

UR1985\_35

1985/35. BORIS - A borehole record information system for water bores  
(Revision 2)

R.G. Richardson

Abstract

The program suite described is used for adding entries to an index of water bore records, changing entries, sorting the index and searching it. The data are entered from the water bore cards and most of the details entered may be searched. This revision simplifies searching by area.

INTRODUCTION

BORIS is a simplification of the WBASE data-base developed by Polya (1982). Data on several thousand Tasmanian underground and surface waters are held by the Department of Mines on cards or within various reports. To search through the data on the basis of one parameter or more using the manual system is extremely difficult and time consuming.

The programs used for BORIS 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.

USING THE PROGRAMS

Data-base maintenance

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

- (i) OWNER - the owner's name (up to 24 characters).  
- to finish input type END here.
- (ii) LOCATION - a description of the bore location (up to 30 characters).
- (iii) DRILLING COMPANY - up to 8 characters.
- (iv) QUADRANGLE NUMBER.
- (v) CARD NUMBER - the card number within the quadrangle.
- (vi) YEAR - the year of drilling (last 3 digits only).
- (vii) GEOLOGIST - the initials of the responsible geologist (up to 3 characters).
- (viii) MAP REFERENCE - a 9 figure AMG co-ordinate giving the bore location to the nearest 100 m.
- (ix) TOTAL DEPTH - total depth (m) (real).
- (x) CASING DEPTH - casing depth (m) (real).
- (xi) DEPTH STRUCK - depth at which water struck (m) (real).
- (xii) STANDING WATER LEVEL - SWL (m) (real).

- (xiii) YIELD - bore yield (l/s) (real)
- (xiv) CONDUCTIVITY - the conductivity measured in the field ( $\mu\text{S}/\text{cm}$ ) (integer).
- (xv) NO. OF ANALYSES - the number of analyses done ( $\geq 0$ ). Only one analysis is stored in the data-base.
- (xvi) GEOLOGY - a 1 or 2 character summary of the geology.
- (xvii) ANALYSIS RESULT - if the number of analyses is not 0 the program then prompts for the data from one analysis.

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

The new data in WATTEMP.TMP must be added to the end of the existing data-base by typing WATMERGE. To ensure that the file remains in quadrangle and card number order, it should be sorted periodically using the WATSORT command.

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

- (i) QUADRANGLE, CARD NO. - the quadrangle number and card number of the record to be altered.  
- if 0, 0 is entered no further cards are altered.
- (ii) A series of opportunities to change particular data values then follow. It should be noted that, if there is now an analysis where there was not one before, this analysis can be added here.

*Data-base searching*

The search phase is entered by typing WATSRCH. An entry will be retrieved only if the specified search options are matched.

The program requests the following data:

- (i) OWNER - one or more names on a single line, each name terminated by /. If a search by owner is not required leave the line blank. e.g. SMITH/JONES/BROWN/
- (ii) LOCATION - details as for OWNER.
- (iii) QUADRANGLE - one or more quadrangle numbers on a single line, each terminated by a /. Leave blank if not required.  
e.g. 27/ only quadrangle 27  
27/32/14/ quadrangles 27, 32 and 14.
- (iv) YEARS - details as for QUADRANGLE.
- (v) GEOLOGIST - details as for OWNER.  
e.g. PCS/WCC/DCG/
- (vi) SEARCH BY AREA - leave blank for no search by area.  
- Y for finding entries inside a specified area. Entries on the boundary of the area will not be found. The

program prompts for the number of vertices. If 2 is entered a simple rectangle parallel to the grid may be entered by specifying the co-ordinates of any two diagonally opposite corners. Otherwise the program prompts for the vertex co-ordinates. Co-ordinates should be in the same form as used on the input sheets.

- (vii) Total depth range - blank if not to be used.
  - MIN, MAX if looking for bores with total depths between MIN and MAX.
- (viii) YIELD RANGE - blank if not to be used.
  - MIN, MAX if looking for bores with certain yields.
- (ix) TDS RANGE - blank if not to be used.
  - MIN, MAX if looking for bores with TDS in a certain range.
- (x) GEOLOGY - one or more geology codes on a single line, each group terminated by /. Leave blank if not required.
  - e.g. 0/4/ Coastal sands or Permian
  - 89/ Cambrian and Precambrian
- (xi) Search on analysis - N for no search using the results of analyses.
  - Y for searching using the results of analyses. If this is chosen then a series of ranges will be requested.
- (xii) 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.
- (xiii) Printout required - Y if required, N otherwise.
- (xiv) Print analyses - Y if analyses are wanted where available.
  - N otherwise.

THE PROGRAMS

*WATADD (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 (WATTEMP.TMP) may then be edited using the standard edit facilities.

*WATMERGE (Appendix 2)*

The data from WATTEMP.TMP are read and the basic data converted to the format of the data base and put in file WATMAIN.DAT. Any analyses are put in file WATANAL.DAT. Accidental data loss is prevented by copying to intermediate files.

*WATSORT (Appendix 3)*

The data are sorted into ascending quadrangle and card number order using a temporary work file.

*WATCHGE (Appendix 4)*

This pair of programs reads data from WATMAIN.DAT and WATANAL.DAT and allows the user to change the data. The temporary work files are then merged with the main files.

*WATSRCH (Appendix 5)*

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

## REFERENCE

POLYA, D.A. 1982. WBASE: a data-base system for Tasmanian waters.  
*Unpubl.Rep.Dep.Mines Tasm.* 1982/40.

[15 July 1985]

## APPENDIX 1

## Program WATADD

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

\*TITL WATADD.FTN - ADD CARDS TO THE WATER BORE FILE

C TAKES KEYBOARD INPUT AND PREPARES A PROOF SHEET

INTEGER\*4 MAPREF, CONDF

REAL\*4 DEPSTR, SWL, CHEM(17)

CHARACTER\*24 NAME

CHARACTER\*30 LOCATN

CHARACTER\*8 DRILCO, PROMPT(17)

INTEGER\*2 QUAD, CARD, YEAR, YIELD, ANAL

CHARACTER\*3 GEOLST

CHARACTER\*2 GEOL

C INPUT IS OWNER, LOACTION, DRILL CO., QUADRANGLE NO., CARD NO.,

C YEAR, GEOLOGIST, MAPREF, TOTAL DEPTH, CASING DEPTH,

C DEPTH STRUCK, STANDING WATER LEVEL, YIELD, CONDUCTIVITY (FIELD

C MEASUREMENT), NO. OF ANALYSES OF WATER SAMPLES, GEOLOGY

C (UP TO 2). IF THERE HAS BEEN AN ANALYSIS MADE THEN ONE

C ANALYSIS IN THE FORM-

PH, CONDUCTIVITY, CO3, HCO3, CL, SO4,

SIO2, CA, MG, FE, AL, K, NA, TDS(LAB),

ALKALINITY, PERM. HARDNESS,

TEMP. HARDNESS.

(THESE ARE EQUIVALENT TO CHEM(17))

LOGICAL IER

DATA PROMPT/'pH', 'CONDUCT', 'CO3', 'HCO3', 'Cl', 'SO4', 'SiO2',

'Ca', 'Mg', 'Fe', 'Al', 'K', 'Na', 'TDS(Lab)', 'Alk', 'P Hard',

'T Hard'/'

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(' OWNER OR END')

READ(5,200) NAME

200 FORMAT(A24)

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(A30)

WRITE(5,102)

102 FORMAT(' DRILLING CO.')

READ(5,202) DRILCO

202 FORMAT(A8)

WRITE(5,103)

103 FORMAT(' QUADRANGLE NO.')

READ(5,\*) QUAD

WRITE(5,104)

104 FORMAT(' CARD NO')

READ(5,\*) CARD

WRITE(5,109)

109 FORMAT(' YEAR')

READ(5,\*) YEAR

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(' TOTAL DEPTH')
      CALL FRINIT(TOTDP, IER, 5)
      WRITE(5,110)
110  FORMAT(' CASING DEPTH')
      CALL FRINIT(CASDP, IER, 5)
      WRITE(5,111)
111  FORMAT(' DEPTH STRUCK')
      CALL FRINIT(DEPSTR, IER, 5)
      WRITE(5,112)
112  FORMAT(' STANDING WATER LEVEL')
      CALL FRINIT(SWL, IER, 5)
      WRITE(5,113)
113  FORMAT(' YIELD')
      CALL FRINIT(AYIELD, IER, 5)
      WRITE(5,114)
114  FORMAT(' CONDUCTIVITY (FIELD)')
      CALL INFREE(CONDF, IER, 5)
      WRITE(5,115)
115  FORMAT(' NO. OF ANALYSES')
      CALL INT2(ANAL)
      WRITE(5,108)
108  FORMAT(' GEOLOGY')
      READ(5,204) GEOL
204  FORMAT(A2)
      IF (ANAL .NE. 0) THEN
C NEED TO READ AN ANALYSIS
      DO 12 I=1,17
      WRITE(5,116) PROMPT(I)
116  FORMAT(1X,A8)
      CALL FRINIT(CHEM(I), IER, 5)
      12 CONTINUE
      ENDIF
C READ ONLY IF NEEDED
C READ ONE COMPLETE DATA SHEET
C
C SO WRITE TO OUTPUT FILE
      WRITE(6,300) NAME, LOCATN, DRILCO, QUAD, CARD, YEAR, GEOLST, MAPREF,
      TOTDP, CASDP, DEPSTR, SWL, AYIELD, CONDF, ANAL, GEOL
300  FORMAT(A24, '@', A30, '@', A8, '@', I2, '@', I3, '@', I2, '@',
      A3, '@', I9, 4('@', F5.1), '@', F5.2, '@', I5,
      '@', I2, '@', A2)
C WRITE ANALYSIS ONLY IF THERE IS ONE
      IF (ANAL .NE. 0) WRITE(6,301) CHEM
301  FORMAT(3X, F5.2, '@', F6.0, 11('@', F7.1), 4('@', F6.0))
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
*TITL  FRINIT - FREE FORMAT INPUT OF ONE REAL
      SUBROUTINE FRINIT(ANUM, IER, ILUN)
C A FREE FORMAT INPUT ROUTINE FOR READING ONE REAL NUMBER FROM ILUN
C ILUN INTEGER*4
C ANUM REAL
C IER LOGICAL - TRUE IF THERE IS AN ERROR ON THE READ

```

