

UR1987_27

1987/27. CARP - Computer assisted retrieval of plans
(Revision 2)

R. G. Richardson

Abstract

The program suite described is used for adding entries to an index of plans, changing entries, sorting the index, searching the index, and producing plan labels. The data are entered from the plan register and most of the details entered may be searched.

INTRODUCTION

The Department of Mines Cartographic Drafting Room holds several thousand plans. The plans are listed numerically in the plan register, and by quadrangle in a card index. Selecting all plans relating to a particular subject, especially where a number of quadrangles may be involved, is extremely difficult and time consuming. CARP provides a convenient method of searching the plan index.

The programs used for CARP are written in FORTRAN 77 and are run on the Geological Survey Perkin-Elmer mini-computer. It is assumed that the user is familiar with the operating system and the standard editing facilities. Revision 2 incorporates changes to simplify multiple searches.

USING THE PROGRAMS

Data base maintenance

Data are input by typing PLANADD. The program requests the following:

- (i) PLAN - the plan number (up to 4 digits and two letters)
- to finish input type END here.
- (ii) TITLE - the plan title (up to 80 characters).
- (iii) LOCALITY - the locality covered by the plan (up to 16 characters).
- (iv) QUADRANGLE NUMBER - from one to four quadrangle numbers separated by commas
- e.g. 43, 44, 55, 56
- plans covering more than four quadrangles should be given the number 00.
- (v) CATEGORY - one or two characters specifying one of the standard categories in CATEGORY.DAT. This item is checked by the program for validity.
- (vi) AUTHOR - the initials of the author of the plan (up to 4 characters). This item is checked by the program against a standard list in AUTHOR.DAT.
- (vii) YEAR - the year of drafting (4 digits).
- (viii) DRAFTSMAN - the initials of the drafter of the plan (up to 4 characters). This item is checked by the program against a standard list in DRAFTS.MEN.

- (ix) REMARKS - up to 17 characters of remarks.
- (x) MASTER FILM NUMBER - up to 3 digits specifying the master film number. A prefix of MIDO/GS/GP/ is presumed.
- (xi) PLAN SIZE - two characters describing the plan size.

At the end of input, a proof-sheet is output and the data saved in file PLANTEMP.TMP. Corrections may be made to this file using the standard editing facilities and a new proof-sheet printed and checked.

The new data in PLANTEMP.TMP must be added to the end of the existing data base by typing PLANMIX. To ensure that the file remains in plan number order, it should be sorted periodically using the PLANSORT command.

Data may be changed by typing PLANCHGE. The program requests:

- (i) PLAN NUMBER - the number of the plan for which alterations are required.
- to finish input type END here.
- (ii) A series of opportunities to change particular data values then follow. The plan number cannot be changed.

Film numbers for plans on one master film may be added by typing FILMADD. The program requests:

- (i) MASTER FILM NUMBER - the three digit master film number.
- (ii) PLAN NUMBER - the plan numbers on the master film.
- to finish input type END here.

Data base searching

The search phase is entered by typing PLANSRCH. An entry will be retrieved only if the specified search options are matched. If a device name is specified after PLANSRCH output is sent to that device.

The program requests the following data:

- (i) FIRST TITLE SELECTION - one or more groups of words on a single line, each group of words terminated by /. If this option is not required leave the line blank.
e.g. GOLDEN GATE/TASMANIA MINE/LEFROY/
Golden Gate or Tasmania Mine or Lefroy.
- (ii) SECOND TITLE SELECTION - as for the first title selection. Both title selections must be satisfied during a search if further items are to be checked, and this provides a 'logical and' for the title search. For example it is possible to look for surface plans from either the Golden Gate, Tasmania Mine or Lefroy by making the first title selection.

GOLDEN GATE/TASMANIA MINE/LEFROY/

and the second title selection

SURFACE PLAN/

- (iii) LOCATION - details as for the first title selection.
- (iv) QUADRANGLE - one or more quadrangle numbers on a single line, each terminated by a /. Leave blank if not required.
To search regional plans use a quadrangle number of 00.
e.g. 27/ only quadrangle 27
 27/32/ quadrangle 27 or quadrangle 32
 27/32/00 quadrangle 27, quadrangle 32, or any regional plan.
- (v) CATEGORY - one or more single character standard category numbers on a single line, each terminated by a /. Leave blank if not required.
- (vi) SUB-CATEGORY - one or more single character standard sub-category numbers on a single line, each terminated by a /. Leave blank if not required.
- (vii) AUTHOR - one or more standard author abbreviations arranged in the same manner as the first title selection.
- (viii) YEAR - details as for quadrangle.
- (ix) DRAFTSMAN - one or more standard drafter abbreviations arranged in the same manner as the first title selection.
- (x) Screen listing required - Y if required, N otherwise. At the end of each screen a choice is made as to whether to list more on the screen or not.
- (xi) Printout required - Y if required, N otherwise.
- (xii) Further searches - Y if required, N otherwise.

Data base listing

To produce a listing type PLANLIST. A list of the entire data base may be produced, or listed from any particular plan number. The data base should be sorted before listing. If a device name is specified after PLANLIST the listing is sent to that device.

The program requests the following data:

STARTING PLAN NUMBER - either the plan number to start listing from or a blank to list the entire data base.

ENDING PLAN NUMBER - either the last plan number to be listed or a blank to list the remainder of the data base.

Label printing

To produce labels for the aperture cards from one or more master films the command PLANLABL may be used.

The program requests the following data:

ROLL NUMBER - the roll number for which labels are to be printed. A zero or blank will terminate the run.

Commands are also required when printing the labels.

THE PROGRAMS

PLANADD (Appendix 1)

This program accepts data from the keyboard and copies it to a file in a format suitable for editing. Prompts are used to guide the user. The file created (PLANTEMP.TMP) may be edited using the standard editing facilities. PLANADD uses three files of standard abbreviations - CATEGORY.DAT, AUTHOR.DAT and DRAFTS.MEN. Appendix 2 shows an example of CATEGORY.DAT and it should be noted that any line with the first 4 columns blank is ignored by the program and can be used for comments. The same is true of AUTHOR.DAT and DRAFTS.MEN.

PLANMIX (Appendix 3)

The data from PLANTEMP.TMP are read and the basic data converted to the format of the data base and put in file PLANMAIN.DAT. Accidental data loss is prevented by copying to intermediate files.

PLANSORT (Appendix 4)

The data are sorted into ascending plan number order using a temporary work file.

PLANCHGE (APPENDIX 5)

This pair of programs reads data from PLANMAIN.DAT and allows the user to change the data. The temporary work file is then merged with the main file.

PLANSRCH (Appendix 6)

The data base is searched for the occurrence of specified strings or numeric values. The output is an extract from the data held for each plan.

PLANLIST (Appendix 7)

For producing a complete list of all plan information.

PLANLABL (Appendix 8)

For producing labels for aperture cards.

FILMADD (Appendix 9)

For adding master film numbers to existing plan records. A temporary main file is used to avoid accidental data loss.

[22 May 1987]

APPENDIX 1

Program PLANADD

```
*PLANADD. CSS
*FOR RUNNING PLANADD AND PRINTING A PROOF SHEET
$IFX PLANTEMP.TMP: $WRITE ** CHECK LAST MERGE: $EXIT: $ENDC
PRE ETM: AL PLANTEMP.TMP,IN,132
* SET UP THE OUTPUT FILE
L PLANADD,3: AS 6,PLANTEMP.TMP: * LOAD PROG AND SET LU 6
ST
PRINT PLANTEMP.TMP
$WRITE FILE IS PLANTEMP.TMP: ENA ETM: $EXIT
```

```

#TITL  PLANADD.FTN  -  ADD PLANS TO THE PLAN INDEX
C TAKES KEYBOARD INPUT AND PREPARES A PROOF SHEET
  CHARACTER*6 PLAN
  CHARACTER*80 TITLE
  CHARACTER*4 AUTHOR, DRAFT, AUTREF(1000), DRAREF(1000)
  CHARACTER*2 CATG, PLNSIZ, CATREF(100)
  CHARACTER*16 LOCATN
  CHARACTER*11 QUAD
  CHARACTER*17 REMARK
  INTEGER FILM, YEAR
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C   QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C   MASTER FILM ROLL NUMBER.
C
  LOGICAL IER, CHECK
C
  OPEN(UNIT=5, FILE='CON:')
C SET THE CONSOLE UP AS LU 5 FOR INPUT
C THE OUTPUT IS TO LU 6 WHICH IS ASSIGNED IN THE COMMAND FILE
  OPEN(UNIT=8, FILE='AUTHOR.DAT', SHARE='ERO')
C CONTAINS LIST OF AUTHORS
  NAUT=1
  10 READ(8,202,END=11) AUTREF(NAUT)
  IF (AUTREF(NAUT) .NE. ' ') NAUT=NAUT+1
C SKIP RECORDS WITH BLANKS IN THE FIRST 4 COLUMNS WHICH ALLOWS
C LINES OF COMMENTS TO BE INSERTED
  IF (NAUT .GT. 1000) STOP 'TOO MANY AUTHORS'
  GOTO 10
  11 CLOSE(UNIT=8, STATUS='KEEP')
C
  OPEN(UNIT=8, FILE='DRAFTS.MEN', SHARE='ERO')
C CONTAINS LIST OF DRAFTSMEN
  NDRAFT=1
  15 READ(8,202,END=16) DRAREF(NDRAFT)
  IF (DRAREF(NDRAFT) .NE. ' ') NDRAFT=NDRAFT+1
C SKIP RECORDS WITH BLANKS IN FIRST 4 COLUMNS
  IF (NDRAFT .GT. 1000) STOP 'TOO MANY DRAFTSMEN'
  GOTO 15
  16 CLOSE(UNIT=8, STATUS='KEEP')
C
  OPEN(UNIT=8, FILE='CATEGORY.DAT', SHARE='ERO')
C CONTAINS LIST OF POSSIBLE CATEGORIES
  NCAT=1
  18 READ(8,104,END=19) CATREF(NCAT)
  IF (CATREF(NCAT) .NE. ' ') NCAT=NCAT+1
C SKIP RECORDS WITH BLANKS IN FIRST 2 COLUMNS
  IF (NCAT .GT. 100) STOP 'TOO MANY CATEGORIES'
  GOTO 18
  19 CLOSE(UNIT=8, STATUS='KEEP')
C
  50 WRITE(5,100)
  100 FORMAT(' PLAN OR END')
  READ(5,200) PLAN
  200 FORMAT(A6)
  IF (INDEX(PLAN,'END ') .NE. 0) GOTO 60
C RECOGNISE END BY HAVING END ON ITS OWN AND PRINT PROOF SHEET
  WRITE(5,101)
  101 FORMAT(' TITLE - FULL LINE')
  READ(5,201) TITLE

