UR1986_40

1986/40. ISOLIBRARY - an isotope data index.

A. L. Telfer

Abstract

The ISOLIBRARY program suite is designed to provide a data base for the storage of isotope data. Programs are provided to add to, sort and search the data base. A sample input form is included.

USING THE PROGRAMS

The programs are run on the Geological Survey Perkin-Elmer mincomputer.

Adding data

Data are recorded on a standard input form (Appendix 1) before entry. Data input is started by typing ISOADD, then the data are entered from the standard input forms in response to the prompts of the program. Input is terminated by entering 0 (zero) as the analysis number.

A proof sheet is printed in the computer room, and the data are saved in in file ISOTEMP.TMP. This file should be corrected using the editing facilities (see Appendix 2) before continuing.

The corrected temporary file ISOTEMP.TMP is added to the end of the existing data base using the ISOMERGE command. The data base should be sorted periodically using the ISOSORT command to maintain the entries in analysis number order.

Corrections to the main data base (ISOIND.DAT) would normally be made by deleting entries using the system editor in hexadecimal mode and then putting the corrected entries in using ISOADD.

Data base searching

The data base is searched by typing ISOSRCH. An entry will be retrieved if each of the specified search options is found in the entry.

The following search options may be specified;

(1) DOM SAMPLE NUMBER: one or more sample numbers on a single line, each terminated by a /, or the first and last numbers of a range of numbers, separated by a - and terminated by a /. The entire six-digit number must be used.

(2) ISOTOPE ANALYSIS NUMBER: one or more analysis numbers, each terminated by a /. The entire number must be used.

(3) ISOTOPE ELEMENT: one or more isotope elements, each terminated by a /. These are one letter options.

(4) SAMPLE TYPE: one or more sample types, each terminated by a /. Any contraction of the sample type entry may be the subject of the search.

(5) ROCK TYPE: one or more rock types, each terminated by a /. Any contraction of the rock type entry may be the subject of the search.

(6) COLLECTOR: one or more collectors, each terminated by a /. Any contraction of the collectors entry may be the subject of the search.

(7) SEARCH AN AREA: the east-north coordinates of diagonally opposite corners of a rectangle, one corner per line.

PROGRAM FUNCTIONS

ISOADD (Appendices 3, 4)

This program prompts the user for data and writes it to a temporary file ISOTEMP.TMP. This file is corrected using the computers editing facilities (see Appendix 2 for the most useful commands).

ISOMERGE (Appendices 5, 6)

The corrected data in ISOTEMP.TMP is converted to hexadecimal format. The data base is copied to a temporary work file and the corrected data in ISOTEMP.TMP is converted to hexadecimal format and added to the end of the work file. The combined file is then copied back to the original data base file.

ISOSORT (Appendices 7, 8)

The data are sorted into ascending DOM sample 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.

ISOSRCH (Appendices 9, 10)

The program prompts the user with the available search options (see *Data-base searching* above), takes the values of the options selected, and searches each record for the occurrence of every selected option.

[3 July 1986]

APPENDIX 1

ISOLIBRARY data base of isotope analysis results

ANALYSIS NUMBER: _ _ _ _ _ (5)

SAMPLE NUMBER: _ _ _ _ _ _ (6)

SAMPLE TYPE: _ _ _ (3)

ROCK TYPE: _ _ _ _ _ _ _ _ _ _ (10)

AMG COORDINATES: _ _ _ _ _ _ 5 _ _ _ _ _ _ (13)

ELEVATION: _ _ _ _ _ (5)

COLLECTOR: _ _ _ (3)

ANALYST: _ _ _ (3)

ELEMENT: _ (1)

DELTA: _ _ _ _ _ _ _ (7)

COMMENT: _
_ _(40)

ISOLIBRARY data base of isotope analysis results

ANALYSIS NUMBER: _ _ _ _ _ (5)

SAMPLE NUMBER: _ _ _ _ _ _ (6)

SAMPLE TYPE: _ _ _ (3)

ROCK TYPE: _ _ _ _ _ _ _ _ _ _ (10)

AMG COORDINATES: _ _ _ _ _ _ 5 _ _ _ _ _ _ (13)

ELEVATION: _ _ _ _ _ (5)

COLLECTOR: _ _ _ (3)

ANALYST: _ _ _ (3)

ELEMENT: _ (1)

DELTA: _ _ _ _ _ _ _ (7)

COMMENT: _
_ _(40)

APPENDIX 2

Editing Facilities

To examine and change a file, type 'EDIT file name' (in this case, 'filename' = ISOTEMP.TMP).

Attached is a short list of the most useful editing commands. A full summary of commands is found in the 'EDIT User Guide' in the computer room.

ALTER COMMAND

The ALTER command modifies entire lines of text.

