UR1983_40

1983/40. CARS - A computer assisted records system

R.G. Richardson

Abstract

The program suite described is used for adding entries to a data-base of Engineering Geology correspondence, sorting the data-base, and searching it. The data is entered in a simple fill-in-the spaces manner and part or all of the details entered may be searched.

USING THE PROGRAMS

The programs are run on the Geological Survey Perkin-Elmer mini-computer, and it is assumed that the user is familiar with the standard data editing facilities.

Data-base maintenance

Data is put into a standard format (Appendix 1) before entry. Input is commenced by typing RECADD and continues until END is typed in response to the name prompt. Data is entered left justified in response to prompts, with the category and geology numbers being entered as either one or two digits on the same line.

At the end of input, a proof-sheet is output and the data saved in file RECTEMP.TMP. Any corrections should be made to this file using the standard editing facilities and a new proof sheet then printed and checked.

The new data is RECTEMP.TMP must be added to the end of the existing data-base by typing RECMERGE. To ensure that the file remains in records reference number order, it should be sorted periodically using the RECSORT command. As sorting is a slow process it should not be carried out more often than necessary.

Data-base searching

The search phase is entered by typing RECSRCH. An entry will be retrieved only if one value in each search option specified is found.

The program requests the following data:

(i) Search type - F for a search on specified option values
- R for a search between a range of records reference numbers. If R is specified, the starting and ending record reference numbers are specified. *e.g.* 1234/78.

(ii) Name - one or more names on a single line, each name terminated by /. If a search by name is not required leave the line blank. *e.g.* BROWN/JONES/SMITH/

(iii) Job location - details as for name.

(iv) Agency name - details as for name.

(v) Agency type - one or more agency numbers on a single line, each terminated by /. If a search by agency type

is not required leave the line blank.
e.g. 0/ find agency type 0
0/2/ find agency types 0 or 2

(vi) Year - one or more years on a single line, each terminated by a /. Leave blank if not required.
e.g. 82/ only year 1982
75/76/77/78/79/80/ years 1975 to 1980 inclusive.

(vii) Geologist - details as for name. *e.g.* PRW/WCC/DJJ/

(viii) Search by area - N for no search by area
- Y for finding entries inside a specified co-ordinated area. Entries on the boundary of the area will not be found. The program prompts for the number of vertices and the vertex co-ordinates in the same form as used on the input sheets.

(ix) Category - one or more category numbers on a single line, each group terminated by /. Leave blank if not required.
e.g. 0/2/ groundwater or expansive soils
12/4/ slope stability and expansive soils or miscellaneous

(x) Geology - one or more geology numbers on a single line, each group terminated by /. Leave blank if not required.
e.g. 0/9/ Quaternary or granite
17/ Tertiary sediments and Tertiary basalt.

(xi) Printout required - Y if the number of entries found can be realistically printed.
- N if the number of entries found is too great to want them printed.

THE PROGRAMS

RECADD (Appendix 2)

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 (RECTEMP.TMP) is then edited using the standard edit facilities.

RECMERGE (Appendix 3)

The data from RECTEMP.TMP is converted to the format of the data-base. The data-base is copied to a temporary work file and the new data is added to the end of this. The combined file is then copied back to the original data-base file.

RECSORT (Appendix 4)

The data is sorted into ascending records reference number order and written to a temporary file in the new order. The temporary file is then copied back to the original data-base file.

RECSRCH (Appendix 5)

The data-base is searched for the occurrence of specified strings and a point-in-polygon is used to locate data from within a specified area.

[19 August 1983]

APPENDIX 1

Data input sheet

CARS (COMPUTER ASSISTED RECORDS SYSTEM)

NAME: .... (16)

JOB LOCATION: .... (34)

AGENCY: .... (20)

0 GOVT. 1 PROF. 2 INDIV.

RECORDS REF. NO.: ...../..

GEOLOGIST: ... (3)

MAP REF:.....5.... (9)

CATEGORY:*

0 GROUNDWATER
1 SLOPE STABILITY
2 EXPANSIVE SOILS
3 FOUNDATION INVESTIGATION
4 MISCELLANEOUS

GEOLOGY:*

Qu	0	Ts	1	Trias	2	Perm	3	L-Pal	4
C-	5	Pre-E	6	Tb	7	Jdl	8	Gran	9

*Tick two (2) maximum

CARS (COMPUTER ASSISTED RECORDS SYSTEM)

NAME: .... (16)

JOB LOCATION: .... (34)

AGENCY: .... (20)

0 GOVT. 1 PROF. 2 INDIV.

RECORDS REF. NO.: ...../..

