UR1982-40

1/62

1982/40. WBASE: a data base system for Tasmanian waters.

D.A. Polya

CONTENTS

1982/40. WBASE : a data-base system for Tasmanian waters.

D.A. Polya

Abstract

A data storage and retrieval system for Tasmanian waters has been devised and implemented on the Department's Perkin Elmer 6/16 computer. Identification, location, chemical, yield and other data may be stored for each water sample.

The data-base may be readily searched by specifying any of the following parameters: location, sample type, depth, flow rate, analysis reliability, rock type, salinity. A search of the database takes approximately 50 seconds/1000 samples stored.

Records found by a specified search may be transferred to a random access file for ready use by programmes that could calculate ionic ratios and chemical speciation and could be potentially used for plotting chemical analyses of selected waters.

INTRODUCTION

Data on several thousand Tasmanian underground and surface waters are held by the Department of Mines. The most commonly held and used information refers to sample location, type, depth, salinity, flow rate and associated rock types. Complete major chemical analyses for several hundred of these samples exist.

This information is disseminated between Underground Water Supply Papers, Bulletins, Technical and Unpublished Reports and a number of other unpublished compilations. Much of the data is available on a card system. The most complete compilation of borehole waters is that of W.L. Matthews.

There is a need for water data to be stored in a single permanent system that is easily searched and in such a form that information (especially numerical data) may be readily transferred to analytical systems. The data-base system (WBASE) has been written to fulfill this need.

The WBASE system consists of a main (random access) data file (WB), several utility files and a number of programme modules. Each programme module executes a specific set of procedures (*e.g.* add data to WB, correct data in WB, write data from WB, search data in WB).

WHAT THE SYSTEM CAN DO

The system stores water sample data. Using the appropriate programmes it is also possible to:

(1) Add water sample data to the data-base
(2) Correct previously entered data
(3) Search the data-base for water samples with specified characteristics (*e.g.* location, salinity)
(4) Print out stored data - either the entire data-base or portions of it
(5) Transfer data of samples found by a search to a separate random access data file (for use by plotting programmes *etc.*)

INFORMATION STORED IN THE DATA BASE

A list of parameters that can be stored in the data-base is given below. Restrictions on the form data can take are also detailed.

Launceston analysis number: a 6 digit reference number given to each water sample analysed by the Department's Launceston laboratory. The first two digits indicate the year of the analysis. If a sample has been analysed elsewhere a value of '000000' is given to this parameter.

Source of data: a 6-character code for the original report of compilation from which the water data was taken. For example, the code used for Jane Mackey's Oatlands spring and borehole sampling programme is MAKY76. A list of these codes should be made.

Reference number in source: up to 8 characters may be assigned to the identification number of the water sample in the cited source of data.

Quadrangle number: number of 1:50 000 map sheet. Numbers range from 1 to 97. A map showing the quadrangle number is included in the back of the report.

100 000 m^2 designation: 2-letter designation of 100 km^2 in the Australian Map Grid. These may be obtained from TASMAP 1:100 000 sheets.

Easting, Northing: 3 digit co-ordinates are used, corresponding to tens of kilometres, kilometres and hundreds of metres. 4-digit co-ordinates are calculated from the Easting, Northing and 100 km^2 designation.

Rock types: up to four rock types may be input by designated 2 digit codes. Some codes already used are:

QQ	Quaternary sediments, undifferentiated
TB	Tertiary basalt
TR	Triassic sediments, undifferentiated
TS, TQ, TF	Triassic units Rs, Rq and Rf respectively according to Oatlands 1:50 000 sheet legend
PM	Permian sediments
JD	Jurassic dolerite

Reliability of rock type data: a single character code for the source of the rock type data.

G	From geologist's report or other communication
D	From driller's log or field assistant's report
M	Guessed from 1:50 000 geological map
	Left blank if unknown

Analysis reliability: a single character code.

R	Reliable complete (pH, TDS, HCO_3, Cl, SO_4, SiO_2, Ca, Mg, Na, K) analysis
D	Reliable complete analysis; some major components below detection limits
E	TDS determined from qualitative estimate
P	Partial analysis
C	TDS from conductivity measurements
T	Complete analysis but poor matching of cation and anion totals
	Left blank if unknown

Location reliability: a single character code.

R	Reliable grid reference
K	Grid reference reliable to within one kilometre
T	Grid reference reliable to within ten kilometres
	Left blank if unknown

Locality: name (up to 16 characters)

Region: up to 16 characters for the name of the region from which the sample was taken. To facilitiate searching the number of region names used should be restricted.

Number of rock types: 1-digit number of rock types associated with the water sample. For a bore water this means the rocks encountered in the bore; for a surface sample the rocks through which the water has run might be considered.

Type of sample: a 1-character code for the sample type may be stored. These codes are:

B	Borehole or well water
S	Spring
R	River or runoff
P	Precipitate
L	Leachate
D	Dam water

Flow: the flow in gallons/hour of the bore, spring *etc.* from which the sample was taken. If no data is available the flow is set to zero.

Depth of borehole: expressed in metres. If no data is available, or if inapplicable this depth is set to zero.

Depth to water: as above.

Conductivity: expressed in µS/cm. The value measured at 20°C should be taken.

Eh: space is set aside for this parameter in the data-files WB and FNDWB. Eh measurements are rarely carried out at present so there is no facility to enter this data. Eh is set to a default value of zero for every water sample.

pH: if data are not available, this parameter is set to zero.

Total dissolved solids, total dissolved carbonate, Cl, SO_4, SiO_2, Ca, Mg, K, Na, Fe, Al, NO_3, F: analyses in ppm (mg/*l*) of these components. If no analysis was done, the value of a particular component is set to zero. If a component was analysed for but was found to be less than the detection limit, then the negative of the detection limit in ppm should be entered for that component (*e.g.* if SO_4 $\gtrsim$5ppm, then enter -5.0 for SO_4).

Note that total dissolved carbonate is stored rather than carbonate and bicarbonate. This is because the storage of all of pH, (CO_3^-) and (HCO_3^-) is redundant. From pH and total dissolved carbonate (TCO3), the concentrations of carbonate and bicarbonate, as well as other aqueous carbonate species, may be calculated.

Calculated molarities: molarities of the components above are calculated when the analyses are entered.

HOW TO USE THE SYSTEM

The data-base system, WBASE, is implemented on the Perkin Elmer 6/16 on the ENGINEERING GEOLOGY PACK 1 disc. This section explains:

(1) General use of the computer : loading disc, running programme modules, running down disc.

(2) Detailed use of each of the system programme modules: ADTOWB, CRCTWB, RITEWB, and SRCHWB.

General use of the computer

A set of instructions is given below for the running of any of the system programmes. It is possible that some of these procedures may change as the computer facility is improved; if in any doubt as to any of the steps, the supervisor should be consulted.

(1) *Load ENGNG GEOL Disc 1*

(2) *Remove write protection of cartridge disc (C6).* Wait until upper PROT light goes out.

(3) *Run appropriate system programme module:* you may run one of the programme modules ADTOWB, CRCTWB, RITEWB, or SRCHWB. This is done by typing in RU, then a blank, then the module name and then pressing the RETURN key.

e.g. *RU ADTOWB

(4) *Respond to programme module prompts:* these are discussed in detail below.

(5) *Close neatly all files on completion of run:* you will know when a programme module run is finished when STOP and EOJ appear on the terminal. When this happens please type in CL and press the RETURN key.

(6) *Run down disc* or *run another programme module:* if you wish to run another programme module or the same module again then repeat steps (3)-(5). If you have completed use of the computer switch the upper PROT switch to PROT and switch from RUN to LOAD.

How to add data to the system: ADTOWB

Run the programme module ADTOWB if you wish to add to the store of water sample information. Data for each sample are entered in response to terminal prompts. At the end of each sample input, the data for that sample are displayed on the terminal so that corrections can be made. Rapid input of data may be facilitated by use of a transcription sheet (fig. 1).