Format:

```
                    ( ⍽          )          NOTE:
                    ( \          )          The characters used to
      -(    cr    ) ( @          )          alter the line can be used
ALTER ( line no ) ( %          )          more than once on the
                    ( %/string/ )          same line.
                    ( x          )
```

cr — entering a carriage return immediately following the the ALTER command readies the current line to be altered.

line no — specifies the line number to be altered.

⍽ — represents a space. Spaces have no effect on the current line. The cursor is moved by depressing the space bar.

\ — is the backslash character that replaces the current character with a space.

@ — is the at sign character that deletes the current character and compresses the rest of the line.

% — is the percent sign character that inserts a single space before the percent sign.

%/string/ — is the percent sign followed by a slash, a character string, and a slash, that inserts the character string before the percent sign.

x — indicates any character other than a blank, \, @, and %. The character replaces the current character in that position.

Note: This decription of the ALTER command applies only when in Edit's NOBIOC mode. In the BIOC mode the ALTER commands functions in a different way, using the CONTROL key to position the cursor and to operate the delete and insert functions. See the 'EDIT User Guide' for details.

Functional Details:

After the ALTER command is entered, Edit responds by printing the specified line as it currently exists. The line number is then printed again. Changes to the original line now can be entered in the corresponding character positions. The alter line can be terminated at any character position. All remaining characters in the line are retained.

A carriage return in the first character position will terminate ALTER.

As soon as 80 characters are typed on a terminal, the line automatically wraps around to the next line. To alter the second part of a line that exceeds 80 characters, space through the first part of the line in order to access the second part.

Examples:

```
>T1-3
    1-  This text is part of a new document
    2   The first thing we must do is revise it
    3   This is an example of a long line that wraps
        around when it is displayed.

>AL 1
    1   This text is part of a new document
    1   >                        old
    1   This text is part of a old document
    1   >                       n
    1   This text is part of anold document
    1   >                        %
    1   This text is part of an old document
        >             %/one /
    1   This text is one part of an old document
    1   >

>AL 2
    2   The first thing we must do is revise it
    2   >                too@@@@@
    2   The first thing toodo is revise it
    2   >                  \
    2   The first thing to do is revise it
    2   >

>AL 3
    3   This is an example of a long line that wraps
        around when it is displayed.
    3   >
        >                   print@@
    3   This is an example of a long line that wraps
        around when it is printed.
    3   >
```

DELETE COMMAND

The DELETE command deletes data from a file.

Format:

DELETE [<line no 1>][-<line no 2>]

Parameters:

line no 1 is the line number, or first line number of a range of lines to be deleted.

line no 2 is the ending line number of a range of data to be deleted.

Examples:

```
>DELETE 10-20
```

Delete lines 10 through 20.

```
>DELETE
```

Delete the current line.

DONE COMMAND

The DONE command saves the current file and ends the editing session.

Format:

DONE

SCREEN COMMAND

The SCREEN command displays a full screen of data starting, ending, or centred at the current line.

Format:

```
        ( E )
SCREEN  ( C )
        ( S )
```

Parameters:

E displays a full screen of data ending at the current line. If this parameter is omitted, the default is S.

C displays a full screen of data centered at the current line. If this parameter is omitted, the default is S.

S displays a full screen of data starting at the current line.

TOP COMMAND

The TOP command moves the current line pointer to the first line of the file.

Format:

```
TOP
```

Examples:

```
>TY 1-
    1    The TOP command
    2    moves the current line pointer
    3    to the first line of the file
>TOP
    1    The TOP command
```

TYPE COMMAND

The TYPE command displays lines of data to the list device.

Format:

```
TYPE  [<line no 1>][-<line no 2>]
```

Parameters:

line no 1 — is the line or first line number of a range of lines to be displayed to the list device.

line no 2 — is the ending line number of the range to be displayed to the list device.

Examples:

```
>TYPE 10-20
```

Display lines 10 through 20 of the current file.

```
>TYPE 8
```

Display line 8 of the current file.

```
>TYPE 20-
```

Display lines 20 to the last line of the current file.
file.

APPENDIX 3

```
*ISOADD.CSS
*FOR RUNNING ISOADD AND PRINTING A PROOF SHEET
$IFX ISOADD.TMP; $WRITE  **CHECK LAST ISOMERGE**; $EXIT; $ENDC
PRE ETM; AL ISOTEMP.TMP,IN,150
*SET UP THE OUTPUT FILE
L ISOADD,3; AS 6,ISOTEMP.TMP; *LOAD PROGRAMME AND ASSIGN LU 6
ST
*PRINT ISOTEMP.TMP
$WRITE  FILE IS ISOTEMP.TMP; ENA ETM; $EXIT
```

