
342    /*-----------------------------------------------------------------*/  /*-----------------------------------------------*/
343         OPEN FILE(CONST) INPUT TITLE(FILN2);                          /* OPEN CONST FOR INPUT                            */
344         DO I=1 TO 26;                                                 /*                                                 */
345            GET FILE(CONST) LIST(AK(1,I));                             /* INPUT AUTHORITY/COUNTRY NUMBERS                 */
346         END;                                                          /*                                                 */
347         AK(2,1)=1;     AK(3,1)=AK(2,1)+AK(1,1)-1;                     /* -> CALCULATE AK KEYS                            */
348         DO I=2 TO 26;                                                 /*                                                 */
349            AK(2,I)=AK(3,I-1)+1;                                       /* -> START POSITIONS                              */
350            AK(3,I)=AK(1,I)+AK(2,I)-1;                                 /* -> LAST POSITIONS                               */
351         END;                                                          /*                                                 */
352         N=SUM(AK(1,*));                                               /* -> CALCULATE STORAGE REQUIRED                   */
353         ALLOCATE AUTR(N),AUTR1(N);                                    /* -> ALLOCATE STORAGE                             */
354         DO K=1 TO 26;                                                 /*                                                 */
355           DO I=AK(2,K) TO AK(3,K);                                    /*                                                 */
```

Dataset Limited

```
356            GET FILE(CONST) EDIT(AUTR(I),AUTR1(I))                    /* -> INPUT AUTHORITY/COUNTRY & CODES        */
357            (COL(1),A(19),COL(40),A(3));                              /*                                           */
358            END;                                                      /*                                           */
359         END;                                                         /*                                           */
360         GET FILE(CONST) EDIT((OAK(K) DO K=1 TO 4))(4(COL(1),A(40))); /* INPUT AREA CODES (PRE 1972)               */
361         GET FILE(CONST) EDIT((OFK(K) DO K=1 TO 6))(6(COL(1),A(30))); /* INPUT RELIGION CODES (PRE 1972)           */
362         GET FILE(CONST) EDIT((OPSOK(K) DO K=1 TO 6))(6(COL(1),A(40))); /* INPUT PARENT/STUDENT OCC.CODES (PRE 1972) */
363         GET FILE(CONST) EDIT((SCK(I) DO I=1 TO 9))(9(COL(1),A(60))); /* INPUT SCHOOL CODES                        */
364         GET FILE(CONST) EDIT((PSOK(I) DO I=1 TO 5))(5(COL(1),A(65))); /* INPUT PARENT/STUDENT OCC. CODES           */
365         GET FILE(CONST) EDIT((STK(I) DO I=1 TO 6))(6(COL(1),A(50))); /* INPUT SETTLEMENT CODES                    */
366         GET FILE(CONST) EDIT((MK(I) DO I=1 TO 3))(3(COL(1),A(12))); /* INPUT MARITAL STATUS CODES                 */
367         GET FILE(CONST) EDIT((OK(I) DO I=1 TO 10))(10(COL(1),A(35))); /* INPUT OVER 25 OCC. CODES                 */
368         GET FILE(CONST) EDIT((RK(I) DO I=1 TO 5))(5(COL(1),A(30))); /* INPUT RELIGION CODES                       */
369         GET FILE(CONST) EDIT((GRK(I) DO I=1 TO 11))(11(COL(1),A(30))); /* INPUT REGION CODES                      */
370         GET FILE(CONST) EDIT(ALPHA)(COL(1),A(26));                   /* INPUT ALPHABET                            */
371         GRC=SUBSTR(ALPHA,1,11);                                      /* GEOGRAPHICAL REGION KEYS                  */
372         GET FILE(CONST) EDIT(NALC)(COL(1),F(3));                     /* INPUT A'LEVEL NUMBER                      */
373         ALLOCATE ATAB(NALC),ATAB1(NALC);                             /* -> ALLOCATE STORAGE                       */
374         GET FILE(CONST) SKIP;                                        /*                                           */
375         GET FILE(CONST) EDIT((ATAB(I),ATAB1(I) DO I=1 TO NALC))      /* -> INPUT A'LEVEL CODES & COLUMNS          */
376         (X(2),A(3),X(1),F(2));                                       /*                                           */
377         DO I=1 TO 2;                                                 /*                                           */
378            DO J=1 TO 16;                                             /*                                           */
379               GET FILE(CONST) LIST((NORM(I,J,K) DO K=1 TO 10));      /* INPUT NORM TABLES                         */
380               GET FILE(CONST) LIST((NORM2(I,J,K) DO K=1 TO 2));      /*                                           */
381            END;                                                      /*                                           */
382         END;                                                         /*                                           */
383         CLOSE FILE(CONST);                                           /* CLOSE CONST                               */
384      END;                                                            /* END PROCEDURE CON                         */
```

Dataset Limited

6.3 COMPILER LISTING

THIS SECTION CONTAINS A LISTING OF THE OUTPUT PROVIDED BY THE PL/1 F COMPILER DURING THE COMPILATION OF PROGRAMME D.

THE JCL WAS AS FOLLOWS:

```
$RUN *PL1 SPUNCH=-D
  ----------------------------
 |                            |
 |                            |
 |  PROGRAMME D  SOURCE DECK  |
 |                            |
 |                            |
  ----------------------------
%PROCESS;
  ----------------------------
 |                            |
 |  PROCEDURE CON SOURCE DECK |
 |                            |
  ----------------------------
$ENDFILE
```

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRO
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST

 *OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
 *OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
 *OPTIONS IN EFFECT*     NEST,OPLIST
```

Dataset Limited

```
STMT LEVEL NEST
                                                                                D0000010
  1                 (SUBRG):(STRG):                                             D0000020
                    D:PROC OPTIONS(MAIN);                                       D0000030
  2     1                DCL 1 RECORD BASED(PTR),                               D0000040
                            2 (DESNO CHAR(8), NAME CHAR(40)),                   D0000050
                            2 GENERAL,                                          D0000060
                              3 (PNAME CHAR(17), (TITLE,YC) CHAR(2), (SEX,REG) CHAR(1)),D0000070
                              3 ((DOBC,MART,SET,POCC,SOCC,OV25,REL,CONF) CHAR(1)),      D0000080
                              3 ((DOB,TTFT(2,2)) CHAR(6),AUTH CHAR(19),AUTHNO CHAR(1)), D0000090
                              3 FPOS CHAR(4),                                   D0000100
                            2 EDUCATION,                                        D0000110
                              3 (SCHNO CHAR(2), SCH(10) CHAR(5)),               D0000120
                              3 (QUALNO CHAR(2), QUAL(30) CHAR(10)),            D0000130
                              3 (UNCOL(2) CHAR(21), FPT(2) CHAR(1), FT(2,2) CHAR(6)),   D0000140
                              3 (UNOTH(2) CHAR(21), DEG(2) CHAR(3), HPCD(2) CHAR(4)),   D0000150
                              3 (SUB(2) CHAR(36), DATT(2) CHAR(6)),             D0000160
                              3 (ADIP CHAR(30), ADIPC CHAR(1)),                 D0000170
                              3 DDIPC CHAR(1),                                  D0000180
                              3 (OQ(2) CHAR(30), OQC(2) CHAR(4)),               D0000190
                              3 ATO CHAR(1),                                    D0000200
                            2 COURSE,                                           D0000210
                              3 (LEN CHAR(1), CODES(4) CHAR(3)),                D0000220
                              3 ((LE,DC(4,24:64)) CHAR(1)),                     D0000230
                              3 (DC1(4,26:30) CHAR(2)),                         D0000240
                              3 ((DC46(4),DC49(4)) CHAR(3)),                    D0000250
                            2 FINALS,                                           D0000260
                              3 ((TC,WS,BE,SUBCAR) CHAR(1)),                    D0000270
                            2 ANNPROG,                                          D0000280
                              3 AP CHAR(802),                                   D0000290
                              3 ((LEAP,FK(100,2)) FIXED BINARY),                D0000300
                            2 INTEL,                                            D0000310
                              3 (INTELB CHAR(2),(VALUES(2,6), DSCALE(2)) FIXED BINARY), D0000320
                              3 ((AH4(2,3),PF16(2,16)) FIXED BINARY),           D0000330
                              3 STEN(2,16) FIXED BINARY;                        D0000340
                    /*-----------------------------------------------------------------*/D0000350
  3     1                DCL ((AUTR(*) CHAR(19), (AUTR1(*),ATAB(*)) CHAR(3), ATAB1(*)      D0000360
                         FIXED BINARY) CONTROLLED, (NALC,AK(3,26)) FIXED BINARY, ALPHA     D0000370
                         CHAR(26)) EXTERNAL;                                    D0000380
  4     1                DCL (FILN1,FILN2,DESNO2,DESNO3,DES) CHAR(8), YE CHAR(4), B BIT(1),D0000390
                         TRANS CHAR(2), C4FLAG BIT(1);                          D0000400
  5     1                DCL CON ENTRY(CHAR(8));                                D0000410
                    /*-----------------------------------------------------------------*/D0000420
  6     1                DCL STUDENT FILE RECORD ENV(F(2416));                  D0000430
  7     1                DCL DIC FILE;                                          D0000440
                    /*****************************************************************/D0000450
  8     1           NMON:PROCEDURE(AMON) RETURNS(CHAR(1));                      D0000460
  9     2                DCL AMON CHAR(2);                                      D0000470
 10     2                IF AMON='10' THEN RETURN('&'); ELSE                    D0000480
 12     2                IF AMON='11' THEN RETURN('-'); ELSE                    D0000490
 14     2                IF AMON='12' THEN RETURN('0'); ELSE                    D0000500
 16     2                RETURN(SUBSTR(AMON,2,1));                              D0000510
 17     2           END;                                                        D0000520
 18     1           DCL NMON ENTRY(CHAR(2)) RETURNS(CHAR(1));                   D0000530
                    /*****************************************************************/D0000540
 19     1           DKEY:PROCEDURE(RC);                                         D0000550
 20     2                DCL RC CHAR(80), MON CHAR(1);                          D0000560
```

```
STMT LEVEL NEST

 21     2              SUBSTR(RC,1,7)=SUBSTR(DESNO,1,2) || SUBSTR(DESNO,4,5);                D0000570
 22     2              IF SEX='M' THEN SUBSTR(RC,8,1)='1'; ELSE SUBSTR(RC,8,1)='2';          D0000580
 25     2              MON=NMON(SUBSTR(DOB,3,2));                                            D0000590
 26     2              SUBSTR(RC,9,5)=SUBSTR(DOB,1,2) || MON || SUBSTR(DOB,5,2);             D0000600
 27     2              SUBSTR(RC,14,6)=SUBSTR(NAME,1,INDEX(NAME,',')-1);                     D0000610
 28     2         END;                                                                       D0000620
                  /*****************************************************************/ D0000630
 29     1         DCOUR:PROCEDURE(RC,I);                                                     D0000640
 30     2              DCL RC CHAR(80), MON CHAR(1), I FIXED BINARY;                         D0000650
 31     2              SUBSTR(RC,21,1)='F';                                  /* DC21  */D0000660
 32     2              IF DC(I,24)¬=' ' THEN SUBSTR(RC,24,1)='1';            /* DC24  */D0000670
 34     2              IF DC(I,25)¬=' ' THEN SUBSTR(RC,25,1)='1';            /* DC25  */D0000680
 36     2              MON=NMON(DC1(I,26));                                  /* DC26  */D0000690
 37     2              SUBSTR(RC,26,3)=MON || DC1(I,27);                     /* 27,28 */D0000700
 38     2              MON=NMON(DC1(I,29));                                  /* DC29  */D0000710
 39     2              SUBSTR(RC,29,3)=MON || DC1(I,30);                     /* 30,31 */D0000720
 40     2              IF LEN¬='1' THEN SUBSTR(RC,32,1)=DC(I,32);           /* DC 32 */ D0000730
 42     2              SUBSTR(RC,34,5)=DC(I,34) || DC(I,35) || DC(I,36) || DC(I,37)          D0000740
                       || DC(I,38);                                          /* 34-38 */D0000750
 43     2              SUBSTR(RC,41,2)=DC(I,41) || DC(I,42);                 /* 41-42 */D0000760
 44     2              IF DC(I,59)=' ' | (DC(I,59)¬=' ' & SUBSTR(UNOTH(1),1,1)='F') THEN D0000770
 45     2              DO; SUBSTR(RC,43,1)=DC(I,43);                         /* DC43 */ D0000780
 47     2    1             IF DC(I,44)¬=' ' THEN SUBSTR(RC,44,1)='1';       /* DC44 */ D0000790
 49     2    1             SUBSTR(RC,45,1)=DC(I,45);                        /* DC45  */D0000800
 50     2    1             SUBSTR(RC,46,6)=DC46(I) || DC49(I);              /* 46,49 */D0000810
 51     2    1             END;                                                            D0000820
 52     2              IF DC(I,59)¬=' ' & SUBSTR(UNOTH(1),1,1)='B' THEN                      D0000830
 53     2              SUBSTR(RC,59,1)=DC(I,59);                             /* DC59  */D0000840
 54     2              SUBSTR(RC,64,1)=DC(I,64);                             /* DC64  */D0000850
 55     2         END;                                                                       D0000860
                  /*****************************************************************/D0000870
 56     1         QUALC:PROCEDURE(C);                                                        D0000880
 57     2              DCL C(5) CHAR(80), (B11,B61) BIT(1), (CAQ,CCQ,CSEQ) PICTURE'99',      D0000890
                       (Q,I,J) FIXED BINARY;                                                 D0000900
 58     2              DCL QUAL2 CHAR(10), QT CHAR(3) DEFINED QUAL2 POS(3), QC CHAR(3)       D0000910
                       DEFINED QUAL2 POS(6), QG CHAR(1) DEFINED QUAL2 POS(9), QP CHAR(1)     D0000920
                       DEFINED QUAL2 POS(10),AQT(20) CHAR(4),OQT(30) CHAR(3);                D0000930
 59     2              COQ=0; CAQ=0; CSEQ=0; AQT='    '; OQT='   ';                          D0000940
 64     2              B61='1'B;                                                             D0000950
 65     2              B11='1'B;                                                             D0000960
 66     2              Q=QUALNO;                                          /* CONVERSION */D0000970
 67     2              DO I=1 TO Q;                                                          D0000980
 68     2    1            QUAL2=QUAL(I);                                                    D0000990
 69     2    1            IF QT='  A' & QP='P' THEN                             /* GCE A */D0001000
 70     2    1            DO; DO J=1 TO CAQ;                                                D0001010
 72     2    3                 IF QC=SUBSTR(AQT(J),1,3) THEN                               D0001020
 73     2    3                 DO; IF QG>SUBSTR(AQT(J),4,1) THEN AQT(J)=QC || QG;          D0001030
 76     2    4                     GOTO L2;                                                D0001040
 77     2    4                 END;                                                        D0001050
 78     2    3               END;                                                          D0001060
 79     2    2               CAQ=CAQ+1;                                                    D0001070
 80     2    2               AQT(CAQ)=QC || QG;                                            D0001080
 81     2    2            END; ELSE                                                         D0001090
 82     2    1            IF (QT='  O' & QP='P') |                              /* GCE O */D0001100
```

Dataset Limited

```
STMT LEVEL NEST

                          (QT='  A' & QP='F' & QG='0') THEN                          D0001110
 83     2     1           DO; DO J=1 TO COQ;                                         D0001120
 85     2     3                IF QC=OQT(J) THEN GOTO L2;                            D0001130
 87     2     3               END;                                                   D0001140
 88     2     2               COQ=COQ+1;  OQT(COQ)=QC;                               D0001150
 90     2     2               GOTO L1;                                               D0001160
 91     2     2           END; ELSE                                                  D0001170
 92     2     1           IF QT='CSE' & QP='P' & QG='1' THEN                 /* CSE 1 */D0001180
 93     2     1           DO; DO J=I+1 TO Q;                                         D0001190
 95     2     3                IF SUBSTR(QUAL(J),3,3)='  0' & SUBSTR(QUAL(J),10,1)='P' D0001200
 96     2     3                & SUBSTR(QUAL(J),6,3)=SUBSTR(QUAL(I),6,3) THEN GOTO L1; D0001210
 97     2     3               END;                                                   D0001220
 98     2     2               CSEQ=CSEQ+1;                                           D0001230
 99     2     2               GOTO L1;                                               D0001240
100     2     2           END; ELSE                                                  D0001250
101     2     1           IF QT='SCC' & QP='P' & (QC='061' | QC='046') THEN   /* SCC   */D0001260
102     2     1           DO; SUBSTR(C(3),20,1)='1';                                 D0001270
104     2     2               GOTO L1;                                               D0001280
105     2     2           END; ELSE                                                  D0001290
106     2     1           IF QG='S' & QP='P' THEN                            /* ORELQ */D0001300
107     2     1           DO; IF SUBSTR(C(4),32,4)='    ' THEN                       D0001310
109     2     2               SUBSTR(C(4),32,4)='0' || QC; ELSE                      D0001320
110     2     2               IF SUBSTR(C(4),36,4)='    ' THEN                       D0001330
111     2     2               SUBSTR(C(4),36,4)='0' || QC;                           D0001340
112     2     2               C4FLAG='1'B;                                           D0001350
113     2     2           END;                                                       D0001360
114     2     1           GOTO L2;                                                   D0001370
115     2     1     L1:   IF QC='061' & B61 THEN                             /* 61&11 */D0001380
116     2     1           DO; B61='0'B;                                              D0001390
118     2     2               SUBSTR(C(3),28,1)='1';                                 D0001400
119     2     2               GOTO L2;                                               D0001410
120     2     2           END;                                                       D0001420
121     2     1           IF QC='011' & B11 THEN                                     D0001430
122     2     1           DO; B11='0'B;                                              D0001440
124     2     2               SUBSTR(C(3),29,1)='1';                                 D0001450
125     2     2           END;                                                       D0001460
126     2     1     L2:  END;                                                        D0001470
127     2                SUBSTR(C(3),21,2)=CSEQ;          /* CSE 1 */      /* CONVERSION */D0001480
128     2                SUBSTR(C(3),23,2)=COQ;           /* GCE 0 */      /* CONVERSION */D0001490
129     2                DO I=1 TO CAQ;                                             D0001500
130     2     1             DO J=1 TO NALC;                                         D0001510
131     2     2                IF SUBSTR(AQT(I),1,3)=ATAB(J) THEN                   D0001520
132     2     2                DO; IF SUBSTR(AQT(I),4,1)='X' THEN SUBSTR(C(3),ATAB1(J),1) D0001530
135     2     3                    ='P'; ELSE SUBSTR(C(3),ATAB1(J),1)=SUBSTR(AQT(I),4,1); D0001540
136     2     3                    GOTO L3;                                        D0001550
137     2     3                END;                                                D0001560
138     2     2             END;                                                    D0001570
139     2     1             PUT EDIT(DESNO,'A.LEVEL CODE',SUBSTR(AQT(I),1,3),'NOT FOUND') D0001580
                            (SKIP,COL(1),A,3(X(1),A));                             D0001590
140     2     1     L3:  END;                                                        D0001600
141     2                SUBSTR(C(3),25,2)=CAQ;           /* GCE A */      /* CONVERSION */D0001610
142     2           END;                                                            D0001620
                    /*****************************************************************/D0001630
143     1           DESADM:PROCEDURE;                                               D0001640
```

```
STMT LEVEL NEST

144    2          DCL C(5) CHAR(80), CNAME CHAR(18), CFNAME CHAR(31), MON CHAR(1),  D0001650
                  FIRST CHAR(1),(I,ESNAME,POS) FIXED BINARY;                        D0001660
145    2          C=' '; CNAME=' '; CFNAME=' ';                                     D0001670
148    2          ESNAME=INDEX(NAME,',');                                 /* NAME  */D0001680
149    2          CNAME=SUBSTR(NAME,1,ESNAME-1);                                    D0001690
150    2          CFNAME=SUBSTR(NAME,ESNAME+2);                           /* FNAME */D0001700
151    2          CALL DKEY(C(5));                                                  D0001710
152    2          DO I=2 TO 4;                                                      D0001720
153    2    1        SUBSTR(C(I),1,19)=SUBSTR(C(5),1,19);                          D0001730
154    2    1     END;                                                             D0001740
155    2          SUBSTR(C(1),1,13)=SUBSTR(C(5),1,13);                              D0001750
156    2          IF DOBC¬=' ' THEN SUBSTR(C(1),14,1)='1';                          D0001760
158    2          SUBSTR(C(1),15,18)=CNAME;                               /* NAME  */D0001770
159    2          SUBSTR(C(1),33,31)=CFNAME;                              /* FNAME */D0001780
160    2          SUBSTR(C(*),74,3)='999';                                          D0001790
161    2          SUBSTR(C(*),77,2)='01';                                           D0001800
162    2          SUBSTR(C(1),79,2)='01';                                 /* C.N01 */D0001810
163    2          SUBSTR(C(2),22,17)=PNAME;                               /* PNAME */D0001820
164    2          MON=NMON(SUBSTR(TTFT(1,1),3,2));                        /* START */D0001830
165    2          SUBSTR(C(2),42,5)=SUBSTR(TTFT(1,1),1,2) || MON ||                 D0001840
                  SUBSTR(TTFT(1,1),5,2);                                            D0001850
166    2          SUBSTR(C(2),47,4)='9625';                               /* ESTNO */D0001860
167    2          SUBSTR(C(2),70,1)=AUTHNO;                               /* AUTH  */D0001870
168    2          FIRST=SUBSTR(AUTH,1,1);                                 /* AUTHC */D0001880
169    2          POS=INDEX(ALPHA,FIRST);                                           D0001890
170    2          IF POS=0 THEN GOTO L6;                                            D0001900
172    2          DO I=AK(2,POS) TO AK(3,POS);                                      D0001910
173    2    1        IF AUTH =AUTR(I) THEN                                         D0001920
174    2    1        DO; SUBSTR(C(2),39,3)=AUTR1(I);                               D0001930
176    2    2            GOTO L6;                                                  D0001940
177    2    2        END;                                                          D0001950
178    2    1     END;                                                             D0001960
179    2          PUT EDIT(DESNO,'AUTHORITY CODE',AUTH,'NOT FOUND')                 D0001970
                  (SKIP,COL(1),A,3(X(1),A));                                        D0001980
180    2     L6:  SUBSTR(C(2),71,1)=OV25;                                 /* OV25  */D0001990
181    2          SUBSTR(C(2),79,2)='02';                                 /* C.N02 */D0002000
182    2          CALL QUALC(C);                                                    D0002010
183    2          SUBSTR(C(3),79,2)='03';                                 /* C.N03 */D0002020
184    2          IF ADIPC¬=' ' | DDIPC¬=' ' | OQC(1)¬='    ' | OQC(2)¬='    '      D0002030
                  | ATO¬=' ' THEN                                                   D0002040
185    2          DO; C4FLAG='1'B;                                                  D0002050
187    2    1         SUBSTR(C(4),22,1)=ADIPC;                             /* ADIPC */D0002060
188    2    1         SUBSTR(C(4),23,1)=DDIPC;                             /* DDIPC */D0002070
189    2    1         SUBSTR(C(4),24,4)=OQC(1);                            /* OADV1 */D0002080
190    2    1         SUBSTR(C(4),28,4)=OQC(2);                            /* OADV2 */D0002090
191    2    1         SUBSTR(C(4),44,1)=ATO;                               /* ATO   */D0002100
192    2    1     END;                                                             D0002110
193    2          SUBSTR(C(4),79,2)='04';                                 /* C.N04 */D0002120
194    2          I=1;                                                              D0002130
195    2          CALL DCOUR(C(5),I);                                               D0002140
196    2          SUBSTR(C(5),79,2)='05';                                 /* C.N05 */D0002150
197    2          IF C4FLAG THEN PUT FILE(SPUNCH) EDIT(C(*))(5(A(80))); ELSE        D0002160
199    2          PUT FILE(SPUNCH) EDIT(C(1),C(2),C(3),C(5))(4(A(80)));             D0002170
200    2     END;                                                                  D0002180
```

Dataset Limited

```
STMT LEVEL NEST

                  /*****************************************************************/D0002190
201     1         DESWOL:PROCEDURE;                                                   D0002200
202     2              DCL C CHAR(80), MON CHAR(1), I FIXED BINARY;                   D0002210
203     2              C=' ';                                                         D0002220
204     2              CALL DKEY(C);                                                  D0002230
205     2              IF TRANS='99' THEN GOTO L1;                                    D0002240
207     2              MON=NMON(SUBSTR(TTFT(1,2),3,2));                               D0002250
208     2              SUBSTR(C,20,5)=SUBSTR(TTFT(1,2),1,2) || MON || SUBSTR(TTFT(1,2),  D0002260
                       5,2);                                                          D0002270
209     2              SUBSTR(C,25,4)='9625';                          /* ESTNC */D0002280
210     2              SUBSTR(C,29,1)=SUBSTR(TTFT(2,1),6,1);           /* WRDCO */D0002290
211     2              MON=NMON(SUBSTR(TTFT(1,2),3,2));                /* WRDDT */D0002300
212     2              SUBSTR(C,30,3)=MON || SUBSTR(TTFT(1,2),5,2);                   D0002310
213     2              GOTO L2;                                                       D0002320
214     2         L1:  IF TC¬=' ' THEN SUBSTR(C,40,1)=TC; ELSE                        D0002330
216     2              IF WS¬=' ' THEN SUBSTR(C,40,1)=WS; ELSE                        D0002340
218     2              IF BE¬=' ' THEN SUBSTR(C,40,1)=BE;              /* FINRS */D0002350
220     2              SUBSTR(C,41,1)='1';                             /* DEGCF */D0002360
221     2              SUBSTR(C,42,1)=SUBCAR;                          /* SUBCR */D0002370
222     2              MON=NMON(SUBSTR(TTFT(1,2),3,2));                /* TODAT */D0002380
223     2              SUBSTR(C,43,5)=SUBSTR(TTFT(1,2),1,2) || MON || SUBSTR(TTFT(1,2),  D0002390
                       5,2);                                                          D0002400
224     2              SUBSTR(C,48,4)='9625';                          /* ESTNO */D0002410
225     2         L2:  SUBSTR(C,77,2)='12';                            /* TRANS */D0002420
226     2              SUBSTR(C,74,3)='999';                                          D0002430
227     2              PUT FILE(SPUNCH) EDIT(C)(A(80));                               D0002440
228     2         END;                                                                D0002450
                  /*****************************************************************/D0002460
229     1         DESCOCH:PROCEDURE;                                                  D0002470
230     2              DCL C(2) CHAR(80), MON CHAR(1), I FIXED BINARY;                D0002480
231     2              C=' ';                                                         D0002490
232     2              CALL DKEY(C(1));                                               D0002500
233     2              SUBSTR(C(2),1,19)=SUBSTR(C(1),1,19);                           D0002510
234     2              SUBSTR(C(*),74,3)='999';                                       D0002520
235     2              SUBSTR(C(*),77,2)=TRANS;                                       D0002530
236     2              SUBSTR(C(1),47,4)='9625';                       /* ESTNO */D0002540
237     2              SUBSTR(C(1),79,2)='02';                         /* C.NC2 */D0002550
238     2              SUBSTR(C(2),79,2)='05';                         /* C.NO5 */D0002560
239     2              IF TRANS='15' THEN I=1; ELSE I=LE;           /* CONVERSION */D0002570
242     2              CALL DCOUR(C(2),I);                                            D0002580
243     2              IF TRANS='15' THEN                                             D0002590
244     2              DO; MON=NMON(SUBSTR(TTFT(1,1),3,2));            /* START */D0002600
246     2     1            SUBSTR(C(1),42,5)=SUBSTR(TTFT(1,1),1,2) || MON ||         D0002610
                           SUBSTR(TTFT(1,1),5,2);                                    D0002620
247     2     1        END;                                                           D0002630
248     2              PUT FILE(SPUNCH) EDIT(C(*))(2(A(80)));                         D0002640
249     2         END;                                                                D0002650
                  /*****************************************************************/D0002660
250     1              ON ENDFILE(DIC) GOTO L49;                                      D0002670
252     1              GET EDIT(FILN1,FILN2,YE)(COL(1),A(8),X(1),A(8),X(1),A(4));     D0002680
253     1              GET DATA;                                                      D0002690
254     1              IF DES='        ' THEN                                         D0002700
255     1              DO; PUT EDIT('TRANSACTION FILE NOT SPECIFIED....EXECUTION',    D0002710
                           ' TERMINATED')(PAGE,COL(1),A,A);                          D0002720
```

```
STMT LEVEL NEST

257     1    1            GOTO L49;                                          DC002730
258     1    1        END;                                                   D0002740
259     1             PUT EDIT('BCRS....BEDE COLLEGE RECORD SYSTEM','PROGRAMME C',   D0002750
                      'YEAR REQUESTED',YE)(PAGE,COL(1),A,SKIP(2),COL(1),A,COL(1),A,  D0002760
                      COL(30),A);                                            D0002770
260     1             PUT SKIP(2);                                           D0002780
261     1             B='0'B;                                                D0002790
262     1             OPEN FILE(STUDENT) INPUT TITLE(FILN1);                 D0002800
263     1             OPEN FILE(DIC) INPUT TITLE(DES);                       D0002810
264     1             CALL CON(FILN2);                                       D0002820
                 /*-----------------------------------------------------------*/D0002830
265     1             ON ENDFILE(STUDENT)                                    D0002840
266     1             BEGIN; CLOSE FILE(STUDENT);                            D0002850
268     2                    OPEN FILE(STUDENT) INPUT TITLE(FILN1);          D0002860
269     2                    GOTO L41;                                       D0002870
270     2             END;                                                   D0002880
271     1        L42: GET FILE(DIC) EDIT(DESNO2,TRANS)(COL(1),A(8),A(2));    D0002890
272     1             IF TRANS¬='01' & TRANS¬='11' & TRANS¬='12' & TRANS¬='15' &  D0002900
                      TRANS¬='99' THEN                                       D0002910
273     1             DO; PUT EDIT(DESNO2,'TRANSACTION IN ERROR....TRANSACTION IGNORED')D0002920
                          (COL(1),A,X(1),A);                                 D0002930
275     1    1            GOTO L42;                                          D0002940
276     1    1        END;                                                   D0002950
277     1        L41: READ FILE(STUDENT) SET(PTR);                           D0002960
278     1             IF DESNO=DESNO2 THEN                                   D0002970
279     1             DO; IF B THEN B='0'B;                                  D0002980
282     1    1            C4FLAG='0'B;                                       D0002990
283     1    1            PUT EDIT(DESNO2,'TRANSACTION',TRANS)(COL(1),A,2(X(1),A));  D0003000
284     1    1            IF TRANS='01' THEN CALL DESADM; ELSE               D0003010
286     1    1            IF TRANS='11' | TRANS='15' THEN CALL DESCOCH; ELSE D0003020
288     1    1             IF TRANS='12' | TRANS='99' THEN CALL DESWOL;      D0003030
290     1    1            PUT EDIT('COMPLETED')(X(1),A);                     D0003040
291     1    1            GOTO L42;                                          D0003050
292     1    1        END; ELSE                                              D0003060
293     1             DO; IF ¬B THEN                                         D0003070
295     1    1            DO; DESNO3=DESNO;                                  D0003080
297     1    2                B='1'B;                                        D0003090
298     1    2            END; ELSE                                          D0003100
299     1    1            DO; IF DESNO=DESNO3 THEN                           D0003110
301     1    2                DO; PUT EDIT(DESNO2,'CANNOT BE FOUND....DATA IGNORED')  D0003120
                                 (COL(1),A,X(1),A);                          D0003130
303     1    3                    B='0'B;                                    D0003140
304     1    3                    GOTO L42;                                  D0003150
305     1    3                END;                                           D0003160
306     1    2            END;                                               D0003170
307     1    1            GOTO L41;                                          D0003180
308     1    1        END;                                                   D0003190
                 /*-----------------------------------------------------------*/D0003200
309     1        L49: PUT PAGE;                                              D0003210
310     1             CLOSE FILE(STUDENT);                                   D0003220
311     1             CLOSE FILE(DIC);                                       D0003230
312     1        END;                                                        D0003240
                                                                             D0003250
```

Dataset Limited

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
2		ADIP	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER
2		ADIPC	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 184,187
2	*********	AH4	(2,3)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
3	*********	AK	(3,26)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 172,172
3		ALPHA	STATIC,EXTERNAL,UNALIGNED,STRING(26),CHARACTER 169
9		AMON	PARAMETER,UNALIGNED,STRING(2),CHARACTER 8,10,12,14,16
2		ANNPROG	IN RECORD,BASED(PTR),STRUCTURE
2		AP	IN ANNPROG IN RECORD,BASED(PTR),UNALIGNED,STRING(802),CHARACTER
58		AQT	(20)AUTOMATIC,UNALIGNED,STRING(4),CHARACTER 62,72,74,75,80,131,133,135,139
3		ATAB	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 131
3	*********	ATAB1	(*)EXTERNAL,CONTROLLED,ALIGNED,BINARY,FIXED(15,0) 134,135
2		ATO	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 184,191
2		AUTH	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(19),CHARACTER 168,173,179
2		AUTHNO	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 167
3		AUTR	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 173
3		AUTR1	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 175
4		B	AUTOMATIC,UNALIGNED,STRING(1),BIT 261,280,281,294,297,303
57		B11	AUTOMATIC,UNALIGNED,STRING(1),BIT 65,121,123
57		B61	AUTOMATIC,UNALIGNED,STRING(1),BIT

Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               64,115,117

2                BE                            IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               218,219

230              C                             (2)AUTOMATIC,UNALIGNED,STRING(80),CHARACTER
                                               231,232,233,233,234,235,236,237,238,242,246,248

202              C                             AUTOMATIC,UNALIGNED,STRING(80),CHARACTER
                                               203,204,208,209,210,212,215,217,219,220,221,223,224,225,226,227