Unfortunately, the design of the program language requires that input must be in a rigorous format. The use of the free format routines INFREE, NUMIN, FRINIT and FREEIN solves this problem to some extent, but

```
STORED DATA FOR THE WATER SAMPLE NUMBER,□□□□□

(1)LAUNCESTON ANALYSIS NUMBER   □□□□□□
(2)DATA SOURCE,IDENTIFICATION IN SOURCE:□□□□□□/□□□□□□□□

(3)QUADRANGLE NUMBER:                      □□          (16)TDS (PPM):□□□□□⊡□
(4)AMG 100M GRID REFERENCE:       □□ □□□/□□□          (17)TCO3(PPM):□□□□□⊡□
(5)LOCALITY:                 □□□□□□□□□□□□□□□□          (18)CL  (PPM):□□□□□⊡□
(6)REGION:                   □□□□□□□□□□□□□□□□          (19)SO4 (PPM):□□□□□⊡□
(7)ROCK TYPE DATA:             □ □□ □□ □□ □□ □          (20)SIO2(PPM):□□□□□⊡□
(8)CODES:                                □ □ □          (21)CA  (PPM):□□□□□⊡□
(9)DATA RELIABILITY:                       □ □          (22)MG  (PPM):□□□□□⊡□
(10)TYPE OF SAMPLE:                          □          (23)K   (PPM):□□□□□⊡□
(11)FLOW(G/H):                          □□□□□⊡          (24)NA  (PPM):□□□□□⊡□
(12)DEPTH OF BOREHOLE(m):               □□□□⊡□          (25)FE  (PPM):□□□□□⊡□
(13)DEPTH TO WATER(m):                  □□□□⊡□          (26)AL  (PPM):□□□□□⊡□
(14)CONDUCTIVITY (uS/cm):              □□□□□⊡□          (27)NO3 (PPM):□□□□□⊡□
(15)pH:                                   □□⊡□          (28)F   (PPM):□□□□□⊡□
```

Figure 1. *Transcription sheet.*

care must nevertheless be taken in the way information is typed in.

Below some prompts, a sequence of asterisks and blanks is displayed. The data should be typed directly below these asterisks.

```
e.g. TYPE IN AMG 100 M GRID REFERENCE: SQUARE, E, N      ) Terminal prompt
     ** ***/***                                          )
     EP 364 291                                          )    response
```

In most cases it is obvious what the nature of the input should be (*e.g.* chemical analyses should be numbers). However

(1) to facilitate searching the input for SOURCE, LOCALITY and REGION should be left-justified;

(2) if the SOURCE REFERENCE parameter is right-justified than this parameter might be more readily used at a later date as a number rather than merely a character string.

Speed of data input might be improved by noting that if a line of input consists entirely of blanks or entirely of numbers whose value is zero, then this line of data may be entered merely pressing the RETURN after the appropriate prompt.

The correction procedure requires you to input the number of parameters you wish to change and then codes for each of these parameters (these code numbers are displayed on the terminal when the data are checked). If you type in less codes than you indicated you would, then the programme will stop. The run may be continued by pressing BREAK twice and then RETURN.

How to correct previously entered data: CRCTWB

Run the programme module CRCTWB if you wish to correct water sample data that have been entered in a previous run. The correction procedure is identical to that in ADTOWB with the exception that a record number must be supplied before a record can be corrected.

How to search the data-base: SRCHWB

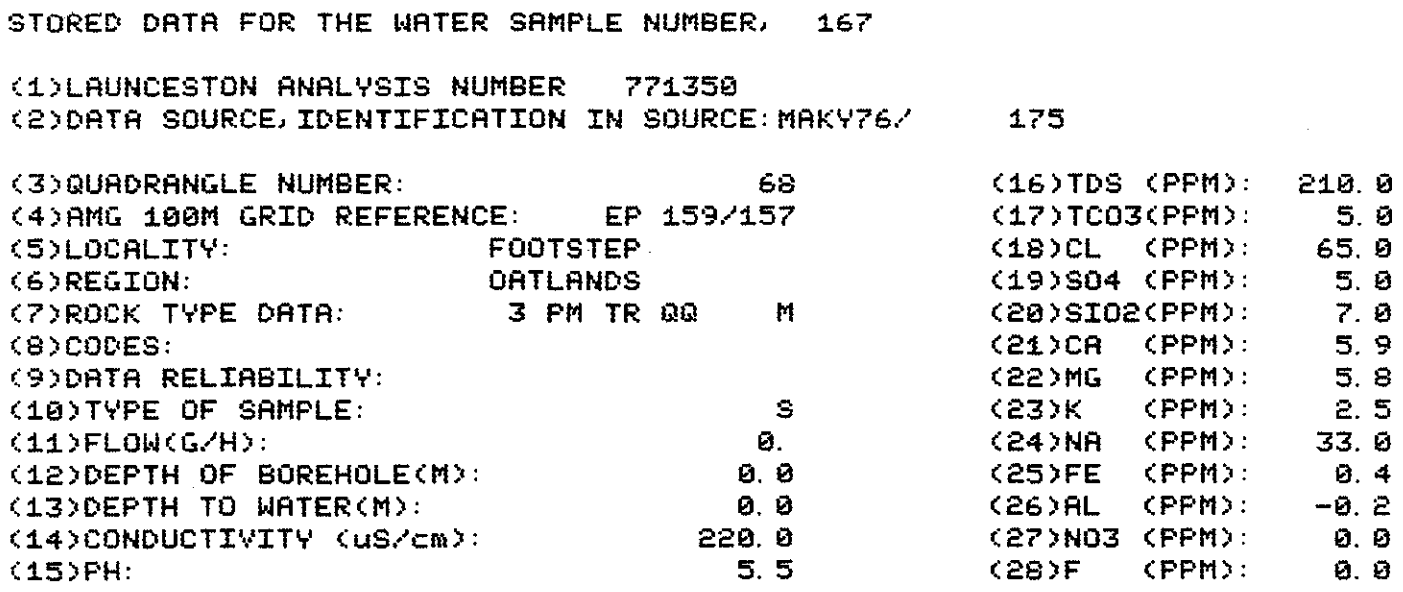
Two types of search may be conducted using the programme module SRCHWB:

(1) *Data-file search:* whereby characteristics of the required water samples are specified. Only the records of those samples satisfying all these specifications will be found by the search. This search takes about 50 seconds/1000 records in the main data-file WB.

(2) *Combination search:* whereby two previous searches may be combined. The method of combination may be a union (that is, all records found by either search) or an intersection (that is, only those records found by both searches). This search takes about 17 seconds/1000 records in WB.

It is possible to search by the following:

(*a*) Location: quadrangle number, region, locality and/or a co-ordinated area (*i.e.* an area defined by corners whose AMG 100 m co-ordinates you specify).



```
STORED DATA FOR THE WATER SAMPLE NUMBER,   167

(1)LAUNCESTON ANALYSIS NUMBER   771350
(2)DATA SOURCE,IDENTIFICATION IN SOURCE:MAKY76/     175

(3)QUADRANGLE NUMBER:                       68          (16)TDS (PPM):  210. 0
(4)AMG 100M GRID REFERENCE:     EP 159/157              (17)TCO3(PPM):    5. 0
(5)LOCALITY:               FOOTSTEP                     (18)CL  (PPM):   65. 0
(6)REGION:                 OATLANDS                     (19)SO4 (PPM):    5. 0
(7)ROCK TYPE DATA:          3 PM TR QQ     M            (20)SIO2(PPM):    7. 0
(8)CODES:                                               (21)CA  (PPM):    5. 9
(9)DATA RELIABILITY:                                    (22)MG  (PPM):    5. 8
(10)TYPE OF SAMPLE:                         S           (23)K   (PPM):    2. 5
(11)FLOW(G/H):                             0.           (24)NA  (PPM):   33. 0
(12)DEPTH OF BOREHOLE(M):                 0. 0          (25)FE  (PPM):    0. 4
(13)DEPTH TO WATER(M):                    0. 0          (26)AL  (PPM):   -0. 2
(14)CONDUCTIVITY (uS/cm):               220. 0          (27)NO3 (PPM):    0. 0
(15)PH:                                   5. 5          (28)F   (PPM):    0. 0
```

Figure 2. *Typical proof sheet for a water sample data record in WB.*

(*b*) Range of flow rates.
(*c*) Range of salinities.
(*d*) Range of water depths.
(*e*) Water sample type.
(*f*) Rock type.
(*g*) Analysis reliability.
(*h*) Codes: any of the three as yet unspecified code parameters.

At the end of each search, the number of records found is displayed. Up to ten searches may be conducted in a single run. Before the end of run, you have the option to transfer the data from just one search to a random access data-file FNDWB; this should be done if a printout of these records is desired.

How to write out records from the data-base: RITEWB

Run the programme module RITEWB if you wish to obtain a printout of all or part of the data-base. The following options are available:

(1) Printout all of the records in the water-analysis data-file.

(2) Printout all of the records added to the water-analysis data-file in the latest update. This option may be used to obtain proof sheets of a large amount of added data for checking.

(3) Printout the records found by a search and transferred to the data-file FNDWB during the run of programme module SRCHWB.

(4) Printout a number of individual records. The number of records cannot exceed 50.

The printout may be either in a rather complete form or in the form of a proof-sheet (fig. 2). The type of printout is set according to the response to the appropriate prompt.