APPENDIX 4

```
$TITL ISOADD.FTN - ADD ISOTOPE ANALYSES TO THE ISOTOPE INDEX
C
C TAKES KEYBOARD INPUT AND PREPARES A PROOF SHEET
C
C
        INTEGER*4 ANALNO
        CHARACTER*40 COMMENT
        CHARACTER*10 ROCTYPE
        CHARACTER*7 DELTA
        CHARACTER*6 SAMPLNO
        CHARACTER*5 ELEVN
        CHARACTER*3 SAMTYPE,COLLECT,ANALYST
        INTEGER*4 NORTH,EAST
        CHARACTER*1 ELEMENT
C
C          INPUT IS ISOTOPE ANALYSIS NUMBER, SAMPLE NUMBER, TYPE,
C          ROCK TYPE, AMG COORDS.,ELEVATION,COLLECTOR,ANALYST,ELEMENT,
C      -   DELTA AND COMMENTS
C
      OPEN (UNIT=5,FILE='CON:')
C
C          OPEN THE CONSOLE FOR INPUT. ASSUME THAT OUTPUT FILE 6 HAS
C          BEEN SET UP ELSEWHERE
C
   20 WRITE(5,100)
  100 FORMAT(' ANALYSIS NUMBER (up to 5 numbers) TYPE *0* TO END')
      READ(5,*) ANALNO
      IF (ANALNO .EQ. 0) GOTO 50
C
C          END OF BATCH SO CLOSE OFF
C
      WRITE(5,101)
  101 FORMAT(' SAMPLE NUMBER (up to 6 numbers)')
      READ(5,201) SAMPLNO
  201 FORMAT(A6)
      WRITE(5,102)
  102 FORMAT(' SAMPLE TYPE eg.CPY,WR (up to 3 alphanumerics)')
      READ(5,202) SAMTYPE
  202 FORMAT(A3)
      WRITE(5,103)
  103 FORMAT(' ROCK TYPE (up to 10 alphanumerics)')
      READ(5,203) ROCTYPE
  203 FORMAT(A10)
      WRITE(5,104)
  104 FORMAT(' AMG COORDINATES (easting = 6 numbers, northing = 7)')
      READ(5,204) EAST,NORTH
  204 FORMAT(I6,I7)
      WRITE(5,105)
  105 FORMAT(' ELEVATION (up to 5 alphanumerics)')
      READ(5,205) ELEVN
  205 FORMAT(A5)
      WRITE(5,106)
  106 FORMAT(' COLLECTOR (up to 3 alphanumerics)')
      READ(5,202) COLLECT
      WRITE(5,107)
  107 FORMAT(' ANALYST (up to 3 alphanumerics)')
```

```
      READ(5,202) ANALYST
      WRITE(5,108)
  108 FORMAT(' ELEMENT (up to 1 alphanumeric)')
      READ(5,208) ELEMENT
  208 FORMAT(A1)
      WRITE(5,109)
  109 FORMAT(' DELTA (up to 7 alphanumerics)')
      READ(5,209) DELTA
  209 FORMAT(A7)
      WRITE(5,110)
  110 FORMAT(' COMMENT (up to 40 alphanumerics)')
      READ(5,210) COMMENT
  210 FORMAT(A40)
C
C         READ ONE DATA SHEET AND WRITE TO THE OUTPUT FILE
C
       WRITE(6,300) ANALNO,SAMPLNO,SAMTYPE,ROCTYPE,EAST,NORTH,
     .              ELEVN,COLLECT,ANALYST,ELEMENT,DELTA,COMMENT
  300 FORMAT(I5,1X,A6,1X,A3,1X,A10,1X,I6,1X,I7,1X,A5,1X,A3,1X,A3,1X,
     .       A1,1X,A7,1X,A40/)
C
C         WRITE WITH A BLANK BETWEEN FIELDS AND A BLANK LINE AFTER
C         EACH ENTRY
C
      GOTO 20
C
C         BACK AROUND, OTHERWISE STOP
C
   50 CLOSE(UNIT=6,STATUS='KEEP')
      STOP
      END
```

APPENDIX 5

```
*ISOMERGE.CSS
* FOR MERGING CORRECTED FILE WITH MAIN FILE
PRE ETM; XDE MRGTMP.TMP; *DELETE SCRATCH FILE
AL MRGTMP.TMP,IN,90; L ISOMERGE,10; * AL SCRATCH FILE AND LOAD PROG
REP ISOTEMP.TMP,FF00; AS 6,ISOTEMP.TMP,ERO; AS 4,MRGTMP.TMP; ST
$IFNE 0; $WRITE TRANSLATE ERROR; ENA ETM;  $EXIT; $ENDC
REP MRGTMP.TMP,FF00
$BUILD COPY.CMD
IN ISOIND.DAT
AL SYSF:TEMP.DAT,IN,90/6/5
OUT SYSF: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 ISOIND.DAT,0; REP SYSF:TEMP.DAT,FF00
DE COPY.CMD,ISOIND.DAT
$BUILD COPY.CMD
IN SYSF:TEMP.DAT
AL ISOIND.DAT,IN,90/10/3
OUT ISOIND.DAT
COPY *,*
REW I
REW 0
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 ISOIND.DAT,FF00; REP SYSF:TEMP.DAT,0; REP ISOTEMP.TMP,0
REP MRGTMP.TMP,0
DE COPY.CMD,SYSF:TEMP.DAT,MRGTMP.TMP,ISOTEMP.TMP
ENA ETM
$EXIT
```