GEOLOGIST: ... (3)

MAP REF: .....5.... (9)

CATEGORY:*

0 GROUNDWATER
1 SLOPE STABILITY
2 EXPANSIVE SOILS
3 FOUNDATION INVESTIGATION
4 MISCELLANEOUS

GEOLOGY:*

Qu 0 Ts 1 Trias 2 Perm 3 L-Pal 4
C- 5 Pre-E 6 Tb 7 Jdl 8 Gran 9

*Tick two (2) maximum

APPENDIX 2

Program RECADD

```
*RECADD.CSS
*FOR RUNNING RECADD AND PRINTING A PROOF SHEET
$IFX RECTEMP.TMP; $WRITE ** CHECK LAST MERGE; $EXIT; $ENDC
PRE ETM; AL RECTEMP.TMP,IN,106
* SET UP THE OUTPUT FILE
L RECADD,3; AS 6,RECTEMP.TMP; * LOAD PROG AND SET LU 6
TEMPFILE 3,IN,80; ST
PRINT RECTEMP.TMP
$WRITE  FILE IS RECTEMP.TMP; ENA ETM; $EXIT
```

```
$TITL  RECADD.FTN  -  ADD LETTERS TO THE RECORDS FILE
C TAKES KEYBOARD INPUT AND PREPARES A PROOF SHEET
      CHARACTER*1 BUFF(9)
      INTEGER*4 MAPREF
      CHARACTER*16 NAME
      CHARACTER*34 LOCATN
      CHARACTER*20 AGNAME
      INTEGER*2 AGNO,RECNO,RECYR
      CHARACTER*3 GEOLST
      CHARACTER*2 CATG,GEOL
C INPUT IS NAME,LOCATION,AGENCY NAME,AGENCY TYPE(AGNO),
C            RECORDS FILE NO (RECNO/RECYR),GEOLOGIST,
C            CATEGORY(UP TO 2),GEOLOGY (UP TO 2)
C
      OPEN(UNIT=5,FILE='CON:')
C SET THE CONSOLE UP AS LU 5 FOR INPUT
C THE OUTPUT IS TO LU 6
   10 WRITE(5,100)
  100 FORMAT(' NAME OR END')
      READ(5,200) NAME
  200 FORMAT(A16)
      IF (INDEX(NAME,'END     ') .NE. 0) GOTO 20
C RECOGNISE END BY HAVING END ON ITS OWN AND PRINT PROOF SHEET
      WRITE(5,101)
  101 FORMAT(' LOCATION')
      READ(5,201) LOCATN
  201 FORMAT(A34)
      WRITE(5,102)
  102 FORMAT(' AGENCY')
      READ(5,202) AGNAME
  202 FORMAT(A20)
      WRITE(5,103)
  103 FORMAT(' AGENCY TYPE')
      READ(5,*) AGNO
      WRITE(5,104)
  104 FORMAT(' REF NO')
      READ(5,109) BUFF
  109 FORMAT(9A1)
      DO 12 I=1,9
      IF (BUFF(I) .EQ. '/') BUFF(I)=','
   12 CONTINUE
C CHANGE / TO , TO COPE WITH FREE FORMAT READ
      REWIND 3
      WRITE(3,109) BUFF
      REWIND 3
      READ(3,*) RECNO,RECYR
      WRITE(5,105)
  105 FORMAT(' GEOLOGIST')
      READ(5,203) GEOLST
  203 FORMAT(A3)
      WRITE(5,106)
  106 FORMAT(' MAP REF')
      READ(5,*) MAPREF
      WRITE(5,107)
  107 FORMAT(' CATEGORY')
      READ(5,204) CATG
  204 FORMAT(A2)
      WRITE(5,108)
  100 FORMAT(' GEOLOGY')
      READ(5,204) GEOL
C READ ONE COMPLETE DATA SHEET
C
C SO WRITE TO OUTPUT FILE
      WRITE(6,300) NAME,LOCATN,AGNAME,AGNO,RECNO,RECYR,GEOLST,MAPREF,
     .           CATG,GEOL
  300 FORMAT(1X,'@',A16,'@',A34,'@',A20,'@',I1,'@',I5,'/',
     .         I2,'@',A3,'@',I9,2('@',A2),'@')
C WRITE WITH DELIMITERS AFTER EACH FIELD
      GOTO 10
C BACK AROUND
C
C OTHERWISE CLOSE UP AND STOP
   20 CONTINUE
      CLOSE(UNIT=6,STATUS='KEEP')
      END
```