```

```

201 FORMAT(A80)
   WRITE(5,105)
105 FORMAT(' LOCALITY           J')
   READ(5,203) LOCATN
203 FORMAT(A16)
   WRITE(5,106)
106 FORMAT(' QUADRANGLE NO')
   READ(5,107) QUAD
107 FORMAT(A11)
   52 WRITE(5,103)
103 FORMAT(' CATEGORY')
   READ(5,104) CATG
104 FORMAT(A2)
   IF (CATG .NE. ' ' .AND. .NOT. CHECK(CATG,CATREF,NCAT)) GOTO 52
   51 WRITE(5,102)
102 FORMAT(' AUTHOR')
   READ(5,202) AUTHOR
202 FORMAT(A4)
   IF (AUTHOR .NE. ' ' .AND. .NOT. CHECK(AUTHOR,AUTREF,NAUT))
       GOTO 51
   WRITE(5,205)
205 FORMAT(' YEAR')
   READ(5,*) YEAR
   IF (YEAR .LT. 100) YEAR=YEAR+1900
C CONVERT TO 20 TH CENTURY IF NEEDED
   54 WRITE(5,110)
110 FORMAT(' DRAFTSMAN')
   READ(5,111) DRAFT
111 FORMAT(A4)
   IF (DRAFT .NE. ' ' .AND. .NOT. CHECK(DRAFT,DRAREF,NDRAFT))
       GOTO 54
   WRITE(5,112)
112 FORMAT(' REMARKS           J')
   READ(5,113) REMARK
113 FORMAT(A17)
   WRITE(5,114)
114 FORMAT(' MASTER FILM ROLL'// ASSUMES PREFIX OF MIDO/GS/GP//)
   CALL INFREE(FILM,IER,5)
   WRITE(5,108)
108 FORMAT(' PLAN SIZE')
   READ(5,109) PLNSIZ
109 FORMAT(A2)
C READ ONE COMPLETE DATA SHEET
C
C GO WRITE TO OUTPUT FILE
   WRITE(6,300) PLAN,TITLE,LOCATN,QUAD,CATG,AUTHOR,YEAR,DRAFT,
       REMARK,FILM,PLNSIZ
300 FORMAT(A6,'@',A80,'@',A16,'@',A11/7X,A2,'@',A4,'@',I4,
       '@',A4,'@',A17,'@',I3,'@',A2)
C WRITE WITH DELIMITERS AFTER EACH FIELD
   GOTO 50
C BACK AROUND
C
C OTHERWISE CLOSE UP AND STOP
   60 CONTINUE
   CLOSE(UNIT=6,STATUS='KEEP')
   END
   LOGICAL FUNCTION CHECK(VAR,VARREF,NVAR)
   CHARACTER *(*) VAR,VARREF(2)
   DO 10 I=1,NVAR

```

```
      IF (VAR .NE. VARREF(I)) GOTO 10  
      CHECK=.TRUE.  
      RETURN  
10  CONTINUE  
C IF FALL THROUGH HAVENT FOUND A MATCH  
  CHECK=.FALSE.  
  RETURN  
  END  
#INCLUDE INFREE.FTN, -
```

```

      SUBROUTINE INFREE(INUM, IER, ILUN)
C A FREE FORMAT INPUT ROUTINE FOR READING ONE INTEGER*4 NUMBER FROM
C ILUN.
C INUM, ILUN ARE BOTH INTEGER*4
C IER IS A LOGICAL
C MAKES USE OF ENCODE TO GO FROM FREE FORMAT TO
C FIXED FORMAT BY WRITING IN A FORM SUITABLE FOR RE-READING
C IN I6 FORMAT.
      INTEGER*2 A(10), BLA, I, J, K, JUP, L
      REAL BUFF(5)
      LOGICAL IER
      DATA BLA/1H /
C USED TO DETERMINE NON-BLANK CHARACTERS
      IER=.TRUE.
      READ(ILUN,100,ERR=101) A
      100 FORMAT(10A1)
C READ IN A CHARACTER FORMAT
      DO 10 I=1,10
      IF (A(I) .NE. BLA) GOTO 20
C LOOK FOR FIRST NON-BLANK CHARACTER
      10 CONTINUE
C IF FALL OUT OF LOOP ALL CHARACTERS ARE BLANK - HENCE INUM=0
      IER=.FALSE.
      INUM=0
      101 RETURN
C THE QUICK RETURN
C
C NOW COPY THE FREE FORMAT INTEGER RIGHT JUSTIFIED INTO THE SAME
C ARRAY
      20 CONTINUE
      DO 11 J=I,10
      IF (A(J) .EQ. BLA) GOTO 21
      11 CONTINUE
C FIND THE END OF THE NUMBER, EITHER BY FINDING A BLANK AT THE END
C OR BY REACHING THE TENTH CHARACTER
      J=11
      21 J=J-1
      K=10
      JUP=J
C NOW COPY RIGHT JUSTIFIED AND FILL LEADING PART WITH BLANKS IF NEEDS BE
      DO 12 L=I, J
      A(K)=A(JUP)
      K=K-1
      JUP=JUP-1
      12 CONTINUE
      IF (K .EQ. 0) GOTO 15
      DO 14 L=1, K
      A(L)=BLA
      14 CONTINUE
C NOW FILLED WITH LEADING BLANKS AND RIGHT JUSTIFIED
C
C SO
C WRITE OUT TO CORE BUFFER AND READ BACK IN I6 FORMAT
      15 CONTINUE
      ENCODE(BUFF,100) A
      DECODE(BUFF,111) INUM
      111 FORMAT(I10)
C NOW HAVE THE NUMBER
      IER=.FALSE.
      RETURN
      END

```

APPENDIX 2

Example of standard abbreviations, CATEGORY.DAT

1 ENGINEERING GEOLOGY

- 10 GROUND WATER
- 11 DAMSITES
- 12 LANDSLIDES
- 13 ROAD, RAIL AND PIPELINES
- 14 FOUNDATIONS
- 15 QUARRIES AND ROCK FACES
- 16 GEOPHYSICS
- 19 MISCELLANEOUS

2 ECONOMIC GEOLOGY

- 20 GEOLOGICAL MAP SECTIONS
- 21 MINE PLAN SECTIONS
- 22 METALLOGENIC
- 23 FUELS
- 24 INDUSTRIAL MINERALS
- 25 PETROLOGICAL, GEOCHEMICAL
- 26 PRODUCTION FIGURES
- 27 STRUCTURAL DIAGRAMS
- 29 MISCELLANEOUS