APPENDIX 6

```
$TITL ISOMERGE.FTN - CONVERT PROOF FILE TO MASTER FILE FORMAT
      INTEGER*4 ANALNO
      CHARACTER*40 COMMENT
      CHARACTER*10 ROCTYPE
      CHARACTER*7 DELTA
      CHARACTER*6 SAMPLNO
      CHARACTER*5 ELEVN
      CHARACTER*3 SAMTYPE,COLLECT,ANALYST
      INTEGER*4 NORTH,EAST
      CHARACTER*1 ELEMENT
C
C       DECLARATION OF INPUT AND OUTPUT VARIABLES
C
      CHARACTER*1 BUFF(90)
      EQUIVALENCE (BUFF(1),COMMENT),(BUFF(41),ROCTYPE),
     .            (BUFF(51),DELTA),(BUFF(58),SAMPLNO),
     .            (BUFF(64),ELEVN),(BUFF(69),EAST),(BUFF(73),ANALNO)
     .           ,(BUFF(77),NORTH),(BUFF(81),SAMTYPE),
     .            (BUFF(84),ANALYST),(BUFF(87),COLLECT),
     .            (BUFF(90),ELEMENT)
C
C       A BINARY BUFFERED ARRAY FOR FASTER DATA-BASE ACCESS
C
      OPEN(UNIT=4,FORM='BINARY',RECL=93)
C
C       OPEN THE OUTPUT FILE FOR BINARY OPERATIONS
C
C
C       READ ISOTEMP.TMP (THE ISOADD DATA FILE) AND REWRITE AS A
C       BINARY FILE.
C
   10 READ(6,300,END=29) ANALNO,SAMPLNO,SAMTYPE,ROCTYPE,EAST,NORTH,
     .            ELEVN,COLLECT,ANALYST,ELEMENT,DELTA,COMMENT
  300 FORMAT(I5,1X,A6,1X,A3,1X,A10,1X,I6,1X,I7,1X,A5,1X,A3,1X,A3,1X,
     .       A1,1X,A7,1X,A40/)
      WRITE(4) BUFF
C
C       BINARY WRITE TO OUTPUT FILE
C
      GOTO 10
C
C       BACK AROUND
C
C
C       WRITE TO THE BINARY FILE, CLOSE ALL FILES AND END.
C
   29 CLOSE(UNIT=4,STATUS='KEEP')
      CLOSE(UNIT=6,STATUS='KEEP')
      END
```

APPENDIX 7

```
*ISOSORT.CSS
* FOR SORTING ISOTOPE DATA LIBRARY FILES INTO REFERENCE NO. ORDER
PRE ETM
L ISOSORT,10; AS 4,ISOIND.DAT,ERO; AL SYSF:TEMP.DAT,IN,90/5/2
AS 6,SYSF:TEMP.DAT; ST
$IFNE 0; $WRITE SORT ERROR; ENA ETM; $EXIT; $ENDC
$BUILD COPY.CMD
IN SYSF:TEMP.DAT
AL ISOIND.DAT,IN,90/10/5
OUT ISOIND.DAT
COPY *,*
REW I
REW O
VERIFY *,*
END
$ENDB
REP SYSF:TEMP.DAT,FF00; REP ISOIND.DAT,0; DE ISOIND.DAT
L COPY32,50; ST ,COM=COPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0; $WRITE SORT-COPY ERROR; ENA ETM; $EXIT; $ENDC
REP ISOIND.DAT,FF00; REP SYSF:TEMP.DAT,0; DE SYSF:TEMP.DAT,COPY.CMD
ENA ETM; $EXIT
```