APPENDIX 3

Program RECMERGE

```
*RECMERGE.CSS
* FOR MERGING CORRECTED FILE WITH MAIN FILE
PRE ETM; XDE MRGTMP.TMP; *DELETE SCRATCH FILE
AL MRGTMP.TMP,IN,88; L RECMERGE,10; * AL SCRATCH FILE AND LOAD PROG
REP RECTEMP.TMP,FF00; AS 6,RECTEMP.TMP,ERO; AS 4,MRGTMP.TMP; ST
$IFNE 0; $WRITE TRANSLATE ERROR; ENA ETM;  $EXIT; $ENDC
REP MRGTMP.TMP,FF00
$BUILD COPY.CMD
IN JOBREC.DAT
AL TEMP:TEMP.DAT,IN,88/6/5
OUT TEMP:TEMP.DAT
COPY *,*
IN MRGTMP.TMP
COPY *,*
END
$ENDB
L COPY32,50; ST ,COM=COPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0; $WRITE COPY-MERGE ERROR; ENA ETM; $EXIT; $ENDC
REP JOBREC.DAT,0; REP TEMP:TEMP.DAT,FF00
DE COPY.CMD,JOBREC.DAT
$BUILD COPY.CMD
IN TEMP:TEMP.DAT
AL JOBREC.DAT,IN,88/10/3
OUT JOBREC.DAT
COPY *,*
REW I
REW O
VERIFY *,*
END
$ENDB
L COPY32,50; ST ,COM=COPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0; $WRITE COPY BACK FAILED; ENA ETM; $EXIT; $ENDC
REP JOBREC.DAT,FF00; REP TEMP:TEMP.DAT,0; REP RECTEMP.TMP,0
REP MRGTMP.TMP,0
DE COPY.CMD,MRGTMP.TMP,RECTEMP.TMP,TEMP:TEMP.DAT
ENA ETM
$EXIT
```

```
$TITL  RECMERGE.FTN - CONVERT PROOF FILE TO MASTER FILE FORMAT
C
      INTEGER*4 NORTH
      INTEGER*2 EAST
C SPLIT MAPREF INTO THE RELEVANT PARTS NOW
      CHARACTER*16 NAME
      CHARACTER*34 LOCATN
      CHARACTER*20 AGNAME
      INTEGER*2 RECNO,RECYR
      CHARACTER*3 GEOLST
      CHARACTER*2 CATG,GEOL
      CHARACTER*1 AGNO,BUFF(88)
C AS FOR RECADD.FTN
C
C NOW SET UP EQUIVALENCE
      EQUIVALENCE (BUFF(1),NAME),(BUFF(17),LOCATN),(BUFF(51),AGNAME),
     . (BUFF(71),RECNO),(BUFF(73),RECYR),(BUFF(75),GEOLST),
     . (BUFF(78),AGNO),(BUFF(79),CATG),(BUFF(81),GEOL),
     . (BUFF(83),EAST),(BUFF(85),NORTH)
C THIS IS USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
      OPEN(UNIT=4,FORM='BINARY',RECL=88)
C OPEN THE OUTPUT FILE FOR BINARY OUTPUT
   10 READ(6,300,END=20) NAME,LOCATN,AGNAME,AGNO,RECNO,RECYR,
     .                   GEOLST,EAST,NORTH,CATG,GEOL
  300 FORMAT(2X,A16,1X,A34,1X,A20,1X,A1,1X,I5,1X,I2,1X,A3,
     .       1X,I4,I5,2(1X,A2))
      WRITE(4) BUFF
C BINARY WRITE TO OUTPUT FILE
      GOTO 10
C BACK ROUND
C
C AT END
   20 CONTINUE
      CLOSE(UNIT=4,STATUS='KEEP')
      CLOSE(UNIT=6,STATUS='KEEP')
      END
```

APPENDIX 4

Program RECSORT

```
*RECSORT.CSS
* FOR SORTING JOB FILE INTO CHRONOLOGICAL ORDER
PRE ETM
L RECSORT,10; AS 4,JOBREC.DAT,ERO; AL TEMP:TEMP.DAT,IN,88/5/2
AS 6,TEMP:TEMP.DAT; ST
$IFNE 0; $WRITE SORT ERROR; ENA ETM; $EXIT; $ENDC
$BUILD COPY.CMD
IN TEMP:TEMP.DAT
AL JOBREC.DAT,IN,88/10/5
OUT JOBREC.DAT
COPY *,*
REW I
REW O
VERIFY *,*
END
$ENDB
REP TEMP:TEMP.DAT,FF00; REP JOBREC.DAT,0; DE JOBREC.DAT
L COPY32,50; ST ,COM=COPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0; $WRITE SORT-COPY ERROR; ENA ETM; $EXIT; $ENDC
REP JOBREC.DAT,FF00; REP TEMP:TEMP.DAT,0; DE TEMP:TEMP.DAT,COPY.CMD
ENA ETM; $EXIT
```

11/22

```
$TITL  RECSORT.FTN - SORT THE JOB FILE INTO CALENDAR ORDER
C
      INTEGER*4 NORTH
      INTEGER*2 EAST
C SPLIT MAPREF INTO THE RELEVANT PARTS NOW
      CHARACTER*16 NAME
      CHARACTER*34 LOCATN
      CHARACTER*20 AGNAME
      INTEGER*2 RECNO,RECYR
      CHARACTER*3 GEOLST
      CHARACTER*2 CATG,GEOL
      CHARACTER*1 AGNO,BUFF(88)
C AS FOR RECADD.FTN
C
C FOLLOWING ARE FOR THE SORT
      INTEGER*2 INDEX(8000)
      INTEGER*4 IA(8000)
C ALLOW FOR 8000 ENTRIES
C
C NOW SET UP EQUIVALENCE
      EQUIVALENCE (BUFF(1),NAME),(BUFF(17),LOCATN),(BUFF(51),AGNAME),
     . (BUFF(71),RECNO),(BUFF(73),RECYR),(BUFF(75),GEOLST),
     . (BUFF(78),AGNO),(BUFF(79),CATG),(BUFF(81),GEOL),
     . (BUFF(83),EAST),(BUFF(85),NORTH)
C THIS IS USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
      OPEN(UNIT=4,FORM='BINARY',ACCESS='DIRECT',RECL=88)
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. 8000) STOP 'MORE THAN 8000 ENTRIES'
      INDEX(IREC)=IREC
C FILL SORT INDEX
      IA(IREC)=RECNO+(RECYR*100000)
C COMBINE RECNO AND RECYR 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 BUBSTI(INDEX,IA,1,IREC)
C SORT INTO ASCENDING NUMBER - I.E. CHRON ORDER
      OPEN(UNIT=6,FORM='BINARY',RECL=88)
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 BUBSTI(IR,IA,IBASE,N)
      INTEGER*4 IA(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 (IA(IRI) .LE. IA(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 RECSRCH