Before printing out, the number of records that will be printed is displayed on the terminal. If you then do not want all the records printed out type in PA and press the RETURN key.

DETAILS OF THE SYSTEM

Introduction, files, variables

The data-base system has been designed for implementation on the Geological Survey Perkin Elmer 6/16. The source files are written in a non-standard form of FORTRAN IV; the following non-standard routines from the FORLIB library are used: SYSCOM, POSITN, INFREE, FREEIN, NUMIN and FRINIT. In addition a modified version of FSTJDN (a collection of subroutines and functions from the Department's programme collection) has been incorporated in the search programme.

The system consists of 17 files: 3 main random-access data-files and 14 utility files (data and source files and programme modules). There are approximately 2300 lines of FORTRAN in the source files and the programme modules occupy over 800 records of code. The utility files use approximately 30 cylinders of disc space whilst the three random access files will use about 90 cylinders per 1000 water samples stored.

Details of the nature and purpose of each of the files are given in Table I.

Table 1. NATURE AND PURPOSE OF WBASE SYSTEM FILES

File name	Type	Record Length	Cyl.	Records	Programme modules in which file is:		Remarks
					read	written over	
WB	RA	218	20	MAXNOS	ADTOWB CRCTWB RITEWB SRCHWB	ADTOWB CRCTWB	The main data-file of water sample information One record per water sample. No file-mark at end.
WBLOG	RA	40	10	MAXNOS	ADTOWB SRCHWB	ADTOWB SRCHWB	Contains results of 10 searches of each water sample record stored in WB. Search results are initially set to FALSE. Data-file consists solely of logical variables. One record per water sample. Ten search results (TRUE or FALSE) per sample. No file-mark at end.
FNDNB	RA	218	20	FMAXNS	RITEWB SRCHWB	SRCHWB	Contains complete records of water samples found by a specified search (using SRCHWB module). No file-mark at end.
WBD1	D	5	1	2	ADTOWB CRCTWB RITEWB SRCHWB	ADTOWB	Stores MAXNOS
WBD2	D	5	1	2	ADTOWB	ADTOWB	Stores MAXNOS
WBD3	D	5	1	2	ADTOWB RITEWB	ADTOWB	Stores PMAXNS
WBD4	D	5	1	2	RITEWB SRCHWB	SRCHWB	Stores FMAXNS
WBD5	D	5	1	2	-	-	Stores a zero
WBD6	D	50	1	2	RITEWB	RITEWB	Stores NFOUND (10)
WBS1	Source	80	3	647	-	-	Source file for ADTOWB
WBS2	Source	80	3	459	-	-	Source file for CRCTWB

Table 1. (continued)

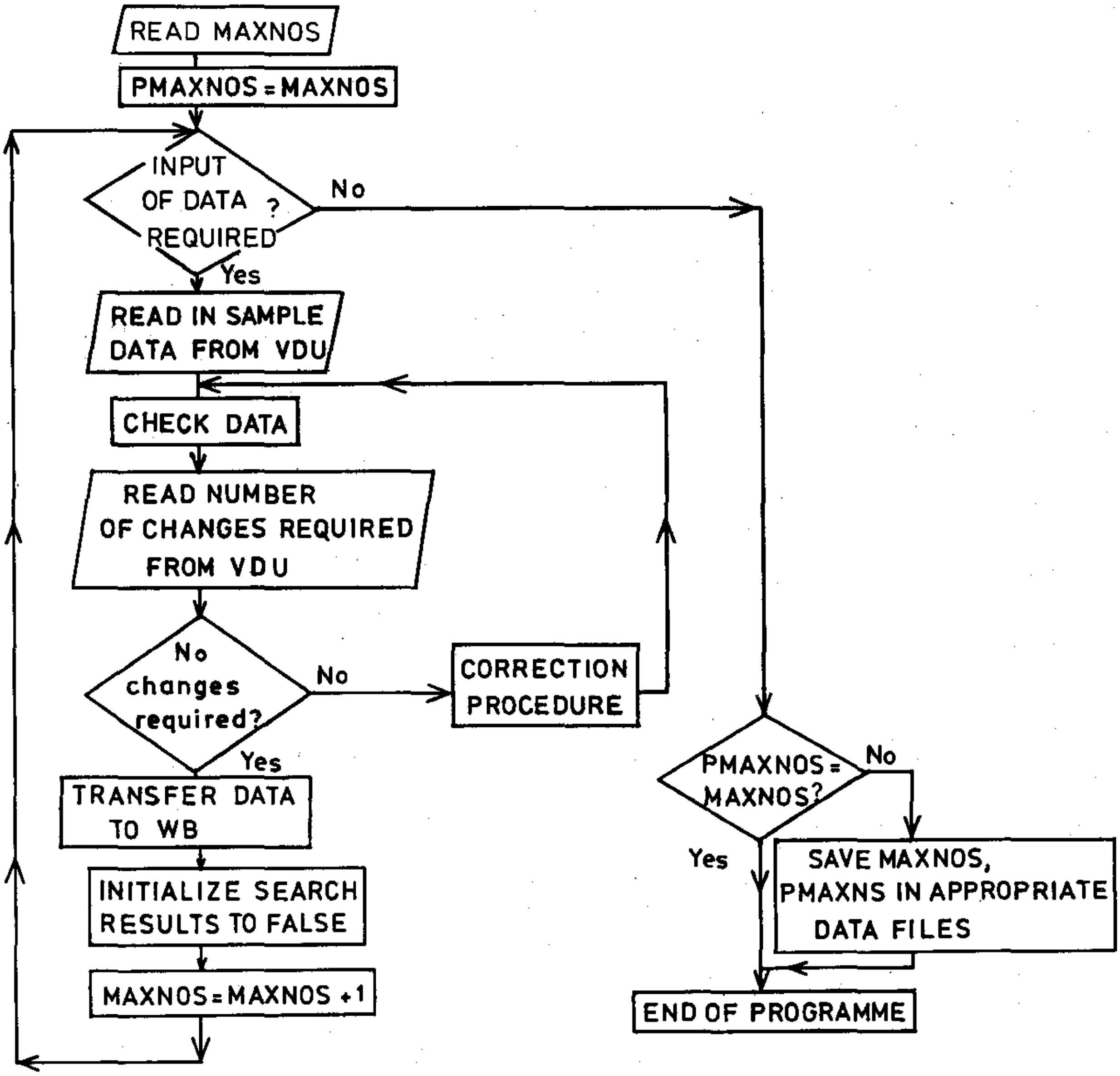
File name	Type	Record Length	Cyl.	Records	Programme modules in which file is:		Remarks
					read	written over	
WBS3	Source	80	3	367	-	-	Source file for RITEWB
WBS4	Source	80	5	814	-	-	Source file for SRCHWB
ADTOWB	Module	108	2	221	-	-	For adding water sample data to WB,
CRCTWB	Module	108	2	184	-	-	For correcting previously entered data in WB
RITEWB	Module	108	2	136	-	-	For printing out records in WB or FNDWB
SRCHWB	Module	108	3	284	-	-	For searching the data-base WB

Cyl. - cylinders

Table 2. VARIABLES STORED IN UTILITY FILES

Variable	Explanation	Storage	Format in data-file	Remarks
MAXNOS	Number of records stored in the main water-analysis data-file WB	WBD1, WBD2	I5	Initially zero. Incremented by one after every transfer of a new record to WB by the module ADTOWB. Storage of MAXNOS in WBD2 is a safety precaution in case of WBD1 being accidently removed.
PMAXNS	Number of records stored in the main water-analysis data-file WB *before* the last update.	WBD3	I5	Initially zero. Set equal to MAXNOS at the beginning of each run of the module ADTOWB. If no records are added during the module run then PMAXNS is set back to its previous value.
FMAXNS	Number of records in the data-file FNDWB.	WBD4	I5	Initially zero. Value is reset after a transfer of a specified search to FNDNB by the module SRCHWB.
--	Zero	WBD5	I5	Zero stored for rapid initialisation of utility files.
NFOUND	An array of 10 integers: corresponding to the number of records found by each of searches 1 to 10.	WBD6	10I5	Initially all values in the array are zero. NFOUND (10) is updated in SRCHWB after every successful search. The updated array is transferred to WBD6 after the last search of each SRCHWB run.

Flow Sheet: WBS1 (ADTOWB)



The three random access files, WB, WBLOG and FNDWB, are the major data files and are manipulated by the programme modules ADTOWB, CRCTWB, SRCHWB and RITEWB (whose names should be self-explanatory). Six small data files (WBD1...WBD6) are used to store file control variables (see Table 2).