```

C USES ENCODE TO GO FROM FREE FORMAT TO F10.0 COMPATIBLE
C FORMAT
  INTEGER*2 POINT, I, J, A(10), BLA, DPOINT, AJ
  REAL ANUM, BUFF(4)
  LOGICAL IER
  DATA BLA/1H /, DPOINT/1H. /
  IER=. TRUE.
  READ(ILUN, 100, ERR=101) A
100 FORMAT(10A1)
C READ IN 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 FAIL OUT BOTTOM OF LOOP ALL CHARACTERS ARE BLANK AND ANUM=0.0
  IER=. FALSE.
  ANUM=0.0
  101 RETURN
C THE QUICK RETURN
C
C NOW FIND THE DECIMAL POINT - IF ANY
  20 POINT=0
  DO 11 J=I, 10
  AJ=A(J)
  IF (AJ .EQ. DPOINT) POINT=1
C FOUND A DECIMAL POINT
  IF (AJ .EQ. BLA) GOTO 21
C FOUND A TRAILING BLANK AT THE END OF THE NUMBER
  11 CONTINUE
  21 IF (POINT .EQ. 0) A(J)=DPOINT
C IF THERE WAS NO DECIMAL POINT INSERT A DEC. POINT AT THE END
C
  IF (J .EQ. 10) GOTO 23
C DONT NEED TO BLANK FILL
  J=J+1
  DO 22 I=J, 10
  A(I)=BLA
  22 CONTINUE
C BLANK FILL
C NOW ENCODE
  23 ENCODE(BUFF, 100) (A(I), I=1, 10)
C AND READ BACK
  DECODE(BUFF, 111) ANUM
  111 FORMAT(F10.1)
C HAVE THE NUMBER
  IER=. FALSE.
  RETURN
  END
  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(5,100,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
  SUBROUTINE INT2(IVAR)
  INTEGER*2 IVAR
  LOGICAL IER
  CALL INFREE(ITMP,IER,5)
  IVAR=ITMP
  RETURN
  END

```

APPENDIX 2

Program WATMERGE

```

*WATMERGE. CSS
* FOR MERGING CORRECTED FILE WITH MAIN FILE
PRE ETM: XDE SYSF:WATMRG1.TMP: *DELETE SCRATCH FILE FOR MAIN ENTRY
XDE SYSF:WATMRG2.TMP: *DELETE SCRATCH FILE FOR ANALYSES
AL SYSF:WATMRG1.TMP,IN,108: AL SYSF:WATMRG2.TMP,IN,68: *AL SCRATCH FILES
L WATMERGE.10: * LOAD PROGRAM
REP WATTEMP.TMP,FF00: AS 6,WATTEMP.TMP,ERO: * THE TEMPORARY FILE
AS 4,WATMAIN.DAT,ERO: AS 2,WATANAL.DAT,ERO: * THE MAIN FILES
AS 3,SYSF:WATMRG1.TMP: AS 1,SYSF:WATMRG2.TMP: * TEMP FILES ON SYSF
ST
$IFNE 0: $WRITE TRANSLATE ERROR: ENA ETM: $EXIT: $ENDC
REP SYSF:WATMRG1.TMP,FF00: REP SYSF:WATMRG2.TMP,FF00: *PROTECT TEMP FILES
REP WATMAIN.DAT,0: REP WATANAL.DAT,0
DE WATMAIN.DAT,WATANAL.DAT
$BUILD WATCOPY.CMD
IN SYSF:WATMRG1.TMP
AL WATMAIN.DAT,IN,108/8/3
OUT WATMAIN.DAT
COPY *,*
REW I
REW 0
VERIFY *,*
IN SYSF:WATMRG2.TMP
AL WATANAL.DAT,IN,68/5/2
OUT WATANAL.DAT
COPY *,*
REW I
REW 0
VERIFY *,*
END
$ENDB
L COPY32.50: ST ,COM=WATCOPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0: $WRITE COPY BACK FAILED: ENA ETM: $EXIT: $ENDC
REP WATMAIN.DAT,FF00: REP WATANAL.DAT,FF00: * PROTECT NEW MAIN FILES
REP SYSF:WATMRG1.TMP,0: REP SYSF:WATMRG2.TMP,0: REP WATTEMP.TMP,0
DE WATCOPY.CMD,SYSF:WATMRG1.TMP,SYSF:WATMRG2.TMP,WATTEMP.TMP
ENA ETM
$EXIT

```

```

$TITL  WATMERGE.FTN - APPEND PROOF FILE TO MASTER FILES
REAL*4  DEPSTR, SWL, CHEM(17)
INTEGER*4  NORTH, TDSC, CONDF
CHARACTER*24  NAME
CHARACTER*30  LOCATN
CHARACTER*8  DRILCO
INTEGER*2  CARD, QUAD, YEAR, TOTDEP, CASDEP, YIELD, ANAL, INANAL, RECCNT,
        EAST
CHARACTER*3  GEOLST
CHARACTER*2  GEOL
C INPUT IS OWNER, LOACTION, DRILL CO., QUADRANGLE NO., CARD NO.,
C YEAR, GEOLOGIST, MAPREF, TOTAL DEPTH, CASING DEPTH,
C DEPTH STRUCK, STANDING WATER LEVEL, YIELD, CONDUCTIVITY (FIELD
C MEASUREMENT), NO. OF ANALYSES OF WATER SAMPLES, GEOLOGY
C (UP TO 2). IF THERE HAS BEEN AN ANALYSIS MADE THEN ONE
C ANALYSIS IN THE FORM-
C PH, CONDUCTIVITY, CO3, HCO3, CL, SO4,
C SIO2, CA, MG, FE, AL, K, NA, TDS(LAB),
C ALKALINITY, PERM. HARDNESS,
C TEMP. HARDNESS.
C (THESE ARE EQUIVALENT TO CHEM(17))
CHARACTER*1  BUFF(108)
EQUIVALENCE (BUFF(1), NAME), (BUFF(25), LOCATN), (BUFF(55), DRILCO),
. (BUFF(65), TDSC), (BUFF(69), CONDF), (BUFF(73), NORTH),
. (BUFF(77), DEPSTR), (BUFF(81), SWL), (BUFF(85), GEOLST),
. (BUFF(89), TOTDEP), (BUFF(91), CASDEP), (BUFF(93), YEAR),
. (BUFF(95), YIELD), (BUFF(97), QUAD), (BUFF(99), CARD),
. (BUFF(101), ANAL), (BUFF(103), GEOL), (BUFF(105), INANAL),
. (BUFF(107), EAST)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
C
REWIND 6
C REWIND WATTEMP.TMP - THE INPUT FILE
OPEN(UNIT=4, FORM='BINARY', RECL=100, SHARE='ERO')
C OPEN MAIN FILE INPUT ONLY
OPEN(UNIT=3, FORM='BINARY', RECL=100)
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 COPY THE ANALYSIS FILE
OPEN(UNIT=2, FORM='BINARY', RECL=68, SHARE='ERO')
C FILE TO READ FROM
OPEN(UNIT=1, FORM='BINARY', RECL=68)
C FILE TO COPY TO
RECCNT=0
12 RECCNT=RECCNT+1
C INCREMENT POINTER TO NEXT RECORD TO BE READ
READ(2, END=14) CHEM
WRITE(1) CHEM
GOTO 12
C END LOOP FOR COPYING

```