144              C                             (5)AUTOMATIC,UNALIGNED,STRING(80),CHARACTER
                                               145,151,153,153,155,155,157,158,159,160,161,162,163,165,166,167,175
                                               180,181,182,183,187,188,189,190,191,193,195,196,198,199,199,199,199

57               C                             (5)PARAMETER,UNALIGNED,STRING(80),CHARACTER
                                               56,103,108,109,110,111,118,124,127,128,134,135,141

4                C4FLAG                        AUTOMATIC,UNALIGNED,STRING(1),BIT
                                               112,186,197,282

57               CAQ                           AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(99)
                                               60,71,79,79,80,129,141

144              CFNAME                        AUTOMATIC,UNALIGNED,STRING(31),CHARACTER
                                               147,150,159

144              CNAME                         AUTOMATIC,UNALIGNED,STRING(18),CHARACTER
                                               146,149,158

2                CODES                         (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER

5                CON                           EXTERNAL,ENTRY,DECIMAL,FLOAT(SINGLE)
                                               264

2                CONF                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

57               COQ                           AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(99)
                                               59,84,88,88,89,128

2                COURSE                        IN RECORD,BASED(PTR),STRUCTURE

57               CSEQ                          AUTOMATIC,UNALIGNED,DECIMAL,PICTURE(99)
                                               61,98,98,127

1                D                             ENTRY,DECIMAL,FLOAT(SINGLE)

2                DATT                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER

2                DC                            (4,24:64)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),
                                               CHARACTER
                                               32,34,41,42,42,42,42,42,43,43,44,44,46,47,49,52,53,54

2                DC1                           (4,26:30)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(2),
```

Dataset Limited

```
DCL NO.        IDENTIFIER                ATTRIBUTES AND REFERENCES

                                         CHARACTER
                                         36,37,38,39

2              DC46                      (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                         50

2              DC49                      (4)IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER
                                         50

29             DCOUR                     ENTRY,DECIMAL,FLOAT(SINGLE)
                                         195,242

2              DDIPC                     IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                         184,188

2              DEG                       (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(3),CHARACTER

4              DES                       AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                         254,263

143            DESADM                    ENTRY,DECIMAL,FLOAT(SINGLE)
                                         285

229            DESCOCH                   ENTRY,DECIMAL,FLOAT(SINGLE)
                                         287

2              DESNO                     IN RECORD,BASED(PTR),UNALIGNED,STRING(8),CHARACTER
                                         21,21,139,179,278,296,300

4              DESNO2                    AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                         271,274,278,283,302

4              DESNO3                    AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                         296,300

201            DESWOL                    ENTRY,DECIMAL,FLOAT(SINGLE)
                                         289

7              DIC                       FILE,EXTERNAL
                                         250,263,271,311

19             DKEY                      ENTRY,DECIMAL,FLOAT(SINGLE)
                                         151,204,232

2              DOB                       IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                         25,26,26

2              DOBC                      IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                         156

2      ********* DSCALE                  (2)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

2              EDUCATION                 IN RECORD,BASED(PTR),STRUCTURE
```

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

144    *********  ESNAME                        AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                                148,149,150

4                 FILN1                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                                252,262,268

4                 FILN2                         AUTOMATIC,UNALIGNED,STRING(8),CHARACTER
                                                252,264

2                 FINALS                        IN RECORD,BASED(PTR),STRUCTURE

144               FIRST                         AUTOMATIC,UNALIGNED,STRING(1),CHARACTER
                                                168,169

2      *********  FK                            (100,2)IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED
                                                (15,0)

2                 FPOS                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER

2                 FPT                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

2                 FT                            (2,2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER

2                 GENERAL                       IN RECORD,BASED(PTR),STRUCTURE

2                 HPCD                          (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER

230    *********  I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                                240,241,242

202    *********  I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)

144    *********  I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                                152,153,172,173,175,194,195

57     *********  I                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                                67,68,94,95,129,131,133,135,139

30     *********  I                             PARAMETER,ALIGNED,BINARY,FIXED(15,0)
                                                29,32,34,36,37,38,39,41,42,42,42,42,42,43,43,44,44,46,47,49,50,50,52
                                                53,54

                  INDEX                         GENERIC,BUILT-IN FUNCTION
                                                27,148,169

2                 INTEL                         IN RECORD,BASED(PTR),STRUCTURE

2                 INTELB                        IN INTEL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER

57     *********  J                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                                71,72,74,75,84,85,94,95,95,95,130,131,134,135

115               L1                            STATEMENT LABEL CONSTANT
                                                90,96,99,104
```

Dataset Limited

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
214	L1	STATEMENT LABEL CONSTANT 206
126	L2	STATEMENT LABEL CONSTANT 76,86,114,119
225	L2	STATEMENT LABEL CONSTANT 213
140	L3	STATEMENT LABEL CONSTANT 136
277	L41	STATEMENT LABEL CONSTANT 269,307
271	L42	STATEMENT LABEL CONSTANT 275,291,304
309	L49	STATEMENT LABEL CONSTANT 251,257
180	L6	STATEMENT LABEL CONSTANT 171,176
2	LE	IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 241
2	********* LEAP	IN ANNPROG IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
2	LEN	IN COURSE IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 40
2	MART	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
230	MON	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 245,246
202	MON	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 207,208,211,212,222,223
144	MON	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 164,165
30	MON	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 36,37,38,39
20	MON	AUTOMATIC,UNALIGNED,STRING(1),CHARACTER 25,26
3	********* NALC	STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 130
2	NAME	IN RECORD,BASED(PTR),UNALIGNED,STRING(40),CHARACTER

Dataset Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               27,27,148,149,150

8                NMON                          ENTRY,STRING(1),CHARACTER
                                               25,36,38,164,207,211,222,245

2                OQ                            (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(30),CHARACTER

2                OQC                           (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(4),CHARACTER
                                               184,184,189,190

58               OQT                           (30)AUTOMATIC,UNALIGNED,STRING(3),CHARACTER
                                               63,85,89

2                OV25                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               180

2      ********* PF16                          (2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

2                PNAME                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(17),CHARACTER
                                               163

2                POCC                          IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER

144    ********* POS                           AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               169,170,172,172

                 PTR                           AUTOMATIC,ALIGNED,POINTER
                                               21,21,22,25,26,26,27,27,32,34,36,37,38,39,40,41,42,42,42,42,42,43,43
                                               44,44,44,46,47,49,50,50,52,52,53,54,66,68,95,95,95,95,139,148,149
                                               150,156,163,164,165,165,167,168,173,179,179,180,184,184,184,184,184
                                               187,188,189,190,191,207,208,208,210,211,212,214,215,216,217,218,219
                                               221,222,223,223,241,245,246,246,277,278,296,300

57     ********* Q                             AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)
                                               66,67,94

58               QC                            AUTOMATIC,DEFINED,UNALIGNED,STRING(3),CHARACTER
                                               72,75,80,85,89,101,101,109,111,115,121

58               QG                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               74,75,80,82,92,106

58               QP                            AUTOMATIC,DEFINED,UNALIGNED,STRING(1),CHARACTER
                                               69,82,82,92,101,106

58               QT                            AUTOMATIC,DEFINED,UNALIGNED,STRING(3),CHARACTER
                                               69,82,82,92,101

2                QUAL                          (30)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(10),CHARACTER
                                               68,95,95,95,95

58               QUAL2                         AUTOMATIC,UNALIGNED,STRING(10),CHARACTER
                                               68
```

Dataset Limited

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
56	QUALC	ENTRY,DECIMAL,FLOAT(SINGLE) 182
2	QUALNO	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER 66
30	RC	PARAMETER,UNALIGNED,STRING(80),CHARACTER 29,31,33,35,37,39,41,42,43,46,48,49,50,53,54
20	RC	PARAMETER,UNALIGNED,STRING(80),CHARACTER 19,21,23,24,26,27
2	RECORD	BASED(PTR),STRUCTURE
2	REG	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
2	REL	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
	SCARDS	FILE,EXTERNAL 252,253
2	SCH	(10)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(5),CHARACTER
2	SCHNO	IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER
2	SET	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
2	SEX	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 22
2	SOCC	IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
	SPRINT	FILE,EXTERNAL 139,179,256,259,260,274,283,290,302,309
	SPUNCH	FILE,EXTERNAL 198,199,227,248
2	********* STEN	(2,16)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)
6	STUDENT	FILE,EXTERNAL,RECORD,ENVIRONMENT(F(2416)) 262,265,267,268,277,310
2	SUB	(2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(36),CHARACTER
2	SUBCAR	IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER 221
	SUBSTR	GENERIC,BUILT-IN FUNCTION 16,21,21,21,23,24,25,26,26,26,27,27,31,33,35,37,39,41,42,43,44,46,48 49,50,52,53,54,72,74,95,95,95,95,103,108,109,110,111,118,124,127,128 131,133,134,135,135,139,141,149,150,153,153,155,155,157,158,159,160 161,162,163,164,165,165,165,166,167,168,175,180,181,183,187,188,189 190,191,193,196,207,208,208,208,209,210,210,211,212,212,215,217,219

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

                                               220,221,222,223,223,223,224,225,226,233,233,234,235,236,237,238,245
                                               246,246,246

2                TC                            IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               214,215

2                TITLE                         IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER

4                TRANS                         AUTOMATIC,UNALIGNED,STRING(2),CHARACTER
                                               205,235,239,243,271,272,272,272,272,272,283,284,286,286,288,288

2                TTFT                          (2,2)IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(6),CHARACTER
                                               164,165,165,207,208,208,210,211,212,222,223,223,245,246,246

2                UNCOL                         (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER

2                UNOTH                         (2)IN EDUCATION IN RECORD,BASED(PTR),UNALIGNED,STRING(21),CHARACTER
                                               44,52

2      ********* VALUES                        (2,6)IN INTEL IN RECORD,BASED(PTR),ALIGNED,BINARY,FIXED(15,0)

2                WS                            IN FINALS IN RECORD,BASED(PTR),UNALIGNED,STRING(1),CHARACTER
                                               216,217

2                YC                            IN GENERAL IN RECORD,BASED(PTR),UNALIGNED,STRING(2),CHARACTER

4                YE                            AUTOMATIC,UNALIGNED,STRING(4),CHARACTER
                                               252,259
```

Dataset Limited

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	AK	156
58	AQT	80
144	C	400
230	C	160
58	OQT	90
2	RECORD	2416

Declassified

STORAGE REQUIREMENTS.

```
THE STORAGE AREA FOR THE PROCEDURE LABELLED D IS 1128 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED NMON IS 928 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED DKEY IS 1040 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED DCOUR IS 1024 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED QUALC IS 1384 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED DESADM IS 1604 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED DESWOL IS 1140 BYTES LONG.
THE STORAGE AREA FOR THE PROCEDURE LABELLED DESCOCH IS 1224 BYTES LONG.
THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 250 IS 896 BYTES LONG.
THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 266 IS 928 BYTES LONG.
THE PROGRAM CSECT IS NAMED D        AND IS 27588 BYTES LONG.
THE STATIC  CSECT IS NAMED ******DA AND IS 4383 BYTES LONG.

*STATISTICS*     SOURCE RECORDS =      325,PROG TEXT STMNTS =      312,OBJECT BYTES =    27588
```

Dataset Limited

COMPILER DIAGNOSTICS.

WARNINGS.

IEM0227I NO FILE/STRING OPTION SPECIFIED IN ONE OR MORE GET/PUT STATEMENTS. SYSIN/SYSPRINT HAS BEEN ASSUMED IN EACH CASE.

IEM0764I ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY ARE FLAGGED '**********' IN THE XREF/ATR LIST.

IEM1790I DATA CONVERSIONS WILL BE DONE BY SUBROUTINE CALL IN THE FOLLOWING STATEMENTS 66, 127, 128, 141, 241.

END OF DIAGNOSTICS.

COMPILE TIME 35.80 SECS

ELAPSED TIME 112.82 SECS

```
THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                 CHAR60
                                                               NOMACRO
                                                                 SOURCE2
                                                               NOMACDCK
                                                                 COMP
                                                                 SOURCE
                                                                 ATR
                                                                 XREF
                                                               NOEXTREF
                                                               NOLIST
                                                               NOLOAD
                                                                 DECK
                                                                 FLAGW
                                                                 STMT
                                                                 SIZE=010P
                                                                 LINECNT=060
                                                                 OPT=01
                                                                 SORMGIN=(001,072)
                                                                 NEST
                                                                 OPLIST

*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTREF,NOLIST,NOLOAD,
*OPTIONS IN EFFECT*     DECK,FLAGW,STMT,SIZE=010P,LINECNT=060,OPT=01,SORMGIN=(001,072),
*OPTIONS IN EFFECT*     NEST,OPLIST
```

Dataset Limited

```
STMT LEVEL NEST
                                                                          D0003270
 1                CCN:PROC(FILN2);                                        D0003280
 2      1              DCL FILN2 CHAR(8);                                 D0003290
 3      1              DCL (I,J,K) FIXED BINARY, AK(3,26) FIXED BINARY EXTERNAL,  D0003300
                       (AUTR(*) CHAR(19), AUTR1(*) CHAR(3), ATAB(*) CHAR(3), ATAB1(*)  D0003310
                       FIXED BINARY) CONTROLLED EXTERNAL, (ALPHA CHAR(26), NALC FIXED  D0003320
                       BINARY) EXTERNAL;                                  D0003330
 4      1              DCL (SCK(9) CHAR(60), PSOK(5) CHAR(65), STK(6) CHAR(50),  D0003340
                       MK(3) CHAR(12), OK(10) CHAR(35), RK(5) CHAR(30), GRC CHAR(11),  D0003350
                       GRK(11) CHAR(30),OAK(4) CHAR(40), ORK(6) CHAR(30), OPSOK(6)  D0003360
                       CHAR(40)) VARYING EXTERNAL;                        D0003370
 5      1              DCL NORM(2,16,10) FIXED BINARY EXTERNAL;           D0003380
 6      1              DCL NORM2(2,16,2) FLOAT DECIMAL EXTERNAL;          D0003390
                  /*-----------------------------------------------------*/D0003400
 7      1              DCL CONST FILE;                                    D0003410
                  /*-----------------------------------------------------*/D0003420
 8      1              OPEN FILE(CONST) INPUT TITLE(FILN2);               D0003430
 9      1              DO I=1 TO 26;                                      D0003440
10      1     1           GET FILE(CONST) LIST(AK(1,I));                  D0003450
11      1     1        END;                                               D0003460
12      1              AK(2,1)=1;     AK(3,1)=AK(2,1)+AK(1,1)-1;          D0003470
14      1              DO I=2 TO 26;                                      D0003480
15      1     1           AK(2,I)=AK(3,I-1)+1;                            D0003490
16      1     1           AK(3,I)=AK(1,I)+AK(2,I)-1;                      D0003500
17      1     1        END;                                               D0003510
18      1              N=SUM(AK(1,*));                                    D0003520
19      1              ALLOCATE AUTR(N),AUTR1(N);                         D0003530
20      1              DO K=1 TO 26;                                      D0003540
21      1     1          DO I=AK(2,K) TO AK(3,K);                         D0003550
22      1     2           GET FILE(CONST) EDIT(AUTR(I),AUTR1(I))          D0003560
                          (COL(1),A(19),COL(40),A(3));                    D0003570
23      1     2           END;                                            D0003580
24      1     1        END;                                               D0003590
25      1              GET FILE(CONST) EDIT((OAK(K) DO K=1 TO 4))(4(COL(1),A(40)));  D0003600
26      1              GET FILE(CONST) EDIT((ORK(K) DO K=1 TO 6))(6(COL(1),A(30)));  D0003610
27      1              GET FILE(CONST) EDIT((OPSOK(K) DO K=1 TO 6))(6(COL(1),A(40)));  D0003620
28      1              GET FILE(CONST) EDIT((SCK(I) DO I=1 TO 9))(9(COL(1),A(60)));  D0003630
29      1              GET FILE(CONST) EDIT((PSOK(I) DO I=1 TO 5))(5(COL(1),A(65)));  D0003640
30      1              GET FILE(CONST) EDIT((STK(I) DO I=1 TO 6))(6(COL(1),A(50)));  D0003650
31      1              GET FILE(CONST) EDIT((MK(I) DO I=1 TO 3))(3(COL(1),A(12)));  D0003660
32      1              GET FILE(CONST) EDIT((OK(I) DO I=1 TO 10))(10(COL(1),A(35)));  D0003670
33      1              GET FILE(CONST) EDIT((RK(I) DO I=1 TO 5))(5(COL(1),A(30)));  D0003680
34      1              GET FILE(CONST) EDIT((GRK(I) DO I=1 TO 11))(11(COL(1),A(30)));  D0003690
35      1              GET FILE(CONST) EDIT(ALPHA)(COL(1),A(26));         D0003700
36      1              GRC=SUBSTR(ALPHA,1,11);                            D0003710
37      1              GET FILE(CONST) EDIT(NALC)(COL(1),F(3));           D0003720
38      1              ALLOCATE ATAB(NALC),ATAB1(NALC);                   D0003730
39      1              GET FILE(CONST) SKIP;                              D0003740
40      1              GET FILE(CONST) EDIT((ATAB(I),ATAB1(I) DO I=1 TO NALC))  D0003750
                       (X(2),A(3),X(1),F(2));                             D0003760
41      1              DO I=1 TO 2;                                       D0003770
42      1     1           DO J=1 TO 16;                                   D0003780
43      1     2              GET FILE(CONST) LIST((NORM(I,J,K) DO K=1 TO 10));  D0003790
44      1     2              GET FILE(CONST) LIST((NORM2(I,J,K) DO K=1 TO 2));  D0003800
45      1     2           END;                                            D0003810
46      1     1        END;                                               D0003820
```

```
STMT LEVEL NEST

 47     1              CLOSE FILE(CONST);                                  D0003830
 48     1         END;                                                     D0003840
```

Phase Limited

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
3	*********	AK	(3,26)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 10,12,13,13,13,15,15,16,16,16,18,21,21
3		ALPHA	STATIC,EXTERNAL,UNALIGNED,STRING(26),CHARACTER 35,36
3		ATAB	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 38,40
3	*********	ATAB1	(*)EXTERNAL,CONTROLLED,ALIGNED,BINARY,FIXED(15,0) 38,40
3		AUTR	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 19,22
3		AUTR1	(*)EXTERNAL,CONTROLLED,UNALIGNED,STRING(*),CHARACTER 19,22
1		CON	ENTRY,DECIMAL,FLOAT(SINGLE)
7		CONST	FILE,EXTERNAL 8,10,22,25,26,27,28,29,30,31,32,33,34,35,37,39,40,43,44,47
2		FILN2	PARAMETER,UNALIGNED,STRING(8),CHARACTER 1,8
4		GRC	STATIC,EXTERNAL,UNALIGNED,STRING(11),CHARACTER,VARYING 36
4		GRK	(11)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING 34
3	*********	I	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 9,10,14,15,15,16,16,16,21,22,22,28,28,29,29,30,30,31,31,32,32,33,33 34,34,40,40,40,41,43,44
3	*********	J	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 42,43,44
3	*********	K	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 20,21,21,25,25,26,26,27,27,43,43,44,44
4		MK	(3)STATIC,EXTERNAL,UNALIGNED,STRING(12),CHARACTER,VARYING 31
	*********	N	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 18,19,19
3	*********	NALC	STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0) 37,38,38,40

Datapoint Limited

```
DCL NO.          IDENTIFIER                    ATTRIBUTES AND REFERENCES

5     ********   NORM                          (2,16,10)STATIC,EXTERNAL,ALIGNED,BINARY,FIXED(15,0)
                                                43

6                NORM2                         (2,16,2)STATIC,EXTERNAL,ALIGNED,DECIMAL,FLOAT(SINGLE)
                                                44

4                OAK                           (4)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                                25

4                OK                            (10)STATIC,EXTERNAL,UNALIGNED,STRING(35),CHARACTER,VARYING
                                                32

4                OPSOK                         (6)STATIC,EXTERNAL,UNALIGNED,STRING(40),CHARACTER,VARYING
                                                27

4                ORK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                                26

4                PSOK                          (5)STATIC,EXTERNAL,UNALIGNED,STRING(65),CHARACTER,VARYING
                                                29

4                RK                            (5)STATIC,EXTERNAL,UNALIGNED,STRING(30),CHARACTER,VARYING
                                                33

4                SCK                           (9)STATIC,EXTERNAL,UNALIGNED,STRING(60),CHARACTER,VARYING
                                                28

4                STK                           (6)STATIC,EXTERNAL,UNALIGNED,STRING(50),CHARACTER,VARYING
                                                30

                 SUBSTP                        GENERIC,BUILT-IN FUNCTION
                                                36

                 SUM                           GENERIC,BUILT-IN FUNCTION
                                                18
```

Dataset Limited

AGGREGATE LENGTH TABLE

STATEMENT NO.	IDENTIFIER	LENGTH IN BYTES
3	AK	156
38	ATAB	ADJUSTABLE
38	ATAB1	ADJUSTABLE
19	AUTR	ADJUSTABLE
19	AUTR1	ADJUSTABLE
4	GRK	330
4	MK	36
5	NORM	640
6	NORM2	256
4	OAK	160
4	OK	350
4	OPSOK	240
4	ORK	180
4	PSOK	325
4	RK	150
4	SCK	540
4	STK	300

STORAGE REQUIREMENTS.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED CON IS 388 BYTES LONG.

THE PROGRAM CSECT IS NAMED CON AND IS 4294 BYTES LONG.

THE STATIC CSECT IS NAMED ****CONA AND IS 1044 BYTES LONG.

STATISTICS SOURCE RECORDS = 58,PROG TEXT STMNTS = 48,OBJECT BYTES = 4294

Dataset Limited

```
COMPILER DIAGNOSTICS.

WARNINGS.
   IEM0526I      1   OPTION MAIN HAS NOT BEEN SPECIFIED FOR THE EXTERNAL PROCEDURE, STATEMENT NUMBER 1
   IEM0764I          ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY
                     ARE FLAGGED '*********' IN THE XREF/ATR LIST.

END OF DIAGNOSTICS.

COMPILE TIME        10.96 SECS

ELAPSED TIME        37.85 SECS
EXECUTION TERMINATED
```

7. THE DATA CONSTANTS FILE.

7.1 FILE LISTING

THIS SECTION CONTAINS A LISTING OF THE DATA CONSTANTS FILE.

```
 1    20 44 35 15 14  9 24 19 14  4  8 19 26 25  3 18  1 13 54 18  7  6 17  0  6  3
 2   ALDERNEY                                   592
 3   ANGLESEY                                   550
 4   ABU DHABI                                  764
 5   ABYSSINIA                                  648
 6   ADEN                                       601
 7   AFARS/ISSAS TERRITORY                      749
 8   AFGHANISTAN                                602
 9   AJMAN                                      764
10   ALBANIA                                    603
11   ALGERIA                                    604
12   ANDORRA                                    605
13   ANGOLA                                     794
14   ANGUILLA                                   736
15   ANTIGUA                                    607
16   ANTILLES (NETHERLANDS)                     637
17   ARGENTINE                                  608
18   ASCENSION                                  735
19   AUSTRALIA                                  609
20   AUSTRIA                                    610
21   AZORES                                     728
22   BARKING                                    301
23   BARNET                                     302
24   BARNSLEY                                   451
25   BARROW-IN-FURNESS                          452
26   BATH                                       453
27   BEDFORDSHIRE                               401
28   BERKSHIRE                                  402
29   BEXLEY                                     303
30   BIRKENHEAD                                 454
31   BIRMINGHAM                                 455
32   BLACKBURN                                  456
33   BLACKPOOL                                  457
34   BOLTON                                     458
35   BOOTLE                                     459
36   BOURNEMOUTH                                460
37   BRADFORD                                   461
38   BRECONSHIRE                                551
39   BRENT                                      304
40   BRIGHTON                                   462
41   BRISTOL                                    463
42   BROMLEY                                    305
43   BUCKINGHAMSHIRE                            403
44   BURNLEY                                    464
45   BURTON-ON-TRENT                            465
46   BURY                                       466
47   BAHAMAS                                    611
48   BALEARIC IS.                               751
49   BAHREIN                                    612
50   BANGLADESH                                 787
```

51	BARBADOS	613
52	BASUTOLAND	690
53	BECHUANALAND	618
54	BELGIUM	614
55	BERMUDA	615
56	BHUTAN	616
57	BOLIVIA	617
58	BORNEO NORTH	739
59	BORNEO SOUTH	673
60	BOTSWANA	618
61	BRAZIL	619
62	BRUNEI	620
63	BULGARIA	621
64	BURMA	622
65	BURUNDI	623
66	CAERNARVONSHIRE	552
67	CAMBRIDGESHIRE/ELY	404
68	CANTERBURY	467
69	CARDIFF	563
70	CARDIGANSHIRE	553
71	CARLISLE	468
72	CARMARTHENSHIRE	554
73	CHESHIRE	405
74	CHESTER	469
75	CHANNEL ISLES	591
76	CORNWALL	406
77	COVENTRY	470
78	CROYDON	306
79	CUMBERLAND	407
80	CAMBODIA	624
81	CAMEROON	625
82	CANADA	626
83	CANARY ISLANDS	751
84	CAPE VERDE IS.	788
85	CAYMAN IS.	789
86	CENTRAL AFRICAN REPUBLIC	627
87	CEYLON	628
88	CHAD	629
89	CHILE	630
90	CHINA, COMMUNIST	631
91	CHINA, NATIONALIST	652
92	COLOMBIA	632
93	CONGO DEMOCRATIC REP.	633
94	CONGO (BRAZZAVILLE)	634
95	COSTA RICA	635
96	CRETE	661
97	CUBA	636
98	CURACAO (DUTCH)	637
99	CYPRUS	638
100	CZECHOSLOVAKIA	639
101	DARLINGTON	472
102	DENBIGHSHIRE	555
103	DERBY	473
104	DERBYSHIRE	408
105	DEVON	409
106	DEWSBURY	474
107	DONCASTER	475
108	DORSET	410
109	DUDLEY	476
110	DURHAM	411

111	DAHOMEY	640
112	DENMARK	641
113	DOMINICA	642
114	DOMINICAN REPUBLIC	643
115	DUBIA	764
116	EALING	307
117	EASTBOURNE	477
118	ENFIELD	308
119	ESSEX	412
120	EXETER	479
121	ECUADOR	645
122	EGYPT	768
123	EIRE	676
124	EL SALVADOR	646
125	ELLICE IS.	660
126	EQUATORIAL GUINEA	799
127	ERITREA	648
128	ESTONIA	772
129	ETHIOPIA	648
130	FLINTSHIRE	556
131	FALKLAND IS.	649
132	FIJI	650
133	FINLAND	651
134	FORMOSA	652
135	FRANCE	653
136	FRENCH GUIANA	791
137	FRENCH WEST INDIES	792
138	FUJAIRAH	764
139	GATESHEAD	480
140	GLAMORGAN	557
141	GLOUCESTER	481
142	GLOUCESTERSHIRE	413
143	GREAT YARMOUTH	482
144	GRIMSBY	483
145	GUERNSEY	593
146	GABOON	654
147	GAMBIA	655
148	GERMANY, WEST	656
149	GERMANY, EAST	657
150	GHANA	658
151	GIBRALTAR	659
152	GILBERT/ELLIS IS.	660
153	GREECE	661
154	GREENLAND	641
155	GRENADA	662
156	GUADELOUPE	792
157	GUATAMALA	663
158	GUIANA (BRITISH)	665
159	GUIANA (DUTCH)	753
160	GUIANA (FRENCH)	791
161	GUINEA	664
162	GUYANA	665
163	HALIFAX	484
164	HAMPSHIRE	414
165	HARINGEY	309
166	HARROW	310
167	HARTLEPOOL	533
168	HASTINGS	485
169	HAVERING	311
170	HEREFORDSHIRE	415

231	MIDDLESBROUGH	494
232	MONMOUTHSHIRE	559
233	MONTGOMERYSHIRE	560
234	MACAO	694
235	MADAGASCAR	695
236	MADEIRA	728
237	MALAGASY	695
238	MALAWI	696
239	MALAYA	697
240	MALDIVE IS.	793
241	MALI	699
242	MALTA	700
243	MARTINIQUE	792
244	MAURITANIA	701
245	MAURITIUS	702
246	MEXICO	703
247	MONACO	653
248	MONGOLIA, OUTER	704
249	MONTSERRAT	705
250	MOROCCO	706
251	MOZAMBIQUE	707
252	MUSCAT/OMAN	708
253	NEWCASTLE	495
254	NEWHAM	316
255	NEWPORT	565
256	NORFOLK	429
257	NORTHAMPTON	496
258	NORTHAMPTONSHIRE	430
259	NORTHERN IRELAND	099
260	NORTHUMBERLAND	431
261	NORWICH	497
262	NOTTINGHAM	498
263	NOTTINGHAMSHIRE	432
264	NAMIBIA	798
265	NEPAL	709
266	NETHERLANDS	710
267	NETHERLANDS ANTILLES	637
268	NEW CALEDONIA	711
269	NEW GUINEA	712
270	NEW HEBRIDES	713
271	NEW ZEALAND	714
272	NICARAGUA	715
273	NIGER	716
274	NIGERIA	717
275	NORWAY	718
276	NOT KNOWN	782
277	NYASALAND	696
278	OLDHAM	499
279	OXFORD	500
280	OXFORDSHIRE	433
281	PEMBROKESHIRE	561
282	PLYMOUTH	501
283	PORTSMOUTH	502
284	PRESTON	503
285	PAKISTAN	721
286	PANAMA	722
287	PAPUA	723
288	PARAGUAY	724
289	PERSIA	674
290	PERU	725

291	PHILIPPINES	726
292	POLAND	727
293	PORTUGAL	728
294	PORTUGUESE EAST AFRICA	707
295	PORTUGUESE WEST AFRICA	794
296	PORTUGUESE GUINEA	794
297	PORTUGUESE TIMOR	786
298	PUERTO RICO	730
299	QUATAR	731
300	RADNORSHIRE	562
301	REDBRIDGE	317
302	RICHMOND-UPON-THAMES	318
303	READING	504
304	ROCHDALE	505
305	ROTHERHAM	506
306	RUTLAND	435
307	RAS AL KHAIMAH	764
308	REUNION	795
309	RHODESIA, NORTHERN	781
310	RHODESIA	732
311	RUMANIA	733
312	RWANDA	734
313	SALFORD	508
314	SALOP	436
315	SHEFFIELD	509
316	SOLIHULL	530
317	SOMERSET	437
318	SOUTHAMPTON	511
319	SOUTHEND-ON-SEA	512
320	SOUTHPORT	513
321	SOUTH SHIELDS	514
322	STAFFORDSHIRE	438
323	ST. HELENS	507
324	STOCKPORT	515
325	STOKE-ON-TRENT	516
326	SUFFOLK, E	439
327	SUFFOLK, W	440
328	SUNDERLAND	517
329	SURREY	441
330	SUTTON	319
331	SUSSEX, E	442
332	SUSSEX, W	443
333	SWANSEA	566
334	ST. HELENA	735
335	ST. KITTS-NEVIS	736
336	ST. LUCIA	737
337	ST. TOME AND PRINCIPE	794
338	ST. VINCENT	738
339	SABAH	739
340	SALVADOR	646
341	SAMOA (U.S.)	796
342	SAN MARINO	678
343	SARAWAK	742
344	SAUDI ARABIA	743
345	SCOTLAND	299
346	SENEGAL	785
347	SEYCHELLES	744
348	SHARJAH	764
349	SIAM	760
350	SIERRA LEONE	745

351	SIKKIM	797
352	SINGAPORE	746
353	SOLOMON IS.	747
354	SOMALI REPUBLIC	748
355	SOMALILAND (FRENCH)	749
356	SOUTH AFRICA	750
357	SOUTH WEST AFRICA	798
358	SOUTH YEMEN REP.	601
359	SPAIN	751
360	STATELESS	783
361	SUDAN	752
362	SURINAM	753
363	SWAZILAND	754
364	SWEDEN	755
365	SWITZERLAND	756
366	SYRIAN ARAB REP.	757
367	TEESSIDE	535
368	TORBAY	534
369	TYNEMOUTH	518
370	TAIWAN	652
371	TANGANYIKA	759
372	TANGIERS	706
373	TANZANIA	759
374	TCHAD	629
375	THAILAND	760
376	TIBET	761
377	TOGO	762
378	TONGA	784
379	TRINIDAD AND TOBAGO	763
380	TRISTAN DA CUNHA	735
381	TRUCIAL STATES	764
382	TUNISIA	765
383	TURKEY	766
384	TURKS AND CAICOS IS.	799
385	UGANDA	767
386	UMMALQUAIWAIN	764
387	UNITED ARAB REPUBLIC	768
388	UPPER VOLTA	769
389	URUGUAY	770
390	U.S.A.	771
391	U.S.S.R.	772
392	VATICAN CITY	678
393	VENEZUELA	773
394	VIETNAM, SOUTH	774
395	VIETNAM, NORTH	775
396	VIRGIN IS. (BRITISH)	776
397	VIRGIN IS. (U.S.)	800
398	WAKEFIELD	519
399	WALLASEY	520
400	WALSALL	521
401	WALTHAM FOREST	320
402	WARLEY	532
403	WARRINGTON	522
404	WARWICKSHIRE	444
405	WEST BROMWICH	523
406	WESTMORLAND	445
407	WIGAN	526
408	WILTSHIRE	446
409	WOLVERHAMPTON	527
410	WORCESTER	528