APPENDIX 8

```
$TITL ISOSORT.FTN - SORT THE ISOTOPE DATA FILES INTO ASCENDING REF NO.
C
        INTEGER*4 ANALNO
        CHARACTER*40 COMMENT
        CHARACTER*10 ROCTYPE
        CHARACTER*7 DELTA
        CHARACTER*6 SAMPLNO
        CHARACTER*5 ELEVN
        CHARACTER*3 SAMTYPE,COLLECT,ANALYST
        INTEGER*4 NORTH,EAST
        CHARACTER*1 ELEMENT
C
C         DECLARATION OF INPUT AND OUTPUT VARIABLES
C
        CHARACTER*1 BUFF(90)
        EQUIVALENCE (BUFF(1),COMMENT),(BUFF(41),ROCTYPE),
     .              (BUFF(51),DELTA),(BUFF(58),SAMPLNO),
     .              (BUFF(64),ELEVN),(BUFF(69),EAST),(BUFF(73),ANALNO)
     .             ,(BUFF(77),NORTH),(BUFF(81),SAMTYPE),
     .              (BUFF(84),ANALYST),(BUFF(87),COLLECT),
     .              (BUFF(90),ELEMENT)
C
C         A BINARY BUFFERED ARRAY FOR FASTER DATA-BASE ACCESS
C
       INTEGER*2 INDEX(8000),IA(8000)
C
C         THESE INDICES ARE FOR THE SORT: UP TO 8000 ENTRIES ALLOWED
C
      OPEN(UNIT=4,FORM='BINARY',RECL=90)
C
C         OPEN THE INPUT FILE FOR RANDOM AND SEQUENTIAL ACCESS
C
       REWIND 4
       IREC=0
C
C         COUNTS NUMBER OF RECORDS
C
  10   READ(4,END=20) BUFF
       IREC=IREC+1
       IF(IREC .GT. 8000) STOP 'MORE THAN 8000 ENTRIES'
       INDEX(IREC)=IREC
C
C         FILL SORT INDEX
C
       IA(IREC)=ANALNO
       GO TO 10
C
C         READ THE NEXT FILE
C
  20   CONTINUE
C
C        AND NOW TO BUSINESS
C
       IF(IREC .EQ. 0) STOP 'NO RECORDS IN DATA FILE'
       CALL BUBSTI(INDEX,IA,IREC)
C
```

```
C       SORT INTO ASCENDING NUMBER - I.E. CHRON ORDER
C
      OPEN(UNIT=6,FORM='BINARY',RECL=90)
C
C       OPEN OUTPUT FILE
C
      DO 30 I=1,IREC
      READ(4,REC=INDEX(I)) BUFF
      WRITE(6) BUFF
C
C       COPY TO NEW ORDER
C
   30 CONTINUE
C
C       CLOSE FILES AND END
C
      CLOSE(UNIT=6,STATUS='KEEP')
      CLOSE(UNIT=4,STATUS='KEEP')
      END
C
C***********************************************
C
      SUBROUTINE BUBSTI(IR,IA,N)
      INTEGER*2 IA(N)
      INTEGER*2 IR(N)
      LOGICAL NSWAP
      IF (N .LE. 1) RETURN
C
C       NOTHING TO SORT
C
      NM1=N-1
      DO 30 J=1,NM1
      NSWAP=.TRUE.
      IRI=IR(1)
      DO 40 I=1,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 9

```
*ISOSRCH.CSS - SEARCH THE ISOTOPE INDEX FILE
L ISOSRCH,10; AS 4,ISOIND.DAT,ERO; XAL SYSF:ISOIND.TMP,IN,150/3/2
AS 6,SYSF:ISOIND.TMP; REW 6; AS 5,CON:; TEMPFILE 3,IN,80; ST
$IFX SYSF:ISOIND.TMP; PRI SYSF:ISOIND.TMP,DEL; $ENDC; $EXIT
```

APPENDIX 10

```
$TITL  ISOSRCH.FTN - SEARCH ISOTOPE INDEX FILE
C
C
        INTEGER*4 ANALNO
        CHARACTER*40 COMMENT
        CHARACTER*10 ROCTYPE
        CHARACTER*7 DELTA
        CHARACTER*6 SAMPLNO
        CHARACTER*5 ELEVN
        CHARACTER*3 SAMTYPE,COLLECT,ANALYST
        INTEGER*4 NORTH,EAST
        CHARACTER*1 ELEMENT
C
C         DECLARATION OF INPUT AND OUTPUT VARIABLES
C
        CHARACTER*1 BUFF(90)
        EQUIVALENCE (BUFF(1),COMMENT),(BUFF(41),ROCTYPE),
     .              (BUFF(51),DELTA),(BUFF(58),SAMPLNO),
     .              (BUFF(64),ELEVN),(BUFF(69),EAST),(BUFF(73),ANALNO)
     .             ,(BUFF(77),NORTH),(BUFF(81),SAMTYPE),
     .              (BUFF(84),ANALYST),(BUFF(87),COLLECT),
     .              (BUFF(90),ELEMENT)
C
C         A BINARY BUFFERED ARRAY FOR FASTER DATA-BASE ACCESS
C
      CHARACTER*80 SANALNO,SELEMENT,SROCTYPE,SCOLLECT,SSAMPLNO
      CHARACTER*80 SSAMTYPE
C
C         STRINGS FOR INPUT DATA
C
      INTEGER*2 FFEED
      DATA FFEED/3072/
C
C        THE FORM FEED CHARACTER IN A1
C
      CHARACTER*1 ALINE(66)
C
C        SCREEN OUTPUT VARIABLE
C
      CHARACTER*6 CSAMPLNO(5000)
      INTEGER*4 IANALNO(40)
C
C        ARRAYS HOLDING THE SAMPLE NUMBERS AND THE INTEGER FORMS OF
C        THE INPUT ANALYSIS NUMBERS
C
      LOGICAL NOTIN
      INTEGER*2 FCOUNT,LCOUNT,CCOUNT
C
C        END OF DECLARATIONS. NOW WE CAN START THE DATA SEARCH.
C
      OPEN(UNIT=4,FORM='BINARY',RECL=90)
C
C        OPEN THE INPUT FILE
C
C
C-----------------------------------------------------------------
```