Explanation of programme modules

ADTOWB (see flow sheet for WBS1): reads in data to be added to WB. At the beginning of the programme MAXNOS, the number of records in WB is read and PMAXNS is set to this value. If data input is required then prompts for reading in data are written to the terminal; at the completion of input, the record for the water sample is displayed in proof-sheet form; if changes are required then a correction procedure enables this to be done. The checking and correction procedures may be repeated indefinitely until it is indicated that the data is OK.

If the data is OK then it is transferred to WB; all search results for the latest record are set to FALSE and these values are transferred to WBLOG; and MAXNOS is incremented by one.

New records may be input indefinitely (provided there is space available) until it is indicated that more input is not required, whereupon the updated values of MAXNOS and PMAXNS are transferred to the appropriate data-files. If no records are added then PMAXNS is reset to its initial value and MAXNOS remains unaltered.

Most of the data is read in by free-format routines. Error loops have been set up in several parts of the input procedure to ensure that:

(1) Quadrangle number lies between 1 and 97.

(2) 100 km^2 designation corresponds to one in Tasmania (*i.e.* Easting letter is B, C, D, E, or F; Northing letter is M, N, P, Q, R, or S).

(3) Rock reliability code is D, G, M, or a blank.

(4) Analysis reliability code is R, D, E, P, C, T, or a blank.

(5) Location reliability code is R, K, T, or a blank.

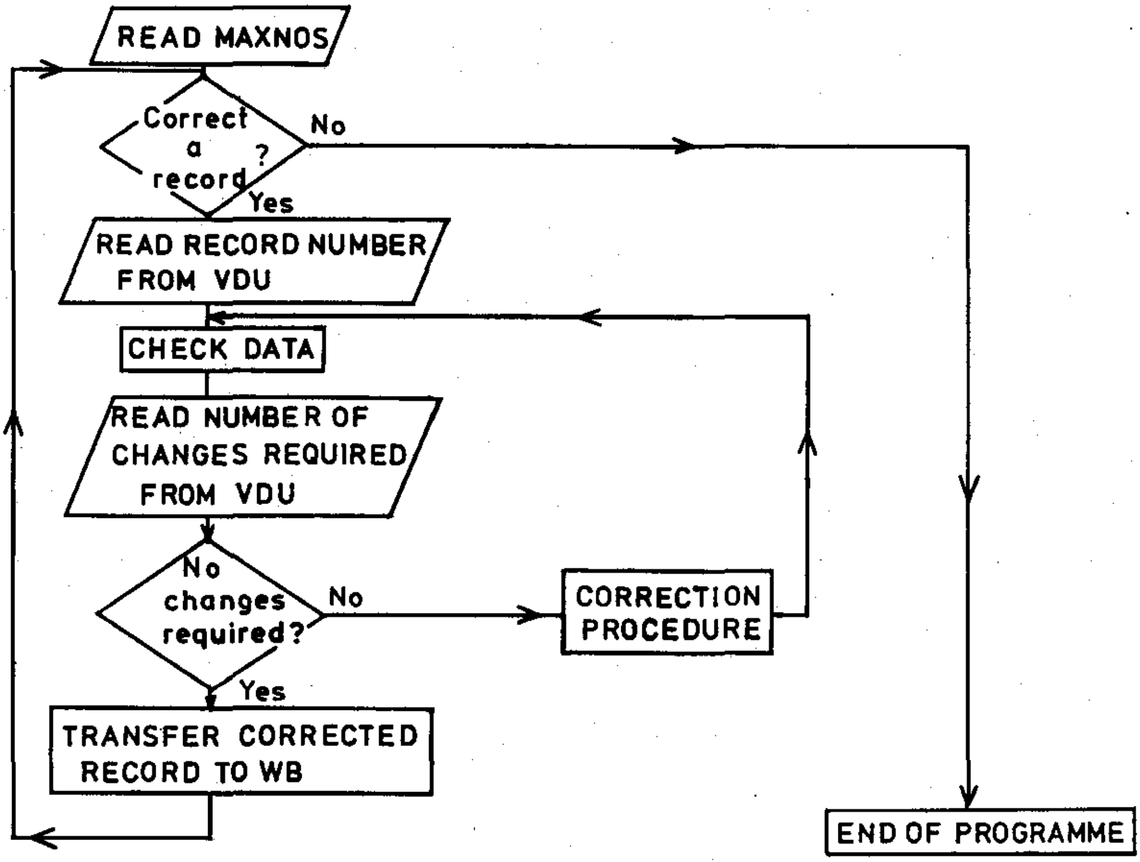
(6) Type code is B, S, D, R, P, L, or a blank.

These checks are required to ensure that all variables printed out by RITEWB are defined.

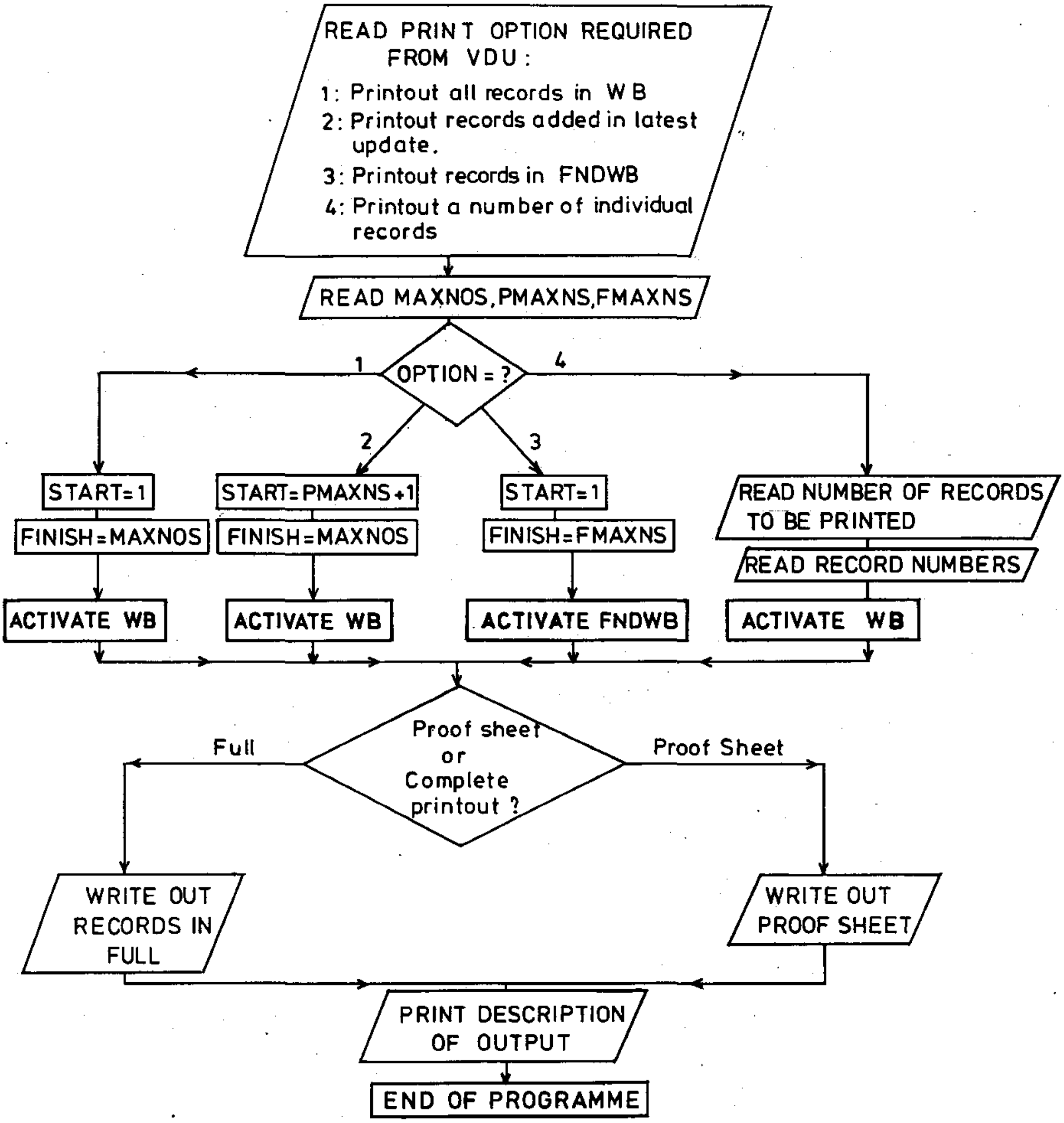
A list of the variables stored in WB is given in the programme listing of WBS1. Note that ALINE (109) is equivalent to all these variables and is merely used to simplify read and write statements.