411	WORCESTERSHIRE	447
412	WEST INDIES	777
413	WESTERN SAMOA	741
414	WINDWARD IS.	778
415	YORK	529
416	YORKS. E RIDING	448
417	YORKS. N RIDING	449
418	YORKS. W RIDING	450
419	YEMEN	779
420	YUGOSLAVIA	780
421	ZAIRE	633
422	ZAMBIA	781
423	ZANZIBAR	759
424	METROPOLITAN (LARGE CITY CENTRE)	
425	SUBURBAN (OUTSKIRTS OF LARGE CITY)	
426	RURAL URBAN (SMALL TOWN OR VILLAGE)	
427	RURAL	
428	C. OF E.	
429	R.C.	
430	NONCONFORMIST	
431	JEWISH	
432	AGNOSTIC/ATHEIST	
433	HUMANIST/OTHER	
434	PROFESSIONAL AND HIGH ADMINISTRATIVE	
435	MANAGERIAL AND EXECUTIVE	
436	NON-MANUAL, HIGHER GRADE	
437	NON-MANUAL, LOWER GRADE	
438	SKILLED MANUAL	
439	SEMI-SKILLED MANUAL	
440	INDEPENDENT OR DIRECT GRANT SCHOOL...MIXED	
441	INDEPENDENT OR DIRECT GRANT SCHOOL...SINGLE SEX	
442	GRAMMAR...MIXED	
443	GRAMMAR...SINGLE SEX	
444	COMPREHENSIVE...MIXED	
445	COMPREHENSIVE ...SINGLE SEX	
446	SEC. MODERN SCHOOL...MIXED	
447	SEC. MODERN SCHOOL...SINGLE SEX	
448	COLL. OF FURTHER EDUCATION/TECHNICAL COLL	
449	HIGHER ADMINISTRATIVE PROFESSIONAL AND MANAGERIAL	
450	FARMERS, INTERMEDIATE ADMINISTRATIVE PROFESSIONAL & MANAGERIAL	
451	ROUTINE NON-MANUAL AND SERVICE. MANUAL SKILLED	
452	SEMI-SKILLED WORKERS	
453	UNSKILLED WORKERS	
454	ROADSIDE OR HAMLET I.E. LESS THAN 150 PEOPLE	
455	VILLAGE OR SMALL TOWN I.E. 150 TO 2,500 PEOPLE	
456	TOWN I.E. 2,500 TO 10,000 PEOPLE	
457	LARGE TOWN I.E. 10,000 TO 500,000 PEOPLE	
458	METROPOLIS I.E. OVER 500,000 PEOPLE	
459	N/A	
460	SINGLE	
461	MARRIED	
462	DIVORCED/WIDOWED	
463	TEACHING OR YOUTH SERVICE	
464	CENTRAL OR LOCAL GOVERNMENT	
465	H.M. FORCES	
466	POLICE	
467	NURSING	
468	OTHER MEDICAL OR WELFARE SERVICE	
469	BANKING/INSURANCE/ACCOUNTANCY	
470	COMMERCE	

```
471   INDUSTRY
472   OTHER
473   C. OF E.
474   R.C.
475   NONCONFORMIST
476   OTHER RELIGION
477   HUMANIST/AGNOSTIC/ATHEIST
478   SCOTLAND AND NORTHERN IRELAND
479   NORTHERN
480   YORKSHIRE AND HUMBERSIDE
481   NORTH WEST
482   EAST MIDLAND
483   WEST MIDLAND
484   WALES AND MONMOUTH
485   SOUTH WEST
486   EAST ANGLIA
487   SOUTH EAST
488   OTHER/ABROAD
489   ABCDEFGHIJKLMNOPQRSTUVWXYZ
490   079
491     060 30  046 31  031 32  030 33  041 34  042 35  044 36  043 37  045 38  033 39
492     100 39  051 40  053 41  098 42  094 43  104 44  092 45  004 46  005 46  029 46
493     032 46  040 46  056 46  007 46  055 46  054 46  008 46  096 46  073 46  ZZZ 47
494     ZZZ 48  012 49  013 50  011 51  014 51  015 51  016 51  017 51  010 51  021 52
495     022 53  ZZZ 54  026 55  023 56  025 57  086 58  088 59  089 60  027 61  048 61
496     047 61  020 61  083 61  024 61  006 61  097 61  090 61  082 61  038 61  037 61
497     034 61  058 61  039 61  067 61  049 62  050 63  091 64  052 65  099 66  067 67
498     101 68  059 69  102 69  002 69  103 69  066 69  093 69  061 46  087 61
499   3  4  6  7  9 11 13 14 16 20  9.80 3.43
500   4  5  0  6  7  8  9 10 11 13  7.72 1.80
501   7  9 11 13 15 17 19 21 22 26 15.50 3.69
502   6  8  9 11 13 16 18 19 21 26 13.94 3.29
503   5  8 10 13 15 17 19 21 23 26 15.73 4.40
504   4  6  9 11 12 14 16 17 19 20 12.73 3.60
505   2  4  7 10 13 16 18 20 22 26 13.01 5.24
506   2  3  5  6  8 10 12 14 15 20  8.79 3.49
507   3  4  6  7  9 11 12 14 15 20  9.47 3.13
508   5  6  8  9 11 13 15 17 18 26 11.68 3.41
509   5  7  8  9 10 12 13 15 16 20 11.07 2.63
510   3  4  6  8  9 11 13 15 17 26 10.65 3.90
511   4  5  6  8  9 10 12 13 15 20  9.64 2.75
512   3  4  6  7  9 11 13 15 17 20  9.97 3.50
513   3  5  6  8 10 11 13 14 16 20 10.14 3.07
514   3  4  7  9 12 14 17 19 21 26 12.01 4.81
515   3  4  6  8  9 11 13 15 16 20 10.08 3.43
516   4  5  6  7  8  9 10 11 12 13  8.83 1.89
517   6  8 11 13 15 17 19 20 22 26 15.14 3.90
518   5  7  9 11 13 15 17 19 21 26 13.93 4.12
519   6  8 11 13 16 18 20 22 23 26 16.44 4.52
520   3  5  7  9 11 13 15 17 18 20 11.95 3.75
521   2  4  7 10 13 17 19 22 23 26 13.75 5.77
522   2  3  4  6  8 11 13 14 16 20  9.24 3.82
523   2  3  5  7  8 10 12 13 15 20  8.97 3.34
524   5  7  8 10 12 14 16 18 19 26 12.69 3.62
525   2  3  5  6  7  9 10 11 13 20  8.25 2.68
526   2  4  6  7  9 12 14 16 18 26 10.18 3.93
527   3  4  6  8  9 11 12 14 15 20 10.16 3.23
528   2  4  6  8  9 11 14 15 17 20 10.28 3.75
529   4  6  8  9 11 13 14 16 17 20 11.72 3.27
530   3  5  7 10 12 15 17 19 21 26 12.77 4.82
```

8. INPUT DATA.

8.1 DATA LISTINGS

THIS SECTION PROVIDES EXAMPLES OF THE INPUT DATA TO BCRS THE EXAMPLES PROVIDED CAN BE SUMMARISED AS FOLLOWS:

8.1.1 DATA LISTINGS FOR PROGRAMME R.

1 ADMISSIONS RECORDS
2 INTELLIGENCE TEST RECORDS
3 ANNUAL UPDATES
4 WITHDRAWALS
5 CHANGES OF COURSE/SUBJECT
6 SPECIFIC FIELD ALTERATIONS

8.1.2 DATA LISTINGS PROGRAMMES F AND I.

1 THE SPECIAL REQUEST, REQUEST='*' IN PROGRAMME F
2 THE SPECIAL REQUEST, REQUEST='*' IN PROGRAMME I

8.1.3 SELECTED OPTION PARAMETER CARDS.

THE FORMAT DESCRIBING THE DATA APPEARING IN SECTION 8.1.1 CAN BE FOUND IN 'BCRS....BEDE COLLEGE RECORD SYSTEM', APPENDIX 2.

8.1.1 DATA LISTINGS FOR PROGRAMME B.

1 ADMISSIONS RECORDS

```
 1   STUD1972 CONSTNTS 1972  1 ;
 2   71/42120FOX, ROBERT                                M1M130353 2509721B302024  1Y601
 3   71/42120                 DURHAM              3092   1 09720775  111   1  092
 4   71/42120   0116364716 9  00613P69  00605P69  00113P69  00207F69  00947F69  00989F
 5   71/42120  69  00239F69  00879F69  00871P69  00203P69  00927F69  00229F59  A0920F
 6   71/42120  69  A0600F69  A0139F69CSE041XP
 7   72/42051ADAMSON, COLIN TREVOR                      M1M251053125097 21C401022  3Y691
 8   72/42051                 YORKS. N RIDING     3052   1 09720775  111   1  052
 9   72/42051   01114657270  00613P70  00513P70  00526P70  00415P70  00425P70  00113P
10   72/42051  70  00226P70  00217F72  00996P72  A0520F72  A091EP
11   72/42052ALDRIDGE, JOHN ROSS CHARLES                M1M0412511250972 1J4010212 1Y651
12   72/42052                 BARNET              3011   1 09720775  111   1  011
13   72/42052   01124637067  00113P68  00613P68  00603P68  00313P68  00415P68  00425P
14   72/42052  68  00525P68  00986P70  A011BP70  A041EP70  A060EP69  E0622P
15   72/42053ATHERTON, JOHN ROBERT                      M1M150353125097 21D201021  3Y681
16   72/42053                 CHESHIRE            3048   1 09720775  111   1  048
17   72/42053   02157636446472      69  00617F69  00118F69  00235P69  00876P69  00528F
18   72/42053  70  00614P70  00115P70  00216P70  00223P70  00246P70  00418F70  00528F
19   72/42053  72  00994P72  A023DP72  A0220F
20   72/42054ATKINSON, GRAHAM DAVID                     M1M3001541250972 1E402041  1N  1
21   72/42054                 BIRMINGHAM          30510611 09720775  111   1  051
22   72/42054   02136656736772      72  S0512P69  00611P69  00983P70  00603P70  00513P
23   72/42054  70  00413P70  00941P70  00113P70  00213P70  00223P72  A060CP72  A091DP
24   72/42054  72  A056DP
25   72/42251BAKER, ANDRE                               M1M13C8451250972 1J202022 01N  1
26   72/42251                 CAMBRIDGESHIRE/ELY 1      1 09720773  111    1      1
27   72/42251C1SUSSEX              F011066150769
28   72/42251C2BSUSSEX             BSCH2.1PSYCHOLOGY                            150769
29   72/42251   01144566360  00613P68  00111P60  00412P61  00521P61  00231P62  00313P
30   72/42251  62  00911P62  00991P62  A0221P62  A0211P62  A0121P62  A0111P63  S0222P
31   72/42251  63  S0212P
32   72/42252BARNES, DAVID PHILIP                       M1M0206511250972 1H501015  5N  1
33   72/42252                 BRISTOL             1      1 09720773  111    1      1
34   72/42252C1CARDIFF             F031069240672
35   72/42252C2BWALES              BSCH2.1POLITICS                              080772
36   72/42252   01102626967  0011XP67  0061XP67  0051XP67  0052XP67  0020XP67  0041XF
37   72/42252  67  0023XF69  A091CP69  A051BP69  A011FF
38   72/42253BELL, MICHAEL RAYMOND                      M1M0809501250972 1B301023  3N  1
39   72/42253                 CUMBERLAND          1      1 09720773  111    1      1
40   72/42253C1DURHAM              F081069250672
41   72/42253C2BDURHAM              BAH2.1HISTORY                               280672
42   72/42253   01133626967  00612P67  00601P67  00563P67  00314P67  00412P67  00113P
43   72/42253  67  00213P67  00223P69  A099CP69  A060BP69  A056BP69  S0562P69  A041EP
44   72/42254BOWREY, PATRICK STEVEN                     M1M0606501250972 1H502031  1N  1
45   72/42254                 BRISTOL             1      1 09720773  111    1      1
46   72/42254C1DURHAM              F021069250672
47   72/42254C2BDURHAM              BAH2.1HISTORY     ANTHROP.    ARCHAEOLOGY 250672
48   72/42254   01162616865  00613P65  00603P65  00144P65  00512P65  00524P65  00413P
49   72/42254  65  00237F65  00317F66  00317F67  A051DP67  A0520F67  A060CP68  A051CP
50   72/42254  68  A052CP67  F062XP68  S051XP
51   ********
```

```
STUD1969 CONSTNTS 1969 2 YT='F' NTT='1' ;
66/02669353630412216127405409408111213151405111616120714120714
67/04251302416323741135424809008101616171026121212081613090508
68/01269303633382914169445810212110713110606171120140911151015
68/45700384128441910141424809014100918200512121210110912051015
68/90750373326314013176465510112120915180511130917141217160514
69/42051372625362927162485610414101214191219120810061507061213
69/42052273223423620144344007416061214201212071009141110071211
69/42053313831403307169445409810081215191016121804121208060915
69/42056403417452915170434008309061608231417100808020816081208
69/42057243237403413131364708308071017190616140812081009100318
69/42058362729402325160354708206061310150621151015091311150711
69/42059403738321617127375409110091820220608141008100910111510
69/42060342729393319162404608612111122221012101612102111131013
69/42061293135311836151354708213051409051709160818160812181210
69/42062342926433414154364908515091715130716130717120911060812
69/42063333419382432133435509816091913201513080808140313121107
69/42064363521342529135476411112092117171515100614131010071205
69/42065253225314720200324607814091217151216061310111308041014
69/42066272019432150198465510116101917131721140315160710121712
69/42067183426381549189404908905051419121516090915080810161417
69/42069252021422250225466210809111310181710131312121007061513
69/42070324724461615176495710620081915221807090810081612080512
69/42071224821373230153415309410101221160614150910061111080916
69/42073403217442225160455710206071110061206141217121110150911
69/42074282936452319124416310411121218220719110816071405110418
69/42075182924502039161415109212081111200809131315131014100608
69/42078362044292625181374608311081120070403171719102212170722
69/42081282626432334183463708315101509220508100912101112080512
69/42082214022323926167323206414071719151008091614141409140414
********
```

3 ANNUAL UPDATES

```
STUD1969 CONSTNTS 1969  3 VP='0' ;
60/059631 *A3=P*    *A4=P*    *A5=P*    *A6=P*    *A7=P*    *A8=P*    *A9=P*
66/026691 *A1=C*    *A3=B+*   *A4=S*    *A5=P*    *A6=A*    *A7=C++*  *A8=C+*   C
66/026691 *A9=C*
68/012691 *A1=B-*   *A3=A-*   *A4=S*    *A5=P*    *A6=B*    *A7=C*    *A8=D*    C
68/012691 *A9=B*
68/457001 *A1=B*    *A3=C*    *A4=S*    *A5=P*    *A6=B*    *A7=C*    *A8=C-*   C
68/457001 *A9=C*
68/907501 *A1=C*    *A3=C+*   *A4=S*    *A5=P*    *A6=C*    *A7=C++*  *A8=C-*   C
68/907501 *A9=D*
69/420511 *A1=B-*   *A3=C*    *A4=S*    *A5=P*    *A6=B*    *A7=C*    *A8=C*    C
69/420511 *A9=C*
69/420521 *A1=C*    *A3=C*    *A4=S*    *A5=P*    *A6=C*    *A7=C+*   *A8=C+*   C
69/420521 *A9=A*
69/420531 *A1=B-*   *A2=B-*   *A3=B-*   *A4=S*    *A5=P*    *A6=B*    *A7=C+*   C
69/420531 *A8=B-*   *A9=B*
69/420561 *A1=C*    *A3=B*    *A4=S*    *A5=P*    *A6=C*    *A7=C+*   *A8=C*    C
69/420561 *A9=C*
********
STUD1969 CONSTNTS 1969  3 VP='0' ;
66/026692 *A10=B*   *A12=B*   *A13=S*   *A15=A*   *A16=A*   *A17=A*   *A19=B-* C
66/026692 *A20=A+*  *A21=A*   *A22=A*   *A23=PASS*
68/012692 *A10=C-*  *A11=E*   *A12=C+*  *A13=S*   *A17=C*   *A23=TO RE COUP P.E*
68/457002 *A10=C+*  *A12=C*   *A13=S*   *A15=B-*  *A16=B*   *A17=B*   *A19=C*   C
68/457002 *A20=B-*  *A21=C*   *A23=PASS*
68/907502 *A10=C*   *A12=C*   *A13=S*   *A15=B-*  *A16=B*   *A17=B*   *A18=C*   C
68/907502 *A19=B*   *A20=B*   *A21=C*   *A22=A*   *A23=PASS*
69/420512 *A10=B-*  *A12=C+*  *A13=S*   *A15=B*   *A17=B-*  *A19=A-*  *A20=A*   C
69/420512 *A21=B*   *A23=PASS*
69/420522 *A10=B*   *A12=C+*  *A13=S*   *A15=B-*  *A16=B*   *A17=C*   *A19=B*   C
69/420522 *A20=B-*  *A21=A*   *A22=A*   *A23=PASS*
69/420532 *A10=B-*  *A11=B-*  *A12=B-*  *A13=S*   *A15=B*   *A17=C*   *A19=B*   C
69/420532 *A20=A*   *A21=A+*  *A23=PASS*
********
STUD1969 CONSTNTS 1969  3 VP='1' ;
66/026693 *A24=C*   *A26=D*   *A27=P*
67/042513 *A24=P*   *A26=P*   *A27=P*
68/012693 *A24=F*   *A26=F*   *A27=P*   *A33=F*   *A34=F*   *A35=F*   *A39=2*
68/457003 *A24=P*   *A26=P*   *A27=P*
68/907503 *A24=P*   *A26=C*   *A27=P*
69/420513 *A24=P*   *A26=P*   *A27=P*
69/420523 *A24=P*   *A26=P*   *A27=P*
69/420533 *A24=P*   *A26=P*   *A27=P*
********
STUD1969 CONSTNTS 1969  3 VP='1' DES='TRANS' ;
60/05963 F          1                                A          080770
66/02669 F          1                                G          080772
67/04251 F          1                                A          080772
68/01269 F          3                                           080772
68/45700 F          1                                C          080772
68/90750 F          1                                A          080772
69/42051 F          1                                A          080772
69/42052 F          1                                A          080772
********
```

4 WITHDRAWALS

```
STUD1971 CONSTNTS 1971  4 VP='1' DES='TRANS' ;
71/42051 051272 9
71/42060 080772 9
71/42066 240473 3
71/42073 111271 9
71/42078 040172 2
71/42080 080772 9
71/42087 080772 2
71/42097 091171 9
71/42106 240372 9
71/42120 080772 5
71/42122 040172 9
********
```

5 CHANGES OF COURSE/SUBJECT

```
1   STUD1970 CONSTNTS 1970  8 VF='1' DES='TRANS' ;
2   70/42053 S                                        1 09720773  111      1       1
3   70/42054 C                                        1 09720773  111      1       1
4   70/42055 S                                        1 09720775  111    1  092
5   70/42056 S                                        1 09720775  111    1  052
6   70/42057 C                                        1 09720775  111    1  011
7   ********
```

6 SPECIFIC FIELD ALTERATIONS

```
1   STUD1972 CONSTNTS 1972  9 VP='1' ;
2   72/42061 RECORD.NAME='SMITH, DAVID' ;
3   72/42062 RECORD.GENERAL.REL='1' RECORD.GENERAL.SET='4' ;
4   72/42068 RECORD.GENERAL.SOCC='5' ;
5   72/42069 RECORD.DESNO='72/42090' ;
6   72/42088 RECORD.GENERAL.FPOS='0203' RECORD.GENERAL.DOB='021046';
7   ********
```

8.1.2 DATA LISTINGS FOR PROGRAMMES F AND I.

1 THE SPECIAL REQUEST, REQUEST='*' IN PROGRAMME F

```
  1   STUD1969 CONSTNTS 1969  2 REQUEST='*' ;
  2   69/42055
  3   69/42056
  4   69/42057
  5   69/42058
  6   69/42059
  7   69/42251
  8   69/42252
  9   69/42253
 10   69/42254
 11   69/42255
 12   ********
```

2 THE SPECIAL REQUEST, REQUEST='*' IN PROGRAMME I

```
  1   STUD1969 CONSTNTS 1969  4 YT='F' REQUEST='*' ;
  2   69/42156
  3   69/45157
  4   69/42180
  5   69/42181
  6   69/42185
  7   69/42199
  8   69/42200
  9   69/42201
 10   69/42202
 11   69/42220
 12   ********
```

2.4 STUD1972 CONSTNTS 1972 2 WRL='111' WCR='*' FREQ='1' REQUEST=' 1>25 ' ;

THIS CARD WILL PRODUCE A LIST OF THE D.E.S.NOS. OF THE STUDENTS IN THE FILE 'STUD1972' WHO WERE OVER 25 ON ADMISSION. A SET OF GENERAL INFORMATION FREQUENCY TABLES WILL ALSO BE GIVEN.

3. PROGRAMME I.

3.1 STUD1970 ******** 1970 1 YT='F' WRL='111' WCR='*' ;

THIS CARD WILL PRODUCE A SET OF TOTALS, MEANS AND STANDARD DEVIATION TABLES OF THE INTELLIGENCE TEST SCORES OF THE STUDENTS IN THE FILE 'STUD1970'.

3.2 STUD1970 ******** 1970 2 YT='F' WRL='111' WCR='*' ;

THIS CARD WILL PRODUCE A SET OF FREQUENCY TABLES OF THE SCORES OF THE STUDENTS IN THE FILE 'STUD1970'.

3.3 STUDENTS ******** 1969 3 YT='F' WRL='100' WCR='*' PI='1' ;

THIS CARD WILL LIST THE INTELLIGENCE TEST SCORES OF THOSE STUDENTS WHO HAVE WITHDRAWN FROM THE FILE 'STUDENTS'.

3.4 STUDENTS ******** 1972 3 YT='F' WRL='111' WCR='*' PI='2' ;

THIS CARD WILL LIST THE D.E.S.NOS. OF STUDENTS IN THE FILE 'STUDENTS' WHO HAVE INTELLIGENCE TEST SCORES.

3.5 STUD1969 ******** 1969 4 YT='F' REQUEST=' 1>25 & 3=2 ' ;

THIS CARD WILL PROVIDE TOTALS, MEANS, STANDARD DEVIATIONS AND FREQUENCY TABLES OF THE STUDENTS IN THE FILE 'STUD1969' WHO WERE OVER 25 AND MARRIED WHEN ADMITTED.

NB NOTE THAT AS PROGRAMME I DOES NOT REQUIRE THE DATA CONSTANTS, THE FILE NAME HAS BEEN SPECIFIED AS '********' IN EACH CASE.

4. PROGRAMME D.

4.1 STUD1972 CONSTNTS 1972 DES='TRANS' ;

THE OPTION PARAMETER CARD FOR PROGRAMME D GIVEN ABOVE INFORMS BCPS THAT THE TRANSACTION FILE IS KNOWN AS 'TRANS', AND THE DATA FILE IS 'STUD1972'.

9. OUTPUT LISTINGS.

9.1 THIS SECTION CONTAINS EXAMPLES OF THE OUTPUT PROVIDED BY THE BCRS PROGRAMMES.

THE FOLLOWING EXAMPLES ARE INCLUDED:

9.1.1 PROGRAMME R
1 '1970' FILE LISTING (OPTION 6, PI='2')
2 '1970' FILE LISTING (OPTION 6, PI='3')
3 '1972' FILE LISTING (OPTION 6, PI='4')

9.1.2 PROGRAMME F
1 '1972' GENERAL FREQUENCIES (OPTION 1)
2 '1972' REQUEST, 'AGE<21' (OPTION 2)
3 '1972' REQUEST & FREQUENCIES, 'AGE<21' (OPTION 2)
4 '1972' REQUEST, 'AGE<25 & MARRIED' (OPTION 2)

9.1.3 PROGRAMME I
1 '1972' TOTALS. MEANS & STANDARD DEVIATIONS (OPTION 1)
2 '1972' FREQUENCIES (OPTION 2)
3 '1972' REQUEST, 'AGE<25' (OPTION 4)

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME R	
OPTION REQUESTED	5
YEAR REQUESTED	1970
PRINT INDICATOR	2

1970 FILE.....215 STUDENTS INCLUDED.....NUMBERS FOLLOWED BY '*' SHOW THAT THE STUDENT HAS WITHDRAWN FROM THE COLLEGE
NUMBERS FOLLOWED BY '+' SHOW THAT THE STUDENT HAS COMPLETED HIS COURSE AND LEFT

60/02246	65/10053	67/02382	67/42053*	70/15512	70/42051+	70/42052+	70/42053+	70/42054+	70/42055
70/42056+	70/42057+	70/42058*	70/42059+	70/42060+	70/42061+	70/42062+	70/42063+	70/42064+	70/42065+
70/42066*	70/42067*	70/42068	70/42069+	70/42070+	70/42071+	70/42072+	70/42073+	70/42074+	70/42075+
70/42076+	70/42077	70/42078+	70/42079+	70/42080*	70/42081+	70/42082+	70/42083+	70/42084+	70/42085+
70/42086	70/42087	70/42088+	70/42089+	70/42090+	70/42091+	70/42092+	70/42093	70/42094+	70/42095
70/42096+	70/42097+	70/42098+	70/42099+	70/42100+	70/42101	70/42102+	70/42103+	70/42104+	70/42105
70/42106+	70/42107+	70/42108	70/42109	70/42110+	70/42111+	70/42112+	70/42113+	70/42114+	70/42115*
70/42116+	70/42117*	70/42118+	70/42119+	70/42120+	70/42121+	70/42122+	70/42123+	70/42124+	70/42125+
70/42126*	70/42127	70/42128	70/42129+	70/42130+	70/42131+	70/42132	70/42133+	70/42134+	70/42135+
70/42136+	70/42137+	70/42138+	70/42139	70/42140+	70/42141+	70/42142	70/42143+	70/42144+	70/42145
70/42146*	70/42147+	70/42148	70/42149+	70/42151+	70/42152+	70/42153*	70/42154+	70/42155*	70/42156+
70/42157	70/42158*	70/42159	70/42160+	70/42161+	70/42162	70/42163+	70/42164+	70/42165+	70/42166+
70/42167	70/42168+	70/42169+	70/42170+	70/42171+	70/42172+	70/42173+	70/42174	70/42175	70/42176*
70/42177	70/42178+	70/42179*	70/42180+	70/42181*	70/42182+	70/42183*	70/42184	70/42185*	70/42186+
70/42187+	70/42188*	70/42189	70/42190	70/42191+	70/42192	70/42193+	70/42194+	70/42195*	70/42196*
70/42197+	70/42198+	70/42199+	70/42200+	70/42201+	70/42202+	70/42203	70/42204	70/42205+	70/42206+
70/42207+	70/42209+	70/42210	70/42211+	70/42212+	70/42213+	70/42214	70/42215+	70/42216+	70/42217+
70/42218*	70/42219+	70/42220*	70/42221	70/42222+	70/42223	70/42224+	70/42225+	70/42226*	70/42227+
70/42228+	70/42229+	70/42230+	70/42231+	70/42232	70/42233+	70/42234+	70/42235+	70/42236+	70/42237
70/42238	70/42239+	70/42240*	70/42241+	70/42242+	70/42243*	70/42244	70/42245+	70/42246	70/42247
70/42248	70/42250	70/42251	70/42252	70/42253	70/42254	70/42255	70/42256	70/42257	70/42258
70/42259	70/42260	70/42261	70/42263	70/42264					

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME R	
OPTION REQUESTED	6
YEAR REQUESTED	1970
PRINT INDICATOR	3

1970 FILE.....215 STUDENTS INCLUDED

NAME	D.E.S.NO.	BIRTH	START	FINISH
ADAMS, PAUL	70/42051	12/12/51	25/09/70	08/07/73
ALBERT, CHRISTOPHER CHARLES G.	70/42141	05/06/52	25/09/70	08/07/73
ALDERSON, JAMES ARTHUR	70/42052	18/03/37	25/09/70	08/07/73
ALLAN, MAURICE	70/42053	09/01/35	25/09/70	08/07/73
APPLEBY, JOHN RAYMOND	67/42053	19/09/47	25/09/70	24/08/72
ARCHER, ANDREW JAMES FRANK	70/42054	24/05/52	25/09/70	08/07/73
ARNOLD, ANTHONY IAN CRICHTON	70/42055	04/11/49	25/09/70	
ASHCROFT, JOHN	70/42253	12/11/39	25/09/70	
ATKINSON, NEIL	70/42056	16/09/51	25/09/70	08/07/73
BAINBRIDGE, IAN	70/42057	10/05/51	25/09/70	08/07/73
BAITEY, GEORGE	70/42058	01/11/51	25/09/70	06/04/71
BALL, GRAHAM ROBERT	70/42059	07/01/51	25/09/70	08/07/73
BATY, EDWARD STANLEY	70/42060	25/11/51	25/09/70	08/07/73
BECK, PAUL ANTHONY	70/42061	04/05/50	25/09/70	08/07/73
BELL, ALAN	70/42062	13/08/48	25/09/70	08/07/73
BELL, IAN DAVID	70/42063	14/11/51	25/09/70	08/07/73
BELL, MALCOLM ARTHUR	70/42064	01/05/52	25/09/70	08/07/73
BENDELOW, WILLIAM EDWARD	70/42070	16/01/39	25/09/70	08/07/73
BEST, WILLIAM	70/42071	23/07/52	25/09/70	08/07/73
BESTWICK, NEIL HAYON	70/42246	23/08/48	25/09/70	
BETTRIDGE, JOHN RICHARD	70/42072	15/08/51	25/09/70	08/07/73
BIRD, OLIVER JAMES	70/42073	06/11/51	25/09/70	08/07/73
BISHOP, CHARLES RICHARD HUGH	70/42074	12/01/50	25/09/70	08/07/73
BLADEN, TERENCE EDWARD	70/42260	26/06/47	25/09/70	
BLAKEMAN, PETER ANDREW	70/42068	11/11/47	25/09/70	
BLOOM, JOHN	70/42069	24/09/48	25/09/70	08/07/73
BOWE, JOSEPH ARTHUR	70/42067	30/06/52	25/09/70	11/12/71
BRENT, GEORGE JOSEPH PETER	70/42066	09/02/48	25/09/70	18/06/71
BRENT, STEPHEN BARRINGTON	70/42065	28/03/51	25/09/70	08/07/73
BROOK, JOHN WILLIAM	70/42075	16/05/50	25/09/70	08/07/73
BROWN, LESLIE FORSTER	70/42076	08/03/51	25/09/70	08/07/73
BROWN, STEWART JAMES	70/42077	30/01/52	25/09/70	
BRYAN, MICHAEL LENNARD	70/42078	11/02/51	25/09/70	08/07/73
BRYSON, DAVID THOMAS WARWICK	70/42079	20/06/50	25/09/70	08/07/73
BUCKNOLE, KIM	70/42080	18/01/51	25/09/70	19/04/71
BUNKER, NIGEL STEPHEN	70/42081	09/07/52	25/09/70	08/07/73
BURTON, COLIN GEORGE	70/42082	05/02/41	25/09/70	08/07/73
CALLADINE, MURRAY JOHN	70/42083	03/02/52	25/09/70	08/07/73
CAWTHORN, IAN OATES	70/42084	05/12/51	25/09/70	08/07/73
CHANCE, PHILIP	70/42085	09/06/51	25/09/70	08/07/73
CHAPMAN, STEPHEN JOSEPH	70/42086	24/05/52	25/09/70	
CHIDGEY, ANTHONY PATRICK	70/42247	11/09/47	25/09/70	
CLAY-DOVE, GILBERT MARK WILLIAM	70/42087	15/03/52	25/09/70	
CLAYTON, DONALD	70/42088	06/01/45	25/09/70	08/07/73
CLELLAND, JOHN	70/42089	05/08/52	25/09/70	08/07/73
COCKSHAW, JOHN	70/42090	25/10/46	25/09/70	08/07/73
CONROY, ALBERT JOSEPH	70/42091	20/04/50	25/09/70	08/07/73
CULLINGTON, NOEL	70/42092	30/01/36	25/09/70	08/07/73
DEGARA, MICHAEL ANDREW	70/42093	13/02/49	25/09/70	
DOBSON, ALAN	70/42094	25/12/47	25/09/70	08/07/73
DOBSON, TREVOR	70/42095	14/05/52	25/09/70	
DUTFIELD, ANDREW	70/42096	08/11/49	25/09/70	08/07/73
EARL, ROLAND DONALD	70/42097	08/07/30	25/09/70	08/07/73
EMMETT, COLIN JOHN	70/42098	13/07/52	25/09/70	08/07/73
ENGLISH, JOHN ALAN	70/42099	19/06/43	25/09/70	08/07/73