3 REGIONAL GEOLOGY

- 30 REGIONAL GENERAL
- 31 STRUCTURAL
- 32 SECTIONS
- 33 COLUMNS
- 39 MISCELLANEOUS

4 GEOPHYSICS

- 40 SEISMIC
- 41 GRAVITY
- 42 MAGNETICS
- 43 AEROMAGNETICS
- 44 RESISTIVITY
- 49 MISCELLANEOUS

5 GEOCHEMICAL

- 5 GEOCHEMICAL

6 PALAEOONTOLOGICAL

- 6 PALAEOONTOLOGICAL

7 PETROLOGICAL/MINERALOGICAL

- 70 MINERAL DATA - COMPOSITIONS
- 71 TRIANGULAR DIAGRAMS
- 72 ISOTOPE DATA
- 73 PETROLOGY DIAGRAMS
- 79 MISCELLANEOUS

8 MISCELLANEOUS

- 8 MISCELLANEOUS

APPENDIX 3

Program PLANMIX

```

*PLANMIX.CSS
* FOR MERGING ADDITIONS FILE WITH MAIN FILE
PRE ETM: XDE SYSF:PLANMRG1.TMP: *DELETE SCRATCH FILE FOR MAIN ENTRY
AL SYSF:PLANMRG1.TMP,IN,146: *AL SCRATCH FILE
L PLANMIX,10: * LOAD PROGRAM
REP PLANTEMP.TMP,FF00: AS 6,PLANTEMP.TMP,ERO: * THE TEMPORARY FILE
AS 4,PLANMAIN.DAT,ERO: * THE MAIN FILE
AS 3,SYSF:PLANMRG1.TMP: * TEMP FILE ON SYSF
ST
$IFNE 0: $WRITE TRANSLATE ERROR: ENA ETM: $EXIT: $ENDC
REP SYSF:PLANMRG1.TMP,FF00: *PROTECT TEMP FILE
REP PLANMAIN.DAT,0
DE PLANMAIN.DAT
$BUILD PLANCOPY.CMD
IN SYSF:PLANMRG1.TMP
AL PLANMAIN.DAT,IN,146/8/3
OUT PLANMAIN.DAT
COPY *,*
REW I
REW 0
VERIFY *,*
END
$ENDB
L COPY32,50: ST ,COM=PLANCOPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0: $WRITE COPY BACK FAILED: ENA ETM: $EXIT: $ENDC
REP PLANMAIN.DAT,FF00: * PROTECT NEW MAIN FILE
REP SYSF:PLANMRG1.TMP,0: REP PLANTEMP.TMP,0
DE PLANCOPY.CMD,SYSF:PLANMRG1.TMP,PLANTEMP.TMP
ENA ETM
$EXIT

```

```

*TTITLE PLANMIX.FTN - APPEND PROOF FILE TO MASTER FILES
CHARACTER*6 PLAN,TPLAN
CHARACTER*80 TITLE
CHARACTER*4 AUTHOR,DRAFT
CHARACTER*2 CATG,PLNSIZ
CHARACTER*16 LOCATN
CHARACTER*11 QUAD
CHARACTER*17 REMARK
INTEGER*2 FILM, YEAR
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C MASTER FILM ROLL NUMBER.
CHARACTER*1 BUFF(146)
EQUIVALENCE (BUFF(1),TITLE), (BUFF(81),REMARK), (BUFF(98),LOCATN),
(BUFF(114),QUAD), (BUFF(125),PLAN), (BUFF(131),AUTHOR),
(BUFF(135),DRAFT), (BUFF(139),CATG), (BUFF(141),PLNSIZ),
(BUFF(143),YEAR), (BUFF(145),FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
REWIND 6
C REWIND PLANTEMP.TMP - THE INPUT FILE
OPEN(UNIT=4,FORM='BINARY',RECL=146,SHARE='ERO')
C OPEN MAIN FILE INPUT ONLY
OPEN(UNIT=3,FORM='BINARY',RECL=146)
C OPEN FILE TO COPY TO
10 READ(4,END=11) BUFF
WRITE(3) BUFF
GOTO 10
C COPY ONE AT A TIME TO THE END
C
11 CONTINUE
C COPIED MAIN FILE
CLOSE(UNIT=4,STATUS='KEEP')
C SO CLOSE IT
C
C NOW TO START ADDING NEW ENTRIES
20 READ(6,300,END=25) PLAN, TITLE, LOCATN, QUAD, CATG, AUTHOR, YEAR, DRAFT,
REMARK, FILM, PLNSIZ
300 FORMAT(A6,1X,A80,1X,A16,1X,A11/7X,A2,1X,A4,1X,I4,
1X,A4,1X,A17,1X,I3,1X,A2)
C WE WANT TO GET THE PLAN NUMBER IN THE FORMAT I4,A2
C AT THE MOMENT IT COULD BE ANYTHING
I=CTOI(PLAN,K)
C K POINTS TO THE LAST NUMERIC DIGIT IN THE FIRST NUMERIC STRING OF
C PLAN. I.E. 102/3 WILL HAVE K=3
TPLAN(5:6)=PLAN(K+1:K+2)
C THATS THE A2 PART DONE
J=4
IK=K
C WILL NOW WORK BACKWARDS THROUGH TPLAN AND PLAN
DO 30 I=1,K
TPLAN(J:J)=PLAN(IK:IK)
J=J-1
IK=IK-1
30 CONTINUE
IF (J.NE. 0) THEN
C HAVENT FILLED RIGHT TO THE FRONT OF TPLAN SO WILL NEED TO FILL
C WITH BLANKS
DO 35 I=1,J

```

```
      TPLAN(I:I)=' '
35  CONTINUE
      ENDIF
      PLAN=TPLAN
      DO 38 I=1,6
      IF (PLAN(I:I) .GT. '0') GOTO 39
      PLAN(I:I)=' '
C  REPLACE LEADING 0 S BY SPACES
      38  CONTINUE
      39  CONTINUE
          WRITE(3) BUFF
C  WRITE THE RECORD
          GOTO 20
C  AND BACK ROUND
C
C  AT END OF FILE
      25  CLOSE(UNIT=6, STATUS='KEEP')
C  ALL NOW AMALGAMATED WITH MASTER FILE IN TEMPORARY FILE
      END
```

APPENDIX 4

Program PLANSORT

```
*PLANSORT.CSS
* FOR SORTING PLAN FILE INTO PLAN NUMBER ORDER
PRE ETM
L PLANSORT,10: AS 4,PLANMAIN.DAT,ERO: AL SYSF:PLTEMP.DAT,IN,146/5/2
AS 6,SYSF:PLTEMP.DAT: ST
$IFNE 0: $WRITE SORT ERROR: ENA ETM: $EXIT: $ENDC
$BUILD WCOPY.CMD
IN SYSF:PLTEMP.DAT
AL PLANMAIN.DAT,IN,146/10/5
OUT PLANMAIN.DAT
COPY *,*
REW I
REW 0
VERIFY *,*
END
$ENDB
REP SYSF:PLTEMP.DAT,FF00: REP PLANMAIN.DAT,0: DE PLANMAIN.DAT
L COPY32,50: ST ,COM=WCOPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0: $WRITE SORT-COPY ERROR: ENA ETM: $EXIT: $ENDC
REP PLANMAIN.DAT,FF00: REP SYSF:PLTEMP.DAT,0: DE SYSF:PLTEMP.DAT,WCOPY.CMD
ENA ETM: $EXIT
```

```

$TITL  PLANSORT.FTN - SORT THE PLAN FILE INTO ORDER
C
    CHARACTER*6 PLAN,TPLAN(10000)
    CHARACTER*80 TITLE
    CHARACTER*4 AUTHOR,DRAFT
    CHARACTER*2 CATG,PLNSIZ
    CHARACTER*16 LOCATN
    CHARACTER*11 QUAD
    CHARACTER*17 REMARK
    INTEGER*2 FILM, YEAR, INDEX(10000)
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C     QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C     MASTER FILM ROLL NUMBER.
    CHARACTER*1 BUFF(146)
    EQUIVALENCE (BUFF(1),TITLE), (BUFF(81),REMARK), (BUFF(98),LOCATN),
    (BUFF(114),QUAD), (BUFF(125),PLAN), (BUFF(131),AUTHOR),
    (BUFF(135),DRAFT), (BUFF(139),CATG), (BUFF(141),PLNSIZ),
    (BUFF(143),YEAR), (BUFF(145),FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
    OPEN(UNIT=4,FORM='BINARY',ACCESS='DIRECT',RECL=146)
C OPEN THE INPUT FILE FOR RANDOM AND SEQUENTIAL ACCESS
    REWIND 4
    IREC=0
C COUNTER FOR NUMBER OF RECORDS
    10 READ(4,END=20) BUFF
    IREC=IREC+1
    IF (IREC .GT. 10000) STOP 'MORE THAN 10000 ENTRIES'
    INDEX(IREC)=IREC
C FILL SORT INDEX
    TPLAN(IREC)=PLAN
C COMBINE CARD AND QUAD TO GET CHRONOLOGICAL NUMER
    GOTO 10
C BACK AROUND
C
C NOW FOR THE WORK
    20 CONTINUE
    IF (IREC .EQ. 0) STOP 'NO RECORDS'
    CALL BBSSTR(INDEX,TPLAN,1,IREC)
C SORT INTO ASCENDING CHARACTER ORDER AS STRINGS - I. E. PLAN NO ORDER
    OPEN(UNIT=6,FORM='BINARY',RECL=146)
C OPEN OUTPUT FILE
    DO 30 I=1,IREC
    READ(4,REC=INDEX(I)) BUFF
    WRITE(6) BUFF
C COPY TO NEW ORDER
    30 CONTINUE
    CLOSE(UNIT=6,STATUS='KEEP')
    CLOSE(UNIT=4,STATUS='KEEP')
    END
    SUBROUTINE BBSSTR(IR,CVAR,IBASE,N)
    CHARACTER*(*) CVAR(N)
    INTEGER*2 IR(N)
    LOGICAL NSWAP
    IF (N .LE. 1) RETURN
C NOTHING TO SORT
    NM1=N-1
    DO 30 J=IBASE,NM1
    NSWAP=.TRUE.