CRCTWB (see flow sheet for WBS2): corrects previously entered data in WB. At the beginning of the programme MAXNOS is read. If correction of a record is required, then the number of this record will be requested. Error loops ensure that this number is non-negative and does not exceed the maximum record number in WB. Data is then checked and corrected by the same procedure as in ADTOWB. Note however that MAXNOS is not altered and the file WBLOG is not accessed. When no further correction of records is required the programme run terminates.

Flow Sheet: WBS2 (CRCTWB)



Flow Sheet: WBS3 (RITEWB)


READ PRINT OPTION REQUIRED FROM VDU:
1: Printout all records in WB
2: Printout records added in latest update.
3: Printout records in FNDWB
4: Printout a number of individual records
READ MAXNOS, PMAXNS, FMAXNS
OPTION = ?
1
4
2
3
START=1
FINISH=MAXNOS
ACTIVATE WB
START=PMAXNS+1
FINISH=MAXNOS
ACTIVATE WB
START=1
FINISH=FMAXNS
ACTIVATE FNDWB
READ NUMBER OF RECORDS TO BE PRINTED
READ RECORD NUMBERS
ACTIVATE WB
Proof sheet or Complete printout ?
Full
Proof Sheet
WRITE OUT RECORDS IN FULL
WRITE OUT PROOF SHEET
PRINT DESCRIPTION OF OUTPUT
END OF PROGRAMME

RITEWB (see flow sheet for WBS3): writes out records from WB and FNDWB. A prompt written to the terminal asks which of four printing options (1 - all records in WB, 2 - latest update of WB, 3 - all records in FNDWB, 4 - a number of individual records) is required. The file control variables MAXNOS, PMAXNS and FMAXNS are read.

According to option chosen one of four routines is carried out. Each routine sets up the counting variables (START, FINISH) for the DO-loops in which printing is carried out. In addition the appropriate data-file is activated. There are a number of checks in the fourth of the procedures to ensure that only records that exist in WB are requested.

According to the response to the appropriate prompt either a rather complete printout or a proof-sheet of all the requested records is printed. In the more complete printout quadrangle numbers and some codes are replaced by full names and explanations.

SRCHWB (see flow sheet for WBS4): searches data in WB. It should be noted that the number of searches is limited to ten and that every time a search is run a search from a previous run is removed.

The data base search is carried out record by record by testing whether or not:

(1) Search of a particular parameter is required.

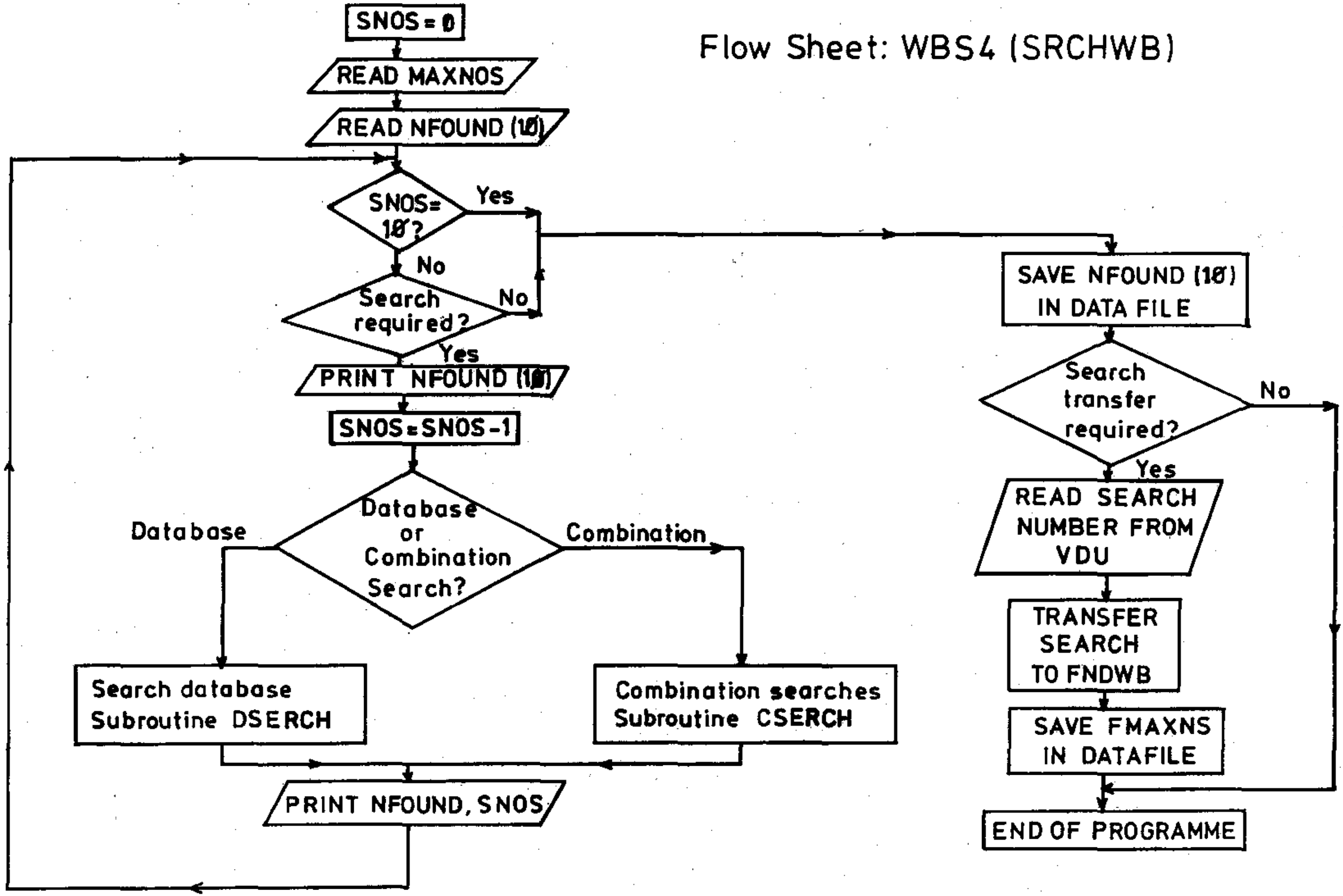
(2) If a search is required, the search parameter is the same as the parameter in the record.

If (1) is false then the search proceeds to the next parameter, if (1) is true then statement (2) is tested. If (2) is false then the search procedure is quite, TFOUND (record number) is set to FALSE and the next record is searched. After all records in WB have been searched the search results, stored in the array TFOUND (5000), are transferred to WBLOG.

Some of the more important variables in this programme are defined below:

SNOS	Number of the search being carried out
TFOUND (5000)	Array of search results for one search of all the records in WB
FOUND (10)	Array of search results for one record in WB
STRANS	Number of search to be transferred to FNDWB