```
C
C
C        ASK FOR THE SEARCH OPTIONS
C
      WRITE(5,500)
  500 FORMAT(' SEARCHING ISOTOPE DATA INDEX',/)
      WRITE(6,600)
  600 FORMAT(' SEARCH OPTIONS ARE:')
      WRITE(5,501)
  501 FORMAT(' For each option enter either a blank line or'/
     . ' the desired values for the option,'
     . ,' each TERMINATED BY A /'/' eg. 2234/4535/7658/'/)
      WRITE(5,503)
  503 FORMAT(' DOMines SAMPLE NUMBER,  .'/,
     .          ' OR RANGE OF NUMBERS SEPARATED BY A -')
      READ(5,202) SSAMPLNO
      IF (SSAMPLNO(1:1) .NE. ' ') WRITE(6,603) SSAMPLNO
  603 FORMAT(' DOMines SAMPLE NUMBER (OR RANGE)',10X,A80)
C
C       DETERMINE IF A RANGE OF SAMPLE NUMBERS ARE INPUT BY SEARCHING
C       FOR A - BETWEEN DELIMITERS (/). THEN USE A DO-LOOP TO FILL
C       AN ARRAY WITH THOSE SAMPLE NUMBERS WITHIN THE STATED RANGE
C
      CCOUNT=0
      JLAST=1
      KLAST=1
   31 KCHAR=INDEX(SSAMPLNO(KLAST:80),'/')
      IF(KCHAR .EQ. 0) GOTO 30
C
C         FIND STRING TERMINATOR OR CONTINUE
C
      JCHAR=INDEX(SSAMPLNO(JLAST:80),'-')
      IF(JCHAR .NE. 0 .AND. JCHAR .LT. (KCHAR+KLAST-1)) THEN
C
C         IF A RANGE IS RECOGNISED WITHIN THE APPROPRIATE
C         TERMINATOR, WRITE THE RANGE NUMBERS TO AN ARRAY.
C
         NSAMPLNO1=CTOI(SSAMPLNO(JLAST:JLAST+JCHAR-1),JTEMP)
         NSAMPLNO2=CTOI(SSAMPLNO(JLAST+JCHAR:JLAST+JCHAR+5),JTEMP)
         DO 33 J=NSAMPLNO1,NSAMPLNO2
               CCOUNT=CCOUNT+1
               CSAMPLNO(CCOUNT)=ITOC(J,JTEMP)
               IF(CCOUNT .GT. 4000) STOP 'MORE THAN 4000 IN RANGE'
   33    CONTINUE
         KLAST=KLAST+KCHAR
         JLAST=JLAST+KCHAR
         GOTO 31
      ENDIF
      CCOUNT=CCOUNT+1
      CSAMPLNO(CCOUNT)=SSAMPLNO(KLAST:KLAST+KCHAR-1)
      KLAST=KLAST+KCHAR
C
C         PLACE ANY SINGLE NUMBERS INTO THE ARRAY
C
      GOTO 31
   30 CONTINUE
C
C         AND SO TO THE REST OF THE INPUT
```

```
C
      WRITE(5,502)
  502 FORMAT(' ISOTOPE ANALYSIS NUMBER?')
      READ(5,202) SANALNO
  202 FORMAT(A80)
      IF (SANALNO(1:5) .EQ. '      ') GOTO 20
C
C         TEST TO SEE IF ANALYSIS NUMBER SEARCH REQUIRED.
C         IF SEARCH REQUIRED, CONVERT THE INPUT ANALYSIS NUMBERS FROM
C         CHARACTER TO INTEGER FORMAT BY LOCATING THE /'S AND
C         THEN CONVERTING EACH NUMBER IN TURN.
C
      NANALNO=0
      WRITE(6,602) SANALNO
  602 FORMAT(' ISOTOPE ANALYSIS NUMBER',10X,A80)
      ILAST=1
   21 ICHAR=INDEX(SANALNO(ILAST:80),'/')
      IF(ICHAR .EQ. 0) GOTO 20
      NANALNO=NANALNO+1
      IANALNO(NANALNO)=CTOI(SANALNO(ILAST:ILAST+ICHAR-1),ITEMP)
C
C         CONVERTED TO AN INTEGER
C
      ILAST=ILAST+ICHAR
      IF (ILAST .NE. 79) GOTO 21
C
C         GO BACK AND CONVERT THE NEXT ANALYSIS NO. TO INTEGER FORMAT
C
C         NOW THE REST OF THE SEARCH OPTIONS
C
   20 CONTINUE
      WRITE(5,504)
  504 FORMAT(' ISOTOPE ELEMENT?')
      READ(5,202) SELEMENT
      IF (SELEMENT(1:1) .NE. ' ') WRITE(6,604) SELEMENT
  604 FORMAT(' ISOTOPE ELEMENT',10X,A80)
      WRITE(5,517)
  517 FORMAT(' SAMPLE TYPE?')
      READ(5,202) SSAMTYPE
      IF (SSAMTYPE(1:1) .NE. ' ') WRITE(6,617) SSAMTYPE
  617 FORMAT(' SAMPLE TYPE',10X,A80)
      WRITE(5,505)
  505 FORMAT(' ROCK TYPE?')
      READ(5,202) SROCTYPE
      IF (SROCTYPE(1:1) .NE. ' ') WRITE(6,605) SROCTYPE
  605 FORMAT(' ROCK TYPE',10X,A80)
      WRITE(5,506)
  506 FORMAT(' COLLECTOR? (eg. GRG)')
      READ(5,202) SCOLLECT
      IF (SCOLLECT(1:1) .NE. ' ') WRITE(6,606) SCOLLECT
  606 FORMAT(' COLLECTOR',10X,A80)
      WRITE(5,507)
  507 FORMAT(' SEARCH FOR SITES WITHIN A SPECIFIED AREA (Y OR N)?')
      READ(5,200) SPRINT
  200 FORMAT(A1)
      IF (SPRINT .EQ. 'Y') GOTO 23
C
C        WANT TO READ COORDS
```