```
*RECSRCH.CSS - SEARCH THE JOB FILE
L RECSRCH,10; AS 4,JOBREC.DAT,ER0; XAL TEMP:ENGJOBS.TMP,IN,132/3/2
AS 6,TEMP:ENGJOBS.TMP; REW 6; AS 5,CON:; TEMPFILE 3,IN,80; ST
$IFX TEMP:ENGJOBS.TMP; PRI TEMP:ENGJOBS.TMP,DEL; $ENDC; $EXIT
```

```
$TITL  RECSRCH.FTN - SEARCH JOB FILE
C
      INTEGER*4 NORTH
      INTEGER*2 EAST,FCOUNT
C SPLIT MAPREF INTO THE RELEVANT PARTS NOW
      CHARACTER*16 NAME
      CHARACTER*34 LOCATN
      CHARACTER*20 AGNAME
      INTEGER*2 RECNO,RECYR
      CHARACTER*3 GEOLST
      CHARACTER*2 CATG,GEOL
      CHARACTER*1 AGNO,BUFF(88)
C AS FOR RECADD.FTN
C
C NOW SET UP EQUIVALENCE
      EQUIVALENCE (BUFF(1),NAME),(BUFF(17),LOCATN),(BUFF(51),AGNAME),
     . (BUFF(71),RECNO),(BUFF(73),RECYR),(BUFF(75),GEOLST),
     . (BUFF(78),AGNO),(BUFF(79),CATG),(BUFF(81),GEOL),
     . (BUFF(83),EAST),(BUFF(85),NORTH)
C THIS IS USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
      CHARACTER*80 SNAME,SLOCN,SAGNAM,SAGNO,SYEAR,SGEOLT,SCAT,SGEOL
      INTEGER*2 IYR(30),RNOSTA,YRSTA,RNOSTO,YRSTO,STYPE
      INTEGER*2 FFEED
C USED TO PUT FORM FEEDS INTO OUTPUT FILE
      REAL X(100),Y(100),YINTVL(100),RSLOPE(100)
      INTEGER*2 SWATH(100,25)
C ABOVE 2 LINES USED FOR FSTJDN POINT IN POLYGON SEARCH
      LOGICAL NOTIN,INOUT
      DATA FFEED/3072/
C THE FORM FEED CHARACTER IN A1
      OPEN(UNIT=4,FORM='BINARY',RECL=88)
C OPEN THE INPUT FILE
   10 WRITE(5,100)
  100 FORMAT(' SEARCHING JOB RECORDS'/' DO YOU NEED A',
     . ' FULL SEARCH (F) OR RECORDS NO.  SEARCH (R)?')
      READ(5,200) STYPE
  200 FORMAT(A1)
      IF (STYPE .EQ. 'R') GOTO 30
C WANT SEARCH ON RECORDS NUMBER
      IF (STYPE .NE. 'F') GOTO 10
C PROMPT AGAIN IF NOT ONE OF THE TWO DESIRED OPTIONS
C
C NOW FOR A FULL SEARCH
      WRITE(6,400)
  400 FORMAT(' SEARCH OPTIONS ARE:')
      WRITE(5,101)
  101 FORMAT(' FOR EACH OPTION 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(' NAME?')
      READ(5,201) SNAME
  201 FORMAT(A80)
      IF (SNAME(1:1) .NE. ' ') WRITE(6,401) SNAME
  401 FORMAT(' NAME',10X,A80)
      WRITE(5,103)
  103 FORMAT(' JOB LOCATION?')
```