```

C
C CLOSE OFF
  14 CONTINUE
    CLOSE(UNIT=2, STATUS='KEEP')
C
C NOW TO START ADDING NEW ENTRIES
  20 READ(6, 300, END=25) NAME, LOCATN, DRILCO, QUAD, CARD, YEAR, GEOLST,
    EAST, NORTH, TOTDP, CASDP, DEPSTR, SWL, AYIELD, CONDF, ANAL, GEOL
300  FORMAT(A24, 1X, A30, 1X, A8, 1X, I2, 1X, I3, 1X, I2, 1X,
    A3, 1X, I4, I5, 4(1X, F5. 1), 1X, F5. 2, 1X, I5,
    1X, I2, 1X, A2)
    YIELD=AYIELD*100. 0+0. 1
C CONVERT TO INTEGER AND STILL KEEP 2 DECIMAL PLACES
C WITH 0. 1 ADDED TO ENSURE ROUNDS CORRECTLY
  TOTDEP=TOTDP*10. 0+0. 1
  CASDEP=CASDP*10. 0+0. 1
C CONVERT TO INTEGER AND KEEP 1 DECIMAL PLACE - ADD 0. 1 FOR ROUNDING
  INANAL=0
  TDSC=-9999
C SET TO A POSITIVE VALUE IF TDS IS TRANSFERRED FROM THE ANALYSIS
  IF (ANAL .GT. 0) THEN
    INANAL=RECCNT
  ENDIF
C STORE WHERE THE ASSOCIATED ANALYSIS IS TO BE FOUND - RECORD COUNT
C STARTS AT 1
  IF (ANAL .NE. 0) THEN
C IF THERE IS AN ANALYSIS IT MUST BE READ AND THEN WRITTEN TO NEW FILE
  RECCNT=RECCNT+1
C INCREMENT ONLY IF HAD AN ANALYSIS
  READ(6, 301) CHEM
  301  FORMAT(3X, F5. 2, 1X, F6. 0, 11(1X, F7. 1), 4(1X, F6. 0))
  TDSC=CHEM(14)
C WRITE THE TDS ON THE FRONT OF THE CARD IF IT EXISTS
  WRITE(1) CHEM
  ENDIF
C COPIED ANY ANALYSIS
  WRITE(3) BUFF
C ALWAYS WRITE THE FIRST LINE
  GOTO 20
C AND BACK ROUND
C
C AT END OF FILE
  25  CLOSE(UNIT=1, STATUS='KEEP')
    CLOSE(UNIT=3, STATUS='KEEP')
    CLOSE(UNIT=6, STATUS='KEEP')
C ALL NOW AMALGAMATED WITH MASTER FILES IN TEMPORARY FILES
  END

```

12/30

### APPENDIX 3

#### Program WATSORT

```
*WATSORT.CSS
* FOR SORTING WATER BORE FILE INTO CHRONOLOGICAL ORDER
PRE ETM
L WATSORT,10: AS 4,WATMAIN.DAT,ERO: AL SYSF:WTEMP.DAT,IN,108/5/2
AS 6,SYSF:WTEMP.DAT: ST
$IFNE 0: $WRITE SORT ERROR: ENA ETM: $EXIT: $ENDC
$BUILD WCOPY.CMD
IN SYSF:WTEMP.DAT
AL WATMAIN.DAT,IN,108/10/5
OUT WATMAIN.DAT
COPY *,*
REW I
REW O
VERIFY *,*
END
$ENDB
REP SYSF:WTEMP.DAT,FF00: REP WATMAIN.DAT,0: DE WATMAIN.DAT
L COPY32,50: ST ,COM=WCOPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0: $WRITE SORT-COPY ERROR: ENA ETM: $EXIT: $ENDC
REP WATMAIN.DAT,FF00: REP SYSF:WTEMP.DAT,0: DE SYSF:WTEMP.DAT,WCOPY.CMD
ENA ETM: $EXIT
```

\*TITL WATSORT.FTN - SORT THE WATER BORE FILE INTO ORDER

C

REAL\*4 DEPSTR, SWL  
INTEGER\*4 NORTH, TDSC, CONDF, IA(8000)  
CHARACTER\*24 NAME  
CHARACTER\*30 LOCATN  
CHARACTER\*8 DRILCO  
INTEGER\*2 CARD, QUAD, YEAR, TOTDEP, CASDEP, YIELD, ANAL, INANAL, RECCNT,  
EAST, INDEX(8000)  
CHARACTER\*3 GEOLST  
CHARACTER\*2 GEOL

C INPUT IS OWNER, LOACTION, DRILL CO., QUADRANGLE NO., CARD NO.,  
C YEAR, GEOLOGIST, MAPREF, TOTAL DEPTH, CASING DEPTH,  
C DEPTH STRUCK, STANDING WATER LEVEL, YIELD, CONDUCTIVITY (FIELD  
C MEASUREMENT), NO. OF ANALYSES OF WATER SAMPLES, GEOLOGY  
C (UP TO 2). IF THERE HAS BEEN AN ANALYSIS MADE THEN ONE  
C ANALYSIS IN THE FORM-

C PH, CONDUCTIVITY, CO3, HCO3, CL, SO4,  
C SiO2, CA, MG, FE, AL, K, NA, TDS(LAB),  
C ALKALINITY, PERM. HARDNESS,  
C TEMP. HARDNESS  
C (THESE ARE EQUIVALENT TO CHEM(17))

CHARACTER\*1 BUFF(108)  
EQUIVALENCE (BUFF(1), NAME), (BUFF(25), LOCATN), (BUFF(55), DRILCO),  
(BUFF(65), TDSC), (BUFF(69), CONDF), (BUFF(73), NORTH),  
(BUFF(77), DEPSTR), (BUFF(81), SWL), (BUFF(85), GEOLST),  
(BUFF(89), TOTDEP), (BUFF(91), CASDEP), (BUFF(93), YEAR),  
(BUFF(95), YIELD), (BUFF(97), QUAD), (BUFF(99), CARD),  
(BUFF(101), ANAL), (BUFF(103), GEOL), (BUFF(105), INANAL),  
(BUFF(107), EAST)

C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS

C

OPEN(UNIT=4, FORM='BINARY', ACCESS='DIRECT', RECL=108)

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)=CARD+(QUAD\*100000)

C COMBINE CARD AND QUAD TO GET CHRONOLOGICAL NUMBER

GOTO 10

C BACK AROUND

C

C NOW FOR THE WORK

20 CONTINUE  
IF (IREC .EQ. 0) STOP 'NO RECORDS'  
CALL SUBSTI(INDEX, IA, 1, IREC)

C SORT INTO ASCENDING NUMBER - I. E. CHRON ORDER

OPEN(UNIT=6, FORM='BINARY', RECL=108)

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

14/3E

```
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 4

Program WATCHGE

```

*WATCHGE. CSS
* FOR CORRECTING MAIN FILES AND ADDING ANALYSES
PRE ETM: XDE SYSF:WATCOR1.TMP: *DELETE SCRATCH FILE FOR MAIN ENTRY
XDE SYSF:WATCOR2.TMP: *DELETE SCRATCH FILE FOR ANALYSES
AL SYSF:WATCOR1.TMP,IN,108: AL SYSF:WATCOR2.TMP,IN,68: *AL SCRATCH FILES
AL WATCRTMP.TMP,IN,108
L WATCHGE,10: AS 2,WATANAL.DAT,ERO: AS 4,WATMAIN.DAT,ERO
AS 6,WATCRTMP.TMP: AS 3,PR:;ST
L WATCHGE2,10: * LOAD PROGRAM
REP WATCRTMP.TMP,FF00: AS 6,WATCRTMP.TMP,ERO: * THE TEMPORARY FILE
AS 4,WATMAIN.DAT,ERO: AS 2,WATANAL.DAT,ERO: * THE MAIN FILES
AS 3,SYSF:WATCOR1.TMP: AS 1,SYSF:WATCOR2.TMP: * TEMP FILES ON SYSF
ST
$IFNE 0: $WRITE TRANSLATE ERROR: ENA ETM: $EXIT: $ENDC
REP SYSF:WATCOR1.TMP,FF00: REP SYSF:WATCOR2.TMP,FF00: *PROTECT TEMP FILES
REP WATMAIN.DAT,0: REP WATANAL.DAT,0
DE WATMAIN.DAT,WATANAL.DAT
$BUILD WATCOPY.CMD
IN SYSF:WATCOR1.TMP
AL WATMAIN.DAT,IN,108/8/3
OUT WATMAIN.DAT
COPY *,*
REW I
REW 0
VERIFY *,*
IN SYSF:WATCOR2.TMP
AL WATANAL.DAT,IN,68/5/2
OUT WATANAL.DAT
COPY *,*
REW I
REW 0
VERIFY *,*
END
$ENDB
L COPY32,50: ST ,COM=WATCOPY.CMD,LI=NULL:,LO=NULL:
$IFNE 0: $WRITE COPY BACK FAILED: ENA ETM: $EXIT: $ENDC
REP WATMAIN.DAT,FF00: REP WATANAL.DAT,FF00: * PROTECT NEW MAIN FILES
REP SYSF:WATCOR1.TMP,0: REP SYSF:WATCOR2.TMP,0: REP WATCRTMP.TMP,0
DE WATCOPY.CMD,SYSF:WATCOR1.TMP,SYSF:WATCOR2.TMP,WATCRTMP.TMP
ENA ETM
$EXIT