```
C
      GOTO 26
C
C        DON'T WANT TO SEARCH FOR SITES IN AN AREA
C
   23 CONTINUE
      WRITE(5,508)
  508 FORMAT(' ENTER COORDS OF TWO DIAGONALLY OPPOSITE CORNERS OF'
     .  'THE RECTANGLE'/' ONE PER LINE IN THE FORM EASTNORTH')
      IEMAX=-999999999
      IEMIN=-IEMAX
      INMAX=IEMAX
      INMIN=IEMIN
C
C        USE THESE TO STORE THE RANGE OF THE AREA
C
      DO 24 I=1,2
      READ(5,509) IE,IN
  509 FORMAT(I6,I7)
C
C         GET INTO 100'S OF METRES EAST AND NORTH
C
      IEMAX=MAX0(IE,IEMAX)
      IEMIN=MIN0(IE,IEMIN)
      INMAX=MAX0(IN,INMAX)
      INMIN=MIN0(IN,INMIN)
C
C        FIND THE RANGE
C
   24 CONTINUE
      WRITE(6,609) IEMIN,IEMAX,INMIN,INMAX
  609 FORMAT(' SEARCH RECTANGLE FROM ',I6,'ME TO ',I6,'ME',/,
     .       'AND FROM ',I7,'MN TO ',I7,'MN')
   26 CONTINUE
C
C
C       END OF OPTION INPUT
C
C
C-------------------------------------------------------------------------
C
C         NOW FOR THE SEARCH
C
      FCOUNT=0
      LCOUNT=9
C
C         COUNTER FOR NUMBER OF FOUND ENTRIES, AND NUMBER OF LINES
C         WRITTEN TO THE OUTPUT FILE.
C
      WRITE(6,610)
  610 FORMAT(/' ANALYSIS',9X,'|ROCK TYPE | COORDINATES  |   COLLECTOR'
     .       ,' | DELTA |',40X,/,' NUMBER|  SAMPLE  |',10X,'|EASTING'
     .        ,6X,'ELEVATION  |   |',7X,'| COMMENT',32X,/,' |     |',
     .        'NUMBER|TYPE',10X,'|      NORTHING|     |  ELEMENT',
     .        '      |',40X/1X,'+-----+------+---+----------+'
     .         '---------------+-----+---+---+-------+',40('-'))
C
C          HEAD UP PAGE THEN TEST FOR A MATCHING STRING FOR EACH
```