Flow Sheet: WBS4 (SRCHWB)



```
$TITL WBS1
$DBUG
C
C SOURCE FILE FOR "ADTOWB"
C
C "ADTOWB" READS WATER SAMPLE ANALYSES AND DATA FROM THE VDU
C AND TRANSFERS THESE DATA TO THE RANDOM ACCESS FILE "WB"
C
C THE INPUT DATA FOR EACH RECORD IN DISPLAYED ON THE VDU
C FOR CHECKING BEFORE EACH TRANSFER IS MADE
C
C A VALUE OF .FALSE. IS GIVEN FOR SEARCHES 1-10 OF EACH NEW RECORD
C AND THESE VALUES ARE TRANSFERRED TO THE SEARCH RESULT FILE "WBLOG"
C
C MAXNOS, THE NUMBER OF SAMPLE RECORDS IN "WB", IS STORED IN
C THE FILES "WBD1" AND "WBD2"(FOR SAFETY)
C
C PMAXNS, THE NUMBER OF SAMPLE RECORDS IN "WB" BEFORE THE LATEST UPDATE
C IS STORED IN THE FILE "WBD3"
C
C LOGICAL UNIT ASSIGNMENTS:
C 01-WBD1  -STORES MAXNOS
C 02-WB    -RANDOM ACCESS DATA FILE
C 03-WBD3  -STORES PMAXNS
C 04-WBLOG -RANDOM ACCESS FILE OF SEARCH RESULTS
C 05-VDU   -INPUT
C 06-VDU   -PROMPTS
C 07-WBD2  -STORES MAXNOS
C 08-99    -CORE BUFFER
C
C SUBROUTINES: 0
C FORLIB ROUTINES: SYSCOM,POSITN,INFREE,NUMIN,FRINIT
C
      DOUBLE PRECISION
     *                LTNNOS,REFNOS,REF
      REAL
     *    LOCLTY(4),REGION(4),
     *    OMEGA,EH,PH,TCO3,CL,SO4,SIO2,CA,MG,K,NA,TDS,FE,AL,NO3,F,
     *    MTCO3,MCL,MSO4,MSIO2,MCA,MMG,MK,MNA,MFE,MAL,MNO3,MF,
     *    FLOW,DHOLE,DWATER
      INTEGER*2
     *         WBNOS,TYPE,QUADNO,SQUARE,EAST,NORTH,
     *         ECOORD,NCOORD,
     *         NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,
     *         CODE1,CODE2,CODE3,ANALRL,LOCRL,
     *         E100KM,N100KM,CHARA,CHARM,CHARN,
     *         CHARB,CHARC,CHARD,CHARE,CHARF,CHARG,CHARK,
     *         CHARL,CHARP,CHARQ,CHARR,CHARS,CHART,BLANK,
     *         MAXNOS,PMAXNS,IREC,ALINE(109),
     *         MOREIP,DATAOK,YES,NO,
     *         CHANGE(28),COUNT,NCHNGE
C
      LOGICAL
     *       FOUND(10),IER,EWRONG,NSRONG,OUTSYD,AWRONG,LWRONG,RECHEQ
```

```
C
      EQUIVALENCE
     *(ALINE(1),LTNNOS),(ALINE(5),REFNOS),(ALINE(9),REF),
     *(ALINE(13),WBNOS),(ALINE(14),TYPE),(ALINE(15),QUADNO),
     *(ALINE(16),SQUARE),(ALINE(17),EAST),(ALINE(18),NORTH),
     *(ALINE(19),ECOORD),(ALINE(20),NCOORD),
     *(ALINE(21),LOCLTY(1)),(ALINE(29),REGION(1)),
     *(ALINE(37),NROCK),(ALINE(38),ROCK1),(ALINE(39),ROCK2),
     *(ALINE(40),ROCK3),(ALINE(41),ROCK4),(ALINE(42),ROCKRL),
     *(ALINE(43),CODE1),(ALINE(44),CODE2),(ALINE(45),CODE3),
     *(ALINE(46),ANALRL),(ALINE(47),LOCRL),
     *(ALINE(48),PH),(ALINE(50),TCO3),(ALINE(52),CL),
     *(ALINE(54),SO4),(ALINE(56),SIO2),(ALINE(58),CA),(ALINE(60),MG),
     *(ALINE(62),K),(ALINE(64),NA),(ALINE(66),TDS),
     *(ALINE(68),FE),(ALINE(70),AL),(ALINE(72),NO3),(ALINE(74),F),
     *(ALINE(76),OMEGA),(ALINE(78),EH),(ALINE(80),MTCO3),
     *(ALINE(82),MCL),(ALINE(84),MSO4),(ALINE(86),MSIO2),(ALINE(88),CA),
     *(ALINE(90),MMG),(ALINE(92),MK),(ALINE(94),MNA),
     *(ALINE(96),MFE),(ALINE(98),MAL),(ALINE(100),MNO3),(ALINE(102),MF),
     *(ALINE(104),FLOW),(ALINE(106),DHOLE),(ALINE(108),DWATER)
C
C
      DATA YES,NO/1HY,1HN/
      DATA CHARA,CHARB,CHARC,CHARD,CHARE,CHARF,CHARG,CHARK,
     *     CHARL,CHARM,CHARN,CHARP,CHARQ,CHARR,CHARS,CHART/
     *1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HK,1HL,1HM,1HN,1HP,1HQ,1HR,1HS,1HT/
      DATA BLANK/1H /
C
C
C LTNNOS   LAUNCESTON WATER ANALYSIS NUMBER
C REFNOS   REFERENCE NUMBER IN DATA SOURCE
C REF      SOURCE OF DATA
C WBNOS    "WB" DATAFILE RECORD NUMBER
C TYPE     SAMPLE TYPE(SPRING,B/HOLE,DAM,RIVER,PRECIPITATION,LEACHATE)
C QUADNO   1:50 000 QUADRANGLE NUMBER
C SQUARE   100 000 M SQUARE DESIGNATION
C EAST     EASTING (10KM-1KM-100M)
C NORTH    NORTHING(10KM-1KM-100M)
C ECOORD   EASTING (100KM-10KM-1KM-100M)
C NCOORD   NORTHING(100KM-10KM-1KM-100M)
C LOCLTY   LOCALITY OF SAMPLE
C REGION   REGION
C NROCK    NUMBER OF MAJOR ROCK TYPES ASSOCIATED WITH SAMPLE
C ROCK1    ROCK TYPE NUMBER 1
C ROCK2    ROCK TYPE NUMBER 2
C ROCK3    ROCK TYPE NUMBER 3
C ROCK4    ROCK TYPE NUMBER 4
C ROCKRL   RELIABILITY OF ROCK-TYPE DATA
C CODE1    UNSPECIFIED CODE 1.
C CODE2    UNSPECIFIED CODE 2.
C CODE3    UNSPECIFIED CODE 3.
C ANALRL   RELIABILITY OF ANALYSIS
C LOCRL    RELIABILITY OF LOCATION DATA
C PH       PH OF SAMPLE
```

```
C TCO3      TOTAL DISSOLVED CARBONATE
C CL        CHLORIDE (PPM)
C SO4       SULPHATE (PPM)
C SIO2      SILICA (PPM)
C CA        CALCIUM (PPM)
C MG        MAGNESIUM (PPM)
C K         POTASSIUM (PPM)
C NA        SODIUM (PPM)
C TDS       TOTAL DISSOLVED SOLIDS (PPM)
C FE        IRON (PPM)
C AL        ALUMINIUM (PPM)
C NO3       NITRATE (PPM)
C F         FLUORIDE (PPM)
C OMEGA     CONDUCTIVITY ( MICRO-SIEMENS/CM  AT 20 C )
C EH        OXIDATION POTENTIAL ( MILLI-VOLTS)
C MTCO3     MOLARITY OF TOTAL DISSOLVED CARBONATE
C MCL       MOLARITY OF CHLORIDE
C MSO4      MOLARITY OF SULPHATE
C MSIO2     MOLARITY OF SILICA
C MCA       MOLARITY OF CALCIUM
C MMG       MOLARITY OF MAGNESIUM
C MK        MOLARITY OF POTASSIUM
C MNA       MOLARITY OF SODIUM
C MFE       MOLARITY OF DISSOLVED IRON
C MAL       MOLARITY OF DISSOLVED ALUMINIUM
C MNO3      MOLARITY OF NITRATE
C MF        MOLARITY OF FLUORIDE
C FLOW      FLOW IN GALLONS/HOUR(SPRINGS,BOREHOLES)
C DHOLE     DEPTH OF BORE-HOLE ( METRES )
C DWATER    DEPTH TO WATER ( METRES )
C
      CALL SYSCOM(I,'AS 0520*')
      CALL SYSCOM(I,'AS 0620*')
      CALL SYSCOM(I,'AS 0899*')
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C READ NUMBER OF RECORDS IN "WB"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      CALL SYSCOM(I,'AC WBD1,1*')
      READ(1,1) MAXNOS
    1 FORMAT(I5)
C SAVE INITIAL VALUE OF MAXNOS AS PMAXNS
      PMAXNS=MAXNOS
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C DETERMINE IF INPUT AND TRANSFER OF AN ANALYSIS IS REQUIRED
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  100 WRITE(6,2)
    2 FORMAT('ADTOWB:  ADDS DATA TO "WB"'/