```
      READ(5,201) SLOCN
      IF (SLOCN(1:1) .NE. ' ') WRITE (6,402) SLOCN
  402 FORMAT(' JOB LOCATION',10X,A80)
      WRITE(5,104)
  104 FORMAT(' AGENCY NAME?')
      READ(5,201) SAGNAM
      IF (SAGNAM(1:1) .NE. ' ') WRITE(6,403) SAGNAM
  403 FORMAT(' AGENCY NAME',10X,A80)
      WRITE(5,105)
  105 FORMAT(' AGENCY TYPE (0,1,2)?')
      READ(5,201) SAGNO
      IF (SAGNO(1:1) .NE. ' ') WRITE(6,404) SAGNO
  404 FORMAT(' AGENCY NO.',10X,A80)
      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
      WRITE(6,405) SYEAR
  405 FORMAT(' YEAR',10X,A80)
      ILAST=1
C COUNT POSITION ALONG INPUT LINE
   20 ICHAR=INDEX(SYEAR(ILAST:80),'/')
C LOOK FOR / AS TERMINATOR OF A YEAR
      IF (ICHAR .EQ. 0) GOTO 21
C NO MORE /
      ENCODE(BUFF,300) SYEAR(ILAST:ILAST+1)
  300 FORMAT(A2)
      NYEAR=NYEAR+1
      DECODE(BUFF,301) IYR(NYEAR)
  301 FORMAT(I2)
C CONVERTED TO AN INTEGER
      ILAST=ILAST+ICHAR
C POSITION TO AFTER THE /
      IF (ILAST .NE. 79) GOTO 20
C ROUND FOR NEXT YEAR
C
C NOW GTE THE NEXT SEARCH OPTION
   21 CONTINUE
      WRITE(5,107)
  107 FORMAT(' GEOLOGIST?')
      READ(5,201) SGEOLT
      IF (SGEOLT(1:1) .NE. ' ') WRITE(6,406) SGEOLT
  406 FORMAT(' GEOLOGIST',10X,A80)
   22 WRITE(5,108)
  108 FORMAT(' SEARCH FOR SITES WITHIN A SPECIFIED AREA (Y OR N)?')
      READ(5,200) STYPE
      IF (STYPE .EQ. 'Y') GOTO 23
C WANT TO READ COORDS
      IF (STYPE .NE. 'N') GOTO 22
C INVALID OPTION
      NCOORD=0
      GOTO 26
C DONT WANT TO SEARCH FOR SITES IN AN AREA
C
   23 CONTINUE
C READ VERTICES
      WRITE(5,109)
  109 FORMAT(' NUMBER OF VERTICES OF AREA?')
      READ(5,*) NCOORD
      IF (NCOORD .LE. 2) GOTO 23
C NEED AT LEAST 3 VERTICES TO DEFINE AN AREA
      WRITE(5,110)
  110 FORMAT(' ENTER VERTEX COORDS IN CLOCKWISE ORDER'/
     .' ONE PER LINE IN FORM EASTNORTH')
      IXMAX=-999999999
      IXMIN=-IXMAX
      IYMAX=IXMAX
      IYMIN=IXMIN
C USE THESE TO STORE THE RANGE OF THE AREA
      DO 24 I=1,NCOORD
      READ(5,*) ITMP
      ENCODE(BUFF,302) ITMP
  302 FORMAT(I9)
      DECODE(BUFF,303) IX,IY
  303 FORMAT(I4,I5)
C GET INTO 100'S OF METRES EAST AND NORTH
      IXMAX=MAX0(IX,IXMAX)
      IXMIN=MIN0(IX,IXMIN)
      IYMAX=MAX0(IY,IYMAX)
      IYMIN=MIN0(IY,IYMIN)
C FIND THE RANGE
      X(I)=IX*0.1
      Y(I)=IY*0.1
C CONVERT TO KM BEFORE STORING
   24 CONTINUE
      IF (X(1) .NE. X(NCOORD) .OR. Y(1) .NE. Y(NCOORD)) THEN
       NCOORD=NCOORD+1
       X(NCOORD)=X(1)
       Y(NCOORD)=Y(1)
C CLOSE AREA
      ENDIF
      WRITE(6,407)
  407 FORMAT(' AREA COORDINATES (KM)')
      WRITE(6,408) (X(I),Y(I),I=1,NCOORD)
  408 FORMAT(20X,4(F12.0,F9.0))
      NCOORD=NCOORD-1
C ROUTINE FOR LOCATING POINTS DOESNT REQUIRE CLOSURE
      CALL PREPLY(X,Y,NCOORD,YINTVL,INTVLS,SWATH,RSLOPE)
C CUT THE SEARCH AREA UP INTO SWATHS
C
   26 CONTINUE
      WRITE(5,111)
  111 FORMAT(' CATEGORY (0-5)?')
      READ(5,201) SCAT
      IF(SCAT(1:1) .NE. ' ') WRITE(6,409) SCAT
  409 FORMAT(' CATEGORY',10X,A80)
      WRITE(5,112)
  112 FORMAT(' GEOLOGY (0-9)?')
      READ(5,201) SGEOL
      IF (SGEOL(1:1) .NE. ' ') WRITE(6,410) SGEOL
  410 FORMAT(' GEOLOGY',10X,A80)
C END OF OPTION INPUT
C
C**************************************************************
C NOW FOR THE SEARCH
      FCOUNT=0
C COUNTER FOR NUMBER OF FOUND ENTRIES
```

17/22