```

```

$TITL WATCHGE.FTN - CHANGE DETAILS OF WATER BORE ENTRIES
INTEGER*2 QUADS, EAST, CARDS, AOK, IBASE
LOGICAL CHANGE, CHGE
REAL*4 DEPSTR, SWL, CHEM(17), BUFF1(4)
INTEGER*4 MAPREF, NORTH, TDSC, CONDF
CHARACTER*24 NAME
CHARACTER*30 LOCATN
CHARACTER*8 DRILCO, PROMPT(17)
INTEGER*2 CARD, QUAD, YEAR, TOTDEP, CASDEP, YIELD, ANAL, INANAL, RECCNT
CHARACTER*3 GEOLST
CHARACTER*2 GEOL
C INPUT IS OWNER, LOACTION, DRILL CO., QUADRANGLE NO., CARD NO.,
C YEAR, GEOLOGIST, MAPREF, TOTAL DEPTH, CASING DEPTH,
C DEPTH STRUCK, STANDING WATER LEVEL, YIELD, CONDUCTIVITY (FIELD
C MEASUREMENT), NO. OF ANALYSES OF WATER SAMPLES, GEOLOGY
C (UP TO 2). IF THERE HAS BEEN AN ANALYSIS MADE THEN ONE
C ANALYSIS IN THE FORM-
C PH, CONDUCTIVITY, CO3, HCO3, CL, SO4,
C SIO2, CA, MG, FE, AL, K, NA, TDS(LAB),
C ALKALINITY, PERM. HARDNESS,
C TEMP. HARDNESS.
C (THESE ARE EQUIVALENT TO CHEM(17))
CHARACTER*1 BUFF(108), BUFF2(108)
EQUIVALENCE (BUFF(1), NAME), (BUFF(25), LOCATN), (BUFF(55), DRILCO),
. (BUFF(65), TDSC), (BUFF(69), CONDF), (BUFF(73), NORTH),
. (BUFF(77), DEPSTR), (BUFF(81), SWL), (BUFF(85), GEOLST),
. (BUFF(89), TOTDEP), (BUFF(91), CASDEP), (BUFF(93), YEAR),
. (BUFF(95), YIELD), (BUFF(97), QUAD), (BUFF(99), CARD),
. (BUFF(101), ANAL), (BUFF(103), GEOL), (BUFF(105), INANAL),
. (BUFF(107), EAST)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
EQUIVALENCE (BUFF2(1), CHEM(1)), (BUFF2(69), IBASE)
C USED TO WRITE TO TEMP OUTPUT FILE TO MAKE LIFE EASIER
DATA PROMPT/'pH', 'CONDUCT', 'CO3', 'HCO3', 'CL', 'SO4', 'SiO2',
. 'Ca', 'Mg', 'Fe', 'Al', 'K', 'Na', 'TDS(Lab)', 'Alk', 'P Hard',
. 'T Hard'/
C USED FOR PROMPTS
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=108, SHARE='ERO')
C OPEN WATMAIN.DAT
OPEN(UNIT=2, FORM='BINARY', ACCESS='DIRECT', RECL=68, SHARE='ERO')
C OPEN WATANAL.DAT
OPEN(UNIT=6, FORM='BINARY', RECL=108)
C TEMP FILE FOR CHANGES
C
10 WRITE(5,100)
100 FORMAT(' QUAD NO. AND CARD NO. IN FORM QUAD, CARD'/
. ' 0, 0 TERMINATES')
READ(5,*) QUADS, CARDS
IF (QUADS .EQ. 0 .AND. CARDS .EQ. 0) GOTO 20
C END OF RUN
IF (QUADS .EQ. 0 .OR. CARDS .EQ. 0) GOTO 10
C SHOULDNT HAVE EITHER OF THESES AS 0
IBASE=0
C COUNTER FOR POSITION IN WATMAIN WHICH IS STORED FOR LATER
C RANDOM ACCESS OF WATMAIN

```

```

REWIND 4
C START AT THE VERY BEGINNING - A VERY GOOD PLACE TO START
  11 CONTINUE
    READ(4,END=10) BUFF
C IF WE HIT THE END MUST HAVE HAD A WRONG QUAD OR CARD
  IBASE=IBASE+1
C COUNTING THROUGH WATMAIN
  IF(QUAD .NE. QUADS .OR. CARD .NE. CARDS) GOTO 11
C NOT THE ONE WE NEED SO READ A BIT MORE
  WRITE(5,101) QUAD,CARD,NAME,LOCATN
  101 FORMAT(1X,I3,I4,1X,A24,A36/' 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
C
C NOW START TO LOOK AT CHANGES
  ANOLD=ANAL
C STORE THE CURRENT NUMBER OF ANALYSES TO ALLOW FOR ANALYSIS WHERE
C NONE WAS KNOWN PREVIOUSLY
C
C NOW ASK FOR CHANGES
  IF (CHANGE('OWNER')) CALL ALPHA1(NAME,'OWNER')
  IF (CHANGE('LOCATION')) CALL ALPHA2(LOCATN,'LOCATION')
  IF (CHANGE('DRILL CO. ')) CALL ALPHA3(DRILCO,'DRILL CO. ')
  IF (CHANGE('QUAD NO. ')) CALL INTG2(QUAD,'QUAD NO. ')
  IF (CHANGE('CARD NO. ')) CALL INTG2(CARD,'CARD NO. ')
  IF (CHANGE('YEAR')) CALL INTG2(YEAR,'YEAR')
  IF (CHANGE('GEOLOGIST')) CALL ALPHA4(GEOLST,'GEOLOGIST')
  IF (CHANGE('MAP REF')) THEN
    CALL INTG4(MAPREF,'MAP REF')
    ENCODE(BUFF1,201) MAPREF
  201 FORMAT(I9)
    DECODE(BUFF1,202) EAST,NORTH
  202 FORMAT(I4,I5)
C SPLIT MAPREF INTO EAST AND NORTH
  ENDIF
  IF (CHANGE('TOTAL DEPTH')) THEN
    CALL REAL4(TOTDP,'TOTAL DEPTH')
    TOTDEP=TOTDP*10.0+0.1
  ENDIF
  IF (CHANGE('CASING DEPTH')) THEN
    CALL REAL4(CASDP,'CASING DEPTH')
    CASDEP=CASDP*10.0+0.1
  ENDIF
  IF (CHANGE('DEPTH STRUCK')) CALL REAL4(DEPSTR,'DEPTH STRUCK')
  IF (CHANGE('SWL')) CALL REAL4(SWL,'STANDING WATER LEVEL')
  IF (CHANGE('YIELD')) THEN
    CALL REAL4(AYIELD,'YIELD')
    YIELD=AYIELD*100.0+0.1
C CONVERT TO INTEGER FOR STORAGE AND ENSURE CORRECT ROUNDING
  ENDIF
  IF (CHANGE('CONDUCTIVITY (FIELD)')) CALL INTG4(CONDF,
    'CONDUCTIVITY (FIELD)')
  IF (CHANGE('NO. OF ANALYSES')) CALL INTG2(ANAL,'NO. OF ANALYSES')
  IF (CHANGE('GEOLOGY')) CALL ALPHA5(GEOL,'GEOLOGY')
C END OF THE UNIFORM SET OF CHANGES
C
C NOW READ ANALYSIS IF IT EXISTS