NAME	D.E.S.NO.	BIRTH	START	FINISH
ERSKINE, MICHAEL ANDREW	70/42100	05/05/51	25/09/70	08/07/73
EVANS, AIDAN WILFRID	70/42101	29/05/52	25/09/70	
EYRES, CHRISTOPHER STEVEN	70/42102	26/01/51	25/09/70	08/07/73
FARROW, DAVID JOHN ALFRED	70/42103	03/08/51	25/09/70	08/07/73
FAULDER, PAUL ANTHONY	70/42104	21/01/52	25/09/70	08/07/73
FAULDS, IAN CRAIG	67/02382	04/07/48	25/09/70	
FAULKS, DAVID ANTHONY	70/42105	23/06/51	25/09/70	
FINCH, TREVOR RAYMOND JAMES	70/42106	06/10/51	25/09/70	08/07/73
FOGGETT, THOMAS LESLIE	70/42107	04/02/52	25/09/70	08/07/73
FRENCH, LESLIE JAMES	70/42108	13/08/52	25/09/70	
GIBSON, THOMAS	70/42254	26/11/45	25/09/70	
GLENTON, RAYMOND	70/42109	09/02/51	25/09/70	
GREENSIT, THOMAS GRAHAM	70/42110	17/01/51	25/09/70	08/07/73
GREY, JOHN DAVID	70/42111	21/01/47	25/09/70	08/07/73
GRUDGINGS, ROBERT GRAHAM	70/42112	26/03/52	25/09/70	08/07/73
HAGUE, NIGEL	70/42113	09/07/51	25/09/70	08/07/73
HALL, DEREK JAMES	70/42114	15/03/47	25/09/70	08/07/73
HAM, JOHN ALISTAIR	70/42261	18/08/48	25/09/70	
HANSON, JOHN BARRIE	70/42115	20/11/50	25/09/70	21/09/71
HARDING, PAUL	70/42116	17/03/52	25/09/70	08/07/73
HARKNESS, PATRICK JOHN	70/42117	22/03/52	25/09/70	19/04/71
HARMER, DAVID LLOYD	70/42118	05/08/52	25/09/70	08/07/73
HARVEY, GEORGE JEFFREY	70/42119	30/05/27	25/09/70	08/07/73
HEBDEN, ALAN RICHARD	70/42120	22/10/51	25/09/70	08/07/73
HENDRY, PETER JAMES BECKER	70/42121	02/04/49	25/09/70	08/07/73
HILL, THOMAS ELLIOTT	70/42122	14/05/51	25/09/70	08/07/73
HINMAN, DAVID FRANK	70/42123	03/04/52	25/09/70	08/07/73
HODGE, JOHN MARSHALL	70/42124	27/05/52	25/09/70	08/07/73
HODGSON, WILLIAM	70/42125	15/02/34	25/09/70	08/07/73
HOLMES, ALAN	70/42126	18/03/48	25/09/70	05/12/70
HOOKWAY, JOHN	70/42127	11/10/50	25/09/70	
HOOSON, GAVIN RODERICK	70/42128	18/09/50	25/09/70	
HOWARD, CHRISTOPHER PATRICK	70/42129	04/12/50	25/09/70	08/07/73
HOWDEN, IAN JAMES	70/42130	21/09/50	25/09/70	08/07/73
HUDSPETH, MALCOLM	70/42131	15/10/49	25/09/70	08/07/73
HUTTON, ANDREW RICHARD	70/42132	13/02/52	25/09/70	
JAGGARD, ANDREW QUENTIN	70/42133	27/04/52	25/09/70	08/07/73
JENKINS, ARCHIBALD AITKEN	70/42134	11/05/52	25/09/70	08/07/73
JOHNSON, IAN LESLIE	70/42135	27/09/51	25/09/70	08/07/73
JOHNSON, JOHN DAVID	70/42136	07/11/50	25/09/70	08/07/73
JOHNSTON, FREDERICK	70/42137	17/07/51	25/09/70	08/07/73
KEELER, STEPHEN ROBERT	70/42138	12/09/51	25/09/70	08/07/73
KERRIDGE, DONALD GEORGE	60/02246	12/12/32	25/09/70	
KIDD, CHARLES WILLIAM	70/42139	23/05/52	25/09/70	
KING, KEITH	70/42140	06/06/50	25/09/70	08/07/73
KINGHAM, MICHAEL ROY	65/10053	17/08/45	25/09/70	
LAUNDER, ROBIN	70/42142	09/10/51	25/09/70	
LEE, ALAN	70/42143	21/01/70	25/09/70	08/07/73
LEE, HENRY	70/42249	24/09/31	25/09/70	
LITTLE, RICHARD DUNCAN	70/42144	14/08/52	25/09/70	08/07/73
LORIMER, GEORGE PETER	70/42145	14/04/51	25/09/70	
LOTHIAN, ALAN CHARLES	70/42146	14/04/51	25/09/70	10/05/71
LUCAS, FRANK	70/42147	24/02/52	25/09/70	08/07/73
LUXTON, WILLIAM HENRY	70/42148	27/11/51	25/09/70	
LYNAS, PAUL	70/42250	20/05/48	25/09/70	
LYONS, HARRY	70/42149	10/10/44	25/09/70	08/07/73

NAME	D.E.S.NO.	BIRTH	START	FINISH
MABEY, MARTIN HILARY	70/42251	04/08/46	25/09/70	
MACRAE, ROSS JAMES PAUL	70/42263	12/12/32	25/09/70	
MADDISON, ANTHONY HAROLD	70/42154	10/07/50	25/09/70	08/07/73
MASON, JAMES FRANCIS	70/42155	11/06/51	25/09/70	09/12/72
MASSEY, CHRISTOPHER HEDLEY H.	70/42156	22/02/51	25/09/70	08/07/73
MATHER, ANDREW IAN	70/42157	19/01/51	25/09/70	
MAY, JOHN ALEXANDER CYRIL	70/42158	18/03/52	25/09/70	01/07/71
MAYCOCK, CHRISTOPHER REX	70/42159	24/06/52	25/09/70	
MCCOY, FRANK	70/15512	24/09/49	25/09/70	
MCEWAN, GRAHAM SHEARER	70/42151	02/01/52	25/09/70	08/07/73
MCHARG, DAVID	70/42255	19/06/42	25/09/70	
MCLEAN, DAVID IAN	70/42152	17/11/51	25/09/70	08/07/73
MCNALLY, DAVID	70/42153	08/08/44	25/09/70	01/07/72
MICKLEWRIGHT, JEFFREY RAYMOND	70/42160	16/06/51	25/09/70	08/07/73
MILLER, PETER EDWARD MCQUEEN	70/42161	15/08/52	25/09/70	08/07/73
MOFFAT, BRIAN	70/42162	02/09/52	25/09/70	
MOON, ROBERT WILLIAM	70/42163	18/03/51	25/09/70	08/07/73
MORTON, PAUL ANTHONY	70/42164	27/10/51	25/09/70	08/07/73
MOXON, PETER	70/42165	13/10/50	25/09/70	08/07/73
MURDAY, JOHN JAMES	70/42166	09/07/52	25/09/70	08/07/73
MURPHY, ALISTAIR DONALDSON	70/42167	11/12/49	25/09/70	
NEAL, JOHN	70/42168	20/02/52	25/09/70	08/07/73
NEASHAM, DAVID WILLIAM	70/42169	20/12/51	25/09/70	08/07/73
OAKES, BRIAN	70/42170	18/06/47	25/09/70	08/07/73
OGDEN, STEWART GALLOWAY	70/42171	23/09/48	25/09/70	08/07/73
OLIVER, RICHARD	70/42172	02/01/52	25/09/70	08/07/73
OLIVER, RICHARD	70/42173	08/11/35	25/09/70	08/07/73
ORMEROD, ERIC KENNETH	70/42174	31/12/51	25/09/70	
PAINE, JOHN HERBERT	70/42175	23/10/51	25/09/70	
PARISH, RICHARD	70/42176	11/10/51	25/09/70	27/11/70
PARSISSON, GEORGE HENRY	70/42177	11/01/52	25/09/70	
PEARSON, DAVID	70/42178	30/01/51	25/09/70	08/07/73
PHILIPS, REGINALD ENOC KIRKLAND	70/42179	22/06/51	25/09/70	09/09/71
PHILLIPPE, TERRY	70/42180	13/12/49	25/09/70	08/07/73
PHILLIPS, JAMES CHRISTOPHER NIGEL	70/42181	13/12/48	25/09/70	18/06/71
PRICE, WILLIAM	70/42182	12/07/52	25/09/70	08/07/73
PURCIFER, DESMOND POWIS	70/42183	27/01/51	25/09/70	09/12/72
RAMSEY, PETER	70/42184	01/10/51	25/09/70	
RAYNER, ADRIAN GEORGE	70/42185	22/09/51	25/09/70	04/03/71
REDDING, ROBERT GEORGE	70/42186	13/02/46	25/09/70	08/07/73
REED, JEFFREY	70/42252	10/11/44	25/09/70	
RICHARDSON, LESLIE	70/42187	29/06/52	25/09/70	08/07/73
RICHMOND, RONALD	70/42256	25/08/29	25/09/70	
ROBERTSON, DAVID HARRY	70/42188	20/01/52	25/09/70	13/08/71
ROBSON, IAN	70/42189	20/07/52	25/09/70	
RUFFLE, JOHN EDWARD	70/42190	03/01/52	25/09/70	
RUTTER, ALAN FRANK	70/42191	16/02/52	25/09/70	08/07/73
SAWYERS, MICHAEL FRANCIS	70/42192	17/03/51	25/09/70	
SCOFFIN, ANTHONY GEORGE	70/42193	31/01/52	25/09/70	08/07/73
SCOTT, ALAN WILLIAM	70/42195	23/12/51	25/09/70	08/02/71
SCOTT, ALLAN CAMERON	70/42194	28/07/52	25/09/70	08/07/73
SCOTT, GRAHAM	70/42196	15/11/50	25/09/70	28/04/73
SECCOMBE, MALCOLM CHARLES	70/42197	23/03/52	25/09/70	08/07/73
SHAW, GRAHAM	70/42198	22/05/52	25/09/70	08/07/73
SHIELDS, JOHN DAVID	70/42199	10/09/51	25/09/70	08/07/73
SHIPPEN, THOMAS	70/42200	10/03/27	25/09/70	08/07/73

NAME	D.F.S.NO.	BIRTH	START	FINISH
SIMPSON, ROBERT NEWTON	70/42201	09/12/51	25/09/70	08/07/73
SLAIN, KEITH	70/42212	25/02/51	25/09/70	08/07/73
SLOAN, DAVID	70/42202	17/03/52	25/09/70	08/07/73
SLOWEY, JOHN	70/42203	06/08/42	25/09/70	
SMITH, ALAN RONALD MONTGOMERY	70/42204	24/03/51	25/09/70	
SMITH, ROY JOHN	70/42205	18/02/35	25/09/70	08/07/73
SOUTHWARD, COLIN	70/42206	05/09/51	25/09/70	08/07/73
SPENCER, JEREMY LLOYD	70/42207	05/02/52	25/09/70	08/07/73
STEPHENSON, ALAN PAUL	70/42209	23/04/52	25/09/70	08/07/73
STEPHENSON, GEORGE WILLIAM	70/42257	25/04/44	25/09/70	
STEWART, IAN	70/42210	09/05/51	25/09/70	
STONE, CHRISTOPHER PAUL	70/42211	21/06/51	25/09/70	08/07/73
SWINNEY, ROBERT ALAN	70/42258	28/11/36	25/09/70	
THOMAS, MALCOLM	70/42213	27/01/51	25/09/70	08/07/73
THOMPSON, MALCOLM	70/42214	30/04/52	25/09/70	
THOMPSON, NEIL ANDREW	70/42215	19/10/51	25/09/70	08/07/73
THOMPSON, PETER	70/42216	16/02/36	25/09/70	08/07/73
THOMPSON, RODNEY WILLIAM	70/42217	22/09/51	25/09/70	08/07/73
THORNTON, DEREK	70/42218	02/10/51	25/09/70	04/03/71
TODD, GRANVILLE	70/42219	29/09/50	25/09/70	08/07/73
TOPP, JEFFERY ROGER	70/42220	12/08/50	25/09/70	11/07/71
UNSWORTH, JOHN PHILIP	70/42221	17/03/51	25/09/70	
URWIN, GORDON WALTER	70/42222	04/09/50	25/09/70	08/07/73
WALLACE, DAVID	70/42223	13/03/52	25/09/70	
WALLS, GEOFFREY	70/42224	16/04/52	25/09/70	08/07/73
WALSH, MICHAEL JULIAN	70/42225	03/09/50	25/09/70	08/07/73
WALT, TREVOR WILLIAM	70/42226	26/04/52	25/09/70	22/06/71
WARD, NIGEL JOHN	70/42227	26/09/51	25/09/70	08/07/73
WATSON, PETER WARDLE	70/42228	26/04/52	25/09/70	08/07/73
WEDDELL, MICHAEL	70/42229	29/04/49	25/09/70	08/07/73
WEDGEWOOD, FREDERICK JAMES S.	70/42230	26/10/51	25/09/70	08/07/73
WENTWORTH, JOHN STUART	70/42231	15/04/47	25/09/70	08/07/73
WHITCOMBE, JOSEPH NOEL	70/42232	25/12/30	25/09/70	
WHITTAKER, EDWARD	70/42233	06/10/51	25/09/70	08/07/73
WILEMAN, WALTER	70/42259	25/09/32	25/09/70	
WILKINSON, BRIAN	70/42234	15/11/33	25/09/70	08/07/73
WILLIAMS, FRANCIS JOHN	70/42235	07/10/51	25/09/70	08/07/73
WILLIAMS, NEVILLE HENRY	70/42236	20/09/51	25/09/70	08/07/73
WILSON, IAN	70/42237	11/12/49	25/09/70	
WINTER, BARRINGTON PETER	70/42238	01/12/51	25/09/70	
WIPER, MICHAEL JOHN	70/42239	18/01/52	25/09/70	08/07/73
WISHART, JOHN MACKEAND	70/42240	06/02/51	25/09/70	01/07/71
WOLSTENHOLME, COLIN	70/42241	24/01/52	25/09/70	08/07/73
WOOD, STEPHEN EDWARD	70/42264	14/12/42	25/09/70	
WOODCOCK, COLIN JOHN LINTON	70/42242	28/11/51	25/09/70	08/07/73
WOODHOUSE, PETER JAMES	70/42243	25/09/47	25/09/70	04/12/70
YOUNG, ROBERT	70/42244	12/06/51	25/09/70	
YUILLE, JOHN LAWRENCE	70/42245	16/03/51	25/09/70	08/07/73

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME R
OPTION REQUESTED 5
YEAR REQUESTED 1972

PRINT INDICATOR 4

1972 FILE.....212 STUDENTS INCLUDED

NO MAIN COURSE RECORDED

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
PGCE	72/42251	BAKER, ANDRE
PGCE	72/42252	BARNES, DAVID PHILIP
PGCE	72/42253	BELL, MICHAEL RAYMOND
PGCE	72/42254	BOWREY, PATRICK STEVEN
PGCE	72/42255	CATLIN, NICHOLAS JOHN
PGCE	72/42257	DUNSCOMBE, DAVID
PGCE	72/42258	EDWARDS, PAUL MARTIN
PGCE	72/42259	FEATHERSTONE-WITTY, MARK PHILIP
PGCE	72/42260	FOSTER, ALASTAIR JOHN
PGCE	72/42261	GREENWELL, BRIAN
PGCE	72/42262	HESLOP, JOHN RONALD
PGCE	72/42263	LLEWELLYN, DAVID GWYN
PGCE	72/42264	MCNAMARA, IAN EDWARD
PGCE	72/42265	MOORE, ALAN PETER
PGCE	72/42266	OGDEN, GRAEME MACLEOD
PGCE	72/42267	PARKER, DAVID GUNSON
PGCE	72/42268	SMITH, RONALD GEORGE
PGCE	72/42269	TAYLOR, ALAN FREDERICK
PGCE	72/42270	WHATMORE, JAMES CHARLES
PGCE	72/42271	YOUNG, FREDERICK DAVID
1	72/42238	ALLAN, DONALD
1	72/42239	ASKEW, MARK RUSSELL
1	72/42240	DALTON, JOHN THOMAS
1	72/42243	HEATLEY, KENNETH JOSEPH
1	72/42244	MAIDMENT, PETER
1	72/42245	MOREHEAD, WILLIAM CHARLES
1	72/42246	PIPER, TERENCE WINSTON
1	72/42247	SANDS, DAVID
1	72/42249	WHITE, WILLIAM SELWYN
1	72/42250	WOLFE, PETER LESLIE

COURSE CODE 001

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
1	72/42241	GOODWIN, DEREK
1	72/42242	GREEN, JOSEPH HUDSON

COURSE CODE 002

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
3	72/42056	AYRE, WILLIAM
3	72/42057	BALLARD, MALCOLM
3	72/42233	BARKER, ALAN MALCOLM
3	72/42061	BATHGATE, JAMES ALAN
3	72/42078	CHAPMAN, ARTHUR ALAN
3	72/42080	COLLINGWOOD, DAVID
3	72/42085	COTTON, DAVID ANTHONY
3	72/42091	DAVISON, GEOFFREY
3	72/42095	DOE, RAYMOND CLIFFORD
3	72/42104	FINDLATER, MAXWELL KERR
3	72/42112	GILFILLAN, HOWARD
3	72/42117	GRAVELING, JOHN EDWARD
3	72/42118	GRAYDON, GEOFFREY
3	72/42123	HAMILTON, IAN
3	72/42132	HINCH, JAMES ROBERTSON
3	72/42142	JACKSON, PAUL
3	72/42157	MCDONALD, JOHN
3	72/42165	MILLS, DAVID
3	72/42167	MONTGOMERY, PAUL
3	72/42169	MORDUE, ROBERT WEDDERBURN
3	72/42170	MORGAN, NEIL
3	72/42171	MURRAY, MITCHELL JOHN
3	72/42173	NEVINS, MICHAEL JOHN
3	72/42174	NICHOLSON, PETER THOMAS
3	72/42178	OLD, BRIAN LESLIE
3	72/42188	RICHARDSON, SYDNEY
3	72/42192	RUTHERFORD, JOHN FRANCIS
3	72/42202	STAYMAN, GEORGE MICHAEL
3	72/42204	STEPHENS, PHILIP MORGAN
3	72/42210	TAYLOR, JOHN BARRY
3	72/42222	WIDDRINGTON, THOMAS
3	72/42228	WILSON, GEOFFREY

COURSE CODE 011

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
3	72/42052	ALDRIDGE, JOHN ROSS CHARLES
3	72/42097	D.SOUZA, ANDREW BERNARD JOHN
3	72/42099	ELVIN, JOHN
3	72/42101	FEATHERSTONE, DENIS RONALD
3	72/42134	HODGKISS, PAUL
3	72/42162	MEDLOCK, STUART ANTONY
3	72/42185	RAMSHAW, MARSHALL
3	72/42196	SIMPSON, JAMES
3	72/42206	STRAW, CHRISTOPHER STEPHEN
3	72/42232	YOUNG, CLIVE PHILIP

COURSE CODE 048

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
3	72/42053	ATHERTON, JOHN ROBERT
3	72/42068	BLACKWELL, ADRIAN ALEXANDER
3	72/42076	CARTER, TREVOR RICHARD EARL
3	72/42077	CATTERSON, PETER
3	72/42084	COPELAND, JOSEPH STEPHEN
3	72/42090	DARWIN, DAVID CHRISTOPHER
3	72/42105	FITZSIMMONS, MALCOLM LEWIS
3	72/42115	GOSLING, BRIAN GEORGE EDWARD
3	72/42120	GRIFFIN, RICHARD CHARLES
3	72/42124	HAMILTON, MALCOLM IAN
3	72/42127	HARDY, KEITH JOHN
3	72/42133	HIRD, PETER GRAHAM
3	72/42149	KNOX, JEFFREY NICHOLAS
3	72/42150	LAUDER, IAN WILLIAM
3	72/42153	LOWEY, KENNETH
3	72/42177	O.DONNELL, PAUL ANTHONY
3	72/42187	REAY, DAVID
3	72/42199	STAINTHORPE, DAVID
3	72/42208	TAIT, ALAN
3	72/42209	TARRANT, GEOFFREY PAUL
3	72/42236	TARRELL, JAMES ANDREW SCOTT
3	72/42219	WATSON, ROBERT JOHN
3	72/42231	WREN, CHRISTOPHER JOHN

COURSE CODE 051

DATE OF LISTING 22/08/73

TYPE	D.F.S.NO	NAME
3	72/42054	ATKINSON, GRAHAM DAVID
3	72/42073	CARR, JOHN CLENNEL
3	72/42088	CROOKS, PHILIP GERARD
3	72/42092	DEBENHAM, PAUL BRUCE
3	72/42096	DREW, GARY RONALD
3	72/42107	FOX, PAUL DOUGLAS
3	72/42108	FRANKLIN, DEREK ANTHONY AIDAN
3	72/42109	GALLIE, DUNCAN ANDREW
3	72/42111	GARDINER, JOHN ROGAN
3	72/42122	HALLIDAY, STEPHEN
3	72/42126	HANSON, JOHN
3	72/42129	HEDLEY, NORMAN ANTHONY
3	72/42140	HULME, ROBIN ANTHONY
3	72/42144	JOHNSTON, DEREK
3	72/42164	MIDDLETON, COLIN TYSON
3	72/42172	MURRAY, PAUL
3	72/42176	NICOL, GARETH JAMES
3	72/42182	PICKERING, DAVID
3	72/42184	RAMSEY, NEIL
3	72/42201	STAPLES, FRANCIS TIMOTHY
3	72/42215	TRACE, DAVID JOHN
3	72/42218	VICKERS, JOHN JAMES
3	72/42223	WILKINSON, IAN DAVID

COURSE CODE 052

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
3	72/42051	ADAMSON, COLIN TREVOR
3	72/42069	BOWES, DEREK ARTHUR
3	72/42075	CARRUTHERS, PAUL
3	72/42083	COOKE, NORMAN MEREDITH
3	72/42161	MATTHEWS, TIMOTHY JOHN
3	72/42156	MCCALL, COLIN WINTER
3	72/42163	MERCER, NICHOLAS OLIVER ANDREW
3	72/42179	OUGHTON, DAVID
3	72/42181	PARTINGTON, PHILLIP ERIC
3	72/42186	RAW, BRIAN
3	72/42190	RIPPON, PHILIP DAVID BARTLE
3	72/42217	UNDERWOOD, HOWARD DOUGLAS
3	72/42221	WHITE, PHILIP WILLIAM
3	72/42225	WILKINSON, STEPHEN ARTHUR
3	72/42226	WILLIAMS, HOWARD MARTYN
3	72/42230	WILSON, KEVIN WILLIAM

COURSE CODE 057

DATE OF LISTING 22/08/73

TYPE	D.F.S.NO	NAME
3	72/42093	DENHAM, JAMES JOHN WILLIAM
3	72/42135	HOLROYD, ANDREW CHARLES
3	72/42158	MACLEOD, TORQUIL ROBERT
3	72/42207	SUMMERS, ROBERT CRUSE
3	72/42229	WILSON, JOHN MARK FLEMING

COURSE CODE C61

DATE OF LISTING 22/08/73

TYPE	D.F.S.NO	NAME
3	72/42058	BAMBEROUGH, PAUL ROBERT
3	72/42059	BARRETT, STEPHEN
3	72/42060	BARTHOLOMEW, GRAHAM JOHN
3	72/42064	BERRY, IVOR JAMES
3	72/42065	BETTLEY, ANDREW
3	72/42066	BEWICK, JAMES
3	72/42072	CAMPBELL, COLIN JOHN MCNAIR
3	72/42081	COLLINS, IAN FRANCIS
3	72/42082	COLLINSON, DAVID THOMAS
3	72/42102	FERGUSON, ANDREW MICHAEL
3	72/42106	FOWKES, ROGER CHARLES
3	72/42110	GARBUTT, IAN
3	72/42113	GILL, RAYMOND CHRISTOPHER
3	72/42114	GOODWIN, TOBY MICHAEL GILES
3	72/42119	GREEN, DAVID WILLIAM
3	72/42136	HOLYOME, JOHN MARK
3	72/42143	JOHN, DAVID NEIL
3	72/42145	JONES, KEITH
3	72/42147	KELLY, NEIL HAROLD
3	72/42148	KIDNEY, LANCE NICHOLAS
3	72/42154	LUMSDEN, DAVID WILLIAM
3	72/42155	LYNCH, CHRISTOPHER MICHAEL
3	72/42160	MARSHALL, STEPHEN JOHN
3	72/42166	MINTO, EDWARD JOHN
3	72/42168	MOONEY, JOHN FRANCIS
3	72/42234	PATMAN, BRIAN SAMUEL
3	72/42272	PEARCEY, PETER FRANK
3	72/42183	PILLEY, NICHOLAS JOHN
3	72/42191	ROWBOTHAM, PHILIP
3	72/42195	SIBBALD, JULIAN
3	72/42200	STANLEY-CLAMP, RICHARD STEPHEN INGLIS
3	72/42211	TAYLOR, TIMOTHY CHARLES HARCOURT
3	72/42212	THOMPSON, ROBERT
3	72/42213	THOMPSON, WILLIAM
3	72/42227	WILSON, ALAN HARDY

COURSE CODE 092

DATE OF LISTING 22/09/73

TYPE	D.F.S.NO	NAME
3	72/42062	BEANEY, MICHAEL JOSEPH
3	72/42071	BURGESS, KENNETH JOHN
3	71/42120	FOX, ROBERT
3	72/42220	WEST, JEREMY LEIGH

COURSE CODE 098

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
3	72/42074	CARRUTHERS, CHRISTOPHER JOHN
3	72/42094	DIXON, DAVID ANTON
3	72/42116	GOW, PETER DRAPPIN
3	72/42121	GUYMER, WILLIAM STEPHENSON
3	72/42130	HESKETH, NEIL WILLIAM
3	72/42198	SPEDDING, LEWIS CHRISTOPHER
3	72/42203	STEAD, JOHN CHESSWAS
3	72/42224	WILKINSON, JAMES EDWARD

COURSE CODE 134

DATE OF LISTING 22/08/73

TYPE	D.F.S.NO	NAME
2	72/42237	WILLIAMSON, KENNETH
3	72/42055	AUTON, RAYMOND
3	72/42063	BELL, JEFFREY LLOYD
3	72/42067	BLACKBURN, KEITH
3	72/42070	BROWN, PETER CHRISTOPHER
3	72/42086	CRAWSHAW, EVAN VAUGHAN
3	72/42087	CROOKS, CHRISTOPHER JOHN
3	72/42089	CURTIS, GRAHAM CLIFFORD
3	72/42100	FAULKS, MARTIN RICHARD
3	72/42103	FERRY, BRIAN WILLIAM
3	72/42137	HOUGHTON, TREVOR KEITH
3	72/42138	HOULDING, ROBIN RICHARD
3	72/42141	HUTCHINSON, STANLEY JAMES
3	72/42146	JONES, NORMAN
3	72/42152	LINE, STEPHEN CHARLES
3	72/42180	PARKER, JOHN DARRELL
3	72/42189	RILEY, STEWART
3	72/42193	SCOPER, ALEXANDER
3	72/42235	TEMPLE, ROBERT KEITH
3	72/42214	THWAITES, MARTIN JOSEPH EDWARD
3	72/42216	TREWARTHA, ALEC BLAIR

COURSE CODE 109

DATE OF LISTING 22/08/73

TYPE	D.E.S.NO	NAME
3	72/42098	EATON, GORDON STANLEY
3	72/42125	HANSEN, PAUL
3	72/42205	STEWART, JONATHAN MALCOLM

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME F
OPTION REQUESTED 1
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED *

72/42097 ERROR RETURNED.....FAMILY POSITION (10/14)

1972 FILE.....BEDE COLLEGE GENERAL FREQUENCIES

219 STUDENTS INCLUDED

71/42120	72/42051	72/42052	72/42053	72/42054	72/42055	72/42056	72/42057	72/42058	72/42059
72/42060	72/42061	72/42062	72/42063	72/42064	72/42065	72/42066	72/42067	72/42068	72/42069
72/42070	72/42071	72/42072	72/42073	72/42074	72/42075	72/42076	72/42077	72/42078	72/42079
72/42080	72/42081	72/42082	72/42083	72/42084	72/42085	72/42086	72/42087	72/42088	72/42089
72/42090	72/42091	72/42092	72/42093	72/42094	72/42095	72/42096	72/42097	72/42098	72/42099
72/42100	72/42101	72/42102	72/42103	72/42104	72/42105	72/42106	72/42107	72/42108	72/42109
72/42110	72/42111	72/42112	72/42113	72/42114	72/42115	72/42116	72/42117	72/42118	72/42119
72/42120	72/42121	72/42122	72/42123	72/42124	72/42125	72/42126	72/42127	72/42129	72/42130
72/42132	72/42133	72/42134	72/42135	72/42136	72/42137	72/42138	72/42139	72/42140	72/42141
72/42142	72/42143	72/42144	72/42145	72/42146	72/42147	72/42148	72/42149	72/42150	72/42151
72/42152	72/42153	72/42154	72/42155	72/42156	72/42157	72/42158	72/42160	72/42161	72/42162
72/42163	72/42164	72/42165	72/42166	72/42167	72/42168	72/42169	72/42170	72/42171	72/42172
72/42173	72/42174	72/42175	72/42176	72/42177	72/42178	72/42179	72/42180	72/42181	72/42182
72/42183	72/42184	72/42185	72/42186	72/42187	72/42188	72/42189	72/42190	72/42191	72/42192
72/42193	72/42195	72/42196	72/42197	72/42198	72/42199	72/42200	72/42201	72/42202	72/42203
72/42204	72/42205	72/42206	72/42207	72/42208	72/42209	72/42210	72/42211	72/42212	72/42213
72/42214	72/42215	72/42216	72/42217	72/42218	72/42219	72/42220	72/42221	72/42222	72/42223
72/42224	72/42225	72/42226	72/42227	72/42228	72/42229	72/42230	72/42231	72/42232	72/42233
72/42234	72/42235	72/42236	72/42237	72/42238	72/42239	72/42240	72/42241	72/42242	72/42243
72/42244	72/42245	72/42246	72/42247	72/42248	72/42249	72/42250	72/42251	72/42252	72/42253
72/42254	72/42255	72/42256	72/42257	72/42258	72/42259	72/42260	72/42261	72/42262	72/42263
72/42264	72/42265	72/42266	72/42267	72/42268	72/42269	72/42270	72/42271	72/42272	

FREQUENCY TABLE KEYS

MARITAL STATUS

1 SINGLE
2 MARRIED
3 DIVORCED/WID

GEOGRAPHICAL REGION

NOT CODED PRIOR TO 1972

A SCOTLAND AND NORTHERN IRELAND
B NORTHERN
C YORKSHIRE AND HUMBERSIDE
D NORTH WEST
E EAST MIDLAND
F WEST MIDLAND
G WALES AND MONMOUTH
H SOUTH WEST
I EAST ANGLIA
J SOUTH EAST
K OTHER/ABROAD

SETTLEMENT

1 ROADSIDE OR HAMLET I.E. LESS THAN 150 PEOPLE
2 VILLAGE OR SMALL TOWN I.E. 150 TO 2,500 PEOPLE
3 TOWN I.E. 2,500 TO 10,000 PEOPLE
4 LARGE TOWN I.E. 10,000 TO 500,000 PEOPLE
5 METROPOLIS I.E. OVER 500,000 PEOPLE
6 N/A

SETTLEMENT PRIOR TO 1972

1 METROPOLITAN (LARGE CITY CENTRE)
2 SUBURBAN (OUTSKIRTS OF LARGE CITY)
3 RURAL URBAN (SMALL TOWN OR VILLAGE)
4 RURAL

OCCUPATION IF OVER 25

0 OTHER
1 TEACHING OR YOUTH SERVICE
2 CENTRAL OR LOCAL GOVERNMENT
3 H.M. FORCES
4 POLICE
5 NURSING
6 OTHER MEDICAL OR WELFARE SERVICE
7 BANKING/INSURANCE/ACCOUNTANCY
8 COMMERCE
9 INDUSTRY

STUDENT/PARENT/GUARDIAN OCCUPATION

1 HIGHER ADMINISTRATIVE PROFESSIONAL AND MANAGERIAL
2 FARMERS, INTERMEDIATE ADMINISTRATIVE PROFESSIONAL & MANAGERIAL
3 ROUTINE NON-MANUAL AND SERVICE. MANUAL SKILLED
4 SEMI-SKILLED WORKERS
5 UNSKILLED WORKERS

RELIGION

1 C. OF E.
2 R.C.
3 NONCONFORMIST
4 OTHER RELIGICN
5 HUMANIST/AGNOSTIC/ATHEIST

SCHOOL TYPE

1 INDEPENDENT OR DIRECT GRANT SCHOOL...MIXED
2 INDEPENDENT OR DIRECT GRANT SCHOOL...SINGLE SEX
3 GRAMMAR...MIXED
4 GRAMMAR...SINGLE SEX
5 COMPREHENSIVE...MIXED
6 COMPREHENSIVE ...SINGLE SEX
7 SEC. MODERN SCHOOL...MIXED
8 SEC. MODERN SCHOOL...SINGLE SEX
9 COLL. OF FURTHER EDUCATION/TECHNICAL COLL

STUDENT/PARENT/GUARDIAN OCCUPATION PRIOR TO 1972

1 PROFESSIONAL AND HIGH ADMINISTRATIVE
2 MANAGERIAL AND EXECUTIVE
3 NON-MANUAL, HIGHER GRADE
4 NON-MANUAL, LOWER GRADE
5 SKILLED MANUAL
6 SEMI-SKILLED MANUAL

RELIGION PRIOR TO 1972

1 C. OF E.
2 R.C.
3 NONCONFORMIST
4 JEWISH
5 AGNOSTIC/ATHEIST
6 HUMANIST/OTHER

FREQUENCIES (TOTALS PERCENTAGES)

SEX FREQUENCY

MALE	219	100.00
FEMALE	0	0.00

AGE FREQUENCY ON ADMISSION

UNDER 25	185	84.47
OVER 25	34	15.52

AVERAGE AGE ON ADMISSION

21

MARITAL STATUS

1	186	84.93
2	33	15.07
3	0	0.00

GEOGRAPHICAL REGION

A	10	4.57
B	96	43.84
C	26	11.87
D	24	10.96
E	11	5.02
F	2	0.91
G	1	0.46
H	9	4.11
I	3	1.37
J	34	15.52
K	3	1.37

NOT CODED PRIOR TO 1972

SETTLEMENT

1	5	2.28
2	42	19.18
3	45	20.55
4	98	44.75
5	27	12.33
6	2	0.91

SETTLEMENT PRIOR TO 1972

1	0	0.00
2	0	0.00
3	0	0.00
4	0	0.00

OCCUPATION IF OVER 25

0	8	3.65
1	1	0.46
2	3	1.37
3	1	0.46
4	0	0.00
5	0	0.00
6	1	0.46
7	2	0.91
8	3	1.37
9	15	6.85
UNDER 25	185	84.47

STUDENT OCCUPATION			PARENT/GUARDIAN OCCUPATION	
1	5	2.28	45	20.55
2	32	14.61	78	35.62
3	26	11.87	72	32.88
4	2	0.91	11	5.02
5	2	0.91	13	5.94
@ SCHOOL	152	69.41		