```

```
IRI=IR(1)
DO 40 I=IBASE,NM1
  IP1=I+1
  IRIP1=IR(IP1)
  IF (CVAR(IRI) .LE. CVAR(IRIP1)) GOTO 40
  NSWAP=.FALSE.
  IR(I)=IRIP1
  IR(IP1)=IRI
  IRIP1=IRI
40 IRI=IRIP1
  IF (NSWAP) RETURN
30 CONTINUE
  RETURN
END
```

APPENDIX 5

Program PLANCHGE

```

*PLANCHGE. CSS
* FOR CORRECTING MAIN FILE
PRE ETM: XDE SYSF: PLANCOR1. TMP: *DELETE SCRATCH FILE FOR MAIN ENTRY
AL SYSF: PLANCOR1. TMP, IN, 146: *AL SCRATCH FILE
AL PLANCTMP. TMP, IN, 148
L PLANCHGE, 10: AS 4, PLANMAIN. DAT, ERO
AS 6, PLANCTMP. TMP: AS 3, PR: ST
L PLANCHG2, 10: * LOAD PROGRAM
REP PLANCTMP. TMP, FF00: AS 6, PLANCTMP. TMP, ERO: * THE TEMPORARY FILE
AS 4, PLANMAIN. DAT, ERO: * THE MAIN FILES
AS 3, SYSF: PLANCOR1. TMP: * TEMP FILE ON SYSF
ST
$IFNE 0: $WRITE TRANSLATE ERROR: ENA ETM: $EXIT: $ENDC
REP SYSF: PLANCOR1. TMP, FF00: *PROTECT TEMP FILES
REP PLANMAIN. DAT, 0
DE PLANMAIN. DAT
$BUILD PLANCOPY. CMD
IN SYSF: PLANCOR1. TMP
AL PLANMAIN. DAT, IN, 146/8/3
OUT PLANMAIN. DAT
COPY *,*
REN I
REN 0
VERIFY *,*
END
$ENDB
L COPY32, 50: ST , COM=PLANCOPY. CMD, LI=NULL: , LO=NULL:
$IFNE 0: $WRITE COPY BACK FAILED: ENA ETM: $EXIT: $ENDC
REP PLANMAIN. DAT, FF00: * PROTECT NEW MAIN FILE
REP SYSF: PLANCOR1. TMP, 0: REP PLANCTMP. TMP, 0
DE PLANCOPY. CMD, SYSF: PLANCOR1. TMP, PLANCTMP. TMP
ENA ETM
$EXIT

```

```

#TITL PLANCHGE.FTN - CHANGE DETAILS OF PLAN ENTRIES
      CHARACTER*6 PLAN, APLAN
      LOGICAL CHANGE, CHGE, START
      CHARACTER*80 TITLE
      CHARACTER*4 AUTHOR, DRAFT
      CHARACTER*2 CATG, PLNSIZ
      CHARACTER*16 LOCATN
      CHARACTER*11 QUAD
      CHARACTER*17 REMARK
      INTEGER*2 FILM, YEAR, IBASE
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C     QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C     MASTER FILM ROLL NUMBER.
      CHARACTER*1 BUFF(146), BUFF2(148)
      EQUIVALENCE (BUFF(1), TITLE), (BUFF(81), REMARK), (BUFF(98), LOCATN),
        (BUFF(114), QUAD), (BUFF(125), PLAN), (BUFF(131), AUTHOR),
        (BUFF(135), DRAFT), (BUFF(139), CATG), (BUFF(141), PLNSIZ),
        (BUFF(143), YEAR), (BUFF(145), FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
      EQUIVALENCE (BUFF2(1), BUFF(1)), (BUFF2(147), IBASE)
C USED TO WRITE TO TEMP OUTPUT FILE TO MAKE LIFE EASIER
C
      CHGE=.FALSE.
C USED TO DECIDE WHEN THE OUTPUT LISTING NEEDS HEADING
      OPEN(UNIT=5, FILE='CON:')
C SET CONSOLE AS LU 5
      OPEN(UNIT=4, FORM='BINARY', RECL=146, SHARE='ERO')
C OPEN WATMAIN.DAT
      OPEN(UNIT=6, FORM='BINARY', RECL=148)
C TEMP FILE FOR CHANGES
C
      REWIND 4
      START=.TRUE.
      IBASE=0
C REWIND FILE AND SET INITIAL PARAMETERS FOR RECORD NUMBER
C AND NUMBER OF TIMES THROUGH FILE ON A SEARCH
      10 WRITE(5,100)
      100 FORMAT(' PLAN NUMBER - END TERMINATES')
      READ(5,101) APLAN
      101 FORMAT(A6)
      IF (APLAN .EQ. 'END ') GOTO 20
C END OF RUN
C NOW FIND THE EXACT LENGTH OF THE PLAN NUMBER
      DO 8 I=1,6
      IF (APLAN(I:I) .GT. '0') GOTO 9
      APLAN(I:I)=' '
C REPLACE LEADING ZEROES BY SPACES
      8 CONTINUE
      9 CONTINUE
      J=6
      DO 12 I=1,6
      IF (APLAN(J:J) .NE. ' ') GOTO 13
      J=J-1
      12 CONTINUE
      GOTO 10
C ALL BLANKS
C THE FIRST NON-BLANK CHARACTER STARTING FROM THE RIGHT
C IS THE END OF THE PLAN NUMBER

```

```

13 CONTINUE
11 CONTINUE
  READ(4,END=15) BUFF
C IF WE HIT THE END MUST HAVE HAD A WRONG PLAN NUMBER
  IBASE=IBASE+1
C COUNTING THROUGH PLANMAIN
  IF (INDEX(PLAN,APLAN(1:J)) .EQ. 0) GOTO 11
C NOT THE ONE WE NEED SO READ A BIT MORE
  GOTO 16
15 CONTINUE
C TO HERE AT THE END OF FILE AND CHECK TO SEE IF STARTED READING
C PART WAY THROUGH - THIS IS SO IF START IS FALSE
  IBASE=0
C RESET THE COUNTER FOR THE POSITION IN PLANMAIN WHICH IS KEPT
C FOR LATER ACCESSES.
  REWIND 4
  IF (START) GOTO 10
C HAVE BEEN THROUGH COMPLETELY
C SO ASK FOR ANOTHER PLAN NUMBER
  START=.TRUE.
C GO THROUGH FOR ANOTHER LOOK
  GOTO 11
16 WRITE(5,102) PLAN, TITLE(1:65), TITLE(66:80), AUTHOR, LOCATN
102 FORMAT(1X,A6,5X,A65/12X,A15,A4,4X,A16// OK (Y OR N)?')
C WAS IT THE ONE WANTED
  READ(5,200) AOK
200 FORMAT(A1)
  IF (AOK .NE. 'Y') GOTO 11
C IT WASNT SO READ SOME MORE
  START=.FALSE.
C HAVE FOUND SOMETHING SO ARE PART WAY THROUGH FILE
C EVEN IF IT IS NOT THE ONE WANTED MAY STILL NEED TO
C SEARCH ENTIRE FILE
C
C NOW START TO LOOK AT CHANGES
C
C NOW ASK FOR CHANGES
  IF (CHANGE('TITLE')) CALL ALPHA1(TITLE,'TITLE')
  IF (CHANGE('LOCALITY')) CALL ALPHA2(LOCATN,'LOCALITY')
  IF (CHANGE('QUADRANGLE')) CALL ALPHA3(QUAD,'QUADRANGLE')
  IF (CHANGE('CATEGORY')) CALL ALPHA4(CATG,'CATEGORY')
  IF (CHANGE('AUTHOR')) CALL ALPHA5(AUTHOR,'AUTHOR')
  IF (CHANGE('YEAR')) CALL INTG2(YEAR,'YEAR')
  IF (YEAR .LT. 100) YEAR=YEAR+1900
C CONVERT BACK TO THE 20TH CENTURY
  IF (CHANGE('DRAFTSMAN')) CALL ALPHA5(DRAFT,'DRAFTSMAN')
  IF (CHANGE('REMARKS')) CALL ALPHA6(REMARK,'REMARKS')
  IF (CHANGE('ROLL NO. ')) CALL INTG2(FILM,'ROLL NO. ')
  IF (CHANGE('PLAN SIZE')) CALL ALPHA4(PLNSIZ,'PLAN SIZE')
  WRITE(6) BUFF2
  IF (.NOT. CHGE) THEN
    WRITE(3,299)
299 FORMAT(' THE FOLLOWING CHANGES ARE BEING MADE//')
    CHGE=.TRUE.
  ENDIF
  WRITE(3,300) PLAN, TITLE, LOCATN, QUAD, CATG, AUTHOR, YEAR, DRAFT,
    REMARK, FILM, PLNSIZ
300 FORMAT(1X,A6,'@',A80,'@',A16,'@',A11/7X,A2,'@',A4,'@',I4,
  '@',A4,'@',A17,'@',I3,'@',A2)
C WRITE WITH DELIMITERS AFTER EACH FIELD