```

```

      IF (ANOLD .GT. 0) READ(2,REC=INANAL) CHEM
C DIRECT READ FROM ANALYSIS FILE
      IF (ANOLD .EQ. 0 .AND. ANAL .GT. 0) THEN
C HAVE PUT IN AN ANALYSIS WHERE THERE WAS NOT ONE BEFORE
C SO READ THE VALUES
      DO 12 I=1,17
        CALL REAL4(CHEM(I),PROMPT(I))
      12 CONTINUE
      INANAL=-10
C NOTE TO REMEMBER THAT THIS IS A NEW ANALYSIS
      ENDIF
C READ A NEW ANALYSIS
C
C NOW SHOULD WE ALTER THE OLD ANALYSIS
      IF(ANOLD .GT. 0) THEN
        WRITE(5,102)
      102 FORMAT(' CHANGE THE ANALYSIS (Y OR N)?')
        READ(5,200) AOK
        IF (AOK .EQ. 'Y') THEN
          DO 14 I=1,17
            IF (CHANGE(PROMPT(I))) CALL REAL4(CHEM(I),PROMPT(I))
          14 CONTINUE
        ENDIF
C GOT ANY CHANGES TO THE ANALYSIS
      ENDIF
      IF (ANAL .GT. 0) TDSC=CHEM(14)
C STORE ANY NEW VALUE FOR TDS ON THE FRONT OF THE VIRTUAL CARD
C NOW BACK FOR COMMON CODE FOR WRITING OUT
C
C WRITE OUT TO FILE AND TO PRINTER
      WRITE(6) BUFF
      WRITE(6) BUFF2
      IF (.NOT. CHGE) THEN
        WRITE(3,299)
      299 FORMAT(' THE FOLLOWING CHANGES ARE BEING MADE')
        CHGE=.TRUE.
      ENDIF
      WRITE(3,300) NAME,LOCATN,DRILCO,QUAD,CARD,YEAR,GEOLST,EAST,
        NORTH,TOTDP,CASDP,DEPSTR,SWL,AYIELD,CONDF,ANAL,GEOL
      300 FORMAT(1X,A24,'@',A30,'@',A8,'@',I2,'@',I3,'@',I2,'@',
        A3,'@',I4,I5,4('@',F5.1),'@',F5.2,'@',I5,
        '@',I2,'@',A2)
C WRITE ANALYSIS ONLY IF THERE IS ONE
      IF (ANAL .GT. 0) WRITE(3,301) CHEM
      301 FORMAT(3X,F5.2,'@',F6.0,11('@',F7.1),4('@',F6.0))
C WRITE WITH DELIMITERS AFTER EACH FIELD
      GOTO 10
C BACK ROUND
C
C NOW TO TIDY UP
      20 CONTINUE
      WRITE(3,302)
      302 FORMAT(1X/)
      CLOSE(UNIT=2,STATUS='KEEP')
      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*24 SVAR
WRITE(5,100) PROMPT
100 FORMAT(1X,A20)
READ(5,101) SVAR
101 FORMAT(A24)
RETURN
END
SUBROUTINE ALPHA2(SVAR, PROMPT)
CHARACTER* (*) PROMPT
CHARACTER*30 SVAR
WRITE(5,100) PROMPT
100 FORMAT(1X,A20)
READ(5,101) SVAR
101. FORMAT(A30)
RETURN
END
SUBROUTINE ALPHA3(SVAR, PROMPT)
CHARACTER* (*) PROMPT
CHARACTER*8 SVAR
WRITE(5,100) PROMPT
100 FORMAT(1X,A20)
READ(5,101) SVAR
101 FORMAT(A8)
RETURN
END
SUBROUTINE ALPHA4(SVAR, PROMPT)
CHARACTER* (*) PROMPT
CHARACTER*3 SVAR
WRITE(5,100) PROMPT
100 FORMAT(1X,A20)
READ(5,101) SVAR
101 FORMAT(A3)
RETURN
END
SUBROUTINE ALPHA5(SVAR, PROMPT)
CHARACTER* (*) PROMPT
CHARACTER*2 SVAR
WRITE(5,100) PROMPT
100 FORMAT(1X,A20)
READ(5,101) SVAR
101 FORMAT(A2)
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 INTG4<IVAR,PROMPT>  
      CHARACTER* (*) PROMPT  
      INTEGER*4 IVAR  
      WRITE<5,100> PROMPT  
100  FORMAT<1X,A20>  
      READ<5,*> IVAR  
      RETURN  
      END  
      SUBROUTINE REAL4<VAR,PROMPT>  
      CHARACTER* (*) PROMPT  
      REAL*4 VAR  
      WRITE<5,100> PROMPT  
100  FORMAT<1X,A20>  
      READ<5,*> VAR  
      RETURN  
      END
```

```

#TITL WATCHGE2.FTN - PUT CHANGES INTO SEQUENCE IN THE MAIN FILES
INTEGER*2 IBASE, EAST
REAL*4 DEPSTR, SWL, CHEM(17)
INTEGER*4 NORTH, TDSC, CONDF
CHARACTER*24 NAME
CHARACTER*30 LOCATN
CHARACTER*8 DRILCO
INTEGER*2 CARD, QUAD, YEAR, TOTDEP, CASDEP, YIELD, ANAL, INANAL, RECCNT
CHARACTER*3 GEOLST
CHARACTER*2 GEOL
C INPUT IS OWNER, LOACTION, DRILL CO., QUADRANGLE NO., CARD NO.,
C YEAR, GEOLOGIST, MAPREF, TOTAL DEPTH, CASING DEPTH,
C DEPTH STRUCK, STANDING WATER LEVEL, YIELD, CONDUCTIVITY (FIELD
C MEASUREMENT), NO. OF ANALYSES OF WATER SAMPLES, GEOLOGY
C (UP TO 2). IF THERE HAS BEEN AN ANALYSIS MADE THEN ONE
C ANALYSIS IN THE FORM-
C PH, CONDUCTIVITY, CO3, HCO3, CL, SO4,
C SiO2, CA, MG, FE, AL, K, NA, TDS(LAB),
C ALKALINITY, PERM. HARDNESS,
C TEMP. HARDNESS.
C (THESE ARE EQUIVALENT TO CHEM(17))
CHARACTER*1 BUFF(108), BUFF2(108)
EQUIVALENCE (BUFF(1), NAME), (BUFF(25), LOCATN), (BUFF(55), DRILCO),
. (BUFF(65), TDSC), (BUFF(69), CONDF), (BUFF(73), NORTH),
. (BUFF(77), DEPSTR), (BUFF(81), SWL), (BUFF(85), GEOLST),
. (BUFF(89), TOTDEP), (BUFF(91), CASDEP), (BUFF(93), YEAR),
. (BUFF(95), YIELD), (BUFF(97), QUAD), (BUFF(99), CARD),
. (BUFF(101), ANAL), (BUFF(103), GEOL), (BUFF(105), INANAL),
. (BUFF(107), EAST)
C USED TO ALLOW BINARY OUTPUT FOR FASTER ACCESS
EQUIVALENCE (BUFF2(1), CHEM(1)), (BUFF2(69), IBASE)
C USED TO WRITE TO TEMP OUTPUT FILE TO MAKE LIFE EASIER
C
OPEN(UNIT=4, FORM='BINARY', RECL=108, SHARE='ERO')
C OPEN MAIN FILE INPUT ONLY
OPEN(UNIT=3, FORM='BINARY', RECL=108, 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
C NOW COPY THE ANALYSIS FILE
OPEN(UNIT=2, FORM='BINARY', RECL=68, SHARE='ERO')
C FILE TO READ FROM
OPEN(UNIT=1, FORM='BINARY', RECL=68, ACCESS='DIRECT')
C FILE TO COPY TO
RECCNT=0
12 RECCNT=RECCNT+1
C INCREMENT POINTER TO NEXT RECORD TO BE READ
READ(2, END=14) CHEM
WRITE(1) CHEM
GOTO 12
C END LOOP FOR COPYING

```

```

C
C CLOSE OFF
  14 CONTINUE
    CLOSE(UNIT=2, STATUS='KEEP')
C
C NOW TO START FINDING THE CHANGES AND PUTTING THEM IN THE RIGHT
C PLACES
    OPEN(UNIT=6, FORM='BINARY', RECL=100, SHARE='ERO')
C OPEN THE TEMP FILE
    REWIND 6
C AND POSITION IT
  20 READ(6, END=25) BUFF
    READ(6) BUFF2
C READ THE TWO RECORDS THAT MAKE UP EACH CHANGE
C
C DEAL WITH THE ANALYSIS FIRST
    IF (ANAL .NE. 0) THEN
      IF (INANAL .LE. 0) THEN
C DEAL WITH THE CASE WHERE THE ANALYSIS DIDNT EXIST BEFORE
C I. E. NEED TO SET INANAL TO POINT TO THE PRESENT END OF THE ANALYSIS
C FILE AND THEN UPDATE THE POINTER TO THE END OF THAT FILE TO
C POINT TO THE NEXT EMPTY LOCATION
        INANAL=RECCNT
        RECCNT=RECCNT+1
      ENDIF
      WRITE(1, REC=INANAL) CHEM
    ENDIF
C WRITTEN THE ANALYSIS
C
C NOW WRITE THE BASIC DATA
    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
C AT END OF FILE
  25 CLOSE(UNIT=1, STATUS='KEEP')
    CLOSE(UNIT=3, STATUS='KEEP')
    CLOSE(UNIT=6, STATUS='KEEP')
C ALL NOW AMALGAMATED WITH MASTER FILES IN TEMPORARY FILES
END

```

20/38

APPENDIX 5  
Program WATSRCH

```
*WATSRCH.CSS - SEARCH THE WATER BORE FILE  
L WATSRCH,10: AS 4,WATMAIN.DAT,ERO: AS 2,WATANAL.DAT,ERO  
AS 6,PR:; AS 5,CON:; ST  
$EXIT
```

\$TITL WATSRCH.FTN - SEARCH WATER BORE FILE

C  
INTEGER\*2 QUADS, EAST, CARDS, AOK, IBASE, FCOUNT, FOUND(5000)  
LOGICAL CHANGE, CHGE  
REAL\*4 DEPSTR, ANUM(2), SWL, CHEM(17)  
INTEGER\*4 NORTH, TDSC, CONDF  
CHARACTER\*24 NAME  
CHARACTER\*30 LOCATN  
CHARACTER\*8 DRILCO, PROMPT(17)  
INTEGER\*2 CARD, QUAD, YEAR, TOTDEP, CASDEP, YIELD, ANAL, INANAL, RECCNT  
CHARACTER\*3 GEOLST  
CHARACTER\*2 GEOL

C INPUT IS OWNER, LOACTION, DRILL CO., QUADRANGLE NO., CARD NO.,  
C YEAR, GEOLOGIST, MAPREF, TOTAL DEPTH, CASING DEPTH,  
C DEPTH STRUCK, STANDING WATER LEVEL, YIELD, CONDUCTIVITY (FIELD  
C MEASUREMENT), NO. OF ANALYSES OF WATER SAMPLES, GEOLOGY  
C (UP TO 2). IF THERE HAS BEEN AN ANALYSIS MADE THEN ONE  
C ANALYSIS IN THE FORM-

PH, CONDUCTIVITY, CO3, HCO3, CL, SO4,  
SIO2, CA, MG, FE, AL, K, NA, TDS(LAB),  
ALKALINITY, PERM. HARDNESS,  
TEMP. HARDNESS  
(THESE ARE EQUIVALENT TO CHEM(17))