```
   27 READ(4,END=40) BUFF
C READ ONE ENTRY
      IF (NOTIN(SNAME,NAME)) GOTO 27
C NOT THE RIGHT NAME
      IF(NOTIN(SLOCN,LOCATN)) GOTO 27
C NOT THE RIGHT LOCATION
      IF(NOTIN(SAGNAM,AGNAME)) GOTO 27
C NOT THE AGENCY
      IF (SAGNO(1:1) .NE. ' ' .AND. INDEX(SAGNO,AGNO) .EQ. 0) GOTO 27
C NOT THE DESIRED AGENCY NUMBER
      IF (NYEAR .NE. 0) THEN
       DO 28 I=1,NYEAR
        IF (IYR(I) .EQ. RECYR) GOTO 29
   28  CONTINUE
        GOTO 27
C NOT FOUND SO FALL OUT
   29  CONTINUE
      ENDIF
C LOOKED FOR YEAR
      IF (NOTIN(SGEOLT,GEOLST)) GOTO 27
C NOT THE GEOLOGIST
      IF (NCOORD .NE. 0) THEN
       IF (EAST .LT. IXMIN .OR. EAST .GT. IXMAX) GOTO 27
       IF (NORTH .LT. IYMIN .OR. NORTH .GT. IYMAX) GOTO 27
C CANT POSSIBLY BE IN AREA
       XTMP=EAST*0.1
       YTMP=NORTH*0.1
C CONVERT TO KM
       IF (.NOT. INOUT(XTMP,YTMP,X,Y,YINTVL,INTVLS,SWATH,RSLOPE))
     . GOTO 27
C NOT IN THE SPECIFIED AREA
      ENDIF
      IF (NOTIN(SCAT,CATG)) GOTO 27
C NOT RIGHT CATEGORY
      IF (NOTIN(SGEOL,GEOL)) GOTO 27
C NOT RIGHT GEOLOGY
C IF FINALLY GET TO HERE HAVE FOUND AN ENTRY THAT MATCHES THE
C SEARCH OPTIONS
C
      IF (MOD(FCOUNT,53) .EQ. 0) CALL HEAD
C HEAD PAGE
      FCOUNT=FCOUNT+1
C INCREMENT FOUND ENTRY COUNTER
      WRITE(6,412) NAME,LOCATN,AGNAME,AGNO,RECNO,RECYR,GEOLST,EAST,
     .             NORTH,CATG,GEOL
  412 FORMAT(A16,' @',A34,' @',A20,' @',2X,A1,' @',I6,'/',I2,
     .       ' @',2X,A3,'  @',2I5,' @',4X,A2,4X,'@',3X,A2)
      GOTO 27
C BACK AROUND
C
C*****************************************************************
C NOW FOR THE CASE OF SEARCHING ONLY ON A RANGE OF RECORD NOS
   30 CONTINUE
      WRITE(5,113)
  113 FORMAT(' ENTER STARTING RECORD NO. IN FORM NNNNN,YR')
      CALL RNOIN(RNOSTA,YRSTA)
      WRITE(5,114)
  114 FORMAT(' ENTER ENDING RECORDS NUMBER')
      CALL RNOIN(RNOSTO,YRSTO)
      IF (YRSTA .GT. YRSTO .OR. YRSTA .EQ. YRSTO .AND.
     .          RNOSTA .GT. RNOSTO) GOTO 30
C NOT IN CHRONOLOGICAL SEQUENCE
      WRITE(6,413) RNOSTA,YRSTA,RNOSTO,YRSTO
  413 FORMAT(' SEARCHING RECORDS NUMBERS FROM',
     .         I6,'/',I2,' TO',I6,'/',I2)
      FCOUNT=0
   31 READ(4,END=40) BUFF
      IF (RECYR .LT. YRSTA .OR. RECYR .GT. YRSTO) GOTO 31
C NOT IN THE YEAR RANGE
      IF (RECYR .EQ. YRSTA .AND. RECNO .LT. RNOSTA) GOTO 31
C IN STARTING YEAR BUT BEFORE NUMBER REQUIRED
      IF (RECYR .EQ. YRSTO .AND. RECNO .GT. RNOSTO) GOTO 31
C IN ENDING YEAR BUT AFTER THE RANGE OF INTEREST
      IF (MOD(FCOUNT,53) .EQ. 0) CALL HEAD
C HEAD UP PAGE
      FCOUNT=FCOUNT+1
      WRITE(6,412) NAME,LOCATN,AGNAME,AGNO,RECNO,RECYR,GEOLST,
     .  EAST,NORTH,CATG,GEOL
      GOTO 31
C ROUND AGAIN
C
C NOW AT THE END OF THE RUN
   40 CONTINUE
      WRITE(5,115) FCOUNT
  115 FORMAT(I7,' RECORDS WERE FOUND'/' DO YOU WANT A',
     .      ' PRINTOUT (Y OR N)?')
      READ(5,200) STYPE
      IF (STYPE .NE. 'Y' .OR. FCOUNT .EQ. 0) THEN
       CLOSE(UNIT=6,STATUS='DELETE')
       STOP
C CLOSE FILE AND DELETE IT THEN STOP
      ENDIF
      WRITE(6,414) FFEED
  414 FORMAT(1A1,
     . /' AGENCY TYPE'
     . /'      0  GOVT.        1 PROF.          2 INDIV'//
     . /' CATEGORY'
     . /'      0 GROUNDWATER   1 SLOPE          2 EXPANSIVE'
     . /'                        STABILITY        SOILS'
     . /'      3 FOUNDATION    4 MISCELLANEOUS'
     . /'        INVESTIGATION'//
     . /' GEOLOGY'
     . /'      0 QUATERNARY    1 TERTIARY       2 TRIASSIC'
     . /'                        SEDIMENTS'
     . /'      3 PERMIAN       4 LOWER          5 CAMBRIAN'
     . /'                        PALAEOZOIC'
     . /'      6 PRECAMBRIAN   7 TERTIARY       8 JURASSIC'
     . /'                        BASALT           DOLERITE'
     . /'      9 GRANITE')
      CLOSE(UNIT=6,STATUS='KEEP')
      STOP
      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+ICHAR
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
      INTEGER*2 FFEED
      DATA FFEED/3072/
C USED TO PUT FORM FEEDS IN OUTPUT FILE
      WRITE(6,100) FFEED
  100 FORMAT(1A1/6X,'NAME',7X,'@',11X,'JOB  LOCATION',11X,'@',10X,
     . 'AGENCY',10X,'@ RECORDS  @ INITS @  MAPREF    @',
     . ' CATEGORY @ GEOLOGY'/17X,'@',35X,'@',8X,'NAME',9X,'@',
     . ' NO @',4X,'NO',4X,'@',7X,'@',11X,'@',10X,'@'/
     . 131('@'))
      RETURN
      END
      SUBROUTINE RNOIN(RNO,RYR)
C USE TO READ RECORDS NO AND YEAR AND AVOID HAVING TO MANUALLY
C REPLACE / WITH , IN 1234/76
      INTEGER*2 RNO,RYR
      CHARACTER*1 BUFF(9)
      READ(5,115) BUFF
  115 FORMAT(9A1)
      DO 10 I=1,9
      IF (BUFF(I) .EQ.  '/') BUFF(I)=','
   10 CONTINUE
      REWIND 3
      WRITE(3,115) BUFF
      REWIND 3
      READ(3,*) RNO,RYR
C PUT OUT TO TEMPORARY FILES AND READ BACK IN IN FREE FORMAT
      RETURN
      END
$TITL  FSTJDN.FTN - POINT IN POLYGON
C PROGRAM COLLECTION FASTJORDAN
C    SALOMON,K.B., 1978.  AN EFFICIENT POINT-IN-POLYGON ALGORITHM
C      COMPUTERS AND GEOSCIENCES, V4, NO.  2, P.  173-178
C
C USE BY READING VERTICES OF POLYGON IN ORDER (BUT NOT CLOSING IT)
C AND CALLING PREPLY ONCE BEFORE THE SEARCH IS BEGUN.   THE ACTUAL
C SEARCHING IS DONE BY MEANS OF THE LOGICAL FUNCTION INOUT.
```