```

```

      GOTO 10
C BACK ROUND
C
C NOW TO TIDY UP
  20 CONTINUE
    IF (CHGE) WRITE(3,302)
  302 FORMAT(1X/)
    CLOSE(UNIT=4, STATUS='KEEP')
    CLOSE(UNIT=6, STATUS='KEEP')
    STOP
    END
    LOGICAL FUNCTION CHANGE(STRING)
    CHARACTER* (*) STRING
    WRITE(5,100) STRING
  100 FORMAT(' CHANGE ',A20// ' Y OR N?')
    READ(5,102) AOK
  102 FORMAT(A1)
    CHANGE=AOK .EQ. 'Y'
    RETURN
    END
    SUBROUTINE ALPHA1(SVAR,PROMPT)
    CHARACTER* (*) PROMPT
    CHARACTER*80 SVAR
    WRITE(5,100) PROMPT
  100 FORMAT(1X,'ENTER ',A20)
    READ(5,101) SVAR
  101 FORMAT(A80)
    RETURN
    END
    SUBROUTINE INTG2(IVAR,PROMPT)
    CHARACTER* (*) PROMPT
    INTEGER*2 IVAR
    WRITE(5,100) PROMPT
  100 FORMAT(1X,A20)
    READ(5,*) IVAR
    RETURN
    END
    SUBROUTINE ALPHA3(SVAR,PROMPT)
    CHARACTER* (*) PROMPT
    CHARACTER*11 SVAR
    WRITE(5,100) PROMPT
  100 FORMAT(1X,'ENTER ',A20)
    READ(5,101) SVAR
  101 FORMAT(A11)
    RETURN
    END
    SUBROUTINE ALPHA2(SVAR,PROMPT)
    CHARACTER* (*) PROMPT
    CHARACTER*16 SVAR
    WRITE(5,100) PROMPT
  100 FORMAT(1X,'ENTER ',A20)
    READ(5,101) SVAR
  101 FORMAT(A16)
    RETURN
    END
    SUBROUTINE ALPHA4(SVAR,PROMPT)
    CHARACTER* (*) PROMPT
    CHARACTER*2 SVAR
    WRITE(5,100) PROMPT
  100 FORMAT(1X,'ENTER ',A20)

```

```
      READ(5,101) SVAR
101  FORMAT(A2)
      RETURN
      END
      SUBROUTINE ALPHAS(SVAR,PROMPT)
      CHARACTER* (*) PROMPT
      CHARACTER*4 SVAR
      WRITE(5,100) PROMPT
100  FORMAT(1X,'ENTER ',A20)
      READ(5,101) SVAR
101  FORMAT(A4)
      RETURN
      END
      SUBROUTINE ALPHA6(SVAR,PROMPT)
      CHARACTER* (*) PROMPT
      CHARACTER*17 SVAR
      WRITE(5,100) PROMPT
100  FORMAT(1X,'ENTER ',A20)
      READ(5,101) SVAR
101  FORMAT(A17)
      RETURN
      END
```

```

$TITL PLANCHG2.FTN - PUT CHANGES INTO SEQUENCE IN THE MAIN FILE
CHARACTER*6 PLAN
CHARACTER*80 TITLE
CHARACTER*4 AUTHOR, DRAFT
CHARACTER*2 CATG, PLNSIZ
CHARACTER*16 LOCATN
CHARACTER*11 QUAD
CHARACTER*17 REMARK
INTEGER*2 FILM, YEAR, IBASE
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C MASTER FILM ROLL NUMBER.
CHARACTER*1 BUFF(146), BUFF2(148)
EQUIVALENCE (BUFF(1), TITLE), (BUFF(81), REMARK), (BUFF(98), LOCATN),
  (BUFF(114), QUAD), (BUFF(125), PLAN), (BUFF(131), AUTHOR),
  (BUFF(135), DRAFT), (BUFF(139), CATG), (BUFF(141), PLNSIZ),
  (BUFF(143), YEAR), (BUFF(145), FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C EQUIVALENCE (BUFF2(1), BUFF(1)), (BUFF2(147), IBASE)
C USED TO WRITE TO TEMP OUTPUT FILE TO MAKE LIFE EASIER
C OPEN(UNIT=4, FORM='BINARY', RECL=146, SHARE='ERO')
C OPEN MAIN FILE INPUT ONLY
OPEN(UNIT=3, FORM='BINARY', RECL=146, ACCESS='DIRECT')
C OPEN FILE TO COPY TO
10 READ(4, END=11) BUFF
WRITE(3) BUFF
GOTO 10
C COPY ONE AT A TIME TO THE END
C 11 CONTINUE
C COPIED MAIN FILE
CLOSE(UNIT=4, STATUS='KEEP')
C SO CLOSE IT
C NOW TO START FINDING THE CHANGES AND PUTTING THEM IN THE RIGHT
C PLACES
OPEN(UNIT=6, FORM='BINARY', RECL=148, SHARE='ERO')
C OPEN THE TEMP FILE
REWIND 6
C AND POSITION IT
20 READ(6, END=25) BUFF2
C READ THE RECORD THAT MAKES UP EACH CHANGE
C WRITE(3, REC=IBASE) BUFF
C IBASE WAS SET IN WATMERGE.FTN AND STORED USING EQUIVALENCE
GOTO 20
C BACK ROUND FOR THE NEXT CORRECTION
C AT END OF FILE
25 CLOSE(UNIT=6, STATUS='KEEP')
CLOSE(UNIT=3, STATUS='KEEP')
C ALL NOW AMALGAMATED WITH MASTER FILE IN TEMPORARY FILE
END

```

APPENDIX 6

Program PLANSRCH

```
*PLANSRCH.CSS - SEARCH THE PLAN FILE
$IFX PLAN.UPD;$WRITE FILE BEING UPDATED;$CLEAR;$ENDC
L PLANSRCH,10; AS 4,PLANMAIN.DAT,ERO
$IFNULL @1; AS 6,CPR:; $ELSE; AS 6,@1; $ENDC
  AS 5,CON:; ST
$EXIT
```

#TITL PLANSRCH.FTN - SEARCH PLAN FILE

C

CHARACTER*6 PLAN,TPLAN
 CHARACTER*80 TITLE
 CHARACTER*4 AUTHOR,DRAFT
 CHARACTER*2 CATG,PLNSIZ
 CHARACTER*16 LOCATN
 CHARACTER*11 QUAD
 CHARACTER*17 REMARK
 INTEGER*2 FILM,YEAR

C INPUT IS PLAN NUMBER,TITLE,AUTHOR,CATEGORY,YEAR,LOCALITY,

C QUADRANGLE NUMBER,PLANSIZE,DRAFTSMAN,REMARKS,

C MASTER FILM ROLL NUMBER.

CHARACTER*1 BUFF(146)

EQUIVALENCE (BUFF(1),TITLE),(BUFF(81),REMARK),(BUFF(98),LOCATN),

. (BUFF(114),QUAD),(BUFF(125),PLAN),(BUFF(131),AUTHOR),

. (BUFF(135),DRAFT),(BUFF(139),CATG),(BUFF(141),PLNSIZ),

. (BUFF(143),YEAR),(BUFF(145),FILM)

C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS

C USED TO PUT FORM FEEDS INTO OUTPUT FILE

LOGICAL NOTIN,PRNT

INTEGER*2 IYR(30),NYEAR,FOUND(5000),FCOUNT

CHARACTER*80 STITL1,SLOCN,SQUAD,SYEAR,SCAT,SUBCAT,STITL2,SAUTHR,

. SDRAFT

C THE FORM FEED CHARACTER IN A1

OPEN(UNIT=4,FORM='BINARY',ACCESS='DIRECT',RECL=146,SHARE='ERO')

C OPEN THE PLANMAIN.DAT FILE

PRNT=.FALSE.

C USED TO INDICATE IF THERE HAS BEEN ANY PRINTED OUTPUT

C

10 CONTINUE

REWIND 4

WRITE(5,101)

101 FORMAT(' FOR OPTIONS ENTER EITHER A BLANK LINE OR'/

. ' THE DESIRED VALUES FOR THE OPTION, EACH TERMINATED BY'

. , ' A '/'/' E.G. FRED/BILL/TOM/SMITH/'/')

WRITE(5,102)

102 FORMAT(' FIRST TITLE SELECTION?')

READ(5,201) STITL1

201 FORMAT(A80)

WRITE(5,105)

105 FORMAT(' SECOND TITLE SELECTION?')

READ(5,201) STITL2

WRITE(5,103)

103 FORMAT(' LOCATION?')

READ(5,201) SLOCN

WRITE(5,104)

104 FORMAT(' QUADRANGLE NO. ?')

READ(5,201) SQUAD

WRITE(5,112)

112 FORMAT(' CATEGORY?')

READ(5,201) SCAT

WRITE(5,107)

107 FORMAT(' SUB-CATEGORY?')

READ(5,201) SUBCAT

WRITE(5,108)