CHARACTER\*1 BUFF(108)  
EQUIVALENCE (BUFF(1), NAME), (BUFF(25), LOCATN), (BUFF(55), DRILCO),  
(BUFF(65), TDSC), (BUFF(69), CONDF), (BUFF(73), NORTH),  
(BUFF(77), DEPSTR), (BUFF(81), SWL), (BUFF(85), GEOLST),  
(BUFF(89), TOTDEP), (BUFF(91), CASDEP), (BUFF(93), YEAR),  
(BUFF(95), YIELD), (BUFF(97), QUAD), (BUFF(99), CARD),  
(BUFF(101), ANAL), (BUFF(103), GEOL), (BUFF(105), INANAL),  
(BUFF(107), EAST)

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, SANAL, STDM, S'YM, STDC, SCDF, SCH(14), IER  
REAL\*4 CHMAX(14), CHMIN(14), YMIN, YMAX  
INTEGER\*2 IQUAD(30), IYR(30), NQUAD, NYEAR, ITDMIN, ITDMAX,  
YLMIN, YLMAX, STYPE  
INTEGER\*4 TDCMIN, TDCMAX, CDFMIN, CDFMAX  
CHARACTER\*80 SNAME, SLOCN, SQUAD, SYEAR, SGEOL, SGEOLT  
DATA FFEED/3072/

C THE FORM FEED CHARACTER IN A1  
DATA PROMPT/'PH', 'CONDUCT', 'CO3', 'HCO3', 'CL', 'SO4', 'SIO2',  
'Ca', 'Mg', 'Fe', 'Al', 'K', 'Na', 'TDS(Lab)', 'Alk', 'P Hard',  
'T Hard'/

C USED FOR PROMPTS  
OPEN(UNIT=4, FORM='BINARY', ACCESS='DIRECT', RECL=108, SHARE='ERO')

C OPEN THE WATMAIN.DAT FILE  
OPEN(UNIT=2, FORM='BINARY', ACCESS='DIRECT', RECL=68, SHARE='ERO')

C OPEN WATANAL.DAT FILE  
WRITE(5, 101)  
101 FORMAT(' FOR CHARACTER OPTIONS ENTER EITHER A BLANK LINE OR/  
' THE DESIRED VALUES FOR THE OPTION, EACH TERMINATED BY/  
' A ///' E.G. FRED/BILL/TOM/SMITH///' FOR NUMERIC OPTIONS',  
' ENTER EITHER THE RANGE REQUESTED/// OR A BLANK LINE')

WRITE(5, 102)  
102 FORMAT(' OWNER?')

```

      READ(5,201) SNAME
201  FORMAT(A80)
      WRITE(5,103)
103  FORMAT(' LOCATION?')
      READ(5,201) SLOCN
      WRITE(5,104)
104  FORMAT(' QUADRANGLE NO. ?')
      READ(5,201) SQUAD
      CALL NMERIC(SQUAD, IQUAD, NQUAD)
      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(5,107)
107  FORMAT(' GEOLOGIST?')
      READ(5,201) SGEOLT
      WRITE(5,108)
108  FORMAT(' SEARCH FOR SITES WITHIN A SPECIFIED AREA (Y)?')
      READ(5,200) STYPE
      IF (STYPE .EQ. 'Y') GOTO 23
C WANT TO READ COORDS
      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?'/
           ' ENTER 2 TO SPECIFY A RECTANGLE')
      READ(5,*) NCOORD
      IF (NCOORD .LT. 2) GOTO 23
C NEED AT LEAST 3 VERTICES TO DEFINE AN AREA
      IF (NCOORD .EQ. 2) THEN
        WRITE(5,515)
515  FORMAT(' ENTER COORDS OF TWO DIAGONALLY OPPOSITE CORNERS OF '
           ' THE RECTANGLE'// ONE PER LINE IN THE FORM EASTNORTH')
        ELSE
          WRITE(5,110)
110  FORMAT(' ENTER VERTEX COORDS IN CLOCKWISE ORDER'//
           ' ONE PER LINE IN FORM EASTNORTH')
        ENDIF
        IXMAX=-999999999
        IXMIN=-IXMAX
        IYMAX=IXMAX
        IYMIN=IXMIN
C USE THESE TO STORE THE RANGE OF THE AREA
      DO 24 I=1, NCOORD
        READ(5,503) IX, IY
503  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 (NCOORD .GT. 2) THEN
      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
      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
      ENDIF
C
      26 CONTINUE
      WRITE(5,105)
105  FORMAT(' TOTAL DEPTH RANGE - MIN,MAX (REAL)')
      CALL RANGE(ANUM,IER,5)
      TDMIN=ANUM(1)
      TDMAX=ANUM(2)
      ITDMIN=TDMIN*10.0+0.1
      ITDMAX=TDMAX*10.0+0.1
      STDM=ITDMAX .GE. ITDMIN .AND. ITDMAX .NE. 0
      WRITE(5,111)
111  FORMAT(' YIELD RANGE - MIN,MAX (REAL)')
      CALL RANGE(ANUM,IER,5)
      YMIN=ANUM(1)
      YMAX=ANUM(2)
      YLMIN=YMIN*100.0+0.1
      YLMAX=YMAX*100.0+0.1
      SYM=YLMAX .GE. YLMIN .AND. YLMAX .NE. 0
      WRITE(5,121)
121  FORMAT(' CONDUCTIVITY (FIELD) RANGE - MIN,MAX (INTEGER)')
      CALL RANGE(ANUM,IER,5)
      CDFMIN=ANUM(1)
      CDFMAX=ANUM(2)
      SCDF=CDFMAX .GE. CDFMIN .AND. CDFMAX .NE. 0
      WRITE(5,113)
113  FORMAT(' TDS RANGE - MIN,MAX (INTEGER)')
      CALL RANGE(ANUM,IER,5)
      TDCMIN=ANUM(1)
      TDCMAX=ANUM(2)
      STDC=TDCMAX .GE. TDCMIN .AND. TDCMAX .NE. 0
      WRITE(5,112)
112  FORMAT(' GEOLOGY (0-9)?')
      READ(5,201) SGEOL
      WRITE(5,114)
114  FORMAT(' SEARCH ON ANALYSIS (Y OR N)?')
      READ(5,200) STYPE
200  FORMAT(A1)
      SANAL=STYPE .EQ. 'Y'
      IF (SANAL) THEN
      DO 20 I=1,14
      WRITE(5,115) PROMPT(I)

```

```

1.15 FORMAT(' RANGE MIN,MAX FOR ',A8)
CALL RANGE(ANUM,IER,5)
CHMIN(I)=ANUM(1)
CHMAX(I)=ANUM(2)
SCH(I)=CHMAX(I) .GE. CHMIN(I) .AND. CHMAX(I) .NE. 0
20 CONTINUE
ENDIF
IF (.NOT. STDC .AND. SCH(14)) THEN
C LOOKING FOR TDS SO MAKE IT LOOK AT THE FRONT OF THE VIRTUAL
C CARD TO SAVE TIME RUSHING OFF TO GET THE BACK OF THE CARD
STDC=. TRUE.
TDCMIN=CHMIN(14)
TDCMAX=CHMAX(14)
ENDIF
C STORED RANGE FOR TDS
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 WATMAIN.DAT
27 READ(4,END=40) BUFF
C READ ONE ENTRY
RECCNT=RECCNT+1
C READ ONE MORE RECORD FROM WATMAIN
IF (SANAL .AND. ANAL .EQ. 0) GOTO 27
C SEARCHING BY ANALYSIS BUT THERE IS NO ANALYSIS
IF (NOTIN(SNAME,NAME)) GOTO 27
C NOT THE RIGHT NAME
IF(NOTIN(SLOCN,LOCATN)) GOTO 27
C NOT THE RIGHT LOCATION
IF (NQDAD .NE. 0) THEN
DO 32 I=1,NQUAD
IF (IQDAD(I) .EQ. QUAD) GOTO 25
32 CONTINUE
GOTO 27
C NOT FOUND SO FALL OUT
25 CONTINUE
ENDIF
C END LOOK FOR QUADRANGLE
IF (NYEAR .NE. 0) THEN
DO 28 I=1,NYEAR
IF (IYR(I) .EQ. YEAR) 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
IF (NCOORD .GT. 2) THEN
XTMP=EAST*0.1
YTMP=NORTH*0.1