     *       'DO YOU WISH TO ENTER SAMPLE DATA?(Y/N)')
      READ(5,3) MOREIP
    3 FORMAT(A1)
      IF ( MOREIP.NE.YES .AND.  MOREIP.NE.NO ) GOTO 100
      IF ( MOREIP.EQ.NO ) GOTO 900
```

```
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%
C READ IN SAMPLE DATA FROM VDU
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%
      WRITE(6,4)
    4 FORMAT(/,'TYPE IN LAUNCESTON ANALYSIS NUMBER'/
     *         'IF NONE, PRESS THE "RETURN" KEY'/
     *         '******')
      READ(5,5) LTNNOS
    5 FORMAT(A6)
C
      WRITE(6,6)
    6 FORMAT(/,'TYPE IN DATA SOURCE AND IDENTIFICATION IN SOURCE'/
     *          '****** ********')
      READ(5,7) REF,REFNOS
    7 FORMAT(A6,X,A8)
C
  110 WRITE(6,8)
    8 FORMAT(/,'TYPE IN QUADRANGLE NUMBER'/'**')
      READ(5,9) QUADNO
    9 FORMAT(I2)
C CHECK THAT QUADRANGLE NUMBER EXISTS
      IF ( QUADNO.LT.0 .OR.  QUADNO.GT.97 ) WRITE(6,10)
   10 FORMAT('THIS QUADRANGLE NUMBER DOES NOT EXIST')
      IF ( QUADNO.LT.0 .OR.  QUADNO.GT.97 ) GOTO 110
C
  120 WRITE(6,11)
   11 FORMAT(/,'TYPE IN AMG 100M GRID REFERENCE:SQUARE,E,N'/
     *          '** ***/***')
      READ(5,12) SQUARE,EAST,NORTH
   12 FORMAT(A2,X,I3,X,I3)
      WRITE(8,13) SQUARE
   13 FORMAT(A2)
      READ(8,14) E100KM,N100KM
   14 FORMAT(2A1)
C CHECK THAT LOCATION IS INSIDE TASMANIA
      OUTSYD=.FALSE.
      EWRONG=E100KM.NE.CHARB .AND.  E100KM.NE.CHARC .AND.  E100KM.NE.CHARD
     * .AND.  E100KM.NE.CHARE .AND.  E100KM.NE.CHARF
      NSRONG=N100KM.NE.CHARM .AND.  N100KM.NE.CHARN .AND.  N100KM.NE.CHARP
     * .AND.  N100KM.NE.CHARQ .AND.  N100KM.NE.CHARR .AND.  N100KM.NE.CHARS
      OUTSYD= EWRONG .OR.  NSRONG
      IF ( OUTSYD ) WRITE(6,15)
   15 FORMAT('THIS GRID COORDINATE IS NOT IN TASMANIA')
      IF ( OUTSYD ) GOTO 120
C CONVERT 100KM SQUARE DESIGNATION TO 100KM PART OF E,N COORDINATES
      ECOORD=EAST + (((E100KM-CHARA)/256) + 1)*1000
      IF ( N100KM.GT.CHARN ) N100KM=N100KM-256
C("O" IS NOT USED AS SQUARE DESIGNATION LETTER)
      NCOORD=NORTH + (((N100KM-CHARM)/256) + 1)*1000
C
      WRITE(6,16)
   16 FORMAT(/,'TYPE IN LOCALITY'/'****************')
      READ(5,17) LOCLTY
   17 FORMAT(4A4)
```

```
C
      WRITE(6,18)
   18 FORMAT(/,'TYPE IN REGION'/'****************')
      READ(5,17) REGION
C
  130 WRITE(6,19)
   19 FORMAT(/,'TYPE IN ROCK DATA:NUMBER OF ROCK TYPES'/
     *          '                  ROCK TYPES 1,2,3,4'/
     *          '                  DATA RELIABILITY'/
     *          '* ** ** ** ** *')
      READ(5,20) NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL
   20 FORMAT(I1,X,4(A2,X),A1)
C CHECK IF RECOGNISABLE ROCK RELIABILITY CODE HAS BEEN ENTERED
      RECHEQ=.FALSE.
      RECHEQ=ROCKRL.NE.CHARD .AND.  ROCKRL.NE.CHARG
     * .AND. ROCKRL.NE.CHARM .AND.  ROCKRL.NE.BLANK
      IF ( RECHEQ ) WRITE(6,22)
   22 FORMAT('UNRECOGNISABLE CODE FOR ROCK DATA RELIABILITY'/
     *       'PLEASE REENTER DATA (D/G/M/ )')
      IF ( RECHEQ ) GOTO 130
C
      WRITE(6,23)
   23 FORMAT(/,'TYPE IN CODES(1,2,3)'/'* * *')
      READ(5,24) CODE1,CODE2,CODE3
   24 FORMAT(A1,X,A1,X,A1)
C
  140 WRITE(6,25)
   25 FORMAT(/,'TYPE IN RELIABILITY DATA:ANALYSIS,LOCATION'/'* *')
      READ(5,26) ANALRL,LOCRL
   26 FORMAT(A1,X,A1)
C CHECK IF RECOGNISABLE RELIABILITY CODES HAVE BEEN ENTERED
      RECHEQ=.FALSE.
      AWRONG=ANALRL.NE.CHARR .AND.  ANALRL.NE.CHARD
     *.AND.  ANALRL.NE.CHARE .AND.  ANALRL.NE.CHARP .AND.  ANALRL.NE.CHARC
     *.AND.  ANALRL.NE.CHART .AND.  ANALRL.NE.BLANK
      LWRONG=LOCRL.NE.CHARR .AND.  LOCRL.NE.CHARK
     * .AND.  LOCRL.NE.CHART .AND.  LOCRL.NE.BLANK
      RECHEQ= AWRONG .OR.  LWRONG
      IF ( RECHEQ ) WRITE(6,27)
   27 FORMAT('UNRECOGNISABLE RELIABILITY CODE'/
     *       'PLEASE REENTER DATA(R/D/E/P/C/T/ ,R/K/T/ )')
      IF ( RECHEQ ) GOTO 140
C
  150 WRITE(6,28)
   28 FORMAT(/,'TYPE IN THE TYPE OF WATER SAMPLE'/'*')
      READ(5,29) TYPE
   29 FORMAT(A1)
C CHECK IF RECOGNISABLE TYPE CODE HAS BEEN ENTERED
      RECHEQ=.FALSE.
      RECHEQ= TYPE.NE.CHARB .AND.  TYPE.NE.CHARS .AND.  TYPE.NE.CHARR
     *  .AND.  TYPE.NE.CHARP .AND.  TYPE.NE.CHARL .AND.  TYPE.NE.BLANK
     *  .AND.  TYPE.NE.CHARD
      IF ( RECHEQ ) WRITE(6,30)
   30 FORMAT('UNRECOGNISABLE CODE FOR TYPE OF WATER SAMPLE'/
     *       'PLEASE D/REENTED/R DATA (B/S/D/R/P/L/ )')
```

```
      IF ( RECHEQ ) GOTO 150
C
      WRITE(6,31)
   31 FORMAT(/,'TYPE IN FLOW(GALLONS/HOUR)')
      CALL FRINIT(FLOW,IER,5)
C
      WRITE(6,32)
   32 FORMAT(/,'TYPE IN DEPTH OF BOREHOLE(METRES)')
      CALL FRINIT(DHOLE,IER,5)
C
      WRITE(6,33)
   33 FORMAT(/,'TYPE IN DEPTH TO WATER(METRES)')
      CALL FRINIT(DWATER,IER,5)
C
      WRITE(6,34)
   34 FORMAT(/,'TYPE IN CONDUCTIVITY(uS/cm)')
      CALL FRINIT(OMEGA,IER,5)
C
      WRITE(6,35)
   35 FORMAT(/,'TYPE IN PH')
      CALL FRINIT(PH,IER,5)
C
      WRITE(6,36)
   36 FORMAT(/,'TYPE IN SALINITY AS PPM TDS')
      CALL FRINIT(TDS,IER,5)
C
      WRITE(6,37)
   37 FORMAT(/,'TYPE IN TOTAL CARBONATE(PPM):')
      CALL FRINIT(TCO3,IER,5)
      MTCO3=TCO3/61017.
C( 1 MOLE/L=1000(MOLECULAR WEIGHT) MG/L )
C
      WRITE(6,38)
   38 FORMAT(/,'TYPE IN CHLORIDE(PPM)')
      CALL FRINIT(CL,IER,5)
      MCL=CL/35453.
C
      WRITE(6,39)
   39 FORMAT(/,'TYPE IN SULPHATE(PPM)')
      CALL FRINIT(SO4,IER,5)
      MSO4=SO4/96036.
C
      WRITE(6,40)
   40 FORMAT(/,'TYPE IN SILICA(PPM)')
      CALL FRINIT(SIO2,IER,5)
      MSIO2=SIO2/60074.
C
      WRITE(6,41)
   41 FORMAT(/,'TYPE IN CALCIUM(PPM)')
      CALL FRINIT(CA,IER,5)
      MCA=CA/40080.
C
      WRITE(6,42)
   42 FORMAT(/,'TYPE IN MAGNESIUM(PPM)')
      CALL FRINIT(MG,IER,5)
      MMG=MG/24305.
C
      WRITE(6,43)
   43 FORMAT(/,'TYPE IN POTASSIUM(PPM)')
      CALL FRINIT(K,IER,5)
      MK=K/39098.
C
      WRITE(6,44)