108 FORMAT(' AUTHOR?')

READ(5,201) SAUTHR

```

        WRITE(5,106)
106 FORMAT(' YEAR?')
        READ(5,201) SYEAR
        NYEAR=0
C COUNTER FOR NUMBER OF YEAR OPTIONS INPUT
        IF (SYEAR(1:2) .EQ. ' ') GOTO 21
C NO YEAR SEARCH DESIRED
        CALL NMERIC(SYEAR,IYR,NYEAR)
C
C NOW GET THE NEXT SEARCH OPTION
21 CONTINUE
        WRITE(5,113)
113 FORMAT(' DRAFTSMAN?')
        READ(5,201) SDRAFT
C END OF OPTION INPUT
C
C*****
C NOW FOR THE SEARCH
        FCOUNT=0
C COUNTER FOR NUMBER OF FOUND ENTRIES
        RECCNT=0
C COUNTER FOR POSITION IN PLANMAIN.DAT
27 READ(4,END=40) BUFF
C READ ONE ENTRY
        RECCNT=RECCNT+1
C READ ONE MORE RECORD FROM PLANMAIN
        IF (NOTIN(STITL1,TITLE)) GOTO 27
C NOT THE RIGHT FIRST TITLE SELECTION
        IF (NOTIN(STITL2,TITLE)) GOTO 27
C NOT THE RIGHT SECOND TITLE SELECTION
        IF (SCAT(1:2) .NE. ' ') THEN
            IF (INDEX(SCAT,CATG(1:1)) .EQ. 0) GOTO 27
        ENDIF
C NOT RIGHT CATEGORY
        IF (SUBCAT(1:2) .NE. ' ') THEN
            IF (INDEX(SUBCAT,CATG(2:2)) .EQ. 0) GOTO 27
        ENDIF
C NOT THE SUB-CATEGORY
        IF (NOTIN(SLOCN,LOCATN)) GOTO 27
C NOT THE RIGHT LOCATION
        IF (NOTIN(SQUAD,QUAD)) GOTO 27
C NOT FOUND SO FALL OUT
        IF (NYEAR .NE. 0) THEN
            DO 28 I=1,NYEAR
                IF (IYR(I) .EQ. YEAR) GOTO 29
            28 CONTINUE
            GOTO 27
        ENDIF
C NOT FOUND SO FALL OUT
29 CONTINUE
        ENDIF
C LOOKED FOR YEAR
        IF (NOTIN(SAUTHR,AUTHOR)) GOTO 27
C NOT THE RIGHT AUTHOR
        IF (NOTIN(SDRAFT,DRAFT)) GOTO 27
C NOT THE RIGHT DRAFTSMAN
C IF FINALLY GET TO HERE HAVE FOUND AN ENTRY THAT MATCHES THE
C SEARCH OPTIONS
C

```

```

C PUT RECORD NUMBER INTO STORAGE
  FCOUNT=FCOUNT+1
C NO. OF FOUND ENTRIES
  IF (FCOUNT .EQ. 5001) STOP 'SEEK HELP - 5000 ENTRY ERROR'
  FOUND(FCOUNT)=RECCNT
C STORE REC NO IN PLANMAIN.DAT
  GOTO 27
C BACK ROUND
C BACK AROUND
C
C
C NOW AT THE END OF THE RUN
  40 CONTINUE
C NOW DECIDE IF IT IS GOING TO BE PUT OUT ON THE SCREEN
  WRITE(5,123) FCOUNT
  123 FORMAT(I7,' RECORDS WERE FOUND'/' DO YOU WANT A',
    .      ' SCREEN LISTING (Y OR N)?')
  READ(5,200) STYPE
  200 FORMAT(A1)
  IF (FCOUNT .EQ. 0) GOTO 20
C NO ENTRIES FOUND SO EXIT ASAP
  IF (STYPE .EQ. 'N') GOTO 47
C NO SCREEN LIST WANTED
C
C TO SCREEN LIST USE DIRECT I/O
  SCOUNT=0
C NO. OF ENTRIES SENT TO SCREEN
  DO 41 I=1,FCOUNT
    READ(4,REC=FOUND(I)) BUFF
    WRITE(5,122) PLAN,LOCATN,AUTHOR,YEAR,REMARK,PLNSIZ,TITLE(1:78)
  122 FORMAT(1X,A6,' ',A16,' ',A4,' ',I5,' ',A17,' ',A2/1X,A78)
    SCOUNT=SCOUNT+1
    IF (SCOUNT .LT. 10) GOTO 41
C ALLOW 10 ENTRIES PER SCREENFULL
  SCOUNT=0
C RESET
  WRITE(5,117)
  117 FORMAT(' LIST MORE ENTRIES ON SCREEN (Y OR N)?')
  READ(5,200) STYPE
  IF (STYPE .EQ. 'N') GOTO 47
C ROUND AGAIN FOR ANOTHER SCREEN FULL
  41 CONTINUE
  47 WRITE(5,118)
  118 FORMAT(' DO YOU WANT A PRINTOUT OF THIS SEARCH (Y OR N)?')
  READ(5,200) STYPE
  IF (STYPE .NE. 'Y') GOTO 20
C STOP IF NOTHING FOUND
C
C OTHERWISE WRITE TO PRINTER
  PRNT=.TRUE.
C REMEMBER THAT WE HAVE PRINTED
  LINE=0
C COUNTER FOR NUMBER OF LINES PRINTED
  DO 42 I=1,FCOUNT
    READ(4,REC=FOUND(I)) BUFF
    IF (MOD(LINE,53) .EQ. 0) CALL HEAD
C HEAD PAGE
  LINE=LINE+1

```

```

        WRITE(6,300) PLAN,TITLE,LOCATN,AUTHOR,YEAR,REMARK(1:12),PLNSIZ
300  FORMAT(1X,A6,' ',A80,' ',A16,' ',A4,' ',I4,' ',A12,' ',A2)
        42 CONTINUE
C END PRINT LIST RESULTING FROM SEARCH
C WRITE THE SEARCH OPTIONS LAST
        WRITE(6,400)
400  FORMAT(/' SEARCH OPTIONS ARE:')
        IF (STITL1(1:1) .NE. ' ') WRITE(6,401) STITL1
401  FORMAT(' FIRST TITLE SELECTION',10X,A80)
        IF (STITL2(1:1) .NE. ' ') WRITE(6,407) STITL2
407  FORMAT(' SECOND TITLE SELECTION',10X,A80)
        IF (SLOCN(1:1) .NE. ' ') WRITE (6,402) SLOCN
402  FORMAT(' LOCATION',10X,A80)
        IF (SQUAD(1:1) .NE. ' ') WRITE(6,403) SQUAD
403  FORMAT(' QUADRANGLE NO.',10X,A80)
        IF (SCAT(1:1) .NE. ' ') WRITE(6,410) SCAT
410  FORMAT(' CATEGORY',10X,A80)
        IF (SUBCAT(1:1) .NE. ' ') WRITE(6,406) SUBCAT
406  FORMAT(' SUB-CATEGORY',10X,A80)
        IF (SAUTHR(1:1) .NE. ' ') WRITE(6,408) SAUTHR
408  FORMAT(' AUTHOR',10X,A80)
        IF (NYEAR .NE. 0) WRITE(6,405) SYEAR
405  FORMAT(' YEAR',10X,A80)
        IF (SDRAFT(1:1) .NE. ' ') WRITE(6,409) SDRAFT
409  FORMAT(' DRAFTSMAN',10X,A80)
C
        20 CONTINUE
        IF (PRNT) WRITE(6,416)
416  FORMAT(1X)
        WRITE(5,124)
124  FORMAT(' MORE SEARCHES (Y OR N)?')
        READ(5,200) ASRCH
        IF (ASRCH .NE. 'Y') THEN
            STOP
        ENDIF
C
        GOTO 10
C BACK ROUND
        END
        LOGICAL FUNCTION NOTIN(SSTRNG,RSTRNG)
C SSTRNG CONTAINS A SERIES OF STRINGS TERMINATED BY /
C E.G.  FRED/BILL/TOM/
C AND RSTRNG IS SEARCHED FOR THE OCCURENCE OF ONE OF THESE STRINGS
C IF ONE IS FOUND TO MATCH NOTIN IS .FALSE.
C IF STRNG IS EMPTY NOTIN IS .FALSE.
        CHARACTER*80 SSTRNG
        CHARACTER* (*) RSTRNG
        NOTIN=.FALSE.
        IF (SSTRNG(1:1) .EQ. ' ') RETURN
C NO NEED TO LOOK
        ILAST=1
C COUNTER FOR POSITION IN SSTRNG
        20 ICHAR=INDEX(SSTRNG(ILAST:80),'/')
C LOOK FOR / AS TERMINATOR OF STRING
        IF (ICHAR .EQ. 0) GOTO 25
C HAVENT FOUND A MATCH BY THE END OF THE LINE
        IF (INDEX(RSTRNG,SSTRNG(ILAST:ILAST+ICHAR-2)) .NE. 0) RETURN
C LOOK BETWEEN /S IN SSTRNG AND IF FIND A MATCH RETURN

```

```

      ILAST=ILAST+ICHR
C POSITION AFTER THE /
      IF (ILAST .LE. 79) GOTO 20
C BACK ROUND IF NOT AT END OF LINE
C
C ELSE END
      25 CONTINUE
      NOTIN=.TRUE.
C NO MATCH
      RETURN
      END
      SUBROUTINE HEAD
C HEAD UP PAGES FROM SEARCH
      WRITE(6,100)
100  FORMAT(1H1/2X,'PLAN ',37X,'TITLE',38X,' ',4X,'LOCATION',
. 4X,'(AUTH;YEAR)',3X,'REMARK',3X,'(SIZ)')
      RETURN
      END
      SUBROUTINE NERIC(STRING,IVAR,NVAR)
      CHARACTER* (*) STRING
      INTEGER*2 IVAR(2),NVAR
      REAL BUFF(2)
      NVAR=0
      ILAST=1
C COUNT POSITION ALONG INPUT LINE
      20 ICHR=INDEX(STRING(ILAST:80),'/')
C LOOK FOR / AS TERMINATOR OF A STRING
      IF (ICHR .EQ. 0) GOTO 21
C NO MORE /
      ENCODE(BUFF,300) STRING(ILAST:ILAST+3)
      300 FORMAT(A4)
      NVAR=NVAR+1
      DECODE(BUFF,301) IVAR(NVAR)
      301 FORMAT(I4)
C CONVERTED TO AN INTEGER
      ILAST=ILAST+ICHR
C POSITION TO AFTER THE /
      IF (ILAST .NE. 79) GOTO 20
C ROUND FOR NEXT IVAR
      21 RETURN
      END