```

28/38

```

C CONVERT TO KM
  IF (.NOT. INOUT(XTMP, YTMP, X, Y, YINTVL, INTVLS, SWATH, RSLOPE))
    GOTO 27
C NOT IN THE SPECIFIED AREA
ENDIF
C FOR THE SIMPLE RECTANGLE DONT NEED THE POLYGON SEARCH
ENDIF
IF (STDM) THEN
  IF (TOTDEP .LT. ITDMIN .OR. TOTDEP .GT. ITDMAX) GOTO 27
ENDIF
C CHECK FOR TOTAL DEPTH IN RANGE
IF (SYM) THEN
  IF (YIELD .LT. YLMIN .OR. YIELD .GT. YLMAX) GOTO 27
ENDIF
C CHECK YIELD IN RANGE
IF (SCDF) THEN
  IF (CONDF .LT. CDFMIN .OR. CONDF .GT. CDFMAX) GOTO 27
ENDIF
C CHECK FIELD CONDUCTIVITY
IF (STDC) THEN
  IF (TDSC .LT. TDCMIN .OR. TDSC .GT. TDCMAX) GOTO 27
ENDIF
C CHECK TDS
IF (NOTIN(SGEOL, GEOL)) GOTO 27
C NOT RIGHT GEOLOGY
IF (SANAL) THEN
  DO 30 I=1,13
C ONLY 13 AS HAVE LOOKED AT TDS ON FRONT OF CARD
  IF (SCH(I)) THEN
    IF (CHEM(I) .LT. CHMIN(I) .OR. CHEM(I) .GT. CHMAX(I)) GOTO 27
C THAT PARTICULAR ITEM NOT IN RANGE
  ENDIF
  30 CONTINUE
C GO ROUND ONE ITEM AT A TIME
ENDIF
C BEEN THROUGH ANALYSIS
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 WATMAIN.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
  IF (FCOUNT .EQ. 0) STOP
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
  TOTDP=TOTDEP*0.1
  CASDP=CASDEP*0.1
  AYIELD=YIELD*0.01
C CONVERT BACK TO LITRES/SEC FROM HUNDREDTHS OF LITRES/SEC
C AND BACK TO METRES FROM DECIMETRES
  IF (TDSC .GT. 0) THEN
  WRITE(5,116) NAME,LOCATN,DRILCO,QUAD,CARD,YEAR,GEOLST,EAST,
    NORTH,TOTDP,CASDP,DEPSTR,SWL,AYIELD,CONDF,TDSC,ANAL,GEOL
116 FORMAT(1X,A24,' ',A30,' ',A8,2(' ',I3)/5X,I2,' ',
    A3,' ',I4,I5,' ',4(F5.1,'m'),
    F5.2,'1/s',I5,' ',I5,'mg/1',I2,' ',A2)
  ELSE
  WRITE(5,122) NAME,LOCATN,DRILCO,QUAD,CARD,YEAR,GEOLST,EAST,
    NORTH,TOTDP,CASDP,DEPSTR,SWL,AYIELD,CONDF,ANAL,GEOL
122 FORMAT(1X,A24,' ',A30,' ',A8,2(' ',I3)/5X,I2,' ',
    A3,' ',I4,I5,' ',4(F5.1,'m'),
    F5.2,'1/s',I5,' ',5X,'mg/1',I2,' ',A2)
  ENDIF
  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 (Y OR N)?')
  READ(5,200) STYPE
  IF (STYPE .NE. 'Y') STOP
C STOP IF NOTHING FOUND
C
C OTHERWISE WRITE TO PRINTER
C FIRST WRITE THE SEARCH OPTIONS
  WRITE(6,400)
400 FORMAT(' SEARCH OPTIONS ARE:')
  IF (SNAME(1:1) .NE. ' ') WRITE(6,401) SNAME
401 FORMAT(' NAME',10X,A80)
  IF (SLOCN(1:1) .NE. ' ') WRITE (6,402) SLOCN
402 FORMAT(' LOCATION',10X,A80)
  IF (NQDAD .NE. 0) WRITE(6,403) SQDAD
403 FORMAT(' QUADRANGLE NO.',10X,A80)
  IF (NYEAR .NE. 0) WRITE(6,405) SYEAR
405 FORMAT(' YEAR',10X,A80)
  IF (SGEOLT(1:1) .NE. ' ') WRITE(6,406) SGEOLT
406 FORMAT(' GEOLOGIST',10X,A80)
  IF (NCOORD .NE. 0) THEN
  IF (NCOORD .GT. 2) THEN
  WRITE(6,407)
407 FORMAT(' AREA COORDINATES (KM)')

```

```

NCOORD=NCOORD+1
C ROUTINE FOR LOCATING POINTS DOESNT REQUIRE CLOSURE
WRITE(6,408) (X(I),Y(I),I=1,NCOORD)
408 FORMAT(20X,4(F14.2,F9.2))
C ROUTINE FOR LOCATING POINTS DOESNT REQUIRE CLOSURE
ELSE
WRITE(6,516) IXMIN,IXMAX,IYMIN,IYMAX
516 FORMAT(' SEARCH RECTANGLE FROM',I5,'00ME TO',I5,'00ME AND FROM',
I6,'00MN TO',I6,'00MN')
ENDIF
ENDIF
IF (STDM) WRITE(6,409) TDMIN,TDMAX
409 FORMAT(' TOTAL DEPTH RANGE',F6.1,' TO',F6.1)
IF (SYM) WRITE(6,411) YMIN,YMAX
411 FORMAT(' YIELD RANGE',F6.2,' TO',F6.2)
IF (SCDF) WRITE(6,412) CDFMIN,CDFMAX
412 FORMAT(' CONDUCTIVITY (FIELD) RANGE',I6,' TO',I6)
IF (STDC) WRITE(6,413) TDCMIN,TDCMAX
413 FORMAT(' TDS RANGE',I6,' TO',I6)
IF (SGEOL(1:1) .NE. ' ') WRITE(6,410) SGEOL
410 FORMAT(' GEOLOGY',10X,A80)
DO 35 I=1,13
IF (SCH(I)) WRITE(6,414) PROMPT(I),CHMIN(I),CHMAX(I)
414 FORMAT(' RANGE FOR ',A8,' IS',F7.1,' TO',F7.1)
35 CONTINUE
LINE=0
C COUNTER FOR NUMBER OF LINES PRINTED
WRITE(5,119)
119 FORMAT(' PRINT ANALYSIS WHERE AVAILABLE (Y OR N)?')
READ(5,200) STYPE
IF (STYPE .NE. 'Y') THEN
C USE DIFFERENT FORMAT IF NOT WORRIED ABOUT ANALYSIS
DO 42 I=1,FCOUNT
READ(4,REC=FOUND(I)) BUFF
AYIELD=YIELD*0.01
TOTDP=TOTDEP*0.1
CASDP=CASDEP*0.1
C CONVERT TO LITRES/SEC AND METRES
IF (MOD(LINE,53) .EQ. 0) CALL HEAD
C HEAD PAGE
LINE=LINE+1
IF (TDSC .GT. 0) THEN
WRITE(6,300) NAME,LOCATN,DRILCO,QUAD,CARD,YEAR,GEOLST,EAST,
NORTH,TOTDP,CASDP,DEPSTR,SWL,AYIELD,CONDF,TDSC,ANAL,GEOL
300 FORMAT(1X,A22,' ',A27,' ',A8,' ',I2,' ',I3,' ',I2,' ',
A3,' ',I4,I5,4(' ',F5.1),' ',F5.2,' ',I5,' ',I5,
' ',I2,' ',A2)
ELSE
WRITE(6,301) NAME,LOCATN,DRILCO,QUAD,CARD,YEAR,GEOLST,EAST,
NORTH,TOTDP,CASDP,DEPSTR,SWL,AYIELD,CONDF,ANAL,GEOL
301 FORMAT(1X,A22,' ',A27,' ',A8,' ',I2,' ',I3,' ',I2,' ',
A3,' ',I4,I5,4(' ',F5.1),' ',F5.2,' ',I5,' ',I5,
' ',I2,' ',A2)
ENDIF
C USE DIFFERENT FORMAT IF THERE IS NO TDS
42 CONTINUE
ELSE
DO 43 I=1,FCOUNT
READ(4,REC=FOUND(I)) BUFF
C READ THE BASIC DATA