STUDENT OCCUPATION PRIOR TO 1972			PARENT/GUARDIAN OCCUPATION PRIOR TO 1972	
1	0	0.00	0	0.00
2	0	0.00	0	0.00
3	0	0.00	0	0.00
4	0	0.00	0	0.00
5	0	0.00	0	0.00
6	0	0.00	0	0.00
	0	0.00		

SCHOOL TYPE

1	10	4.57
2	58	26.48
3	71	32.42
4	68	31.05
5	32	14.61
6	4	1.83
7	34	15.52
8	18	8.22
9	93	42.47
UNKNOWN	18	8.22

RELIGION

1	135	61.64
2	13	5.94
3	25	11.42
4	18	8.22
5	28	12.79

RELIGION PRIOR TO 1972

1	0	0.00
2	0	0.00
3	0	0.00
4	0	0.00
5	0	0.00
6	0	0.00

POSITION IN FAMILY

	1	2	3	4	5	6	7	8	9	10
1	30	48	20	12	2	1	1	0	0	0
2	X	33	20	7	3	0	0	0	0	0
3	X	X	19	2	1	1	0	1	0	0
4	X	X	X	6	3	1	0	0	1	0
5	X	X	X	X	2	0	1	0	0.	0
6	X	X	X	X	X	0	0	0	1	0
7	X	X	X	X	X	X	0	0	0	0
8	X	X	X	X	X	X	X	0	1	0
9	X	X	X	X	X	X	X	X	1	0
10	X	X	X	X	X	X	X	X	X	0

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME F
OPTION REQUESTED 2
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED *

FREQUENCY COUNT FOR REQUEST 1<21

144 STUDENTS INCLUDED

71/42120	72/42051	72/42053	72/42054	72/42056	72/42058	72/42059	72/42060	72/42061	72/42062
72/42063	72/42064	72/42065	72/42068	72/42069	72/42070	72/42071	72/42072	72/42073	72/42074
72/42076	72/42078	72/42080	72/42081	72/42082	72/42083	72/42084	72/42085	72/42087	72/42089
72/42090	72/42091	72/42092	72/42093	72/42094	72/42095	72/42096	72/42098	72/42099	72/42100
72/42101	72/42104	72/42105	72/42106	72/42107	72/42109	72/42110	72/42112	72/42113	72/42116
72/42117	72/42118	72/42119	72/42120	72/42122	72/42123	72/42124	72/42125	72/42127	72/42129
72/42130	72/42132	72/42133	72/42134	72/42135	72/42136	72/42137	72/42138	72/42139	72/42140
72/42141	72/42142	72/42143	72/42144	72/42145	72/42146	72/42147	72/42148	72/42150	72/42151
72/42152	72/42153	72/42158	72/42160	72/42161	72/42162	72/42163	72/42165	72/42166	72/42167
72/42168	72/42169	72/42171	72/42172	72/42173	72/42174	72/42175	72/42176	72/42177	72/42178
72/42179	72/42180	72/42181	72/42182	72/42183	72/42184	72/42186	72/42187	72/42188	72/42191
72/42192	72/42193	72/42195	72/42198	72/42200	72/42201	72/42202	72/42203	72/42204	72/42206
72/42207	72/42208	72/42209	72/42210	72/42211	72/42213	72/42214	72/42215	72/42216	72/42217
72/42218	72/42220	72/42221	72/42223	72/42225	72/42226	72/42228	72/42229	72/42230	72/42231
72/42232	72/42233	72/42234	72/42235						

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME F
OPTION REQUESTED 2
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED *
FREQUENCY TABLES REQUIRED

FREQUENCY COUNT FOR REQUEST 1<21

144 STUDENTS INCLUDED

71/42120	72/42051	72/42053	72/42054	72/42056	72/42058	72/42059	72/42060	72/42061	72/42062
72/42063	72/42064	72/42065	72/42068	72/42069	72/42070	72/42071	72/42072	72/42073	72/42074
72/42076	72/42078	72/42080	72/42081	72/42082	72/42083	72/42084	72/42085	72/42087	72/42089
72/42090	72/42091	72/42092	72/42093	72/42094	72/42095	72/42096	72/42098	72/42099	72/42100
72/42101	72/42104	72/42105	72/42106	72/42107	72/42109	72/42110	72/42112	72/42113	72/42116
72/42117	72/42118	72/42119	72/42120	72/42122	72/42123	72/42124	72/42125	72/42127	72/42129
72/42130	72/42132	72/42133	72/42134	72/42135	72/42136	72/42137	72/42138	72/42139	72/42140
72/42141	72/42142	72/42143	72/42144	72/42145	72/42146	72/42147	72/42148	72/42150	72/42151
72/42152	72/42153	72/42158	72/42160	72/42161	72/42162	72/42163	72/42165	72/42166	72/42167
72/42168	72/42169	72/42171	72/42172	72/42173	72/42174	72/42175	72/42176	72/42177	72/42178
72/42179	72/42180	72/42181	72/42182	72/42183	72/42184	72/42186	72/42187	72/42188	72/42191
72/42192	72/42193	72/42195	72/42198	72/42200	72/42201	72/42202	72/42203	72/42204	72/42206
72/42207	72/42208	72/42209	72/42210	72/42211	72/42213	72/42214	72/42215	72/42216	72/42217
72/42218	72/42220	72/42221	72/42223	72/42225	72/42226	72/42228	72/42229	72/42230	72/42231
72/42232	72/42233	72/42234	72/42235						

FREQUENCY TABLE KEYS

MARITAL STATUS

1	SINGLE
2	MARRIED
3	DIVORCED/WID

GEOGRAPHICAL REGION

A	SCOTLAND AND NORTHERN IRELAND
B	NORTHERN
C	YORKSHIRE AND HUMBERSIDE
D	NORTH WEST
E	EAST MIDLAND
F	WEST MIDLAND
G	WALES AND MONMOUTH
H	SOUTH WEST
I	EAST ANGLIA
J	SOUTH EAST
K	OTHER/ABROAD

NOT CODED PRIOR TO 1972

SETTLEMENT

1	ROADSIDE OR HAMLET I.E. LESS THAN 150 PEOPLE
2	VILLAGE OR SMALL TOWN I.E. 150 TO 2,500 PEOPLE
3	TOWN I.E. 2,500 TO 10,000 PEOPLE
4	LARGE TOWN I.E. 10,000 TO 500,000 PEOPLE
5	METROPOLIS I.E. OVER 500,000 PEOPLE
6	N/A

SETTLEMENT PRIOR TO 1972

1	METROPOLITAN (LARGE CITY CENTRE)
2	SUBURBAN (OUTSKIRTS OF LARGE CITY)
3	RURAL URBAN (SMALL TOWN OR VILLAGE)
4	RURAL

OCCUPATION IF OVER 25

0	OTHER
1	TEACHING OR YOUTH SERVICE
2	CENTRAL OR LOCAL GOVERNMENT
3	H.M. FORCES
4	POLICE
5	NURSING
6	OTHER MEDICAL OR WELFARE SERVICE
7	BANKING/INSURANCE/ACCOUNTANCY
8	COMMERCE
9	INDUSTRY

STUDENT/PARENT/GUARDIAN OCCUPATION

1 HIGHER ADMINISTRATIVE PROFESSIONAL AND MANAGERIAL
2 FARMERS, INTERMEDIATE ADMINISTRATIVE PROFESSIONAL & MANAGERIAL
3 ROUTINE NON-MANUAL AND SERVICE. MANUAL SKILLED
4 SEMI-SKILLED WORKERS
5 UNSKILLED WORKERS

RELIGION

1 C. OF E.
2 R.C.
3 NONCONFORMIST
4 OTHER RELIGION
5 HUMANIST/AGNOSTIC/ATHEIST

SCHOOL TYPE

1 INDEPENDENT OR DIRECT GRANT SCHOOL...MIXED
2 INDEPENDENT OR DIRECT GRANT SCHOOL...SINGLE SEX
3 GRAMMAR...MIXED
4 GRAMMAR...SINGLE SEX
5 COMPREHENSIVE...MIXED
6 COMPREHENSIVE ...SINGLE SEX
7 SEC. MODERN SCHOOL...MIXED
8 SEC. MODERN SCHOOL...SINGLE SEX
9 COLL. OF FURTHER EDUCATION/TECHNICAL COLL

STUDENT/PARENT/GUARDIAN OCCUPATION PRIOR TO 1972

1 PROFESSIONAL AND HIGH ADMINISTRATIVE
2 MANAGERIAL AND EXECUTIVE
3 NON-MANUAL, HIGHER GRADE
4 NON-MANUAL, LOWER GRADE
5 SKILLED MANUAL
6 SEMI-SKILLED MANUAL

RELIGION PRIOR TO 1972

1 C. OF E.
2 R.C.
3 NONCONFORMIST
4 JEWISH
5 AGNOSTIC/ATHEIST
6 HUMANIST/OTHER

FREQUENCIES (TOTALS PERCENTAGES)

SEX FREQUENCY

MALE	144	100.00
FEMALE	0	0.00

AGE FREQUENCY ON ADMISSION

UNDER 25	144	100.00
OVER 25	0	0.00

AVERAGE AGE ON ADMISSION

19

MARITAL STATUS

1	144	100.00
2	0	0.00
3	0	0.00

GEOGRAPHICAL REGION

A	9	6.25
B	61	42.36
C	16	11.11
D	19	13.19
E	8	5.55
F	1	0.69
G	1	0.69
H	3	2.08
I	2	1.39
J	23	15.97
K	1	0.69

NOT CODED PRIOR TO 1972

SETTLEMENT

1	3	2.08
2	30	20.83
3	28	19.44
4	65	45.14
5	17	11.80
6	1	0.69

SETTLEMENT PRIOR TO 1972

1	0	0.00
2	0	0.00
3	0	0.00
4	0	0.00

OCCUPATION IF OVER 25

0	0	0.00
1	0	0.00
2	0	0.00
3	0	0.00
4	0	0.00
5	0	0.00
6	0	0.00
7	0	0.00
8	0	0.00
9	0	0.00
UNDER 25	144	100.00

STUDENT OCCUPATION			PARENT/GUARDIAN OCCUPATION	
1	0	0.00	33	22.92
2	7	4.86	57	39.58
3	6	4.17	41	28.47
4	1	0.69	7	4.86
5	1	0.69	6	4.17
@ SCHOOL	129	89.58		

STUDENT OCCUPATION PRIOR TO 1972			PARENT/GUARDIAN OCCUPATION PRIOR TO 1972	
1	0	0.00	0	0.00
2	0	0.00	0	0.00
3	0	0.00	0	0.00
4	0	0.00	0	0.00
5	0	0.00	0	0.00
6	0	0.00	0	0.00
	0	0.00		

SCHOOL TYPE

1	5	3.47
2	33	22.92
3	53	36.80
4	46	31.94
5	30	20.83
6	4	2.78
7	23	15.97
8	8	5.55
9	22	15.28
UNKNOWN	7	4.86

RELIGION

1	98	68.05
2	6	4.17
3	13	9.03
4	14	9.72
5	13	9.03

RELIGION PRIOR TO 1972

1	0	0.00
2	0	0.00
3	0	0.00
4	0	0.00
5	0	0.00
6	0	0.00

POSITION IN FAMILY

	1	2	3	4	5	6	7	8	9	10
1	22	30	17	11	1	1	1	0	0	0
2	X	21	13	5	2	0	0	0	0	0
3	X	X	10	2	1	1	0	1	0	0
4	X	X	X	3	1	0	0	0	0	0
5	X	X	X	X	0	0	1	0	0	0
6	X	X	X	X	X	0	0	0	0	0
7	X	X	X	X	X	X	0	0	0	0
8	X	X	X	X	X	X	X	0	0	0
9	X	X	X	X	X	X	X	X	0	0
10	X	X	X	X	X	X	X	X	X	0

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME F
OPTION REQUESTED 2
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED *

FREQUENCY COUNT FOR REQUEST 1<25 & 3=2

8 STUDENTS INCLUDED

72/42157 72/42170 72/42196 72/42222 72/42255 72/42257 72/42261 72/42266

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME I
OPTION REQUESTED 1
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED *
YEAR OF TEST F

71/42120 HAS NO CORRESPONDING DATA
72/42092 HAS NO CORRESPONDING DATA
72/42094 HAS NO CORRESPONDING DATA
72/42114 HAS NO CORRESPONDING DATA
72/42147 HAS NO CORRESPONDING DATA
72/42175 HAS NO CORRESPONDING DATA
72/42176 HAS NO CORRESPONDING DATA
72/42191 HAS NO CORRESPONDING DATA
72/42203 HAS NO CORRESPONDING DATA
72/42236 HAS NO CORRESPONDING DATA
72/42251 HAS NO CORRESPONDING DATA
72/42252 HAS NO CORRESPONDING DATA
72/42253 HAS NO CORRESPONDING DATA
72/42254 HAS NO CORRESPONDING DATA
72/42255 HAS NO CORRESPONDING DATA
72/42256 HAS NO CORRESPONDING DATA
72/42257 HAS NO CORRESPONDING DATA
72/42258 HAS NO CORRESPONDING DATA
72/42259 HAS NO CORRESPONDING DATA
72/42260 HAS NO CORRESPONDING DATA
72/42261 HAS NO CORRESPONDING DATA
72/42262 HAS NO CORRESPONDING DATA
72/42263 HAS NO CORRESPONDING DATA
72/42264 HAS NO CORRESPONDING DATA
72/42265 HAS NO CORRESPONDING DATA
72/42266 HAS NO CORRESPONDING DATA
72/42267 HAS NO CORRESPONDING DATA
72/42268 HAS NO CORRESPONDING DATA
72/42269 HAS NO CORRESPONDING DATA
72/42270 HAS NO CORRESPONDING DATA
72/42271 HAS NO CORRESPONDING DATA

1972 FILE.....INTELLIGENCE TEST TOTALS (MEANS) ST. DEV

188 STUDENTS INCLUDED

72/42051	72/42052	72/42053	72/42054	72/42055	72/42056	72/42057	72/42058	72/42059	72/42060
72/42061	72/42062	72/42063	72/42064	72/42065	72/42066	72/42067	72/42068	72/42069	72/42070
72/42071	72/42072	72/42073	72/42074	72/42075	72/42076	72/42077	72/42078	72/42079	72/42080
72/42081	72/42082	72/42083	72/42084	72/42085	72/42086	72/42087	72/42088	72/42089	72/42090
72/42091	72/42093	72/42095	72/42096	72/42097	72/42098	72/42099	72/42100	72/42101	72/42102
72/42103	72/42104	72/42105	72/42106	72/42107	72/42108	72/42109	72/42110	72/42111	72/42112
72/42113	72/42115	72/42116	72/42117	72/42118	72/42119	72/42120	72/42121	72/42122	72/42123
72/42124	72/42125	72/42126	72/42127	72/42129	72/42130	72/42132	72/42133	72/42134	72/42135
72/42136	72/42137	72/42138	72/42139	72/42140	72/42141	72/42142	72/42143	72/42144	72/42145

72/42146	72/42148	72/42149	72/42150	72/42151	72/42152	72/42153	72/42154	72/42155	72/42156
72/42157	72/42158	72/42159	72/42161	72/42162	72/42163	72/42164	72/42165	72/42166	72/42167
72/42168	72/42169	72/42170	72/42171	72/42172	72/42173	72/42174	72/42177	72/42178	72/42179
72/42180	72/42181	72/42182	72/42183	72/42184	72/42185	72/42186	72/42187	72/42188	72/42189
72/42190	72/42192	72/42193	72/42195	72/42196	72/42197	72/42198	72/42199	72/42200	72/42201
72/42202	72/42204	72/42205	72/42206	72/42207	72/42208	72/42209	72/42210	72/42211	72/42212
72/42213	72/42214	72/42215	72/42216	72/42217	72/42218	72/42219	72/42220	72/42221	72/42222
72/42223	72/42224	72/42225	72/42226	72/42227	72/42228	72/42229	72/42230	72/42231	72/42232
72/42233	72/42234	72/42235	72/42237	72/42238	72/42239	72/42240	72/42241	72/42242	72/42243
72/42244	72/42245	72/42246	72/42247	72/42248	72/42249	72/42250	72/42272		

TEST	TOTAL(MEAN)ST.DEV	TOTAL(MEAN)ST.DEV
VALUES(1)	6014(31.99) 6.02	
(2)	5894(31.35) 7.19	
(3)	4983(26.51) 8.21	
(4)	7714(41.03) 6.45	
(5)	4632(24.64) 6.35	
(6)	4433(23.58) 11.51	
D.SCALE	29621(157.56) 23.42	
A.H.4(1)	8020(42.66) 6.75	
(2)	9333(49.64) 8.49	STEN SCORES
P.F.16(A)	1992(10.60) 2.91	1148(6.11) 1.68
(B)	1544(8.21) 2.02	1153(6.13) 2.15
(C)	2627(13.97) 3.82	889(4.73) 1.96
(E)	2570(13.67) 4.22	1011(5.38) 2.11
(F)	3111(16.55) 4.91	1152(6.13) 2.27
(G)	2191(11.65) 4.01	936(4.98) 2.14
(H)	2364(12.57) 5.13	995(5.29) 1.90
(I)	2105(11.20) 3.35	1290(6.86) 1.86
(L)	1892(10.06) 3.47	1104(5.87) 2.16
(M)	2405(12.79) 3.44	1152(6.13) 1.89
(N)	1807(9.61) 3.18	859(4.57) 2.26
(O)	2024(10.77) 4.10	1121(5.96) 2.23
(Q1)	1919(10.21) 2.83	1121(5.96) 2.02
(Q2)	1966(10.46) 3.63	1107(5.89) 1.99
(Q3)	1831(9.74) 3.18	994(5.29) 1.98
(Q4)	2337(12.43) 4.79	1049(5.58) 1.95

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME I
OPTION REQUESTED 2
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED +
YEAR OF TEST F

71/42120 HAS NO CORRESPONDING DATA
72/42092 HAS NO CORRESPONDING DATA
72/42094 HAS NO CORRESPONDING DATA
72/42114 HAS NO CORRESPONDING DATA
72/42147 HAS NO CORRESPONDING DATA
72/42175 HAS NO CORRESPONDING DATA
72/42176 HAS NO CORRESPONDING DATA
72/42191 HAS NO CORRESPONDING DATA
72/42203 HAS NO CORRESPONDING DATA
72/42236 HAS NO CORRESPONDING DATA
72/42251 HAS NO CORRESPONDING DATA
72/42252 HAS NO CORRESPONDING DATA
72/42253 HAS NO CORRESPONDING DATA
72/42254 HAS NO CORRESPONDING DATA
72/42255 HAS NO CORRESPONDING DATA
72/42256 HAS NO CORRESPONDING DATA
72/42257 HAS NO CORRESPONDING DATA
72/42258 HAS NO CORRESPONDING DATA
72/42259 HAS NO CORRESPONDING DATA
72/42260 HAS NO CORRESPONDING DATA
72/42261 HAS NO CORRESPONDING DATA
72/42262 HAS NO CORRESPONDING DATA
72/42263 HAS NO CORRESPONDING DATA
72/42264 HAS NO CORRESPONDING DATA
72/42265 HAS NO CORRESPONDING DATA
72/42266 HAS NO CORRESPONDING DATA
72/42267 HAS NO CORRESPONDING DATA
72/42268 HAS NO CORRESPONDING DATA
72/42269 HAS NO CORRESPONDING DATA
72/42270 HAS NO CORRESPONDING DATA
72/42271 HAS NO CORRESPONDING DATA

1972 FILE.....INTELLIGENCE TEST FREQUENCY COUNTS

188 STUDENTS INCLUDED

72/42051	72/42052	72/42053	72/42054	72/42055	72/42056	72/42057	72/42058	72/42059	72/42060
72/42061	72/42062	72/42063	72/42064	72/42065	72/42066	72/42067	72/42068	72/42069	72/42070
72/42071	72/42072	72/42073	72/42074	72/42075	72/42076	72/42077	72/42078	72/42079	72/42080
72/42081	72/42082	72/42083	72/42084	72/42085	72/42086	72/42087	72/42088	72/42089	72/42090
72/42091	72/42093	72/42095	72/42096	72/42097	72/42098	72/42099	72/42100	72/42101	72/42102
72/42103	72/42104	72/42105	72/42106	72/42107	72/42108	72/42109	72/42110	72/42111	72/42112
72/42113	72/42115	72/42116	72/42117	72/42118	72/42119	72/42120	72/42121	72/42122	72/42123
72/42124	72/42125	72/42126	72/42127	72/42129	72/42130	72/42132	72/42133	72/42134	72/42135
72/42136	72/42137	72/42138	72/42139	72/42140	72/42141	72/42142	72/42143	72/42144	72/42145

72/42146	72/42148	72/42149	72/42150	72/42151	72/42152	72/42153	72/42154	72/42155	72/42156
72/42157	72/42158	72/42160	72/42161	72/42162	72/42163	72/42164	72/42165	72/42166	72/42167
72/42168	72/42169	72/42170	72/42171	72/42172	72/42173	72/42174	72/42177	72/42178	72/42179
72/42180	72/42181	72/42182	72/42183	72/42184	72/42185	72/42186	72/42187	72/42188	72/42189
72/42190	72/42192	72/42193	72/42195	72/42196	72/42197	72/42198	72/42199	72/42200	72/42201
72/42202	72/42204	72/42205	72/42206	72/42207	72/42208	72/42209	72/42210	72/42211	72/42212
72/42213	72/42214	72/42215	72/42216	72/42217	72/42218	72/42219	72/42220	72/42221	72/42222
72/42223	72/42224	72/42225	72/42226	72/42227	72/42228	72/42229	72/42230	72/42231	72/42232
72/42233	72/42234	72/42235	72/42237	72/42238	72/42239	72/42240	72/42241	72/42242	72/42243
72/42244	72/42245	72/42246	72/42247	72/42248	72/42249	72/42250	72/42272		

VALUES SCORE	FREQUENCY					
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	3
7	0	0	0	0	0	3
8	0	0	0	0	1	3
9	0	0	0	0	0	6
10	0	0	0	1	1	5
11	0	0	0	3	0	1
12	0	0	0	3	0	5
13	1	0	0	3	0	3
14	0	0	0	3	5	6
15	0	1	0	4	2	8
16	0	3	0	4	3	10
17	0	3	0	4	7	8
18	2	1	0	9	9	10
19	2	1	0	6	9	9
20	1	4	0	6	15	5
21	2	5	0	4	7	4
22	1	4	0	11	11	9
23	3	5	0	11	9	3
24	4	6	0	13	12	10
25	7	14	0	8	15	5
26	14	3	0	7	10	4
27	8	7	1	9	12	6
28	13	10	0	6	13	5
29	13	8	3	6	4	5
30	6	11	2	10	6	2
31	11	3	5	5	4	3
32	13	10	6	8	9	5
33	10	14	4	6	9	6
34	7	13	9	4	3	3
35	9	8	6	8	0	1
36	14	11	5	4	3	2
37	12	5	8	1	3	6
38	10	8	6	2	0	2
39	9	4	12	3	2	1
40	4	4	9	3	0	1
41	2	5	11	3	0	0
42	3	4	17	5	0	3
43	4	1	12	1	2	0
44	2	6	14	2	0	1
45	0	3	14	1	0	1
46	1	2	5	0	0	1
47	0	1	11	0	0	2
48	1	0	4	0	0	1
49	0	0	9	1	0	1
50	0	0	5	0	0	0
51	0	0	2	0	0	1
53	0	0	2	0	0	2
54	0	0	2	0	0	3
55	0	0	2	0	0	0
56	0	0	0	0	0	1

D.SCALE SCORE	FREQUENCY
89	1
93	1
95	1
96	1
103	1
109	1
111	2
114	1
116	3
118	1
119	1
125	1
128	1
129	2
130	2
131	1
132	3
133	2
134	2
135	1
136	3
137	3
138	3
139	2
140	2
141	2
142	4
143	2
145	1
146	4
147	2
148	2
149	4
150	1
151	3
152	4
153	5
154	2
155	3
156	2
157	2
158	5
159	2
160	6
161	5
162	4
163	3
164	3
165	7
166	3
167	4
168	5
169	4
170	4
171	2
173	1
174	1

175	4
176	1
177	3
179	4
180	2
181	1
182	2
183	1
184	1
185	3
186	3
187	2
188	2
189	3
190	2
191	2
192	2
195	1
200	1
203	1
204	1
213	1
237	1

A.H.4

SCORE	FREQUENCY	
19	1	0
23	1	1
24	1	0
27	1	0
28	1	0
29	3	1
30	2	0
31	1	1
32	0	2
33	6	5
34	5	0
35	3	6
36	7	3
37	6	1
38	6	6
39	10	1
40	8	2
41	10	4
42	8	6
43	15	7
44	21	4
45	12	6
46	14	4
47	6	4
48	9	9
49	5	8
50	5	7
51	5	15
52	5	8
53	3	12
54	3	4
55	0	9
56	0	6
57	1	12
58	2	3
59	1	8
60	1	3
61	0	7
62	0	11
63	0	1
64	0	1

P.F.16

SCORE	FREQUENCY															
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	1	0	1	3	0	0	0	0
2	0	0	0	0	1	1	2	0	1	0	1	2	0	2	2	3
3	1	1	0	1	0	2	4	0	1	0	2	2	0	0	2	1
4	4	5	0	1	1	8	5	3	3	2	9	5	3	3	4	2
5	4	12	1	0	0	5	8	7	12	1	3	9	3	6	5	6
6	6	18	4	3	2	2	6	9	17	1	19	4	12	15	21	7
7	10	25	8	7	5	8	5	10	16	4	15	13	16	20	16	9
8	21	40	1	10	6	13	8	13	14	11	17	16	23	17	18	16
9	17	35	7	12	4	13	14	15	12	13	20	15	19	18	21	10
10	33	29	15	6	9	24	13	22	30	21	23	21	28	15	25	16
11	15	12	12	18	5	9	18	20	12	9	27	24	22	18	18	18
12	31	10	16	20	7	19	10	19	25	28	20	15	24	24	16	14
13	12	0	26	19	9	18	10	19	10	24	10	9	13	15	13	12
14	19		12	16	10	19	13	23	14	22	11	19	11	11	13	8
15	9		19	15	13	8	13	2	5	13	4	7	8	2	8	12
16	3		17	12	13	15	15	19	9	13	3	7	3	8	4	13
17	1		20	15	15	12	12	3	5	6	2	5	2	6	2	11
18	1		11	10	11	7	8	3	1	9	0	4	1	5	0	7
19	1		5	3	15	1	7	1	0	5	1	4	0	3	0	10
20	0		3	6	19	3	4	0	0	3	0	3	0	0	0	4
21			4	6	11		3			0		1				3
22			4	3	11		4			3		0				2
23			3	3	13		1			0		0				3
24			0	1	6		3			0		0				0
25			0	1	1		1			0		0				0
26			0	0	1		0			0		0				1

STEN SCORE	FREQUENCY															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	7	13	5	2	12	3	0	3	3	16	7	3	2	4	4
2	4	12	8	17	13	7	9	0	3	1	34	5	3	3	9	2
3	10	0	27	12	13	34	19	10	29	15	17	13	12	21	21	22
4	10	18	42	24	21	33	35	9	16	13	20	29	39	20	34	26
5	38	25	31	39	23	19	38	23	26	30	23	15	19	35	46	48
6	48	40	37	43	28	37	41	37	42	52	47	45	28	33	18	20
7	43	35	16	25	26	23	20	39	25	35	10	24	46	39	29	36
8	19	29	7	3	30	12	11	42	24	19	15	26	13	13	13	17
9	12	12	4	12	24	8	7	2	5	9	3	12	19	14	12	7
10	3	10	3	8	8	3	5	26	15	11	3	12	6	8	2	6

DATE OF LISTING 22/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME I
OPTION REQUESTED 4
YEAR REQUESTED 1972

WRL REQUESTED 111
WCR REQUESTED *
YEAR OF TEST F

71/42120 HAS NO CORRESPONDING DATA
72/42092 HAS NO CORRESPONDING DATA
72/42094 HAS NO CORRESPONDING DATA
72/42114 HAS NO CORRESPONDING DATA
72/42147 HAS NO CORRESPONDING DATA
72/42175 HAS NO CORRESPONDING DATA
72/42176 HAS NO CORRESPONDING DATA
72/42191 HAS NO CORRESPONDING DATA
72/42203 HAS NO CORRESPONDING DATA
72/42236 HAS NO CORRESPONDING DATA
72/42251 HAS NO CORRESPONDING DATA
72/42252 HAS NO CORRESPONDING DATA
72/42253 HAS NO CORRESPONDING DATA
72/42254 HAS NO CORRESPONDING DATA
72/42255 HAS NO CORRESPONDING DATA
72/42256 HAS NO CORRESPONDING DATA
72/42257 HAS NO CORRESPONDING DATA
72/42258 HAS NO CORRESPONDING DATA
72/42259 HAS NO CORRESPONDING DATA
72/42260 HAS NO CORRESPONDING DATA
72/42261 HAS NO CORRESPONDING DATA
72/42262 HAS NO CORRESPONDING DATA
72/42263 HAS NO CORRESPONDING DATA
72/42264 HAS NO CORRESPONDING DATA
72/42265 HAS NO CORRESPONDING DATA
72/42266 HAS NO CORRESPONDING DATA
72/42267 HAS NO CORRESPONDING DATA
72/42268 HAS NO CORRESPONDING DATA
72/42269 HAS NO CORRESPONDING DATA
72/42270 HAS NO CORRESPONDING DATA
72/42271 HAS NO CORRESPONDING DATA

TOTALS,MEANS,STANDARD DEVIATIONS AND FREQUENCY COUNTS FOR REQUEST 1<21

136 STUDENTS INCLUDED

72/42051	72/42053	72/42054	72/42056	72/42058	72/42059	72/42060	72/42061	72/42062	72/42063
72/42064	72/42065	72/42068	72/42069	72/42070	72/42071	72/42072	72/42073	72/42074	72/42076
72/42078	72/42080	72/42081	72/42082	72/42083	72/42084	72/42085	72/42087	72/42089	72/42090
72/42091	72/42093	72/42095	72/42096	72/42098	72/42099	72/42100	72/42101	72/42104	72/42105
72/42106	72/42107	72/42109	72/42110	72/42112	72/42113	72/42116	72/42117	72/42118	72/42119
72/42120	72/42122	72/42123	72/42124	72/42125	72/42127	72/42129	72/42130	72/42132	72/42133
72/42134	72/42135	72/42136	72/42137	72/42138	72/42139	72/42140	72/42141	72/42142	72/42143
72/42144	72/42145	72/42146	72/42148	72/42150	72/42151	72/42152	72/42153	72/42158	72/42160
72/42161	72/42162	72/42163	72/42165	72/42166	72/42167	72/42168	72/42169	72/42171	72/42172

72/42173	72/42174	72/42177	72/42178	72/42179	72/42180	72/42181	72/42182	72/42183	72/42184
72/42186	72/42187	72/42188	72/42192	72/42193	72/42195	72/42198	72/42200	72/42201	72/42202
72/42204	72/42206	72/42207	72/42208	72/42209	72/42210	72/42211	72/42213	72/42214	72/42215
72/42216	72/42217	72/42218	72/42220	72/42221	72/42223	72/42225	72/42226	72/42228	72/42229
72/42230	72/42231	72/42232	72/42233	72/42234	72/42235				

TEST	TOTAL(MEAN)ST.DEV	TOTAL(MEAN)ST.DEV
VALUES(1)	4316(31.74) 5.93	
(2)	4318(31.75) 6.87	
(3)	3740(27.50) 7.64	
(4)	5556(40.85) 6.62	
(5)	3332(24.50) 6.37	
(6)	3112(22.88) 10.94	
D.SCALE	21830(160.51) 21.81	
A.H.4(1)	5837(42.92) 6.56	
(2)	6874(50.54) 7.90	STEN SCORES
P.F.16(A)	1430(10.51) 2.92	825(6.07) 1.70
(B)	1107(8.14) 1.96	826(6.07) 2.07
(C)	1912(14.06) 3.70	648(4.76) 1.90
(E)	1882(13.84) 4.18	745(5.48) 2.11
(F)	2302(16.93) 4.85	857(6.30) 2.25
(G)	1552(11.41) 3.99	665(4.89) 2.11
(H)	1688(12.41) 5.12	709(5.21) 1.90
(I)	1524(11.21) 3.38	933(6.86) 1.86
(L)	1426(10.49) 3.50	836(6.15) 2.17
(M)	1744(12.82) 3.44	837(6.15) 1.89
(N)	1297(9.54) 3.24	616(4.53) 2.28
(O)	1502(11.04) 3.91	832(6.12) 2.10
(Q1)	1381(10.15) 2.92	803(5.90) 2.08
(Q2)	1395(10.26) 3.52	785(5.77) 1.93
(Q3)	1281(9.42) 3.15	691(5.08) 1.97
(Q4)	1763(12.96) 4.59	790(5.81) 1.85