```

APPENDIX 7

Program PLANLIST

```
*PLANLIST.CSS - LIST THE PLAN FILE
$IFX PLAN.UPD;$WRITE FILE BEING UPDATED;$CLEAR;$ENDC
L PLANLIST,10; AS 4,PLANMAIN.DAT,ERO
$IFNULL @1; AS 6,CPR:; $ELSE; AS 6,@1; $ENDC
  AS 5,CON:; ST
$EXIT
```

\$TITL PLANLIST.FTN - LIST PLAN FILE

```

C
  CHARACTER*6 PLAN, APLAN, BPLAN
  CHARACTER*80 TITLE
  CHARACTER*4 AUTHOR, DRAFT
  CHARACTER*2 CATG, PLNSIZ
  CHARACTER*16 LOCATN
  CHARACTER*11 QUAD
  CHARACTER*17 REMARK
  INTEGER*2 FILM, YEAR, PAGE
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C   QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C   MASTER FILM ROLL NUMBER.
  CHARACTER*1 BUFF(146)
  LOGICAL FOUND, THEEND
C USE TO REMEMBER IF HAVE FOUND THE FIRST PLAN NUMBER AND IF HAVE TO
C LIST TO THE END OF THE FILE
  EQUIVALENCE (BUFF(1), TITLE), (BUFF(81), REMARK), (BUFF(98), LOCATN),
  . (BUFF(114), QUAD), (BUFF(125), PLAN), (BUFF(131), AUTHOR),
  . (BUFF(135), DRAFT), (BUFF(139), CATG), (BUFF(141), PLNSIZ),
  . (BUFF(143), YEAR), (BUFF(145), FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
  INTEGER*2 FFEED
C USED TO PUT FORM FEEDS INTO OUTPUT FILE
  DATA FFEED/3072/
C THE FORM FEED CHARACTER IN A1
  OPEN(UNIT=4, FORM='BINARY', ACCESS='DIRECT', RECL=146, SHARE='ERO')
C OPEN THE PLANMAIN.DAT FILE
  THEEND=.TRUE.
C SET TO FALSE IF HAVE A TARGET FOR THE END OF THE PLAN LISTING
  WRITE(5,100)
  100 FORMAT(' PLAN NUMBER TO START LISTING FROM'/
  . ' LEAVE BLANK TO LIST ALL')
  READ(5,101) APLAN
  101 FORMAT(A6)
  FOUND=APLAN .EQ. ' '
C IF ALL BLANKS WANT TO LIST THE LOT
  IF (.NOT. FOUND) THEN
C WANT TO FIND THE RANGE OF THE PLAN NUMBER WITHIN APLAN
  WRITE(5,200)
  200 FORMAT(' FINISHING PLAN NUMBER'/' LEAVE BLANK TO LIST'
  . ' ALL PLANS AFTER THE STARTING NUMBER')
  READ(5,101) BPLAN
  THEEND=APLAN .EQ. ' '
C IF ALL BLANKS WANT TO LIST TO THE END
  DO 8 I=1,6
  IF (APLAN(I:I) .GT. '0') GOTO 9
  APLAN(I:I)=' '
C REPLACE LEADING ZEROES BY SPACES
  8 CONTINUE
  9 CONTINUE
  J=6
  DO 12 I=1,6
  IF (APLAN(J:J) .NE. ' ') GOTO 13
  J=J-1
  12 CONTINUE
C FIND THE FIRST NON-BLANK CHARACTER STARTING FROM THE RIGHT
  13 CONTINUE

```

```

ENDIF
IF (.NOT. THEEND) THEN
C WANT TO FIND THE RANGE OF THE PLAN NUMBER WITHIN BPLAN
DO 18 I=1,6
  IF (BPLAN(I:I) .GT. '0') GOTO 19
  BPLAN(I:I)=' '
C REPLACE LEADING ZEROES BY SPACES
18 CONTINUE
19 CONTINUE
  JJ=6
  DO 22 I=1,6
    IF (BPLAN(JJ:JJ) .NE. ' ') GOTO 23
    JJ=JJ-1
  22 CONTINUE
C FIND THE FIRST NON-BLANK CHARACTER STARTING FROM THE RIGHT
  23 CONTINUE
ENDIF
LINE=0
C COUNTER FOR NUMBER OF LINES PRINTED
PAGE=0
C SET THE PAGE NUMBER TO 0 INITIALLY
CALL HEAD(PAGE)
C HEAD UP THE FIRST PAGE
41 READ(4,END=42) BUFF
  IF (.NOT. FOUND) FOUND=INDEX(PLAN,APLAN(1:J)) .NE. 0
C IF WE HAVE FOUND THE PLAN THEN WANT TO PRINT ALL THE REST
  IF (FOUND) THEN
    IF (MOD(LINE,18) .EQ. 0 .AND. LINE .NE. 0) CALL HEAD(PAGE)
C HEAD PAGE
    LINE=LINE+1
    WRITE(6,300) PLAN,TITLE,LOCATN,AUTHOR,YEAR,REMARK,CATG,QUAD,
      . PLNSIZ,DRAFT,FILM
300 FORMAT(1X,A6,' ',A80,' ',A16,' ',A4,' ',I4/
      . 9X,' ',A17,' ',A2,' ',A11,' ',A2,' ',A4,' ',I3/)
    IF (.NOT. THEEND .AND. INDEX(PLAN,BPLAN(1:JJ)) .NE. 0) GOTO 42
C LOOKING FOR A STOPPING PLAN NUMBER AND HAVE FOUND IT
  ENDIF
  GOTO 41
42 CONTINUE
  WRITE(6,305)
305 FORMAT(1X)
C END PRINT LIST RESULTING FROM SEARCH
  STOP
  END
  SUBROUTINE HEAD(PAGE)
C HEAD UP PAGES FROM SEARCH
  INTEGER*2 FFEED,PAGE
  DATA FFEED/3072/
C USED TO PUT FORM FEEDS IN OUTPUT FILE
  PAGE=PAGE+1
  WRITE(6,100) PAGE
100 FORMAT(1H1/100X,'PAGE',I4/2X,'PLAN ',37X,'TITLE',38X,' ',
  . 4X,'LOCATION',4X,'AUTH|YEAR'/9X,' ',5X,'REMARKS',5X,'CAT|',
  . 4X,'QUAD',3X,'|SIZ|DRAFT|ROLL'/1X,131(' '))
  RETURN
  END

```

APPENDIX 8

Program PLANLABL

```
*PLANLABL.CSS - PRODUCE PLAN LABELS FOR A FILM ROLL  
L PLANLABL,10: AS 4,PLANMAIN.DAT,ERO  
AL LABELZZZZ.DAT,IN,36  
AS 8,LABELZZZZ.DAT: AS 5,CON: ST  
SE GR 221  
L STICKLBL1/G,10: ST  
$EXIT
```

\$TITLE PLANLABL.FTN - PRODUCE PLAN LABELS

```

C
  CHARACTER*6 PLAN
  CHARACTER*80 TITLE
  CHARACTER*4 AUTHOR, DRAFT
  CHARACTER*2 CATG, PLNSIZ
  CHARACTER*16 LOCATN
  CHARACTER*11 QUAD
  CHARACTER*17 REMARK
  INTEGER*2 FILM, YEAR, PAGE
  LOGICAL IER
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C   QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C   MASTER FILM ROLL NUMBER.
  CHARACTER*1 BUFF(146)
  EQUIVALENCE (BUFF(1), TITLE), (BUFF(81), REMARK), (BUFF(98), LOCATN),
    (BUFF(114), QUAD), (BUFF(125), PLAN), (BUFF(131), AUTHOR),
    (BUFF(135), DRAFT), (BUFF(139), CATG), (BUFF(141), PLNSIZ),
    (BUFF(143), YEAR), (BUFF(145), FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
  OPEN(UNIT=4, FORM='BINARY', ACCESS='DIRECT', RECL=146, SHARE='ERO')
C OPEN THE PLANMAIN.DAT FILE
  OPEN(UNIT=8, RECL=36)
C OPEN THE LABEL FILE
  10 WRITE(5,100)
  100 FORMAT(' ENTER ROLL NUMBER FOR WHICH LABELS ARE TO BE PRINTED'/
    ' TERMINATE BY A BLANK LINE')
  CALL INFREE(IROLL, IER, 5)
  IF (IROLL .EQ. 0) GOTO 42
  REWIND 4
  41 READ(4, END=10) BUFF
  IF (IROLL .NE. FILM) GOTO 41
C NOT THE RIGHT ROLL
  DO 30 I=1, 2
  WRITE(8, 300) PLAN, TITLE(1:36), TITLE(37:72), LOCATN, YEAR, REMARK
  300 FORMAT(' ', 15X, A6//2(A36//)2X, A16, 8X, I4/A17)
  30 CONTINUE
C PRINT TWO COPIES OF EACH LABEL
  GOTO 41
  42 CONTINUE
C END PRINT LABELS RESULTING FROM SEARCH
  CLOSE(UNIT=8, STATUS='KEEP')
  STOP
  END
#INCLUDE INFREE.FTN, -