```

```

AYIELD=YIELD*0.01
TOTDP=TOTDEP*0.1
CASDP=CASDEP*0.1
C CONVERT BACK TO LITRES/SEC AND METRES
IF (ANAL.NE.0) THEN
  READ(2,REC=INANAL) CHEM
  IF (MOD(LINE,55).EQ.0) CALL HEAD2
C HEAD PAGE
  LINE=LINE+5
  IF (TDSC.GT.0) THEN
    WRITE(6,302) NAME,LOCATN,EAST,NORTH,QUAD,CARD,GEOLST,GEOL,YEAR,
      TOTDP,DEPSTR,SWL,CASDP,AYIELD,TDSC,CONDF,ANAL,
      DRILCO
302 FORMAT(' OWNER: ',A24,' LOCN: ',A30,' AMG: ',2I5,
  ' QUAD: ',I3,' CARD: ',I3,' INIT: ',A3,' GEOLOGY: ',A2,' YR: ',I3
  ' DEPTH: -TOTAL: ',F5.1,' STRUCK: ',F5.1,' SWL: ',F5.1,
  ' CASING: ',F5.1,' YIELD: ',F5.2,' TDS: ',I5,
  ' COND(FIELD): ',I6,' ANALYSES: ',I2,' DRILL CO: ',A8)
  ELSE
    WRITE(6,303) NAME,LOCATN,EAST,NORTH,QUAD,CARD,GEOLST,GEOL,YEAR,
      TOTDP,DEPSTR,SWL,CASDP,AYIELD,CONDF,ANAL,
      DRILCO
303 FORMAT(' OWNER: ',A24,' LOCN: ',A30,' AMG: ',2I5,
  ' QUAD: ',I3,' CARD: ',I3,' INIT: ',A3,' GEOLOGY: ',A2,' YR: ',I3
  ' DEPTH: -TOTAL: ',F5.1,' STRUCK: ',F5.1,' SWL: ',F5.1,
  ' CASING: ',F5.1,' YIELD: ',F5.2,' TDS: ',5X,
  ' COND(FIELD): ',I6,' ANALYSES: ',I2,' DRILL CO: ',A8)
  ENDIF
  WRITE(6,304) CHEM
304 FORMAT(' pH: ',F4.1,' CONDUCTIVITY: ',F7.1,' CO3: ',F7.1,
  ' HCO3: ',F7.1,' Cl: ',F7.1,' SO4: ',F7.1,' SiO2: ',F7.1,
  ' Ca: ',F7.1,' Mg: ',F7.1,' Fe: ',F7.1,' Al: ',F7.1,
  ' K: ',F7.1,' Na: ',F7.1,' TDS: ',F7.1,' Alk: ',F7.1,
  ' TOT OR PERM HARDNESS: ',F7.1,' TEMP HARDNESS: ',F7.1)
  ELSE
    IF (MOD(LINE,54).EQ.0) CALL HEAD2
    LINE=LINE+3
C HEAD PAGE
  IF (TDSC.GE.0) THEN
    WRITE(6,302) NAME,LOCATN,EAST,NORTH,QUAD,CARD,GEOLST,GEOL,YEAR,
      TOTDP,DEPSTR,SWL,CASDP,AYIELD,TDSC,CONDF,ANAL,
      DRILCO
  ELSE
    WRITE(6,303) NAME,LOCATN,EAST,NORTH,QUAD,CARD,GEOLST,GEOL,YEAR,
      TOTDP,DEPSTR,SWL,CASDP,AYIELD,CONDF,ANAL,
      DRILCO
  ENDIF
  ENDIF
  WRITE(6,305)
305 FORMAT(1X)
43 CONTINUE
  ENDIF
C END PRINT EITHER CLOSE SPACED SHORT LIST OR FULL LIST
  WRITE(6,415) FFEED
415 FORMAT(1A1,
  ' // ALL DEPTHS IN METRES'//
  ' // ALL ANALYSES IN mg/l'//
  ' // ALL CONDUCTIVITIES IN uS/cm'//
  ' // ALL YIELDS IN l/sec'//
  ' // GEOLOGY'

```

```

. /'      0 COASTAL          1 QUATERNARY          2 TERTIARY'
. /'      SANDS             ALLUVIUM              SEDIMENTS'
. /'      3 TRIASSIC        4 PERMIAN            5 MATHINNA'
. /'      SEDIMENTS'
. /'      6 ORDOVICIAN      7 GORDON             8 CAMBRIAN'
. /'      LIMESTONE'
. /'      9 PRECAMBRIAN     A PRECAMBRIAN        B TERTIARY'
. /'      DOLOMITE          BASALT'
. /'      C JURASSIC        D GRANITE'
. /'      DOLERITE'
WRITE(6,416)
416 FORMAT(1X)
STOP
END
SUBROUTINE NNERIC(STRING, IVAR, NVAR)
CHARACTER* (*) STRING
INTEGER*2 IVAR(2), NVAR
NVAR=0
ILAST=1
C COUNT POSITION ALONG INPUT LINE
  20 ICHAR=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+1)
300 FORMAT(A2)
  NNVAR=NNVAR+1
  DECODE(BUFF,301) IVAR(NVAR)
301 FORMAT(I2)
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
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 (ICHR .EQ. 0) GOTO 25
C HAVENT FOUND A MATCH BY THE END OF THE LINE
  IF (INDEX(RSTRNG, SSTRNG(ILAST:ILAST+ICHR-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
  INTEGER*2 FFEED
  DATA FFEED/3072/
C USED TO PUT FORM FEEDS IN OUTPUT FILE
  WRITE(6,100) FFEED
100 FORMAT(1A1/9X, 'OWNER', 9X, '1', 9X, 'LOCATION', 10X, '1 DRILCO 1',
. 2X, 'CARD', 'INIT', 9X, '1 TOT | CAS |DEPTH|', 11X, '1FIELD|', 5X, '1',
. 'ANI', /23X, '1', 27X, '1', 8X, 'QUAD ↑ YEAR ↑ | MAPREF ',
. '1 DEP | DEP | STR | SWL | YLD |COND | TDS |ALIGE|',
. /1X, 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
  SUBROUTINE HEAD2
C PUT A FORM FEED AT THE HEAD OF THE PAGE
  INTEGER*2 FFEED
  DATA FFEED/3072/
  WRITE(6,100) FFEED
100 FORMAT(A1)
  RETURN
  END
$TITL RANGE - FREE FORMAT OF 2 POSITIVE REALS SEPARATED BY , OR -
  SUBROUTINE RANGE(ANUM, IER, ILUN)
C A ROUTINE FOR INPUTTING 2 REAL VALUES IN ANUM.
C THE INPUT FILE IS SCANNED UNTIL 2 VALUES ARE READ OR THERE IS AN
C END OF LINE. NUMBERS ON THE SAME LINE MAY BE SEPARATED BY , OR -.
C THE ROUTINE REQUIRES POSITIVE NUMBERS. THE MAXIMUM LENGTH IS 80 CHAR
C ANUM(2) REAL
C ILUN INTEGER*4
C IER LOGICAL
C IER IS TRUE IF THERE IS AN ERROR OR EOF
C COUNT IS THE ACTUAL NUMBER OF VALUES SUCCESSFULLY TRANSFERRED
C USES ENCODE TO REFORMAT SUITABLE FOR F10.0
C
  REAL ANUM(2)
  INTEGER*2 A(10), ALINE(80), PLUS, MINUS,
  DPOINT, AL, K, I, J, IBOT, ZERO, NINE, POINT

```

```

INTEGER*4 ILUN, COUNT
REAL*4 BUFF(4)
LOGICAL IER
DATA PLUS/'1H+/', MINUS/'1H-/', DPOINT/'1H./', ZERO/'1H0/', NINE/'1H9'/
IER=.TRUE.
COUNT=0
C USED TO KEEP TRACK OF NUMBER OF VALUES FOUND
  READ(ILUN, 100, END=101, ERR=101) ALINE
  100 FORMAT(80A1)
C READ IN CHARACTER FORMAT - 80 CHARS
  IBOT=1
C USED TO MOVE STEADILY ALONG THE LINE
  29 CONTINUE
C TO HERE TO LOOK FURTHER ALONG A LINE
  IF (IBOT .GE. 81) GOTO 28
  DO 10 I=IBOT, 80
  AL=ALINE(I)
  IF (AL .EQ. PLUS .OR. AL .EQ. DPOINT
    .OR. (AL .GE. ZERO .AND. AL .LE. NINE)) GOTO 20
C LOOK FOR +, ., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 AS THE START OF A NUMBER
  10 CONTINUE
C IF FALL THROUGH HERE ARE AT THE END OF A LINE
  GOTO 28
C ONLY ONE NUMBER ON LINE WHEN SHOULD BE 2 SO SET BOTH TO 0
C
C NOW COPY THE NUMBER TO ITS END
  20 K=0
C COUNTER FOR A ARRAY
  POINT=0
C ENSURE HAVE ONE AND ONLY ONE DECIMAL POINT
  DO 11 J=I, 80
  AL=ALINE(J)
  IF (AL .EQ. DPOINT .AND. POINT .EQ. 1) GOTO 21
C FIND A SECOND DEC PT SO AT THE END OF THE NUMBER
  IF (AL .EQ. DPOINT) POINT=1
C FOUND A DECIMAL POINT
  IF (AL .NE. PLUS .AND. AL .NE. DPOINT
    .AND. (AL .LT. ZERO .OR. AL .GT. NINE)) GOTO 21
C TERMINATE NUMBER BY NOT A +, ., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  K=K+1
  IF (K .GE. 11) GOTO 101
C TOO MANY CHARACTERS
  A(K)=AL
  11 CONTINUE
C AT END OF NUMBER
  J=81
  21 IBOT=J
C NOTE SCAN START FOR NEXT NUMBER
  IF (POINT .EQ. 1) GOTO 22
C ALREADY HAVE A DECIMAL POINT
  K=K+1
  A(K)=DPOINT
C ADD A DECIMAL POINT IF NEEDED
C
C AND WRITE TO BUFFER AND REFORMAT
  22 IF (K .EQ. 10) GOTO 24
C NUMBER IS 10 CHARACTERS LONG
  K=K+1
  DO 23 I=K, 10
  A(I)=ZERO

```

```

23 CONTINUE
C FILL WITH ZEROS
24 CONTINUE
  ENCODE(BUFF,100) (A(I),I=1,10)
  DECODE(BUFF,111) ATMP
111 FORMAT(F10.0)
C HAVE THE NUMBER
  COUNT=COUNT+1
  ANUM(COUNT)=ATMP
  IF (COUNT .LT. 2) GOTO 29
  IER=.FALSE.
101 RETURN
20 ANUM(1)=0.
  ANUM(2)=0.
  GOTO 101
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 ORDERED FROM LEFT-TO-RIGHT.
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

```

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  

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
```