VALUES SCORE	FREQUENCY					
4	0	0	0	0	0	1
5	0	0	0	0	0	2
6	0	0	0	0	1	1
7	0	0	0	0	0	1
8	0	0	0	0	1	2
9	0	0	0	0	0	5
10	0	0	1	0	1	4
11	0	0	0	0	1	1
12	0	0	1	0	0	3
13	1	0	2	0	0	3
14	0	0	0	0	4	5
15	0	0	1	2	1	8
16	0	1	4	0	3	7
17	0	3	1	0	4	5
18	1	1	4	0	7	6
19	2	0	4	0	6	8
20	1	4	6	0	10	3
21	2	3	2	0	2	3
22	0	0	11	0	9	7
23	2	5	9	0	5	3
24	2	5	10	0	8	8
25	6	7	7	0	14	5
26	12	1	4	0	8	3
27	5	5	5	1	12	5
28	10	7	5	0	9	3
29	7	7	5	3	4	2
30	4	10	10	2	4	2
31	9	3	4	4	2	1
32	10	8	7	4	8	5
33	8	10	5	3	5	4
34	7	10	4	7	1	2
35	8	6	5	3	0	1
36	9	9	3	2	2	2
37	6	2	1	4	0	4
38	7	6	0	3	0	2
39	6	4	3	11	2	0
40	2	4	3	5	0	0
41	2	3	1	8	0	0
42	2	3	3	13	0	2
43	4	1	1	10	2	0
44	0	4	2	10	0	0
45	0	3	1	11	0	0
46	1	1	0	5	0	0
47	0	0	0	9	0	0
48	0	0	0	3	0	1
49	0	0	1	5	0	1
50	0	0	0	3	0	0
51	0	0	0	2	0	0
53	0	0	0	1	0	2
54	0	0	0	1	0	2
55	0	0	0	1	0	0
56	0	0	0	0	0	1

D.SCALE SCORE	FREQUENCY
89	1
96	1
109	1
111	1
116	1
119	1
128	1
129	1
130	1
131	1
132	1
133	1
134	1
135	1
136	1
137	1
138	3
139	2
140	1
141	2
142	3
143	2
145	1
146	3
147	1
148	2
149	1
150	1
151	1
152	3
153	5
154	2
155	2
156	2
157	1
158	5
159	1
160	5
161	4
162	3
163	3
164	3
165	5
166	3
167	3
168	3
169	3
170	4
171	2
173	1
174	1
175	2
176	1
177	3
179	3
180	1
181	1

182	2
183	1
184	1
185	3
186	3
187	1
188	1
190	1
191	1
192	2
195	1
200	1
203	1
204	1
213	1
237	1

A.H.4

SCORE	FREQUENCY	
19	1	0
23	1	1
24	1	0
27	1	0
29	2	1
30	1	0
33	3	2
34	4	0
35	3	3
36	4	1
37	5	0
38	4	4
39	6	1
40	5	0
41	3	3
42	4	5
43	13	7
44	19	2
45	10	6
46	10	2
47	6	3
48	7	7
49	4	7
50	4	4
51	5	12
52	4	6
53	3	10
54	1	2
55	0	5
56	0	6
57	1	9
58	1	3
59	0	7
60	0	2
61	0	6
62	0	7
63	0	1
64	0	1

P.F.16

SCORE	FREQUENCY															
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0
2	0	0	0	0	0	1	2	0	0	0	1	1	0	1	2	1
3	1	1	0	1	0	2	2	0	1	0	1	1	0	0	2	0
4	2	1	0	0	0	6	5	3	2	1	8	4	3	2	3	0
5	3	10	1	0	0	2	4	4	11	1	2	5	3	5	5	3
6	5	15	2	2	2	2	6	6	6	0	12	2	9	12	16	5
7	9	18	5	4	5	8	5	6	13	2	10	8	11	16	11	5
8	17	32	1	7	6	12	5	11	8	8	14	11	16	13	14	10
9	13	22	3	10	2	6	10	12	9	11	16	12	14	13	16	9
10	20	21	9	4	4	17	9	17	22	18	19	18	21	9	19	15
11	11	11	10	10	3	5	13	15	8	7	19	16	14	14	14	10
12	22	4	12	15	3	17	6	12	18	18	11	14	18	18	10	11
13	7	0	24	16	3	16	8	13	7	17	7	8	8	11	8	10
14	14		7	13	9	10	10	16	12	15	7	13	8	6	7	5
15	8		14	9	10	7	8	2	5	7	2	5	7	2	6	8
16	3		15	8	11	9	11	12	9	10	3	3	1	5	2	12
17	0		14	12	11	8	9	3	4	5	2	5	2	5	1	8
18	1		7	7	6	5	7	3	1	8	0	2	1	3	0	6
19	0		2	2	12	1	4	1	0	4	1	4	0	1	0	7
20	0		0	5	15	1	4	0	0	2	0	2	0	0	0	4
21			4	5	8		3			0		1				2
22			3	2	9		1			2		0				1
23			3	3	11		1			0		0				3
24			0	1	4		1			0		0				0
25			0	0	1		1			0		0				0
26			0	0	1		0			0		0				1

STEN SCORE	FREQUENCY															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	3	8	3	0	10	3	0	1	2	13	3	3	1	4	1
2	2	10	4	11	13	4	7	0	2	0	22	4	3	2	8	0
3	8	0	19	10	6	26	15	7	17	10	14	7	9	17	16	13
4	9	15	36	14	9	22	24	6	13	11	16	19	27	16	25	19
5	30	18	21	31	19	17	27	17	17	25	19	12	14	26	35	36
6	31	32	29	30	22	26	29	29	30	35	30	34	21	23	14	15
7	29	22	9	19	18	16	16	27	18	22	7	22	32	29	18	28
8	14	21	4	2	23	8	8	29	19	15	0	18	8	8	7	13
9	11	11	3	10	20	6	4	2	5	8	3	8	15	10	8	6
10	1	4	3	6	6	1	3	19	14	8	3	9	4	4	1	5

DATE OF LISTING 22/08/73

10. D.E.S. OUTPUT LISTINGS.

10.1 THIS SECTION CONTAINS A LISTING OF THE OUTPUT REQUIRED BY THE D.E.S. FOR STUDENT ADMISSIONS.

THE FOLLOWING EXAMPLES ARE INCLUDED:

1 '1972' FILE LISTING (PROGRAMME R, OPTION 5)
2 '1972' ADMISSIONS (PROGRAMME R OPTION 10)
3 '1972' ADMISSIONS (PROGRAMME D)
4 '1972' ADMISSION CARDS LISTING

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME R
OPTION REQUESTED 5
YEAR REQUESTED 1972

71/42120 ADMITTED IN YEAR 1971

.....BEDE COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE___DATES	DEGREES HELD		
PGCE STUDENT(S): FOREIGN DEGREES OR EQUIVALENT							
PGCE	72/42256	DEMBLING, ARTHUR GARY	05/08/49	09/72 TO 07/73	BA	1970	PENNSYLVANIA STATE
PGCE	72/42269	TAYLOR, ALAN FREDERICK	00/10/39	09/72 TO 07/73		1967	INST. OF METALL.

.....BEDE COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE___DATES	DEGREES HELD		
PGCE	72/42251	BAKER, ANDRE	13/08/46	09/72 TO 07/73	BSC	1969	SUSSEX
PGCE	72/42252	BARNES, DAVID PHILIP	02/06/51	09/72 TO 07/73	BSC	1972	WALES
PGCE	72/42253	BELL, MICHAEL RAYMOND	06/09/51	09/72 TO 07/73	BA	1972	DURHAM
PGCE	72/42254	BOWREY, PATRICK STEVEN	06/06/52	09/72 TO 07/73	BA	1972	DURHAM
PGCE	72/42255	CATLIN, NICHOLAS JOHN	19/01/50	09/72 TO 07/73	BSC	1972	EAST ANGLIA
PGCE	72/42257	DUNSCOMBE, DAVID	22/11/49	09/72 TO 07/73	BA	1971	NEWCASTLE-UPON-TYNE
PGCE	72/42258	EDWARDS, PAUL MARTIN	20/08/49	09/72 TO 07/73	BA	1970	LONDON
PGCE	72/42259	FEATHERSTONE-WITTY, MARK PHILIP	02/06/46	09/72 TO 07/73	BA	1972	DURHAM
PGCE	72/42260	FOSTER, ALASTAIR JOHN	13/03/51	09/72 TO 07/73	BA	1972	DURHAM
PGCE	72/42261	GREENWELL, BRIAN	11/12/48	09/72 TO 07/73	BSC	1972	C.N.A.A.
PGCE	72/42262	HESLOP, JOHN RONALD	22/12/47	09/72 TO 07/73	BA	1970	ESSEX
PGCE	72/42263	LLEWELLYN, DAVID GWYN	27/12/47	09/72 TO 07/73	BA	1970	NEWCASTLE-UPON-TYNE
PGCE	72/42264	MCNAMARA, IAN EDWARD	02/05/50	09/72 TO 07/73	BA	1971	YORK
PGCE	72/42265	MOORE, ALAN PETER	09/08/45	09/72 TO 07/73	BA	1969	DURHAM
PGCE	72/42266	OGDEN, GRAEME MACLEOD	06/08/50	09/72 TO 07/73	BSC	1972	HULL
PGCE	72/42267	PARKER, DAVID GUNSON	25/11/46	09/72 TO 07/73	BA	1972	DURHAM
PGCE	72/42268	SMITH, RONALD GEORGE	29/12/34	09/72 TO 07/73	BA	1956	DURHAM
PGCE	72/42270	WHATMORE, JAMES CHARLES	21/04/49	09/72 TO 07/73	BA	1972	DURHAM
PGCE	72/42271	YOUNG, FREDERICK DAVID	26/08/50	09/72 TO 07/73	BA	1972	DURHAM

.....BEDE COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE___DATES	DEGREES HELD
1	72/42238	ALLAN, DONALD	13/05/49	09/72 TO 07/73	
1	72/42239	ASKEW, MARK RUSSELL	17/12/39	09/72 TO 07/73	
1	72/42240	DALTON, JOHN THOMAS	12/02/47	09/72 TO 07/73	
1	72/42241	GOODWIN, DEREK	16/04/42	09/72 TO 07/73	
1	72/42242	GREEN, JOSEPH HUDSON	03/01/28	09/72 TO 07/73	
1	72/42243	HEATLEY, KENNETH JOSEPH	22/06/43	09/72 TO 07/73	
1	72/42244	MAIDMENT, PETER	08/12/49	09/72 TO 07/73	
1	72/42245	MOREHEAD, WILLIAM CHARLES	12/11/32	09/72 TO 07/73	
1	72/42246	PIPER, TERENCE WINSTON	06/12/47	09/72 TO 07/73	
1	72/42247	SANDS, DAVID	30/10/44	09/72 TO 07/73	
1	72/42248	SHORT, HARRY	29/09/28	09/72 TO 07/73	
1	72/42249	WHITE, WILLIAM SELWYN	12/04/46	09/72 TO 07/73	
1	72/42250	WOLFE, PETER LESLIE	01/03/32	09/72 TO 07/73	

.....BEDE COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE___DATES	DEGREES HELD
3	72/42051	ADAMSON, COLIN TREVOR	25/10/53	09/72 TO 07/75	
3	72/42052	ALDRIDGE, JOHN ROSS CHARLES	04/12/51	09/72 TO 07/75	
3	72/42053	ATHERTON, JOHN ROBERT	15/03/53	09/72 TO 07/75	
3	72/42054	ATKINSON, GRAHAM DAVID	30/01/54	09/72 TO 07/75	
3	72/42055	AUTON, RAYMOND	05/06/36	09/72 TO 07/75	
3	72/42056	AYRE, WILLIAM	07/05/52	09/72 TO 07/75	
3	72/42057	BALLARD, MALCOLM	13/09/48	09/72 TO 07/75	
3	72/42058	BAMBROUGH, PAUL ROBERT	13/02/54	09/72 TO 07/75	
3	72/42233	BARKER, ALAN MALCOLM	18/06/54	09/72 TO 07/75	
3	72/42059	BARRETT, STEPHEN	07/04/54	09/72 TO 07/75	
3	72/42060	BARTHOLOMEW, GRAHAM JOHN	13/03/54	09/72 TO 07/75	
3	72/42061	BATHGATE, JAMES ALAN	13/08/52	09/72 TO 07/75	
3	72/42062	BEANEY, MICHAEL JOSEPH	12/02/54	09/72 TO 07/75	
3	72/42063	BELL, JEFFREY LLOYD	12/12/53	09/72 TO 07/75	
3	72/42064	BERRY, IVOR JAMES	14/04/54	09/72 TO 07/75	
3	72/42065	BETTLEY, ANDREW	28/05/53	09/72 TO 07/75	
3	72/42066	BEWICK, JAMES	01/03/42	09/72 TO 07/75	
3	72/42067	BLACKBURN, KEITH	10/01/38	09/72 TO 07/75	
3	72/42068	BLACKWELL, ADRIAN ALEXANDER	15/01/54	09/72 TO 07/75	
3	72/42069	BOWES, DEREK ARTHUR	26/12/53	09/72 TO 7/75	
3	72/42070	BROWN, PETER CHRISTOPHER	04/04/54	09/72 TO 07/75	
3	72/42071	BURGESS, KENNETH JOHN	17/03/54	09/72 TO 07/75	
3	72/42072	CAMPBELL, COLIN JOHN MCNAIR	06/04/53	09/72 TO 07/75	
3	72/42073	CARR, JOHN CLENNEL	24/09/53	09/72 TO 07/75	
3	72/42074	CARRUTHERS, CHRISTOPHER JOHN	07/02/53	09/72 TO 07/75	
3	72/42075	CARRUTHERS, PAUL	28/07/49	09/72 TO 07/75	
3	72/42076	CARTER, TREVOR RICHARD EARL	20/08/52	09/72 TO 07/75	
3	72/42077	CATTERSON, PETER	25/10/35	09/72 TO 07/75	
3	72/42078	CHAPMAN, ARTHUR ALAN	07/10/53	09/72 TO 07/75	
3	72/42079	CHISHOLM, ALISTAIR	11/03/50	09/72 TO 07/75	
3	72/42080	COLLINGWOOD, DAVID	24/07/54	09/72 TO 07/75	
3	72/42081	COLLINS, IAN FRANCIS	26/02/53	09/72 TO 07/75	
3	72/42082	COLLINSON, DAVID THOMAS	23/11/52	09/72 TO 07/75	
3	72/42083	COOKE, NORMAN MEREDITH	16/01/54	09/72 TO 07/75	
3	72/42084	COPELAND, JOSEPH STEPHEN	06/07/54	09/72 TO 07/75	
3	72/42085	COTTON, DAVID ANTHONY	04/05/54	09/72 TO 07/75	
3	72/42086	CRAWSHAW, EVAN VAUGHAN	30/06/51	09/72 TO 07/75	
3	72/42087	CROOKS, CHRISTOPHER JOHN	19/09/53	09/72 TO 07/75	
3	72/42088	CROOKS, PHILIP GERARD	28/03/51	09/72 TO 07/75	
3	72/42089	CURTIS, GRAHAM CLIFFORD	01/05/54	09/72 TO 07/75	
3	72/42097	D.SOUZA, ANDREW BERNARD JOHN	19/10/51	09/72 TO 07/75	
3	72/42090	DARWIN, DAVID CHRISTOPHER	25/12/52	09/72 TO 07/75	
3	72/42091	DAVISON, GEOFFREY	09/04/54	09/72 TO 07/75	
3	72/42092	DEBENHAM, PAUL BRUCE	28/04/54	09/72 TO 07/75	
3	72/42093	DENHAM, JAMES JOHN WILLIAM	08/08/54	09/72 TO 07/75	
3	72/42094	DIXON, DAVID ANTON	13/04/54	09/72 TO 07/75	
3	72/42095	DOE, RAYMOND CLIFFORD	03/07/53	09/72 TO 07/75	
3	72/42096	DREW, GARY RONALD	04/05/53	09/72 TO 07/75	
3	72/42098	EATON, GORDON STANLEY	14/12/52	09/72 TO 07/75	
3	72/42099	ELVIN, JOHN	04/04/53	09/72 TO 07/75	
3	72/42100	FAULKS, MARTIN RICHARD	17/02/54	09/72 TO 07/75	
3	72/42101	FEATHERSTONE, DENIS RONALD	08/12/53	09/72 TO 07/75	
3	72/42102	FERGUSON, ANDREW MICHAEL	11/11/51	09/72 TO 07/75	
3	72/42103	FERRY, BRIAN WILLIAM	22/07/51	09/72 TO 07/75	
3	72/42104	FINDLATER, MAXWELL KERR	18/07/54	09/72 TO 07/75	

.....BEDF COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE___DATES	DEGREES HELD
3	72/42105	FITZSIMMONS, MALCOLM LEWIS	15/03/53	09/72 TO 07/75	
3	72/42106	FOWKES, ROGER CHARLES	14/10/52	09/72 TO 07/75	
3	72/42107	FOX, PAUL DOUGLAS	23/04/53	09/72 TO 07/75	
3	72/42108	FRANKLIN, DEREK ANTHONY AIDAN	21/03/24	09/72 TO 07/75	
3	72/42109	GALLIE, DUNCAN ANDREW	23/03/53	09/72 TO 07/75	
3	72/42110	GARBUTT, IAN	04/10/53	09/72 TO 07/75	
3	72/42111	GARDINER, JOHN ROGAN	21/05/43	09/72 TO 07/75	
3	72/42112	GILFILLAN, HOWARD	01/09/52	09/72 TO 07/75	
3	72/42113	GILL, RAYMOND CHRISTOPHER	28/01/52	09/72 TO 07/75	
3	72/42114	GOODWIN, TOBY MICHAEL GILES	25/07/50	09/72 TO 07/75	
3	72/42115	GOSLING, BRIAN GEORGE EDWARD	11/01/48	09/72 TO 07/75	
3	72/42116	GOW, PETER DRAEFIN	19/10/52	09/72 TO 07/75	
3	72/42117	GRAVELING, JOHN EDWARD	13/01/54	09/72 TO 07/75	
3	72/42118	GRAYDON, GEOFFREY	18/06/52	09/72 TO 07/75	
3	72/42119	GREEN, DAVID WILLIAM	10/06/53	09/72 TO 07/75	
3	72/42120	GRIFFIN, RICHARD CHARLES	01/05/53	09/72 TO 07/75	
3	72/42121	GUYMER, WILLIAM STEPHENSON	01/11/49	09/72 TO 07/75	
3	72/42122	HALLIDAY, STEPHEN	08/08/54	09/72 TO 07/75	
3	72/42123	HAMILTON, IAN	01/09/54	09/72 TO 07/75	
3	72/42124	HAMILTON, MALCOLM IAN	12/07/54	09/72 TO 07/75	
3	72/42125	HANSEN, PAUL	01/11/52	09/72 TO 07/75	
3	72/42126	HANSON, JOHN	28/06/34	09/72 TO 07/75	
3	72/42127	HARDY, KEITH JOHN	24/11/52	09/72 TO 07/75	
3	72/42129	HEDLEY, NORMAN ANTHONY	04/04/54	09/72 TO 07/75	
3	72/42130	HESKETH, NEIL WILLIAM	29/10/53	09/72 TO 07/75	
3	72/42132	HINCH, JAMES ROBERTSON	02/12/53	09/72 TO 07/75	
3	72/42133	HIRD, PETER GRAHAM	03/01/53	09/72 TO 07/75	
3	72/42134	HODGKISS, PAUL	21/09/53	09/72 TO 07/75	
3	72/42135	HOLROYD, ANDREW CHARLES	20/06/53	09/72 TO 07/75	
3	72/42136	HOLYOME, JOHN MARK	16/07/53	09/72 TO 07/75	
3	72/42137	HOUGHTON, TREVOR KEITH	16/05/54	09/72 TO 07/75	
3	72/42138	HOULDING, ROBIN RICHARD	24/04/54	09/72 TO 07/75	
3	72/42139	HOWLAND, JONATHAN VICTOR	22/12/52	09/72 TO 07/75	
3	72/42140	HULME, ROBIN ANTHONY	18/12/52	09/72 TO 07/75	
3	72/42141	HUTCHINSON, STANLEY JAMES	08/07/54	09/72 TO 07/75	
3	72/42142	JACKSON, PAUL	12/04/53	09/72 TO 07/75	
3	72/42143	JOHN, DAVID NEIL	15/10/53	09/72 TO 07/75	
3	72/42144	JOHNSTON, DEREK	23/05/52	09/72 TO 07/75	
3	72/42145	JONES, KEITH	05/02/54	09/72 TO 07/75	
3	72/42146	JONES, NORMAN	26/07/54	09/72 TO 07/75	
3	72/42147	KELLY, NEIL HAROLD	07/05/52	09/72 TO 07/75	
3	72/42148	KIDNEY, LANCE NICHOLAS	24/10/52	09/72 TO 07/75	
3	72/42149	KNOX, JEFFREY NICHOLAS	18/12/51	09/72 TO 07/75	
3	72/42150	LAUDER, IAN WILLIAM	29/09/52	09/72 TO 07/75	
3	72/42151	LAWSON, CHARLES WILLIAM PAGAN	23/06/54	09/72 TO 07/75	
3	72/42152	LINE, STEPHEN CHARLES	04/06/53	09/72 TO 07/75	
3	72/42153	LOWEY, KENNETH	24/04/54	09/72 TO 07/75	
3	72/42154	LUMSDEN, DAVID WILLIAM	22/07/44	09/72 TO 07/75	
3	72/42155	LYNCH, CHRISTOPHER MICHAEL	23/09/51	09/72 TO 07/75	
3	72/42158	MACLEOD, TORQUIL ROBERT	08/04/53	09/72 TO 07/75	
3	72/42160	MARSHALL, STEPHEN JOHN	13/05/53	09/72 TO 07/75	
3	72/42161	MATTHEWS, TIMOTHY JOHN	16/09/53	09/72 TO 07/75	
3	72/42156	MCCALL, COLIN WINTER	06/05/44	09/72 TO 07/75	
3	72/42157	MCDONALD, JOHN	22/04/50	09/72 TO 07/75	
3	72/42162	MEDLOCK, STUART ANTONY	16/02/54	09/72 TO 07/75	
3	72/42163	MERCER, NICHOLAS OLIVER ANDREW	19/04/53	09/72 TO 07/75	

.....BEDF COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE DATES	DEGREES HELD
3	72/42164	MIDDLETON, COLIN TYSON	04/05/49	09/72 TO 07/75	
3	72/42165	MILLS, DAVID	16/04/54	09/72 TO 07/75	
3	72/42166	MINTO, EDWARD JOHN	06/04/54	09/72 TO 07/75	
3	72/42167	MONTGOMERY, PAUL	28/09/52	09/72 TO 07/75	
3	72/42168	MOONEY, JOHN FRANCIS	21/01/52	09/72 TO 07/75	
3	72/42169	MORDUE, ROBERT WEDDERBURN	23/09/52	09/72 TO 07/75	
3	72/42170	MORGAN, NEIL	15/03/49	09/72 TO 07/75	
3	72/42171	MURRAY, MITCHELL JOHN	28/07/54	09/72 TO 07/75	
3	72/42172	MURRAY, PAUL	07/08/54	09/72 TO 07/75	
3	72/42173	NEVINS, MICHAEL JOHN	24/12/52	09/72 TO 07/75	
3	72/42174	NICHOLSON, PETER THOMAS	25/12/53	09/72 TO 07/75	
3	72/42175	NICKLIN, JOHN GILSTIN	14/11/53	09/72 TO 07/75	
3	72/42176	NICOL, GARETH JAMES	06/06/54	09/72 TO 07/75	
3	72/42177	O.DONNELL, PAUL ANTHONY	16/01/53	09/72 TO 07/75	
3	72/42178	OLD, BRIAN LESLIE	23/12/53	09/72 TO 07/75	
3	72/42179	OUGHTON, DAVID	02/03/54	09/72 TO 07/75	
3	72/42180	PARKER, JOHN DAPRELL	01/05/53	09/72 TO 07/75	
3	72/42181	PARTINGTON, PHILLIP ERIC	14/05/54	09/72 TO 07/75	
3	72/42234	PATMAN, BRIAN SAMUEL	23/10/53	09/72 TO 07/75	
3	72/42272	PEARCEY, PETER FRANK	19/08/46	09/72 TO 07/75	
3	72/42182	PICKERING, DAVID	26/06/54	09/72 TO 07/75	
3	72/42183	PILLEY, NICHOLAS JOHN	14/05/53	09/72 TO 07/75	
3	72/42184	RAMSEY, NEIL	24/06/54	09/72 TO 07/75	
3	72/42185	RAMSHAW, MARSHALL	10/01/24	09/72 TO 07/75	
3	72/42186	RAW, BRIAN	03/02/54	09/72 TO 07/75	
3	72/42187	REAY, DAVID	22/01/54	09/72 TO 07/75	
3	72/42188	RICHARDSON, SYDNEY	05/11/53	09/72 TO 07/75	
3	72/42189	RILEY, STEWART	29/01/47	09/72 TO 07/75	
3	72/42190	RIPPON, PHILIP DAVID BARTLE	11/08/50	09/72 TO 07/75	
3	72/42191	ROWBOTHAM, PHILIP	12/09/52	09/72 TO 07/75	
3	72/42192	RUTHERFORD, JOHN FRANCIS	08/02/54	09/72 TO 07/75	
3	72/42193	SCORER, ALEXANDER	20/12/52	09/72 TO 07/75	
3	72/42195	SIBBALD, JULIAN	29/11/52	09/72 TO 07/75	
3	72/42196	SIMPSON, JAMES	05/10/50	09/72 TO 07/75	
3	72/42197	SMITH, PHILIP	10/09/50	09/72 TO 07/75	
3	72/42198	SPEDDING, LEWIS CHRISTOPHER	18/07/54	09/72 TO 07/75	
3	72/42199	STAINTHORPE, DAVID	20/11/50	09/72 TO 07/75	
3	72/42200	STANLEY-CLAMP, RICHARD STEPHEN INGLIS	27/10/53	09/72 TO 07/75	
3	72/42201	STAPLES, FRANCIS TIMOTHY	03/03/53	09/72 TO 07/75	
3	72/42202	STAYMAN, GEORGE MICHAEL	23/10/52	09/72 TO 07/75	
3	72/42203	STEAD, JOHN CHESSWAS	18/03/52	09/72 TO 07/75	
3	72/42204	STEPHENS, PHILIP MORGAN	17/03/53	09/72 TO 07/75	
3	72/42205	STEWART, JONATHAN MALCOLM	02/11/51	09/72 TO 07/75	
3	72/42206	STRAW, CHRISTOPHER STEPHEN	08/03/53	09/72 TO 07/75	
3	72/42207	SUMMERS, ROBERT CRUSE	13/08/53	09/72 TO 07/75	
3	72/42208	TAIT, ALAN	29/05/54	09/72 TO 07/75	
3	72/42209	TARRANT, GEOFFREY PAUL	13/01/54	09/72 TO 07/75	
3	72/42236	TARRELL, JAMES ANDREW SCOTT	27/12/51	09/72 TO 07/75	
3	72/42210	TAYLOR, JOHN BARRY	27/02/54	09/72 TO 07/75	
3	72/42211	TAYLOR, TIMOTHY CHARLES HARCOURT	15/04/52	09/72 TO 07/75	
3	72/42235	TEMPLE, ROBERT KEITH	11/03/54	09/72 TO 07/75	
3	72/42212	THOMPSON, ROBERT	09/06/40	09/72 TO 07/75	
3	72/42213	THOMPSON, WILLIAM	12/08/53	09/72 TO 07/75	
3	72/42214	THWAITES, MARTIN JOSEPH EDWARD	06/07/54	09/72 TO 07/75	
3	72/42215	TRACE, DAVID JOHN	17/11/52	09/72 TO 07/75	
3	72/42216	TREWARTHA, ALEC BLAIR	09/02/52	09/72 TO 07/75	

.....BEDE COLLEGE DURHAM 1972.....

TYPE	D.E.S.NO	NAME	DATE OF BIRTH	COURSE___DATES	DEGREES HELD
3	72/42217	UNDERWOOD, HOWARD DOUGLAS	23/04/54	09/72 TO 07/75	
3	72/42218	VICKERS, JOHN JAMES	11/12/52	09/72 TO 07/75	
3	72/42219	WATSON, ROBERT JOHN	20/11/51	09/72 TO 07/75	
3	72/42220	WEST, JEREMY LEIGH	29/12/53	09/72 TO 07/75	
3	72/42221	WHITE, PHILIP WILLIAM	21/05/53	09/72 TO 07/75	
3	72/42222	WIDDRINGTON, THOMAS	26/10/50	09/72 TO 07/75	
3	72/42223	WILKINSON, IAN DAVID	23/04/53	09/72 TO 07/75	
3	72/42224	WILKINSON, JAMES EDWARD	03/03/47	09/72 TO 07/75	
3	72/42225	WILKINSON, STEPHEN ARTHUR	11/03/52	09/72 TO 07/75	
3	72/42226	WILLIAMS, HOWARD MARTYN	31/07/54	09/72 TO 07/75	
3	72/42227	WILSON, ALAN HARDY	11/02/51	09/72 TO 07/75	
3	72/42228	WILSON, GEOFFREY	07/02/54	09/72 TO 07/75	
3	72/42229	WILSON, JOHN MARK FLEMING	24/12/52	09/72 TO 07/75	
3	72/42230	WILSON, KEVIN WILLIAM	23/03/54	09/72 TO 07/75	
3	72/42231	WREN, CHRISTOPHER JOHN	04/05/54	09/72 TO 07/75	
3	72/42232	YOUNG, CLIVE PHILIP	08/12/53	09/72 TO 07/75	

NO. OF STUDENTS 218

DATE OF LISTING 29/08/73

EXECUTION TERMINATED

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME R
OPTION REQUESTED 19
YEAR REQUESTED 1972

DES O/P REQUIRED

71/42120 ADMITTED IN YEAR 1971

DATE OF LISTING 29/08/73

BCRS....BEDE COLLEGE RECORD SYSTEM

PROGRAMME D
YEAR REQUESTED 1972

72/42051 TRANSACTION 01 COMPLETED
72/42052 TRANSACTION 01 COMPLETED
72/42053 TRANSACTION 01 COMPLETED
72/42054 TRANSACTION 01 COMPLETED
72/42055 TRANSACTION 01 COMPLETED
72/42056 TRANSACTION 01 COMPLETED
72/42057 TRANSACTION 01 COMPLETED
72/42058 TRANSACTION 01 COMPLETED
72/42059 TRANSACTION 01 COMPLETED
72/42060 TRANSACTION 01 COMPLETED
72/42061 TRANSACTION 01 COMPLETED
72/42062 TRANSACTION 01 COMPLETED
72/42063 TRANSACTION 01 COMPLETED
72/42064 TRANSACTION 01 COMPLETED
72/42065 TRANSACTION 01 COMPLETED
72/42066 TRANSACTION 01 COMPLETED
72/42067 TRANSACTION 01 COMPLETED
72/42068 TRANSACTION 01 COMPLETED
72/42069 TRANSACTION 01 COMPLETED
72/42070 TRANSACTION 01 COMPLETED
72/42071 TRANSACTION 01 COMPLETED
72/42072 TRANSACTION 01 COMPLETED
72/42073 TRANSACTION 01 COMPLETED
72/42074 TRANSACTION 01 COMPLETED
72/42075 TRANSACTION 01 COMPLETED
72/42076 TRANSACTION 01 COMPLETED
72/42077 TRANSACTION 01 COMPLETED
72/42078 TRANSACTION 01 COMPLETED
72/42079 TRANSACTION 01 COMPLETED
72/42080 TRANSACTION 01 COMPLETED
72/42081 TRANSACTION 01 COMPLETED
72/42082 TRANSACTION 01 COMPLETED
72/42083 TRANSACTION 01 COMPLETED
72/42084 TRANSACTION 01 COMPLETED
72/42085 TRANSACTION 01 COMPLETED
72/42086 TRANSACTION 01 COMPLETED
72/42087 TRANSACTION 01 COMPLETED
72/42088 TRANSACTION 01 COMPLETED
72/42089 TRANSACTION 01 COMPLETED
72/42090 TRANSACTION 01 COMPLETED
72/42091 TRANSACTION 01 COMPLETED
72/42092 TRANSACTION 01 COMPLETED
72/42093 TRANSACTION 01 COMPLETED
72/42094 TRANSACTION 01 COMPLETED
72/42095 TRANSACTION 01 COMPLETED
72/42096 TRANSACTION 01 COMPLETED
72/42097 TRANSACTION 01 COMPLETED
72/42098 TRANSACTION 01 COMPLETED
72/42099 TRANSACTION 01 COMPLETED
72/42100 TRANSACTION 01 COMPLETED
72/42101 TRANSACTION 01 COMPLETED
72/42102 TRANSACTION 01 COMPLETED
72/42103 TRANSACTION 01 COMPLETED
72/42104 TRANSACTION 01 COMPLETED
72/42105 TRANSACTION 01 COMPLETED