```

```

      SUBROUTINE INFREE(INUM, IER, ILUN)
C A FREE FORMAT INPUT ROUTINE FOR READING ONE INTEGER*4 NUMBER FROM
C ILUN.
C INUM, ILUN ARE BOTH INTEGER*4
C IER IS A LOGICAL
C MAKES USE OF ENCODE TO GO FROM FREE FORMAT TO
C FIXED FORMAT BY WRITING IN A FORM SUITABLE FOR RE-READING
C IN I6 FORMAT.
      INTEGER*2 A(10), BLA, I, J, K, JUP, L
      REAL BUFF(5)
      LOGICAL IER
      DATA BLA/1H /
C USED TO DETERMINE NON-BLANK CHARACTERS
      IER=.TRUE.
      READ(ILUN,100,ERR=101) A
      100 FORMAT(10A1)
C READ IN A CHARACTER FORMAT
      DO 10 I=1,10
        IF (A(I) .NE. BLA) GOTO 20
C LOOK FOR FIRST NON-BLANK CHARACTER
      10 CONTINUE
C IF FALL OUT OF LOOP ALL CHARACTERS ARE BLANK - HENCE INUM=0
      IER=.FALSE.
      INUM=0
      101 RETURN
C THE QUICK RETURN
C
C NOW COPY THE FREE FORMAT INTEGER RIGHT JUSTIFIED INTO THE SAME
C ARRAY
      20 CONTINUE
      DO 11 J=I,10
        IF (A(J) .EQ. BLA) GOTO 21
      11 CONTINUE
C FIND THE END OF THE NUMBER, EITHER BY FINDING A BLANK AT THE END
C OR BY REACHING THE TENTH CHARACTER
      J=11
      21 J=J-1
      K=10
      JUP=J
C NOW COPY RIGHT JUSTIFIED AND FILL LEADING PART WITH BLANKS IF NEEDS BE
      DO 12 L=I,J
        A(K)=A(JUP)
        K=K-1
        JUP=JUP-1
      12 CONTINUE
      IF (K .EQ. 0) GOTO 15
      DO 14 L=1,K
        A(L)=BLA
      14 CONTINUE
C NOW FILLED WITH LEADING BLANKS AND RIGHT JUSTIFIED
C
C SO
C WRITE OUT TO CORE BUFFER AND READ BACK IN I6 FORMAT
      15 CONTINUE
      ENCODE(BUFF,100) A
      DECODE(BUFF,111) INUM
      111 FORMAT(I10)
C NOW HAVE THE NUMBER
      IER=.FALSE.
      RETURN
      END

```

APPENDIX 9

Program FILMADD

```
*FILMADD.CSS
* FOR ADDING FILM ROLL NUMBERS TO PLAN DATA
PRE ETM
L FILMADD,10: AS 4,PLANMAIN.DAT,ERO: AL SYSF:FILTEMP.DAT,IN,146/5/2
AS 0,SYSF:FILTEMP.DAT: AS 6,PR: ST
$IFNE 0: $WRITE FILMADD ERROR: ENA ETM: $EXIT: $ENDC
$BUILD WCOPY.CMD
IN SYSF:FILTEMP.DAT
AL PLANMAIN.DAT,IN,146/10/5
OUT PLANMAIN.DAT
COPY *,*
REN I
REN 0
VERIFY *,*
END
$ENDB
REP SYSF:FILTEMP.DAT,FF00: REP PLANMAIN.DAT,0: DE PLANMAIN.DAT
L COPY32,50: ST ,COM=WCOPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0: $WRITE FILMADD-COPY ERROR: ENA ETM: $EXIT: $ENDC
REP PLANMAIN.DAT,FF00: REP SYSF:FILTEMP.DAT,0: DE SYSF:FILTEMP.DAT,WCOPY.CMD
ENA ETM: $EXIT
```

```

$TITL FILMADD.FTN - ADD THE FILM NUMBER TO THE PLAN DETAILS
CHARACTER*6 PLAN, TPLAN, APLAN(530)
LOGICAL CHANGE, CHGE, START
CHARACTER*80 TITLE
CHARACTER*4 AUTHOR, DRAFT
CHARACTER*2 CATG, PLNSIZ
CHARACTER*16 LOCATN
CHARACTER*11 QUAD
CHARACTER*17 REMARK
INTEGER*2 FILM, YEAR
C INPUT IS PLAN NUMBER, TITLE, AUTHOR, CATEGORY, YEAR, LOCALITY,
C QUADRANGLE NUMBER, PLANSIZE, DRAFTSMAN, REMARKS,
C MASTER FILM ROLL NUMBER.
CHARACTER*1 BUFF(146)
EQUIVALENCE (BUFF(1), TITLE), (BUFF(81), REMARK), (BUFF(98), LOCATN),
(BUFF(114), QUAD), (BUFF(125), PLAN), (BUFF(131), AUTHOR),
(BUFF(135), DRAFT), (BUFF(139), CATG), (BUFF(141), PLNSIZ),
(BUFF(143), YEAR), (BUFF(145), FILM)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
CHGE=.FALSE.
C USED TO DECIDE WHEN THE OUTPUT LISTING NEEDS HEADING
OPEN(UNIT=5, FILE='CON:')
C SET CONSOLE AS LU 5
OPEN(UNIT=4, FORM='BINARY', RECL=146, SHARE='ZERO')
C OPEN WATMAIN.DAT
OPEN(UNIT=8, FORM='BINARY', RECL=146)
C AMENDED OUTPUT FILE
IPLAN=0
C COUNTER FOR NUMBER OF PLAN NUMBERS READ
C
WRITE(5,100)
100 FORMAT(' FILM NUMBER - THREE DIGITS MAX')
READ(5,*) IFILM
10 WRITE(5,101)
101 FORMAT(' PLAN NUMBER - END TERMINATES')
READ(5,102) PLAN
102 FORMAT(A6)
IF (PLAN .EQ. 'END ') GOTO 50
C END OF RUN
C WE WANT TO GET THE PLAN NUMBER IN THE FORMAT I4.A2
C AT THE MOMENT IT COULD BE ANYTHING
I=CTOI(PLAN,K)
C K POINTS TO THE LAST NUMERIC DIGIT IN THE FIRST NUMERIC STRING OF
C PLAN. I.E. 102/3 WILL HAVE K=3
TPLAN(5:6)=PLAN(K+1:K+2)
C THATS THE A2 PART DONE
J=4
IK=K
C WILL NOW WORK BACKWARDS THROUGH TPLAN AND PLAN
DO 30 I=1,K
TPLAN(J:J)=PLAN(IK:IK)
J=J-1
IK=IK-1
30 CONTINUE
IF (J .NE. 0) THEN
C HAVENT FILLED RIGHT TO THE FRONT OF TPLAN SO WILL NEED TO FILL
C WITH BLANKS
DO 35 I=1,J

```

```

      TPLAN(I:I)=' '
35  CONTINUE
      ENDIF
      PLAN=TPLAN
      DO 38 I=1,6
      IF (PLAN(I:I) .GT. '0') GOTO 39
      PLAN(I:I)=' '
C  REPLACE LEADING 0 S BY SPACES
      38 CONTINUE
      39 CONTINUE
      IPLAN=IPLAN+1
      APLAN(IPLAN)=PLAN
      GOTO 10
C  ROUND FOR THE NEXT PLAN
      50 CONTINUE
C  NOW GET READY TO SCAN THE FILE FOR THE PLANS
      READ(4,END=80) BUFF
C  IF WE HIT THE END MUST HAVE FINISHED
      IF (IPLAN .NE. 0) THEN
      DO 60 I=1,IPLAN
      IF (APLAN(I) .NE. PLAN) GOTO 60
C  LOOK FOR MATCHING PLAN NUMBER
      IF (.NOT. CHGE) THEN
      WRITE(6,229) IFILM
      229 FORMAT(' THE FOLLOWING PLANS HAVE BEEN ASSIGNED TO FILM',I4)
      CHGE=.TRUE.
      ENDIF
C  HEAD UP ONLY ONCE
      WRITE(6,300) PLAN
      300 FORMAT(1X,A6)
C  WRITE THE PLAN NUMBER AS CONFIRMATION
      FILM=IFILM
C  LOAD THE FILM NUMBER
      GOTO 70
C  HAVE MATCHED THE PLAN SO GO AROUND
      60 CONTINUE
      ENDIF
      70 WRITE(8) BUFF
      GOTO 50
C  BACK ROUND
C
C  NOW TO TIDY UP
      80 CONTINUE
      IF (CHGE) WRITE(6,302)
      302 FORMAT(1X/)
      CLOSE(UNIT=4, STATUS='KEEP')
      CLOSE(UNIT=8, STATUS='KEEP')
      END

```