```
C
      SUBROUTINE PREPLY(X,Y,NUVERT,YINTVL,INTVLS,SWATH,RSLOPE)
C
C***********************************************************************************************
C  THIS ROUTINE PREPARES THE POLYGON CONSISTING OF THE NUVERT VERTICES
C  (X(I),Y(I)) BY FIRST SORTING THE SEGMENT Y-END POINTS INTO
C  DECREASING ORDER AND FORMING AN INTERVAL FOR EACH CONSECUTIVE PAIR:
C  (YINTVL(I),YINTVL(I+1)), I=1,INTVLS.  THIS IS PERFORMED BY CALLING
C  SORT.
C       THE CODE CONSISTING OF THE DO 100 AND DO 200 LOOPS CONSTRUCTS,
C  FOR EACH INTERVAL I, THE LIST OF SEGMENTS TO BE TESTED BY INOUT.
C  THIS LIST IS PLACED IN THE I-TH ROW OF SWATH.  THE FIRST ENTRY,
C  SWATH(I,1), WILL BE SET TO THE NUMBER OF SEGMENTS IN THE ROW.  NOTE
C  THAT AS YINTVL CONTAINS NO REDUNDANCIES, I.E.  YINTVL(I) IS STRICTLY
C  GREATER THAN YINTVL(I+1), NO HORIZONTAL SEGMENTS WILL BE PLACED IN
C  THE LIST.
C       THE CODE CONSISTING OF THE DO 300 LOOP ESTABLISHES THE
C  RECIPROCAL SLOPE FOR EACH NON-HORIZONTAL SEGMENT.  THIS IS TO BE
C  USED BY INOUT.  FINALLY, THE SEGMENTS WITHIN A ROW OF SWATH ARE
C  OREDERED FROM LEFT-TO-RIGHT.
C
C***********************************************************************************************
C
      INTEGER*2 SWATH(100,25)
      REAL X(100),Y(100),YINTVL(100),RSLOPE(100)
      CALL SORT(Y,NUVERT,YINTVL,INTVLS)
      IF (INTVLS .LE. 0) GOTO 400
      X(NUVERT+1)=X(1)
      Y(NUVERT+1)=Y(1)
      DO 100 I=1,INTVLS
  100 SWATH(I,1)=0
      DO 200 I=1,INTVLS
         DO 200 J=1,NUVERT
         IF (Y(J).GE.YINTVL(I) .AND.  YINTVL(I+1).GE.Y(J+1) .OR.
     *        Y(J+1).GE.YINTVL(I) .AND.  YINTVL(I+1).GE.Y(J))
     *                                              CALL INCLUD(SWATH,I,J)
  200 CONTINUE
      DO 300 I=1,NUVERT
      IF (Y(I).EQ.Y(I+1)) GOTO 300
      RSLOPE(I)=(X(I+1)-X(I))/(Y(I+1)-Y(I))
  300 CONTINUE
      CALL ORDER(X,Y,YINTVL,INTVLS,SWATH,RSLOPE)
      RETURN
  400 WRITE(7,401)
  401 FORMAT(' ******* PREP OF POLYGON ABORTED SINCE NO INTERVALS',
     * ' CONSTRUCTED')
      STOP
      END
      SUBROUTINE SORT(Y,NUVERT,YINTVL,INTVLS)
C
C**********************************************************************************************
C
C  ROUTINE ESTABLISHES THE INTERVALS OF THE Y-AXIS DEFINED BY THE
C  ENDPOINTS OF THE POLYGON'S SEGMENTS.  THE DO 100 LOOP INITIALLISES
C  YSORT FROM THE SEGMENT Y-END POINTS.  THE DO 200 LOOPS SORT YSORT
C INTO DESCENDING ORDER.  THE DO 300 LOOP ELIMINATES REDUNDANCIES IN
C  YSORT AND PLACES IRREDUNDANT SORTED Y'S INTO YINTVL.  IT ALSO SETS
C  INTVLS TO THE TRUE NUMBER OF Y INTERVALS.  JUST PRIOR TO RETURNING
C  A FINAL INTERVAL EXTENDING TO '-INFINITY' IS ESTABLISHED.
C
```