72/42106 TRANSACTION 01 COMPLETED
72/42107 TRANSACTION 01 COMPLETED
72/42108 TRANSACTION 01 COMPLETED
72/42109 TRANSACTION 01 COMPLETED
72/42110 TRANSACTION 01 COMPLETED
72/42111 TRANSACTION 01 COMPLETED
72/42112 TRANSACTION 01 COMPLETED
72/42113 TRANSACTION 01 COMPLETED
72/42114 TRANSACTION 01 COMPLETED
72/42115 TRANSACTION 01 COMPLETED
72/42116 TRANSACTION 01 COMPLETED
72/42117 TRANSACTION 01 COMPLETED
72/42118 TRANSACTION 01 COMPLETED
72/42119 TRANSACTION 01 COMPLETED
72/42120 TRANSACTION 01 COMPLETED
72/42121 TRANSACTION 01 COMPLETED
72/42122 TRANSACTION 01 COMPLETED
72/42123 TRANSACTION 01 COMPLETED
72/42124 TRANSACTION 01 COMPLETED
72/42125 TRANSACTION 01 COMPLETED
72/42126 TRANSACTION 01 COMPLETED
72/42127 TRANSACTION 01 COMPLETED
72/42129 TRANSACTION 01 COMPLETED
72/42130 TRANSACTION 01 COMPLETED
72/42132 TRANSACTION 01 COMPLETED
72/42133 TRANSACTION 01 COMPLETED
72/42134 TRANSACTION 01 COMPLETED
72/42135 TRANSACTION 01 COMPLETED
72/42136 TRANSACTION 01 COMPLETED
72/42137 TRANSACTION 01 COMPLETED
72/42138 TRANSACTION 01 COMPLETED
72/42139 TRANSACTION 01 COMPLETED
72/42140 TRANSACTION 01 COMPLETED
72/42141 TRANSACTION 01 COMPLETED
72/42142 TRANSACTION 01 COMPLETED
72/42143 TRANSACTION 01 COMPLETED
72/42144 TRANSACTION 01 COMPLETED
72/42145 TRANSACTION 01 COMPLETED
72/42146 TRANSACTION 01 COMPLETED
72/42147 TRANSACTION 01 COMPLETED
72/42148 TRANSACTION 01 COMPLETED
72/42149 TRANSACTION 01 COMPLETED
72/42150 TRANSACTION 01 COMPLETED
72/42151 TRANSACTION 01 COMPLETED
72/42152 TRANSACTION 01 COMPLETED
72/42153 TRANSACTION 01 COMPLETED
72/42154 TRANSACTION 01 COMPLETED
72/42155 TRANSACTION 01 COMPLETED
72/42156 TRANSACTION 01 COMPLETED
72/42157 TRANSACTION 01 COMPLETED
72/42158 TRANSACTION 01 COMPLETED
72/42160 TRANSACTION 01 COMPLETED
72/42161 TRANSACTION 01 COMPLETED
72/42162 TRANSACTION 01 COMPLETED
72/42163 TRANSACTION 01 COMPLETED
72/42164 TRANSACTION 01 COMPLETED
72/42165 TRANSACTION 01 COMPLETED
72/42166 TRANSACTION 01 COMPLETED
72/42167 TRANSACTION 01 COMPLETED
72/42168 TRANSACTION 01 COMPLETED