```
C          OF THE SEARCH OPTIONS (USING THE SUBROUTINE 'NOTIN')
C
   27 READ(4,END=40) BUFF
      IF (NANALNO .EQ. 0) GOTO 29
      DO 28 I=1,NANALNO
        IF(IANALNO(I) .EQ. ANALNO) GOTO 29
   28 CONTINUE
      GOTO 27
   29 CONTINUE
C
C          IF MATCHING ANALYSIS NUMBER NOT FOUND, READ NEXT LINE.
C          OTHERWISE TEST OTHER SEARCH OPTIONS.
C
      IF (NOTIN(SELEMENT,ELEMENT)) GOTO 27
      IF (NOTIN(SROCTYPE,ROCTYPE)) GOTO 27
      IF (NOTIN(SSAMTYPE,SAMTYPE)) GOTO 27
      IF (NOTIN(SCOLLECT,COLLECT)) GOTO 27
      IF (CCOUNT .EQ. 0) GOTO 35
      DO 34 K=1,CCOUNT
         IF(INDEX(SAMPLNO,CSAMPLNO(K)) .NE. 0) GOTO 35
   34 CONTINUE
      GOTO 27
   35 IF (SPRINT .NE. 'Y') GOTO 37
      IF (EAST .LT. IEMIN .OR. EAST .GT. IEMAX) GOTO 27
      IF (NORTH .LT. INMIN .OR. NORTH .GT. INMAX) GOTO 27
C
C          THE REMAINING SEARCH OPTIONS ARE TESTED. IF A MISMATCH IS
C          FOUND, A NEW DATA LINE (BUFF) IS READ. IF A MATCH IS FOUND,
C          THE PROGRAMME CONTINUES.
C
   37 IF (LCOUNT .GE. 58) THEN
        WRITE(6,611) FFEED
  611 FORMAT(1A1/' ANALYSIS',9X,'¦ROCK TYPE ¦ COORDINATES  ¦   ',
     .       'COLLECTOR',
     .        ' ¦ DELTA ¦',40X/,' NUMBER¦  SAMPLE  ¦',10X,'¦EASTING'
     .        ,6X,'ELEVATION  ¦   ¦',7X,'¦ COMMENT',32X/,' ¦     ¦',
     .        'NUMBER¦TYPE',10X,'¦      NORTHING¦     ¦  ELEMENT',
     .        '      ¦',40X/1X,'+-----+------+---+----------+'
     .         '--------------+-----+---+---+-------+',40('-'))
       LCOUNT=4
      ENDIF
      FCOUNT=FCOUNT+1
C
C          HEAD UP PAGE (IF NECESSARY) - INCREMENT FOUND ENTRY AND
C          WRITTEN LINE COUNTER.
C
      WRITE(6,612) ANALNO,SAMPLNO,SAMTYPE,ROCTYPE,EAST,NORTH,ELEVN,
     .             COLLECT,ELEMENT,DELTA,COMMENT
  612 FORMAT(2X,I5,1X,A6,1X,A3,1X,A10,1X,I6,1X,I7,1X,A5,1X,A3,2X,A1,
     .       2X,A7,1X,A40)
      GOTO 27
C
C          WRITE THE DATA TO THE OUTPUT FILE, THEN BACK TO THE START
C          FOR THE NEXT DATA LINE.
C
   40 CONTINUE
C
C          AT THE END OF THE RUN, CHECK IF A SCREEN LISTING IS WANTED.
```

```
C
      WRITE(5,513) FCOUNT
  513 FORMAT(I7,' RECORDS WERE FOUND'/' DO YOU WANT A',
     .       ' SCREEN LISTING (Y OR N)?')
      READ(5,200) SPRINT
      IF (SPRINT .EQ. 'N') GOTO 36
C
C         IF NO SCREEN LIST WANTED, CHECK FOR PRINTOUT.
C
      REWIND (6)
C
C         REWIND THE FILE SO IT CAN BE SENT TO THE SCREEN
C
  402 ICOUNT=0
C
C         COUNTER FOR NUMBER OF LINES ON SCREEN
C
  401 READ(6,614,END=36) ALINE
      WRITE(5,614) ALINE
  614 FORMAT(66A1)
      ICOUNT=ICOUNT+1
      IF (ICOUNT .LT. 20) GOTO 401
C
C         ALLOW 20 ENTRIES PER SCREENFULL
C
      WRITE(5,515)
  515 FORMAT(' LIST MORE ENTRIES ON SCREEN (Y OR N)?')
      READ(5,200) SPRINT
      IF (SPRINT .EQ. 'Y') GOTO 402
C
C          ROUND AGAIN FOR ANOTHER SCREEN FULL, OR ASK ABOUT PRINTOUT.
C
   36 WRITE(5,516) FCOUNT
  516 FORMAT(I7,' RECORDS, DO YOU WANT A PRINTOUT?')
      READ(5,200) SPRINT
      IF (SPRINT .NE. 'Y' .OR. FCOUNT .EQ. 0) THEN
       CLOSE(UNIT=6,STATUS='DELETE')
       STOP
C
C          PRINTOUT NOT WANTED, SO CLOSE AND DELETE FILE THEN STOP
C
      ENDIF
      CLOSE(UNIT=4,STATUS='KEEP')
C
C          CLOSE THE INPUT FILE.
C
      STOP
      END
C
C*********************************************************************
C
      LOGICAL FUNCTION NOTIN(SSTRNG,RSTRNG)
C
C          SSTRNG CONTAINS A SERIES OF STRINGS TERMINATED BY /'S
C          E.G.   CPY/CP/CAL/
C          AND RSTRNG IS SEARCHED FOR THE OCCURENCE OF ONE OF THESE
C          STRINGS. IF ONE IS FOUND TO MATCH NOTIN IS .FALSE.
C          IF STRNG IS EMPTY NOTIN IS .FALSE.
```

```
C
      CHARACTER*80 SSTRNG
      CHARACTER* (*) RSTRNG
      NOTIN=.FALSE.
      IF (SSTRNG(1:2) .EQ. '  ') RETURN
C
C          NOTHING IN STRING, SO NO NEED TO LOOK
C
      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
```