```
C*************************************************************************
C
      REAL Y(100),YINTVL(100),YSORT(100)
      INTEGER*2 UPPER
      DO 100 I=1,NUVERT
  100 YSORT(I)=Y(I)
      UPPER=NUVERT-1
      DO 200 I=1,UPPER
         IPLS1=I+1
         DO 200 J=IPLS1,NUVERT
         IF (YSORT(I).GE.YSORT(J)) GOTO 200
            TEMP=YSORT(I)
            YSORT(I)=YSORT(J)
            YSORT(J)=TEMP
  200 CONTINUE
      YINTVL(1)=YSORT(1)
      INTVLS=0
      DO 300 I=1,UPPER
         IF (YSORT(I).EQ.YSORT(I+1)) GOTO 300
         INTVLS=INTVLS+1
         YINTVL(INTVLS+1)=YSORT(I+1)
  300 CONTINUE
      YINTVL(INTVLS+2)=-1.0E75
      RETURN
      END
      SUBROUTINE INCLUD(SWATH,I,J)
C
C*************************************************************************
C  ROUTINE PLACES THE J-TH POLYGON SEGMENT INTO THE NEXT AVAILABLE
C  LOCATION IN ROW I OF SWATH.
C
C*************************************************************************
C
      INTEGER*2 SWATH(100,25),POINTR
      SWATH(I,1)=SWATH(I,1)+1
      POINTR=SWATH(I,1)
      SWATH(I,POINTR+1)=J
      RETURN
      END
      SUBROUTINE ORDER(X,Y,YINTVL,INTVLS,SWATH,RSLOPE)
C
C*************************************************************************
C
C  FOR EACH INTERVAL, A HORIZONTAL LINE IS PASSED THROUGH THE MIDDLE
C  (YMID) OF THE INTERVAL.  THE DO 100 LOOP PLACES THE X-INTERSECTION
C  OF EACH SEGMENT IN THIS SWATH SO THAT THESE INTERSECTIONS OCCUR
C  FROM LEFT-TO-RIGHT.
C
C*************************************************************************
C
      REAL X(100),Y(100),YINTVL(100),RSLOPE(100),XINTSC(25)
      INTEGER*2 SWATH(100,25),POINTR,SEGNO,UPPER
      LOGICAL VERTSG
      DO 200 INTVAL=1,INTVLS
         NMBSEG=SWATH(INTVAL,1)
         YMID=(YINTVL(INTVAL)+YINTVL(INTVAL+1))/2.0
         DO 100 POINTR=1,NMBSEG
            SEGNO=SWATH(INTVAL,POINTR+1)
            VERTSG=ABS(X(SEGNO+1)-X(SEGNO)) .LT.  1.0E-5
            IF (VERTSG) XINTSC(POINTR)=X(SEGNO)
            IF (.NOT. VERTSG) XINTSC(POINTR)=X(SEGNO)+
     *                                RSLOPE(SEGNO)*(YMID-Y(SEGNO))
 100     CONTINUE
         IF (NMBSEG.LT.2 .OR. MOD(NMBSEG,2).NE.0) GOTO 300
         UPPER=NMBSEG-1
         DO 200 I=1,UPPER
            IPLS1=I+1
            DO 200 J=IPLS1,NMBSEG
               IF (XINTSC(I).LE.XINTSC(J)) GOTO 200
               TEMP=XINTSC(I)
               XINTSC(I)=XINTSC(J)
               XINTSC(J)=TEMP
               ITEMP=SWATH(INTVAL,I+1)
               SWATH(INTVAL,I+1)=SWATH(INTVAL,J+1)
               SWATH(INTVAL,J+1)=ITEMP
 200  CONTINUE
      RETURN
 300  WRITE(7,301) INTVAL
 301  FORMAT(' ** PREP OF POLYGON ABORTED. INTERVAL ',I5/
     * ' HAS EITHER LESS THAN TWO SEGMENTS OR AN ODD NUMBER OF THEM')
      STOP
      END
      LOGICAL FUNCTION INOUT(XP,YP,X,Y,YINTVL,INTVLS,SWATH,RSLOPE)
C
C************************************************************************
C
C  THE FOUR LINES ENCLOSED IN DASHES DETERMINE THE INTERVAL CONTAINING
C  YP. THE DO 400 LOOP CONTINUES UNTIL THE FIRST SEGMENT WITHIN THE
C  INTERVAL FALL TO THE LEFT OF (XP,YP). IN THIS EVENT, INOUT IS SET
C  .TRUE. IFF AN EVEN NUMBER OF SEGMENTS HAS BEEN TESTED.
C
C******************************************************************************
C
      REAL X(100),Y(100),YINTVL(100),RSLOPE(100)
      INTEGER*2 SWATH(100,25),SEGNO
      INOUT=.FALSE.
C-------------------------------------------------------
      INTVAL=0
 100  INTVAL=INTVAL+1
      IF (YINTVL(INTVAL) .GT. YP) GOTO 100
      INTVAL=INTVAL-1
C--------------------------------------------------------
 300  IF (INTVAL.LT.1 .OR. INTVAL.GT.INTVLS) RETURN
      NMBSEG=SWATH(INTVAL,1)+1
      DO 400 I=2,NMBSEG
         SEGNO=SWATH(INTVAL,I)
         IF(XP-X(SEGNO).LE.(YP-Y(SEGNO))*RSLOPE(SEGNO)) GOTO 500
 400  CONTINUE
      RETURN
 500  INOUT=MOD(I,2) .EQ. 1
      RETURN
      END
```