72/42169 TRANSACTION 01 COMPLETED
72/42170 TRANSACTION 01 COMPLETED
72/42171 TRANSACTION 01 COMPLETED
72/42172 TRANSACTION 01 COMPLETED
72/42173 TRANSACTION 01 COMPLETED
72/42174 TRANSACTION 01 COMPLETED
72/42175 TRANSACTION 01 COMPLETED
72/42176 TRANSACTION 01 COMPLETED
72/42177 TRANSACTION 01 COMPLETED
72/42178 TRANSACTION 01 COMPLETED
72/42179 TRANSACTION 01 COMPLETED
72/42180 TRANSACTION 01 COMPLETED
72/42181 TRANSACTION 01 COMPLETED
72/42182 TRANSACTION 01 COMPLETED
72/42183 TRANSACTION 01 COMPLETED
72/42184 TRANSACTION 01 COMPLETED
72/42185 TRANSACTION 01 COMPLETED
72/42186 TRANSACTION 01 COMPLETED
72/42187 TRANSACTION 01 COMPLETED
72/42188 TRANSACTION 01 COMPLETED
72/42189 TRANSACTION 01 COMPLETED
72/42190 TRANSACTION 01 COMPLETED
72/42191 TRANSACTION 01 COMPLETED
72/42192 TRANSACTION 01 COMPLETED
72/42193 TRANSACTION 01 COMPLETED
72/42195 TRANSACTION 01 COMPLETED
72/42196 TRANSACTION 01 COMPLETED
72/42197 TRANSACTION 01 COMPLETED
72/42198 TRANSACTION 01 COMPLETED
72/42199 TRANSACTION 01 COMPLETED
72/42200 TRANSACTION 01 COMPLETED
72/42201 TRANSACTION 01 COMPLETED
72/42202 TRANSACTION 01 COMPLETED
72/42203 TRANSACTION 01 COMPLETED
72/42204 TRANSACTION 01 COMPLETED
72/42205 TRANSACTION 01 COMPLETED
72/42206 TRANSACTION 01 COMPLETED
72/42207 TRANSACTION 01 COMPLETED
72/42208 TRANSACTION 01 COMPLETED
72/42209 TRANSACTION 01 COMPLETED
72/42210 TRANSACTION 01 COMPLETED
72/42211 TRANSACTION 01 COMPLETED
72/42212 TRANSACTION 01 COMPLETED
72/42213 TRANSACTION 01 COMPLETED
72/42214 TRANSACTION 01 COMPLETED
72/42215 TRANSACTION 01 COMPLETED
72/42216 TRANSACTION 01 COMPLETED
72/42217 TRANSACTION 01 COMPLETED
72/42218 TRANSACTION 01 COMPLETED
72/42219 TRANSACTION 01 COMPLETED
72/42220 TRANSACTION 01 COMPLETED
72/42221 TRANSACTION 01 COMPLETED
72/42222 TRANSACTION 01 COMPLETED
72/42223 TRANSACTION 01 COMPLETED
72/42224 TRANSACTION 01 COMPLETED
72/42225 TRANSACTION 01 COMPLETED
72/42226 TRANSACTION 01 COMPLETED
72/42227 TRANSACTION 01 COMPLETED
72/42228 TRANSACTION 01 COMPLETED
72/42229 TRANSACTION 01 COMPLETED

```
72/42230 TRANSACTION 01 COMPLETED
72/42231 TRANSACTION 01 COMPLETED
72/42232 TRANSACTION 01 COMPLETED
72/42233 TRANSACTION 01 COMPLETED
72/42234 TRANSACTION 01 COMPLETED
72/42235 TRANSACTION 01 COMPLETED
72/42236 TRANSACTION 01 COMPLETED
72/42237 TRANSACTION 01 COMPLETED
72/42238 TRANSACTION 01 COMPLETED
72/42239 TRANSACTION 01 COMPLETED
72/42240 TRANSACTION 01 COMPLETED
72/42241 TRANSACTION 01 COMPLETED
72/42242 TRANSACTION 01 COMPLETED
72/42243 TRANSACTION 01 COMPLETED
72/42244 TRANSACTION 01 COMPLETED
72/42245 TRANSACTION 01 COMPLETED
72/42246 TRANSACTION 01 COMPLETED
72/42247 TRANSACTION 01 COMPLETED
72/42248 TRANSACTION 01 COMPLETED
72/42249 TRANSACTION 01 COMPLETED
72/42250 TRANSACTION 01 COMPLETED
72/42251 TRANSACTION 01 COMPLETED
72/42252 TRANSACTION 01 COMPLETED
72/42253 TRANSACTION 01 COMPLETED
72/42254 TRANSACTION 01 COMPLETED
72/42255 TRANSACTION 01 COMPLETED
72/42256 TRANSACTION 01 COMPLETED
72/42257 TRANSACTION 01 COMPLETED
72/42258 TRANSACTION 01 COMPLETED
72/42259 TRANSACTION 01 COMPLETED
72/42260 TRANSACTION 01 COMPLETED
72/42261 TRANSACTION 01 COMPLETED
72/42262 TRANSACTION 01 COMPLETED
72/42263 TRANSACTION 01 COMPLETED
72/42264 TRANSACTION 01 COMPLETED
72/42265 TRANSACTION 01 COMPLETED
72/42266 TRANSACTION 01 COMPLETED
72/42267 TRANSACTION 01 COMPLETED
72/42268 TRANSACTION 01 COMPLETED
72/42269 TRANSACTION 01 COMPLETED
72/42270 TRANSACTION 01 COMPLETED
72/42271 TRANSACTION 01 COMPLETED
72/42272 TRANSACTION 01 COMPLETED
```

```
$LIST -RES
     1    7242051125&531ADAMSON            COLIN TREVOR                                    9990101
     2    7242051125&53ADAMSO                   449259729625                          1    9990102
     3    7242051125&53ADAMSO C0C801 11                                        F          9990103
     4    7242051125&53ADAMSO F  1 972775   111      1  052                               9990105
     5    7242052104051ALDRIDGE            JOHN ROSS CHARLES                               9990101
     6    7242052104051ALDRID                   302259729625                          1    9990102
     7    7242052104051ALDRID C00803 11E   E                 B                              9990103
     8    7242052104051ALDRID F  1 972775   111      1  011                               9990105
     9    7242053115353ATHERTON            JOHN ROBERT                                     9990101
    10    7242053115353ATHERT                   405259729625                          1    9990102
    11    7242053115353ATHERT C0C801 11                                  D                 9990103
    12    7242053115353ATHERT F  1 972775   111      1  048                               9990105
    13    7242054130154ATKINSON            GRAHAM DAVID                                    9990101
    14    7242054130154ATKINS                   455259729625                          1    9990102
    15    7242054130154ATKINS C00903 11C                D                        D          9990103
    16    7242054130154ATKINS F  1 972775   111      1  051                               9990105
    17    7242055105636AUTON               RAYMOND                                         9990101
    18    7242055105636AUTON                    411259729625                         19    9990102
    19    7242055105636AUTON  C00600 11                                                    9990103
    20    7242055105636AUTON  F  1 972775   111      1  104                               9990105
    21    7242056107552AYRE                WILLIAM                                         9990101
    22    7242056107552AYRE                     411259729625                          1    9990102
    23    7242056107552AYRE   C0C402 1           D  E                                      9990103
    24    7242056107552AYRE   F  1 972775   111      1  002                               9990105
    25    7242057113948BALLARD             MALCOLM                                         9990101
    26    7242057113948BALLAR                   535259729625                          1    9990102
    27    7242057113948BALLAR 0006C0 11                                                    9990103
    28    7242057113948BALLAR F  1 972775   111      1  002                               9990105
    29    7242058113254BAMBERDUGH          PAUL ROBERT                                     9990101
    30    7242058113254BAMBER                   411259729625                          1    9990102
    31    7242058113254BAMBER C00701 11                                          D         9990103
    32    7242058113254BAMBER F  1 972775   111      1  061                               9990105
    33    7242059107454BARRETT             STEPHEN                                         9990101
    34    7242059107454BARRET                   461259729625                          1    9990102
    35    7242059107454BARRET C0C404 1 C         B E                                E      9990103
    36    7242059107454BARRET F  1 972775   111      1  061                               9990105
    37    7242060113354BARTHOLOMEW         GRAHAM JOHN                                     9990101
    38    7242060113354BARTHO                   303259729625                          1    9990102
    39    7242060113354BARTHO C0C701 1 C                                                   9990103
    40    7242060113354BARTHO F  1 972775   111      1  061                               9990105
    41    7242061113852BATHGATE            JAMES ALAN                                      9990101
    42    7242061113852BATHGA                   299259729625                          2    9990102
    43    7242061113852BATHGA 000000                                                       9990103
    44    7242061113852BATHGA F  1 972775   111      1  002                               9990105
    45    7242062112254BEANEY              MICHAEL JOSEPH                                  9990101
    46    7242062112254BEANEY                   411259729625                          1    9990102
    47    7242062112254BEANEY C0C500 1                                                     9990103
    48    7242062112254BEANEY F  1 972775   111      1  092                               9990105
    49    7242063112053BELL                JEFFREY LLOYD                                   9990101
    50    7242063112053BELL                     411259729625                          1    9990102
    51    7242063112053BELL   C00802 11                                     BB             9990103
    52    7242063112053BELL   F  1 972775   111      1  104                               9990105
    53    7242064114454BERRY               IVOR JAMES                                      9990101
    54    7242064114454BERRY                    480259729625                          1    9990102
    55    7242064114454BERRY  C00901 11E                                                   9990103
    56    7242064114454BERRY  F  1 972775   111      1  061                               9990105
    57    7242065128553BETTLEY             ANDREW                                          9990101
    58    7242065128553BETTLE                   413259729625                          1    9990102
```

```
59   7242065128553BETTLE 000701 11E                                                  9990103
60   7242065128553BETTLE F  1 972775   111      1  061                               9990105
61   7242066101342BEWICK              JAMES                                          9990101
62   7242066101342BEWICK                    480259729625                        18   9990102
63   7242066101342BEWICK 000500 1                                                    9990103
64   7242066101342BEWICK F  1 972775   111      1  061                               9990105
65   7242067110138BLACKBURN           KEITH                                          9990101
66   7242067110138BLACKB                    411259729625                        19   9990102
67   7242067110138BLACKB 000300  1                                                   9990103
68   7242067110138BLACKB F  1 972775   111      1  104                               9990105
69   7242068115154BLACKWELL           ADRIAN ALEXANDER                               9990101
70   7242068115154BLACKW                    473259729625                         1   9990102
71   7242068115154BLACKW 000601 1 E                                                  9990103
72   7242068115154BLACKW F  1 972775   111      1  048                               9990105
73   7242069126053BOWES               DEREK ARTHUR                                   9990101
74   7242069126053BOWES                     299259729625                         2   9990102
75   7242069126053BOWES  000000                                                      9990103
76   7242069126053BOWES  F  1 972775   111      1  052                               9990105
77   7242070104454BROWN               PETER CHRISTOPHER                              9990101
78   7242070104454BROWN                     468259729625                         1   9990102
79   7242070104454BROWN  020500 1                                                    9990103
80   7242070104454BROWN  F  1 972775   111      1  104                               9990105
81   7242071117354BURGESS             KENNETH JOHN                                   9990101
82   7242071117354BURGES                    480259729625                         1   9990102
83   7242071117354BURGES 000700 1                                                    9990103
84   7242071117354BURGES F  1 972775   111      1  092                               9990105
85   7242072106453CAMPBELL            COLIN JOHN MCNAIR                              9990101
86   7242072106453CAMPBE                    431259729625                         1   9990102
87   7242072106453CAMPBE 000503 1 E          E                                E      9990103
88   7242072106453CAMPBE F  1 972775   111      1  061                               9990105
89   7242073124953CARR                JOHN CLENNEL                                   9990101
90   7242073124953CARR                      411259729625                         1   9990102
91   7242073124953CARR   000701 11B                                                  9990103
92   7242073124953CARR   F  1 972775   111      1  051                               9990105
93   7242074107253CARRUTHERS          CHRISTOPHER JOHN                               9990101
94   7242074107253CARRUT                    407259729625                         1   9990102
95   7242074107253CARRUT 010400 11                                                   9990103
96   7242074107253CARRUT F  1 972775   111      1  098                               9990105
97   7242075128749CARRUTHERS          PAUL                                           9990101
98   7242075128749CARRUT                    411259729625                         1   9990102
99   7242075128749CARRUT 000502 1                                            EE      9990103
100  7242075128749CARRUT F  1 972775   111      1  052                               9990105
101  7242076120852CARTER              TREVOR RICHARD EARL                            9990101
102  7242076120852CARTER                    305259729625                         1   9990102
103  7242076120852CARTER 000700 11                                                   9990103
104  7242076120852CARTER F  1 972775   111      1  048                               9990105
105  7242077125835CATTERSON           PETER                                          9990101
106  7242077125835CATTER                    411259729625                        19   9990102
107  7242077125835CATTER 000600 11                                                   9990103
108  7242077125835CATTER F  1 972775   111      1  048                               9990105
109  7242078107853CHAPMAN             ARTHUR ALAN                                    9990101
110  7242078107853CHAPMA                    411259729625                         1   9990102
111  7242078107853CHAPMA 000700 11                                                   9990103
112  7242078107853CHAPMA F  1 972775   111      1  002                               9990105
113  7242079111350CHISHOLM            ALISTAIR                                       9990101
114  7242079111350CHISHO                    407259729625                         1   9990102
115  7242079111350CHISHO 000500  1                                                   9990103
116  7242079111350CHISHO F  1 972775   111      1  061                               9990105
117  7242080124754COLLINGWOOD         DAVID                                          9990101
118  7242080124754COLLIN                    431259729625                         1   9990102
```

```
119   7242080124754COLLIN C00600 1                                                  9990103
120   7242080124754COLLIN F  1 972775   111      1  002                            9990105
121   7242081126253COLLINS            IAN FRANCIS                                   9990101
122   7242081126253COLLIN                   41259729625                        1    9990102
123   7242081126253COLLIN 016502 1 D                                   F            9990103
124   7242081126253COLLIN F  1 972775   111      1  061                            9990105
125   7242082123-52COLLINSON          DAVID THOMAS                                  9990101
126   7242082123-52COLLIN                   411259729625                       1    9990102
127   7242082123-52COLLIN 000703 11A         E                                 E    9990103
128   7242082123-52COLLIN F  1 972775   111      1  061                            9990105
129   7242083116154COOKE              NORMAN MEREDITH                               9990101
130   7242083116154COOKE                    405259729625                       1    9990102
131   7242083116154COOKE  000601 11                                    E            9990103
132   7242083116154COOKE  F  1 972775   111      1  052                            9990105
133   7242084106754COPELAND           JOSEPH STEPHEN                                9990101
134   7242084106754COPELA                   411259729625                       1    9990102
135   7242084106754COPELA C00700 11                                                 9990103
136   7242084106754COPELA F  1 972775   111      1  048                            9990105
137   7242085104554COTTON             DAVID ANTHONY                                 9990101
138   7242085104554COTTON                   411259729625                       1    9990102
139   7242085104554COTTON C00500 11                                                 9990103
140   7242085104554COTTON F  1 972775   111      1  002                            9990105
141   7242086130651CRAWSHAW           EVAN VAUGHAN                                  9990101
142   7242086130651CRAWSH                   409259729625                       1    9990102
143   7242086130651CRAWSH C00701 11                                    D            9990103
144   7242086130651CRAWSH F  1 972775   111      1  104                            9990105
145   7242087119953CROOKS             CHRISTOPHER JOHN                              9990101
146   7242087119953CROOKS                   421259729625                       3    9990102
147   7242087119953CROOKS 000702 11          E  F                                  9990103
148   7242087119953CROOKS F  1 972775   111      1  104                            9990105
149   7242088128351CROOKS             PHILIP GERARD                                 9990101
150   7242088128351CROOKS                   495259729625                       1    9990102
151   7242088128351CROOKS C00603 1           C                              CD     9990103
152   7242088128351CROOKS F  1 972775   111      1  051                            9990105
153   7242089101554CURTIS             GRAHAM CLIFFORD                               9990101
154   7242089101554CURTIS                   421259729625                       1    9990102
155   7242089101554CURTIS C00501 11             C                                 9990103
156   7242089101554CURTIS F  1 972775   111      1  104                            9990105
157   7242090125052DARWIN             DAVID CHRISTOPHER                             9990101
158   7242090125052DARWIN                   411259729625                       1    9990102
159   7242090125052DARWIN C00801 11                                    C            9990103
160   7242090125052DARWIN F  1 972775   111      1  048                            9990105
161   7242091109454DAVISON            GEOFFREY                                      9990101
162   7242091109454DAVISO                   517259729625                       1    9990102
163   7242091109454DAVISO C00701 11                                D                9990103
164   7242091109454DAVISO F  1 972775   111      1  002                            9990105
165   7242092128454DEBENHAM           PAUL BRUCE                                    9990101
166   7242092128454DEBENH                   441259729625                       1    9990102
167   7242092128454DEBENH 000700 11                                                 9990103
168   7242092128454DEBENH F  1 972775   111      1  051                            9990105
169   7242093108854DENHAM             JAMES JOHN WILLIAM                            9990101
170   7242093108854DENHAM                   437259729625                       1    9990102
171   7242093108854DENHAM C00902 11E                                   D            9990103
172   7242093108854DENHAM F  1 972775   111      1  057                            9990105
173   7242094113454DIXON              DAVID ANTON                                   9990101
174   7242094113454DIXON                       259729625                            9990102
175   7242094113454DIXON  000804 11D         E                                ED    9990103
176   7242094113454DIXON  F  1 972775   111      1  098                            9990105
177   7242095103753DOE                RAYMOND CLIFFORD                              9990101
178   7242095103753DOE                      302259729625                       1    9990102
```

```
179   724209510375 3DOE    000500 1                                                    9990103
180   724209510375 3DOE    F  1 972775   111      1  092                              9990105
181   724209610455 3DREW                GARY RONALD                                    9990101
182   724209610455 3DREW                      514259729625                        1    9990102
183   724209610455 3DREW    010501 1 E                                                 9990103
184   724209610455 3DREW    F  1 972775   111      1  051                              9990105
185   724209711985 11D.SOUZA            ANDREW BERNARD JOHN                            9990101
186   724209711985 1D.SOUZ                    305259729625                        1    9990102
187   724209711985 1D.SOUZ 001002 11                          E          E               9990103
188   724209711985 1D.SOUZ F  1 972775   111      1  011                              9990105
189   724209811405 21EATON              GORDON STANLEY                                 9990101
190   724209811405 2EATON                     517259729625                        1    9990102
191   724209811405 2EATON  000701 11                      E                            9990103
192   724209811405 2EATON  F  1 972775   111      1  109                              9990105
193   724209910445 31ELVIN              JOHN                                           9990101
194   724209910445 3ELVIN                     439259729625                        1    9990102
195   724209910445 3ELVIN  000804 11                      A BCD                        9990103
196   724209910445 3ELVIN  F  1 972775   111      1  011                              9990105
197   724210011725 41FAULKS             MARTIN RICHARD                                 9990101
198   724210011725 4FAULKS                    423259729625                        1    9990102
199   724210011725 4FAULKS 000603 1 E         D                           E            9990103
200   724210011725 4FAULKS F  1 972775   111      1  104                              9990105
201   724210110805 31FEATHERSTONE       DENIS RONALD                                   9990101
202   724210110805 3FEATHE                    535259729625                        1    9990102
203   724210110805 3FEATHE 000801 11                                          F        9990103
204   724210110805 3FEATHE F  1 972775   111      1  011                              9990105
205   724210211-51 1FERGUSON            ANDREW MICHAEL                                 9990101
206   724210211-51 FERGUS                     431259729625                        1    9990102
207   724210211-51 FERGUS 000803 1      CB                                      C      9990103
208   724210211-51 FERGUS F  1 972775   111      1  061                              9990105
209   724210312275 11FERRY              BRIAN WILLIAM                                  9990101
210   724210312275 1FERRY                     517259729625                        1    9990102
211   724210312275 1FERRY  000401 1           E                                        9990103
212   724210312275 1FERRY  F  1 972775   111      1  104                              9990105
213   724210411875 41FINDLATER          MAXWELL KERR                                   9990101
214   724210411875 4FINDLA                    299259729625                        2    9990102
215   724210411875 4FINDLA 000000                                                      9990103
216   724210411875 4FINDLA F  1 972775   111      1  002                              9990105
217   724210511635 31FITZSIMMONS        MALCOLM LEWIS                                  9990101
218   724210511635 3FITZSI                    405259729625                        1    9990102
219   724210511635 3FITZSI 000702 11                          E          E               9990103
220   724210511635 3FITZSI F  1 972775   111      1  048                              9990105
221   724210611485 21FOWKES             ROGER CHARLES                                  9990101
222   724210611485 2FOWKES                    430259729625                        1    9990102
223   724210611485 2FOWKES 000503 1 E          E                          E            9990103
224   724210611485 2FOWKES F  1 972775   111      1  061                              9990105
225   724210712845 31FOX                PAUL DOUGLAS                                   9990101
226   724210712845 3FOX                       509259729625                        1    9990102
227   724210712845 3FOX    000603 1 B          A                          D            9990103
228   724210712845 3FOX    F  1 972775   111      1  051                              9990105
229   724210812132 41FRANKLIN           DEREK ANTHONY AIDAN                            9990101
230   724210812132 4FRANKL                    517259729625                       18    9990102
231   724210812132 4FRANKL 000600 1                                                    9990103
232   724210812132 4FRANKL F  1 972775   111      1  098                              9990105
233   724210912835 31GALLIE             DUNCAN ANDREW                                  9990101
234   724210912835 3GALLIE                    427259729625                        1    9990102
235   724210912835 3GALLIE 000503 1 E          B  D                                    9990103
236   724210912835 3GALLIE F  1 972775   111      1  051                              9990105
237   724211010485 31GARBUTT            IAN                                            9990101
238   724211010485 3GARBUT                    535259729625                        1    9990102
```

```
239   7242110104853GARBUT C0C801 11                                  E        9990103
240   7242110104853GARBUT F  1 972775   111     1  061                        9990105
241   7242111121543IGARDINER          JOHN ROGAN                              9990101
242   7242111121543GARDIN                   422259729625                  12  9990102
243   7242111121543GARDIN C0C603 1                 E                BE       9990103
244   7242111121543GARDIN F  1 972775   111     1  051                        9990105
245   7242112101952IGILFILLAN         HOWARD                                  9990101
246   7242112101952GILFIL                   411259729625                   1  9990102
247   7242112101952GILFIL 020502 11                 E                  B     9990103
248   7242112101952GILFIL F  1 972775   111     1  002                        9990105
249   7242113128152IGILL              RAYMOND CHRISTOPHER                     9990101
250   7242113128152GILL                     403259729625                   1  9990102
251   7242113128152GILL   C0C502 1 E                                  E      9990103
252   7242113128152GILL   F  1 972775   111     1  061                        9990105
253   7242114125750IGOODWIN           TOBY MICHAEL GILES                      9990101
254   7242114125750GOODWI                   427259729625                   1  9990102
255   7242114125750GOODWI 000602 1   E       D                               9990103
256   7242114125750GOODWI F  1 972775   111     1  061                        9990105
257   7242115111148IGOSLING           BRIAN GEORGE EDWARD                     9990101
258   7242115111148GOSLIN                   437259729625                   1  9990102
259   7242115111148GOSLIN C00700 11                                           9990103
260   7242115111148GOSLIN F  1 972775   111     1  C48                        9990105
261   7242116118652IGOW               PETER DRAFFIN                           9990101
262   7242116118652GOW                      442259729625                   1  9990102
263   7242116118652GOW    C0C803 1 E         F                           D    9990103
264   7242116118652GOW    F  1 972775   111     1  098                        9990105
265   7242117113154IGRAVELING         JOHN EDWARD                             9990101
266   7242117113154GRAVEL                   518259729625                   1  9990102
267   7242117113154GRAVEL C0C500 11                                           9990103
268   7242117113154GRAVEL F  1 972775   111     1  C02                        9990105
269   7242118118652IGRAYDON           GEOFFREY                                9990101
270   7242118118652GRAYDO                   431259729625                   1  9990102
271   7242118118652GRAYDO 000500 11                                           9990103
272   7242118118652GRAYDO F  1 972775   111     1  002                        9990105
273   7242119110653IGREEN             DAVID WILLIAM                           9990101
274   7242119110653GREEN                    427259729625                   1  9990102
275   7242119110653GREEN  C00702 11E         C                               9990103
276   7242119110653GREEN  F  1 972775   111     1  061                        9990105
277   7242120101553IGRIFFIN           RICHARD CHARLES                         9990101
278   7242120101553GRIFFI                   472259729625                   1  9990102
279   7242120101553GRIFFI C00701 11                          E               9990103
280   7242120101553GRIFFI F  1 972775   111     1  C48                        9990105
281   7242121101-491GUYMER            WILLIAM STEPHENSON                      9990101
282   7242121101-49GUYMER                   411259729625                   1  9990102
283   7242121101-49GUYMER 000500  1                                           9990103
284   7242121101-49GUYMER F  1 972775   111     1  098                        9990105
285   7242122108854IHALLIDAY          STEPHEN                                 9990101
286   7242122108854HALLID                   472259729625                   1  9990102
287   7242122108854HALLID 040201 11          E                               9990103
288   7242122108854HALLID F  1 972775   111     1  051                        9990105
289   7242123101954IHAMILTON          IAN                                     9990101
290   7242123101954HAMILT                   445259729625                   1  9990102
291   7242123101954HAMILT 000502  1             B                       C     9990103
292   7242123101954HAMILT F  1 972775   111     1  G02                        9990105
293   7242124112754IHAMILTON          MALCOLM IAN                             9990101
294   7242124112754HAMILT                   533259729625                   1  9990102
295   7242124112754HAMILT C00600 11                                           9990103
296   7242124112754HAMILT F  1 972775   111     1  048                        9990105
297   7242125101-521HANSEN            PAUL                                    9990101
298   7242125101-52HANSEN                   495259729625                   1  9990102
```

```
299   7242125101-52HANSEN 010401 1          C                                  9990103
300   7242125101-52HANSEN F  1 972775   111     1  109                         9990105
301   7242126128634HANSON             JOHN                                     9990101
302   7242126128634HANSON                   411259729625                   17  9990102
303   7242126128634HANSON 000700 11                                            9990103
304   7242126128634HANSON F  1 972775   111     1  051                         9990105
305   7242127124-521HARDY             KEITH JOHN                               9990101
306   7242127124-52HARDY                    514259729625                    1  9990102
307   7242127124-52HARDY  000700 1                                             9990103
308   7242127124-52HARDY  F  1 972775   111     1  048                         9990105
309   7242129104454HEDLEY             NORMAN ANTHONY                           9990101
310   7242129104454HEDLEY                   514259729625                    1  9990102
311   7242129104454HEDLEY 000401 1                                   F         9990103
312   7242129104454HEDLEY F  1 972775   111     1  051                         9990105
313   7242130129&531HESKETH           NEIL WILLIAM                             9990101
314   7242130129&53HESKET                   492259729625                    1  9990102
315   7242130129&53HESKET 000801 11                                  F         9990103
316   7242130129&53HESKET F  1 972775   111     1  098                         9990105
317   7242132102053 1HINCH            JAMES ROBERTSON                          9990101
318   7242132102053HINCH                    509259729625                    1  9990102
319   7242132102053HINCH  000601 11          E                                 9990103
320   7242132102053HINCH  F  1 972775   111     1  002                         9990105
321   7242133103153 1HIRD             PETER GRAHAM                             9990101
322   7242133103153HIRD                     472259729625                    1  9990102
323   7242133103153HIRD   000802 11                      EE                     9990103
324   7242133103153HIRD   F  1 972775   111     1  048                         9990105
325   7242134121953 1HODGKISS         PAUL                                     9990101
326   7242134121953HODGKI                   535259729625                    1  9990102
327   7242134121953HODGKI 040300 11                                            9990103
328   7242134121953HODGKI F  1 972775   111     1  011                         9990105
329   7242135120653 1HOLROYD          ANDREW CHARLES                           9990101
330   7242135120653HOLROY                   411259729625                    1  9990102
331   7242135120653HOLROY 000902 1 E   E                                       9990103
332   7242135120653HOLROY F  1 972775   111     1  057                         9990105
333   7242136116753 1HOLYOME          JOHN MARK                                9990101
334   7242136116753HOLYOM                   427259729625                    1  9990102
335   7242136116753HOLYOM 000801 11B                                           9990103
336   7242136116753HOLYOM F  1 972775   111     1  061                         9990105
337   7242137116554 1HOUGHTON         TREVOR KEITH                             9990101
338   7242137116554HOUGHT                   416259729625                    1  9990102
339   7242137116554HOUGHT 000803 11      D                 E  B                 9990103
340   7242137116554HOUGHT F  1 972775   111     1  104                         9990105
341   7242138124454 1HOULDING         ROBIN RICHARD                            9990101
342   7242138124454HOULDI                   411259729625                    1  9990102
343   7242138124454HOULDI 000601 1           E                                 9990103
344   7242138124454HOULDI F  1 972775   111     1  104                         9990105
345   7242139122052 1HOWLAND          JONATHAN VICTOR                          9990101
346   7242139122052HOWLAN                   449259729625                    1  9990102
347   7242139122052HOWLAN 000602  1          E                     C           9990103
348   7242139122052HOWLAN F  1 972775   111     1  002                         9990105
349   7242140118052 1HULME            ROBIN ANTHONY                            9990101
350   7242140118052HULME                    405259729625                    1  9990102
351   7242140118052HULME  010603 11          B                 D       D        9990103
352   7242140118052HULME  F  1 972775   111     1  051                         9990105
353   7242141108754 1HUTCHINSON       STANLEY JAMES                            9990101
354   7242141108754HUTCHI                   517259729625                    1  9990102
355   7242141108754HUTCHI 000800 11                                            9990103
356   7242141108754HUTCHI F  1 972775   111     1  104                         9990105
357   7242142112453 1JACKSON          PAUL                                     9990101
358   7242142112453JACKSO                   535259729625                    1  9990102
```

```
359   7242142112453JACKSO 000602 11                          E     F                  9990103
360   7242142112453JACKSO F  1 972775   111      1  002                              9990105
361   7242143115&531JOHN                DAVID NEIL                                    9990101
362   7242143115&53JOHN                      423259729625                        1    9990102
363   7242143115&53JOHN   000602 115 F                                               9990103
364   7242143115&53JOHN   F  1 972775   111      1  061                              9990105
365   7242144123552 1JOHNSTON           DEREK                                         9990101
366   7242144123552JOHNST                    514259729625                        1    9990102
367   7242144123552JOHNST 000603 11B          B                            D         9990103
368   7242144123552JOHNST F  1 972775   111      1  051                              9990105
369   7242145105254 1JONES              KEITH                                         9990101
370   7242145105254JONES                     461259729625                        1    9990102
371   7242145105254JONES  000602 11D          E                                       9990103
372   7242145105254JONES  F  1 972775   111      1  061                              9990105
373   7242146126754 1JONES              NORMAN                                        9990101
374   7242146126754JONES                     411259729625                        1    9990102
375   7242146126754JONES  000601 11                                  C               9990103
376   7242146126754JONES  F  1 972775   111      1  104                              9990105
377   7242147107552 1KELLY              NEIL HAROLD                                   9990101
378   7242147107552KELLY                     488259729625                        1    9990102
379   7242147107552KELLY  000603 11C   C  D                                           9990103
380   7242147107552KELLY  F  1 972775   111      1  061                              9990105
381   7242148124&521KIDNEY              LANCE NICHOLAS                                9990101
382   7242148124&52KIDNEY                    314259729625                        1    9990102
383   7242148124&52KIDNEY 000703 11C   D      E                                       9990103
384   7242148124&52KIDNEY F  1 972775   111      1  061                              9990105
385   7242149118051 1KNOX               JEFFREY NICHOLAS                              9990101
386   7242149118051KNOX                      517259729625                        1    9990102
387   7242149118051KNOX   001102 11                          E          B             9990103
388   7242149118051KNOX   F  1 972775   111      1  048                              9990105
389   7242150129952 1LAUDER             IAN WILLIAM                                   9990101
390   7242150129952LAUDER                    514259729625                        1    9990102
391   7242150129952LAUDER 000601 11                          D                        9990103
392   7242150129952LAUDER F  1 972775   111      1  048                              9990105
393   7242151123654 1LAWSON             CHARLES WILLIAM PAGAN                         9990101
394   7242151123654LAWSON                    299259729625                        2    9990102
395   7242151123654LAWSON 000000                                                     9990103
396   7242151123654LAWSON F  1 972775   111      1  002                              9990105
397   7242152104653 1LINE               STEPHEN CHARLES                               9990101
398   7242152104653LINE                      437259729625                        1    9990102
399   7242152104653LINE   030200 1                                                    9990103
400   7242152104653LINE   F  1 972775   111      1  104                              9990105
401   7242153124454 1LOWEY              KENNETH                                       9990101
402   7242153124454LOWEY                     449259729625                        1    9990102
403   7242153124454LOWEY  000802 11                          C        B               9990103
404   7242153124454LOWEY  F  1 972775   111      1  048                              9990105
405   7242154122744 1LUMSDEN            DAVID WILLIAM                                 9990101
406   7242154122744LUMSDE                    411259729625                        12   9990102
407   7242154122744LUMSDE 000700 11                                                  9990103
408   7242154122744LUMSDE F  1 972775   111      1  061                              9990105
409   7242155123951 1LYNCH              CHRISTOPHER MICHAEL                           9990101
410   7242155123951LYNCH                     455259729625                        1    9990102
411   7242155123951LYNCH  010503 11E                                 E  E             9990103
412   7242155123951LYNCH  F  1 972775   111      1  061                              9990105
413   7242156106544 1MCCALL             COLIN WINTER                                  9990101
414   7242156106544MCCALL                    299259729625                        20   9990102
415   7242156106544MCCALL 000000                                                     9990103
416   7242156106544MCCALL F  1 972775   111      1  052                              9990105
417   7242157122450 1MCDONALD           JOHN                                          9990101
418   7242157122450MCDONA                    315259729625                        1    9990102
```

```
419   7242157122453MCDONA C0C500 11                                          9990103
420   7242157122453MCDONA F  1 972775   111     1  002                       9990105
421   7242158108453IMACLEOD            TORQUIL ROBERT                        9990101
422   7242158108453MACLEO                   411259729625                 1   9990102
423   7242158108453MACLEO C0C501 1           F                               9990103
424   7242158108453MACLEO F  1 972775   111     1  057                       9990105
425   7242160118553IMARSHALL           STEPHEN JOHN                          9990101
426   7242160118553MARSHA                   307259729625                 1   9990102
427   7242160118553MARSHA 010301 1                  D                        9990103
428   7242160118553MARSHA F  1 972775   111     1  061                       9990105
429   7242161116953IMATTHEWS           TIMOTHY JOHN                          9990101
430   7242161116953MATTHE                   449259729625                 1   9990102
431   7242161116953MATTHE C00702 11             C                     D      9990103
432   7242161116953MATTHE F  1 972775   111     1  052                       9990105
433   7242162116254IMEDLOCK            STUART ANTONY                         9990101
434   7242162116254MEDLOC                   411259729625                 1   9990102
435   7242162116254MEDLOC 000701 11                        E                 9990103
436   7242162116254MEDLOC F  1 972775   111     1  011                       9990105
437   7242163119453IMERCER             NICHOLAS OLIVER ANDREW                9990101
438   7242163119453MERCER                   422259729625                 1   9990102
439   7242163119453MERCER 000602 11          D                     D         9990103
440   7242163119453MERCER F  1 972775   111     1  052                       9990105
441   7242164104549IMIDDLETON          COLIN TYSON                           9990101
442   7242164104549MIDDLE                   500259729625                 1   9990102
443   7242164104549MIDDLE C00602 1 D   E                                     9990103
444   7242164104549MIDDLE F  1 972775   111     1  051                       9990105
445   7242165116454IMILLS              DAVID                                 9990101
446   7242165116454MILLS                    518259729625                 1   9990102
447   7242165116454MILLS  C00700 11                                          9990103
448   7242165116454MILLS  F  1 972775   111     1  002                       9990105
449   7242166106454IMINTO              EDWARD JOHN                           9990101
450   7242166106454MINTO                    411259729625                 1   9990102
451   7242166106454MINTO  C00702 11E   E                                     9990103
452   7242166106454MINTO  F  1 972775   111     1  061                       9990105
453   7242167128952IMONTGOMERY         PAUL                                  9990101
454   7242167128952MONTGO                   411259729625                 1   9990102
455   7242167128952MONTGO C0C500 11                                          9990103
456   7242167128952MONTGO F  1 972775   111     1  002                       9990105
457   7242168121152IMOONEY             JOHN FRANCIS                          9990101
458   7242168121152MOONEY                   535259729625                 1   9990102
459   7242168121152MOONEY C00901 11D                                         9990103
460   7242168121152MOONEY F  1 972775   111     1  061                       9990105
461   7242169129952IMORDUE             ROBERT WEDDERBURN                     9990101
462   7242169129952MORDUE                   431259729625                 1   9990102
463   7242169129952MORDUE 000701 1                         E                 9990103
464   7242169129952MORDUE F  1 972775   111     1  002                       9990105
465   7242170115349IMORGAN             NEIL                                  9990101
466   7242170115349MORGAN                   411259729625                 1   9990102
467   7242170115349MORGAN C00600 11                                          9990103
468   7242170115349MORGAN F  1 972775   111     1  002                       9990105
469   7242171128754IMURRAY             MITCHELL JOHN                         9990101
470   7242171128754MURRAY                   299259729625                 2   9990102
471   7242171128754MURRAY C00000                                             9990103
472   7242171128754MURRAY F  1 972775   111     1  002                       9990105
473   7242172107854IMURRAY             PAUL                                  9990101
474   7242172107854MURRAY                   411259729625                 1   9990102
475   7242172107854MURRAY 000600 1                                           9990103
476   7242172107854MURRAY F  1 972775   111     1  051                       9990105
477   7242173124052INEVINS             MICHAEL JOHN                          9990101
478   7242173124052NEVINS                   431259729625                 1   9990102
```

```
479   7242173124052NEVINS C00701 11E                                          9990103
480   7242173124052NEVINS F  1 972775   111     1  002                        9990105
481   7242174125053INICHOLSON         PETER THOMAS                            9990101
482   7242174125053NICHOL                   416259729625                  1   9990102
483   7242174125053NICHOL 010600 1                                            9990103
484   7242174125053NICHOL F  1 972775   111     1  002                        9990105
485   7242175114-531NICKLIN           JOHN GILSTIN                            9990101
486   7242175114-53NICKLI                   517259729625                  1   9990102
487   7242175114-53NICKLI 010401 11           C                               9990103
488   7242175114-53NICKLI F  1 972775   111     1  104                        9990105
489   7242176106654INICOL             GARETH JAMES                            9990101
490   7242176106654NICOL                    514259729625                  1   9990102
491   7242176106654NICOL  000502 1           C                    D          9990103
492   7242176106654NICOL  F  1 972775   111     1  051                        9990105
493   7242177116153IO.DONNELL         PAUL ANTHONY                            9990101
494   7242177116153O.DONN                   505259729625                  1   9990102
495   7242177116153O.DONN 010502  1                        E     B            9990103
496   7242177116153O.DONN F  1 972775   111     1  048                        9990105
497   7242178123053IOLD               BRIAN LESLIE                            9990101
498   7242178123053OLD                      411259729625                  1   9990102
499   7242178123053OLD    010701 11           E                               9990103
500   7242178123053OLD    F  1 972775   111     1  002                        9990105
501   7242179102354IOUGHTON           DAVID                                   9990101
502   7242179102354OUGHTO                   411259729625                  1   9990102
503   7242179102354OUGHTO 000802 11                        E          D        9990103
504   7242179102354OUGHTO F  1 972775   111     1  052                        9990105
505   7242180101553IPARKER            JOHN DARRELL                            9990101
506   7242180101553PARKER                   519259729625                  1   9990102
507   7242180101553PARKER 000701 11           D                               9990103
508   7242180101553PARKER F  1 972775   111     1  104                        9990105
509   7242181114554IPARTINGTON        PHILLIP ERIC                            9990101
510   7242181114554PARTIN                   304259729625                  1   9990102
511   7242181114554PARTIN 030300 1                                            9990103
512   7242181114554PARTIN F  1 972775   111     1  052                        9990105
513   7242182126654IPICKERING         DAVID                                   9990101
514   7242182126654PICKER                   535259729625                  1   9990102
515   7242182126654PICKER 000602 11           E                    E          9990103
516   7242182126654PICKER F  1 972775   111     1  051                        9990105
517   7242183114553IPILLEY            NICHOLAS JOHN                           9990101
518   7242183114553PILLEY                   470259729625                  1   9990102
519   7242183114553PILLEY 000602 11C          D                               9990103
520   7242183114553PILLEY F  1 972775   111     1  061                        9990105
521   7242184124654IRAMSEY            NEIL                                    9990101
522   7242184124654RAMSEY                   411259729625                  1   9990102
523   7242184124654RAMSEY 000803 1 E          D                    E          9990103
524   7242184124654RAMSEY F  1 972775   111     1  051                        9990105
525   7242185110124IRAMSHAW           MARSHALL                                9990101
526   7242185110124RAMSHA                   411259729625                 19   9990102
527   7242185110124RAMSHA 000700  1                                           9990103
528   7242185110124RAMSHA F  1 972775   111     1  011                        9990105
529   7242186103254IRAW               BRIAN                                   9990101
530   7242186103254RAW                      411259729625                  1   9990102
531   7242186103254RAW    000801 11                            C          9990103
532   7242186103254RAW    F  1 972775   111     1  052                        9990105
533   7242187122154IREAY              DAVID                                   9990101
534   7242187122154REAY                     412259729625                  1   9990102
535   7242187122154REAY   001001 11                        B                   9990103
536   7242187122154REAY   F  1 972775   111     1  048                        9990105
537   7242188105-531RICHARDSON        SYDNEY                                  9990101
538   7242188105-53RICHAR                   411259729625                  1   9990102
```

```
539  7242188105-53RICHAR C0C9C0 11                                          9990103
540  7242188105-53RICHAR F  1 972775   111     1  002                       9990105
541  7242189129147RILEY              STEWART                                9990101
542  7242189129147RILEY                   48J259729625                  1S  9990102
543  7242189129147RILEY  000600 1                                           9990103
544  7242189129147RILEY  F  1 972775   111     1  104                       9990105
545  7242190111850RIPPON             PHILIP DAVID BARTLE                    9990101
546  7242190111850RIPPON                  431259729625                  1   9990102
547  7242190111850RIPPON 000401 11                                  F       9990103
548  7242190111850RIPPON F  1 972775   111     1  052                       9990105
549  7242191112952ROWBOTHAM          PHILIP                                 9990101
550  7242191112952ROWBOT                  4C5259729625                  1   9990102
551  7242191112952ROWBOT 020500 11                                          9990103
552  7242191112952ROWBOT F  1 972775   111     1  061                       9990105
553  7242192108254RUTHERFORD         JOHN FRANCIS                           9990101
554  7242192108254RUTHER                  517259729625                  1   9990102
555  7242192108254RUTHER C0C801 11                                  D       9990103
556  7242192108254RUTHER F  1 972775   111     1  C02                       9990105
557  7242193120052SCORER             ALEXANDER                              9990101
558  7242193120052SCORER                  411259729625                  1   9990102
559  7242193120052SCORER 000701 11        C                                 9990103
560  7242193120052SCORER F  1 972775   111     1  104                       9990105
561  7242195129-52SIBBALD            JULIAN                                 9990101
562  7242195129-52SIBBAL                  422259729625                  1   9990102
563  7242195129-52SIBBAL 000701 11C                                         9990103
564  7242195129-52SIBBAL F  1 972775   111     1  C61                       9990105
565  7242196105850SIMPSON            JAMES                                  9990101
566  7242196105850SIMPSO                  411259729625                  1   9990102
567  7242196105850SIMPSO 000600 11                                          9990103
568  7242196105850SIMPSO F  1 972775   111     1  011                       9990105
569  7242197110950SMITH              PHILIP                                 9990101
570  7242197110950SMITH                   498259729625                  1   9990102
571  7242197110950SMITH  000600 11                                          9990103
572  7242197110950SMITH  F  1 972775   111     1  052                       9990105
573  7242198118754SPEDDING           LEWIS CHRISTOPHER                      9990101
574  7242198118754SPEDDI                  468259729625                  1   9990102
575  7242198118754SPEDDI 000500 1                                           9990103
576  7242198118754SPEDDI F  1 972775   111     1  098                       9990105
577  7242199120-50STAINTHORPE        DAVID                                  9990101
578  7242199120-50STAINT                  533259729625                  1   9990102
579  7242199120-50STAINT 000803 11                     BEE                  9990103
580  7242199120-50STAINT F  1 972775   111     1  048                       9990105
581  7242200127853STANLEY-CLAMP      RICHARD STEPHEN INGLIS                 9990101
582  7242200127853STANLE                  498259729625                  1   9990102
583  7242200127853STANLE C00702 1 C        E                                9990103
584  7242200127853STANLE F  1 972775   111     1  061                       9990105
585  7242201103353STAPLES            FRANCIS TIMOTHY                        9990101
586  7242201103353STAPLE                  441259729625                  1   9990102
587  7242201103353STAPLE 000600 1                                           9990103
588  7242201103353STAPLE F  1 972775   111     1  051                       9990105
589  7242202123852STAYMAN            GEORGE MICHAEL                         9990101
590  7242202123852STAYMA  CRIMMEN         489259729625                  1   9990102
591  7242202123852STAYMA C00601 11                                  E       9990103
592  7242202123852STAYMA F  1 972775   111     1  002                       9990105
593  7242203118352STEAD              JOHN CHESSWAS                          9990101
594  7242203118352STEAD                   535259729625                  1   9990102
595  7242203118352STEAD  000700 11                                          9990103
596  7242203118352STEAD  F  1 972775   111     1  C98                       9990105
597  7242204117353STEPHENS           PHILIP MORGAN                          9990101
598  7242204117353STEPHE                  422259729625                  1   9990102
```

```
599   7242204117353STEPHE 040403 11                                  EE F        9990103
600   7242204117353STEPHE F  1 972775   111     1  002                           9990105
601   7242205102-511STEWART            JONATHAN MALCOLM                          9990101
602   7242205102-51STEWAR                   479259729625                   1     9990102
603   7242205102-51STEWAR 000802 11                 F                   E        9990103
604   7242205102-51STEWAR F  1 972775   111     1  109                           9990105
605   7242206108353STRAW               CHRISTOPHER STEPHEN                       9990101
606   7242206108353STRAW                    498259729625                   1     9990102
607   7242206108353STRAW  010400 11                                               9990103
608   7242206108353STRAW  F  1 972775   111     1  011                           9990105
609   7242207113853SUMMERS             ROBERT CRUSE                              9990101
610   7242207113853SUMMER                   441259729625                   1     9990102
611   7242207113853SUMMER 000603 11B   D       E                                  9990103
612   7242207113853SUMMER F  1 972775   111     1  057                           9990105
613   7242208129554TAIT                ALAN                                      9990101
614   7242208129554TAIT                     411259729625                   1     9990102
615   7242208129554TAIT   000801 11                      E                         9990103
616   7242208129554TAIT   F  1 972775   111     1  048                           9990105
617   7242209113154TARRANT             GEOFFREY PAUL                             9990101
618   7242209113154TARRAN                   450259729625                   1     9990102
619   7242209113154TARRAN 000902 11                         D         E           9990103
620   7242209113154TARRAN F  1 972775   111     1  048                           9990105
621   7242210127254TAYLOR              JOHN BARRY                                9990101
622   7242210127254TAYLOR                   411259729625                   1     9990102
623   7242210127254TAYLOR 000500 11                                               9990103
624   7242210127254TAYLOR F  1 972775   111     1  002                           9990105
625   7242211115452 TAYLOR             TIMOTHY CHARLES HARCOURT                  9990101
626   7242211115452TAYLOR                   446259729625                   1     9990102
627   7242211115452TAYLOR 000602 1 C                                  E           9990103
628   7242211115452TAYLOR F  1 972775   111     1  061                           9990105
629   7242212109640 THOMPSON           ROBERT                                    9990101
630   7242212109640THOMPS                   480259729625                   17    9990102
631   7242212109640THOMPS 000200 1                                                9990103
632   7242212109640THOMPS F  1 972775   111     1  061                           9990105
633   7242213112853THOMPSON            WILLIAM                                   9990101
634   7242213112853THOMPS                   497259729625                   3     9990102
635   7242213112853THOMPS 000702 11C         E                                    9990103
636   7242213112853THOMPS F  1 972775   111     1  061                           9990105
637   7242214106754THWAITES            MARTIN JOSEPH EDWARD                      9990101
638   7242214106754THWAIT                   407259729625                   1     9990102
639   7242214106754THWAIT 000801 11                                E              9990103
640   7242214106754THWAIT F  1 972775   111     1  104                           9990105
641   7242215117-521TRACE              DAVID JOHN                                9990101
642   7242215117-52TRACE                    447259729625                   1     9990102
643   7242215117-52TRACE  000702 11          E                            D       9990103
644   7242215117-52TRACE  F  1 972775   111     1  051                           9990105
645   7242216109252TREWARTHA           ALEC BLAIR                                9990101
646   7242216109252TREWAR                   450259729625                   1     9990102
647   7242216109252TREWAR 000600 11                                               9990103
648   7242216109252TREWAR F  1 972775   111     1  104                           9990105
649   7242217123454UNDERWOOD           HOWARD DOUGLAS                            9990101
650   7242217123454UNDERW                   500259729625                   1     9990102
651   7242217123454UNDERW 040500 11                                               9990103
652   7242217123454UNDERW F  1 972775   111     1  052                           9990105
653   7242218111052VICKERS             JOHN JAMES                                9990101
654   7242218111052VICKER                   422259729625                   1     9990102
655   7242218111052VICKER 000602 11                                   CD          9990103
656   7242218111052VICKER F  1 972775   111     1  051                           9990105
657   7242219120-511WATSON             ROBERT JOHN                               9990101
658   7242219120-51WATSON                   450259729625                   1     9990102
```

```
659   7242219120-51WATSON 000702 11                                E          E        9990103
660   7242219120-51WATSON F  1 972775   111      1  048                                 9990105
661   7242220129053WEST                JEREMY LEIGH                                     9990101
662   7242220129053WEST                     443259729625                          3     9990102
663   7242220129053WEST   000702 11    E                                D               9990103
664   7242220129053WEST   F  1 972775   111      1  092                                 9990105
665   7242221121553WHITE               PHILIP WILLIAM                                   9990101
666   7242221121553WHITE                    423259729625                          1     9990102
667   7242221121553WHITE  000804 1 B         E                              EE          9990103
668   7242221121553WHITE  F  1 972775   111      1  052                                 9990105
669   7242222126850WIDDRINGTON         THOMAS                                           9990101
670   7242222126850WIDDRI                   480259729625                          1     9990102
671   7242222126850WIDDRI 000500 11                                                     9990103
672   7242222126850WIDDRI F  1 972775   111      1  002                                 9990105
673   7242223123453WILKINSON           IAN DAVID                                        9990101
674   7242223123453WILKIN                   504259729625                          1     9990102
675   7242223123453WILKIN 010503 11F         C                              E           9990103
676   7242223123453WILKIN F  1 972775   111      1  051                                 9990105
677   7242224103347WILKINSON           JAMES EDWARD                                     9990101
678   7242224103347WILKIN                   411259729625                          19    9990102
679   7242224103347WILKIN 000700 11                                                     9990103
680   7242224103347WILKIN F  1 972775   111      1  098                                 9990105
681   7242225111352WILKINSON           STEPHEN ARTHUR                                   9990101
682   7242225111352WILKIN                   509259729625                          1     9990102
683   7242225111352WILKIN 000602 1                                             CC        9990103
684   7242225111352WILKIN F  1 972775   111      1  052                                 9990105
685   7242226131754WILLIAMS            HOWARD MARTYN                                    9990101
686   7242226131754WILLIA                   551259729625                          1     9990102
687   7242226131754WILLIA 000501 1           E                                          9990103
688   7242226131754WILLIA F  1 972775   111      1  052                                 9990105
689   7242227111251WILSON              ALAN HARDY                                       9990101
690   7242227111251WILSON                   517259729625                          1     9990102
691   7242227111251WILSON 000502 1 C         E                                          9990103
692   7242227111251WILSON F  1 972775   111      1  061                                 9990105
693   7242228107254WILSON              GEOFFREY                                         9990101
694   7242228107254WILSON                   431259729625                          1     9990102
695   7242228107254WILSON 000500 11                                                     9990103
696   7242228107254WILSON F  1 972775   111      1  002                                 9990105
697   7242229124052WILSON              JOHN MARK FLEMING                                9990101
698   7242229124052WILSON                   443259729625                          1     9990102
699   7242229124052WILSON 000701 1           E                                          9990103
700   7242229124052WILSON F  1 972775   111      1  057                                 9990105
701   7242230123354WILSON              KEVIN WILLIAM                                    9990101
702   7242230123354WILSON                   411259729625                          1     9990102
703   7242230123354WILSON 000603 11D         D                              C           9990103
704   7242230123354WILSON F  1 972775   111      1  052                                 9990105
705   7242231104554WREN                CHRISTOPHER JOHN                                 9990101
706   7242231104554WREN                     422259729625                          1     9990102
707   7242231104554WREN   000702 11                          EE                          9990103
708   7242231104554WREN   F  1 972775   111      1  048                                 9990105
709   7242232108053YOUNG               CLIVE PHILIP                                     9990101
710   7242232108053YOUNG                    405259729625                          1     9990102
711   7242232108053YOUNG  001100 1                                                      9990103
712   7242232108053YOUNG  F  1 972775   111      1  011                                 9990105
713   7242233118654BARKER              ALAN MALCOLM                                     9990101
714   7242233118654BARKER                   535259729625                          1     9990102
715   7242233118654BARKER 000600 11                                                     9990103
716   7242233118654BARKER F  1 972775   111      1  002                                 9990105
717   7242234123853PATMAN              BRIAN SAMUEL                                     9990101
718   7242234123853PATMAN                   299259729625                          2     9990102
```

```
719   724223412365 3PATMAN 000000                                                    9990103
720   724223412365 3PATMAN F  1 972775   111      1  061                            9990105
721   724223511135 41TEMPLE              ROBERT KEITH                               9990101
722   724223511135 4TEMPLE                     411259729625                     1    9990102
723   724223511135 4TEMPLE 000602 11                              C B               9990103
724   724223511135 4TEMPLE F  1 972775   111      1  104                            9990105
725   724223612705 11TARRELL             JAMES ANDREW SCOTT                         9990101
726   724223612705 1TARREL                     433259729625                     1    9990102
727   724223612705 1TARREL 000601 1                                 E                9990103
728   724223612705 1TARREL F  1 972775   111      1  048                            9990105
729   724223711463 71WILLIAMSON          KENNETH                                    9990101
730   724223711463 7WILLIA                     411259729625                     19   9990102
731   724223711463 7WILLIA 000100 1                                                 9990103
732   724223711463 7WILLIA                           1                              9990104
733   724223711463 7WILLIA F  1 9727741  111      1  104                            9990105
734   724223811354 91ALLAN               DONALD                                     9990101
735   724223811354 9ALLAN                      495259729625                     1    9990102
736   724223811354 9ALLAN  000600 11                                                9990103
737   724223811354 9ALLAN  F  1 972773   111        1                              9990105
738   724223911703 91ASKEW               MARK RUSSELL                               9990101
739   724223911703 9ASKEW                      535259729625                     19   9990102
740   724223911703 9ASKEW  000100 1                                                 9990103
741   724223911703 9ASKEW                            1                              9990104
742   724223911703 9ASKEW  F  1 972773   111        1                              9990105
743   724224011224 71DALTON              JOHN THOMAS                                9990101
744   724224011224 7DALTON                     535259729625                     12   9990102
745   724224011224 7DALTON 000903 11                              BA  D             9990103
746   724224011224 7DALTON F  1 972773   111        1                              9990105
747   724224111644 21GOODWIN             DEREK                                      9990101
748   724224111644 2GOODWI                     535259729625                     19   9990102
749   724224111644 2GOODWI 000200 1                                                 9990103
750   724224111644 2GOODWI                           1                              9990104
751   724224111644 2GOODWI F  1 972773   111        1                              9990105
752   724224210312 81GREEN               JOSEPH HUDSON                              9990101
753   724224210312 8GREEN                      411259729625                     10   9990102
754   724224210312 8GREEN  000500 1                                                 9990103
755   724224210312 8GREEN  F  1 972773   111        1                              9990105
756   724224312264 31HEATLEY             KENNETH JOSEPH                             9990101
757   724224312264 3HEATLE                     449259729625                     19   9990102
758   724224312264 3HEATLE 000500                                                   9990103
759   724224312264 3HEATLE F  1 972773   111        1                              9990105
760   724224410804 91MAIDMENT            PETER                                      9990101
761   724224410804 9MAIDME                     503259729625                     1    9990102
762   724224410804 9MAIDME 000801 11                                C               9990103
763   724224410804 9MAIDME F  1 972773   111        1                              9990105
764   724224511 2-321MOREHEAD            WILLIAM CHARLES                            9990101
765   724224511 2-32MOREHE                     414259729625                     13   9990102
766   724224511 2-32MOREHE1000301 11                                E               9990103
767   724224511 2-32MOREHE F  1 972773   111        1                              9990105
768   724224610604 71PIPER               TERENCE WINSTON                            9990101
769   724224610604 7PIPER                      490259729625                     10   9990102
770   724224610604 7PIPER  000601 11E                                               9990103
771   724224610604 7PIPER  F  1 972773   111        1                              9990105
772   724224713084 41SANDS               DAVID                                      9990101
773   724224713084 4SANDS                      535259729625                     19   9990102
774   724224713084 4SANDS  000300 11                                                9990103
775   724224713084 4SANDS                            1                              9990104
776   724224713084 4SANDS  F  1 972773   111        1                              9990105
777   724224812992 81SHORT               HARRY                                      9990101
778   724224812992 8SHORT                      533259729625                     19   9990102
```

```
779  724224812992 8SHORT  000000                                                  9990103
780  724224812992 8SHORT                          1                              9990104
781  724224812992 8SHORT  F  1 972773   111       1                              9990105
782  724224911244 61WHITE              WILLIAM SELWYN                            9990101
783  724224911244 6WHITE                    431259729625                     19  9990102
784  724224911244 6WHITE  000601 11                        F                     9990103
785  724224911244 6WHITE  F  1 972773   111       1                              9990105
786  724225010133 21WOLFE              PETER LESLIE                              9990101
787  724225010133 2WOLFE                    533259729625                     19  9990102
788  724225010133 2WOLFE 100000 1                                                9990103
789  724225010133 2WOLFE                          1                              9990104
790  724225010133 2WOLFE  F  1 972773   111       1                              9990105
791  724225111384 51BAKER              ANDRE                                     9990101
792  724225111384 5BAKER                    404259729625                     10  9990102
793  724225111384 5BAKER  000804 11                     1 111                    9990103
794  724225111384 5BAKER  F  1 972773   111                    1                9990105
795  724225210265 11BARNES             DAVID PHILIP                              9990101
796  724225210265 1BARNES                   463259729625                      1  9990102
797  724225210265 1BARNES 000502 11          B                      C           9990103
798  724225210265 1BARNES F  1 972773   111                    1                9990105
799  724225310895 01BELL               MICHAEL RAYMOND                           9990101
800  724225310895 0BELL                     407259729625                      1  9990102
801  724225310895 0BELL   000804 11B   E          B                     C       9990103
802  724225310895 0BELL   F  1 972773   111                    1                9990105
803  724225410665 01BOWREY             PATRICK STEVEN                            9990101
804  724225410665 0BOWREY                   463259729625                      1  9990102
805  724225410665 0BOWREY 000603 1 C         D                      C           9990103
806  724225410665 0BOWREY F  1 972773   111                    1                9990105
807  724225511915 01CATLIN             NICHOLAS JOHN                             9990101
808  724225511915 0CATLIN                   416259729625                      1  9990102
809  724225511915 0CATLIN 020703 11                     D  CD                    9990103
810  724225511915 0CATLIN F  1 972773   111                    1                9990105
811  724225610584 81DEMBLING           ARTHUR GARY                               9990101
812  724225610584 8DEMBLI                   771259729625                      4  9990102
813  724225610584 8DEMBLI 000000                                                 9990103
814  724225610584 9DEMBLI F  1 972773   111       1                              9990105
815  724225712249 10UNSCOMBE           DAVID                                     9990101
816  724225712249 0DUNSCO                   493259729625                      1  9990102
817  724225712249 0DUNSCO 000802 11                         B        A           9990103
818  724225712249 0DUNSCO F  1 972773   111                    1                9990105
819  724225812084 91EDWARDS            PAUL MARTIN                               9990101
820  724225812084 9EDWARD                   535259729625                      1  9990102
821  724225812084 9EDWARD 000603 11B                                CD           9990103
822  724225812084 9EDWARD F  1 972773   111                    1                9990105
823  724225910264 61FEATHERSTONE-WITTYMARK PHILIP                                9990101
824  724225910264 6FEATHE                   500259729625                      10 9990102
825  724225910264 6FEATHE 000703 1 B         D                      B           9990103
826  724225910264 6FEATHE F  1 972773   111                    1                9990105
827  724226011335 11FOSTER             ALASTAIR JOHN                             9990101
828  724226011335 1FOSTER                   407259729625                      1  9990102
829  724226011335 1FOSTER 000603 1   E C     D                                   9990103
830  724226011335 1FOSTER F  1 972773   111                    1                9990105
831  724226111104 81GREENWELL          BRIAN                                     9990101
832  724226111104 8GREENW                   517259729625                      1  9990102
833  724226111104 8GREENW 000000                                                 9990103
834  724226111104 8GREENW F  1 972773   111                    1                9990105
835  724226212047 1HESLOP              JOHN RONALD                               9990101
836  724226212047 HESLOP                    306259729625                      10 9990102
837  724226212047 HESLOP 000802 11D                         A                    9990103
838  724226212047 HESLOP F  1 972773   111                    1                9990105
```

```
839     7242263127047LLEWELLYN          DAVID GWYN                                  9990101
840     7242263127047LLEWEL                   511259729625                     10   9990102
841     7242263127047LLEWEL C0C603 11                                   CCB         9990103
842     7242263127047LLEWEL F  1 972773   111                         1             9990105
843     7242264102550MCNAMARA           IAN EDWARD                                  9990101
844     7242264102550MCNAMA                   462259729625                      1   9990102
845     7242264102550MCNAMA C01003 11  C B   D                                      9990103
846     7242264102550MCNAMA F  1 972773   111                         1             9990105
847     7242265109845MOORE              ALAN PETER                                  9990101
848     7242265109845MOORE                    415259729625                     11   9990102
849     7242265109845MOORE  C00803 11C                B                   E         9990103
850     7242265109845MOORE  F  1 972773   111                         1             9990105
851     7242266106850OGDEN              GRAEME MACLEOD                              9990101
852     7242266106850OGDEN                    518259729625                      1   9990102
853     7242266106850OGDEN  C0C602 11                            ER                 9990103
854     7242266106850OGDEN  F  1 972773   111                         1             9990105
855     7242267125-461PARKER            DAVID GUNSON                                9990101
856     7242267125-46PARKER                   411259729625                     18   9990102
857     7242267125-46PARKER C00802 11          D     C                              9990103
858     7242267125-46PARKER F  1 972773   111                         1             9990105
859     7242268129034SMITH              RONALD GEORGE                               9990101
860     7242268129034SMITH                    535259729625                     16   9990102
861     7242268129034SMITH  C00502 11P           P                                  9990103
862     7242268129034SMITH  F  1 972773   111                         1             9990105
863     7242269109839TAYLOR             ALAN FREDERICK                              9990101
864     7242269109839TAYLOR                   535259729625                     19   9990102
865     7242269109839TAYLOR C0C300  1                                               9990103
866     7242269109839TAYLOR F  1 972773   111                                       9990105
867     7242270121449WHATMORE           JAMES CHARLES                               9990101
868     7242270121449WHATMO                   315259729625                      1   9990102
869     7242270121449WHATMO C00803 11B   B  C                                        9990103
870     7242270121449WHATMO F  1 972773   111                         1             9990105
871     7242271126850YOUNG              FREDERICK DAVID                             9990101
872     7242271126850YOUNG                    437259729625                      1   9990102
873     7242271126850YOUNG  C11002 11E           D                                  9990103
874     7242271126850YOUNG  F  1 972773   111                         1             9990105
875     7242272119846 PEARCEY           PETER FRANK                                 9990101
876     7242272119846PEARCE                   427259729625                     10   9990102
877     7242272119846PEARCE 000802 1 D          E                                   9990103
878     7242272119846PEARCE F  1 972775   111      1  C61                           9990105
END OF FILE
```

DURHAM UNIVERSITY
SCIENCE
- 9 JAN
SECTION
LIB