   44 FORMAT(/,'TYPE IN SODIUM(PPM)')
      CALL FRINIT(NA,IER,5)
      MNA=NA/22990.
C
      WRITE(6,45)
   45 FORMAT(/,'TYPE IN IRON(PPM)')
      CALL FRINIT(FE,IER,5)
      MFE=FE/55847.
C
      WRITE(6,46)
   46 FORMAT(/,'TYPE IN ALUMINIUM(PPM)')
      CALL FRINIT(AL,IER,5)
      MAL=AL/26981.54
C
      WRITE(6,47)
   47 FORMAT(/,'TYPE IN NITRATE(PPM)')
      CALL FRINIT(NO3,IER,5)
      MNO3=NO3/62004.9
C
      WRITE(6,48)
   48 FORMAT(/,'TYPE IN FLUORIDE(PPM)')
      CALL FRINIT(F,IER,5)
      MF=F/18998.4
C
      EH=0.0
C
      WBNOS=MAXNOS
C
C %%%%%%%%%%
C CHECK DATA
C %%%%%%%%%%
  400 WRITE(6,49)  WBNOS,LTNNOS,REF,REFNOS,QUADNO,TDS,
     *             SQUARE,EAST,NORTH,TCO3,LOCLTY,CL,REGION,SO4,
     *             NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,SIO2,
     *             CODE1,CODE2,CODE3,CA,ANALRL,LOCRL,MG,
     *             TYPE,K,FLOW,NA,DHOLE,FE,DWATER,AL,OMEGA,NO3,PH,F
   49 FORMAT(
     *'STORED DATA FOR THE WATER SAMPLE NUMBER,'I5//
     *'(1)LAUNCESTON ANALYSIS NUMBER    ',A6/
     *'(2)DATA SOURCE,IDENTIFICATION IN SOURCE:',A6,'/',A8//
     *'(3)QUADRANGLE NUMBER:',18X,I2,10X,'(16)TDS (PPM):',F7.1/
     *'(4)AMG 100M GRID REFERENCE:',4X,A2,X,I3,'/',I3,10X,
     *                   '(17)TCO3(PPM):',F7.1/
     *'(5)LOCALITY:',13X,4A4,10X,'(18)CL   (PPM):',F7.1/
     *'(6)REGION:',15X,4A4,10X,'(19)SO4 (PPM):',F7.1/
     *'(7)ROCK TYPE DATA:',8X,I1,X,4(A2,X),A1,10X,'(20)SIO2(PPM):',F7.1/
     *'(8)CODES:',26X,3(X,A1),10X,'(21)CA   (PPM):',F7.1/
     *'(9)DATA RELIABILITY:',17X,2(X,A1),10X,'(22)MG   (PPM):',F7.1/
     *'(10)TYPE OF SAMPLE:',21X,A1,10X,'(23)K    (PPM):',F7.1/
     *'(11)FLOW(G/H):',21X,F6.0,10X,'(24)NA  (PPM):',F7.1/
     *'(12)DEPTH OF BOREHOLE(M):',10X,F6.1,10X,'(25)FE  (PPM):',F7.1/
     *'(13)DEPTH TO WATER(M):',13X,F6.1,10X,'(26)AL  (PPM):',F7.1/
     *'(14)CONDUCTIVITY (uS/cm):',9X,F7.1,10X,
     *                     '(27)NO3 (PPM):'F7.1/
     *'(15)PH:',30X,F4.1,10X,'(28)F    (PPM):',F7.1//
     *'IF THE DATA IS OK, THEN TYPE IN A ZERO: "0"'/
     *'IF YOU WISH TO CHANGE SOME DATA'/
     *'TYPE IN THE NUMBER OF PARAMETER(GROUPS) YOU WISH TO CORRECT')
      CALL INFREE(NCHNGE,IER,5)
C
C IF NO CHANGES REQUIRED THEN CORRECTION PROCEDURE IS SKIPPED
      IF ( NCHNGE.EQ.0 ) GOTO 500
C
C %%%%%%%%%%%%%%%%%%%%
C CORRECTION PROCEDURE
C %%%%%%%%%%%%%%%%%%%%
C FORMAT STATEMENTS FOR SAMPLE DATA PROMPTS,INPUT AND ERROR MESSAGES
C ARE IN THE PREVIOUS INPUT PROCEDURE ( 4-48 )
      WRITE(6,50)
   50 FORMAT(
     *'TYPE IN THE NUMBERS OF THE PARAMETER GROUPS YOU WISH TO CHANGE'/
     *'(EG 15,16,21 TO CHANGE PH,TDS AND CA)')
C
      CALL NUMIN(CHANGE,NCHNGE,IER,5,COUNT)
C
      DO 450 I=1,COUNT
         NVAR=CHANGE(I)
         GOTO (401,402,403,404,405,406,407,408,409,
     *           410,411,412,413,414,415,416,417,418,419,
     *           420,421,422,423,424,425,426,427,428),NVAR
C
  401    WRITE(6,4)
         READ(5,5) LTNNOS
         GOTO 450
C
  402    WRITE(6,6)
         READ(5,7) REF,REFNOS
         GOTO 450
C
  403    WRITE(6,8)
         READ(5,9) QUADNO
C CHECK THAT QUADRANGLE NUMBER EXISTS
         IF ( QUADNO.LT.0 .OR.  QUADNO.GT.97 ) WRITE(6,10)
         IF ( QUADNO.LT.0 .OR.  QUADNO.GT.97 ) GOTO 403
         GOTO 450
C
  404    WRITE(6,11)
         READ(5,12) SQUARE,EAST,NORTH
         WRITE(8,13) SQUARE
         READ(8,14) E100KM,N100KM
C CHECK THAT LOCATION IS INSIDE TASMANIA
         OUTSYD=.FALSE.
         EWRONG= E100KM. NE. CHARB . AND.  E100KM. NE. CHARC
     *     . AND.  E100KM. NE. CHARD . AND.  E100KM. NE. CHARE
     *     . AND.  E100KM. NE. CHARF
         NSRONG= N100KM. NE. CHARM . AND.  N100KM. NE. CHARN
     *     . AND.  N100KM. NE. CHARP . AND.  N100KM. NE. CHARQ
     *     . AND.  N100KM. NE. CHARR . AND.  N100KM. NE. CHARS
         OUTSYD= EWRONG . OR.  NSRONG
         IF ( OUTSYD ) WRITE(6,15)
         IF ( OUTSYD) GOTO 404
C CONVERT 100KM SQUARE DESIGNATION TO 100KM PART OF E,N COORDINATES
         ECOORD=EAST + (((E100KM-CHARA)/256) + 1)*1000
         IF ( N100KM. GT. CHARN ) N100KM=N100KM-256
C("O" IS NOT USED AS SQUARE DESIGNATION LETTER)
         NCOORD=NORTH + (((N100KM-CHARM)/256) + 1)*1000
         GOTO 450
C
  405    WRITE(6,16)
         READ(5,17) LOCLTY
         GOTO 450
C
  406    WRITE(6,18)
         READ(5,17) REGION
         GOTO 450
C
  407    WRITE(6,19)
         READ(5,20) NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL
C CHECK IF RECOGNISABLE ROCK RELIABILITY CODE HAS BEEN ENTERED
         RECHEQ=. FALSE.
         RECHEQ=ROCKRL. NE. CHARD . AND.  ROCKRL. NE. CHARG
     *    . AND.  ROCKRL. NE. CHARM . AND.  ROCKRL. NE. BLANK
         IF ( RECHEQ ) WRITE(6,22)
         IF ( RECHEQ) GOTO 407
         GOTO 450
C
  408    WRITE(6,23)
         READ(5,24) CODE1,CODE2,CODE3
         GOTO 450
C
  409    WRITE(6,25)
         READ(5,26) ANALRL,LOCRL
C CHECK IF RECOGNISABLE RELIABILITY CODES HAVE BEEN ENTERED
         RECHEQ=. FALSE.
         AWRONG=ANALRL. NE. CHARR . AND.  ANALRL. NE. CHARD
     *    . AND.  ANALRL. NE. CHARE . AND.  ANALRL. NE. CHARP
     *    . AND.  ANALRL. NE. CHARC . AND.  ANALRL. NE. CHART
     *    . AND.  ANALRL. NE. BLANK
         LWRONG=LOCRL. NE. CHARR . AND.  LOCRL. NE. CHARK
     *    . AND.  LOCRL. NE. CHART . AND.  LOCRL. NE. BLANK
         RECHEQ= AWRONG . OR.  LWRONG
         IF ( RECHEQ ) WRITE(6,27)
         IF ( RECHEQ ) GOTO 409
         GOTO 450
C
  410    WRITE(6,28)
         READ(5,29) TYPE
```

```
C CHECK IF RECOGNISABLE TYPE CODE HAS BEEN ENTERED
        RECHEQ=.FALSE.
        RECHEQ=TYPE.NE.CHARB .AND.  TYPE.NE.CHARS .AND.  TYPE.NE.CHARR
     *   .AND.  TYPE.NE.CHARP .AND.  TYPE.NE.CHARL .AND.  TYPE.NE.BLANK
     *  .AND.  TYPE.NE.CHARD
        IF ( RECHEQ ) WRITE(6,30)
        IF ( RECHEQ) GOTO 410
        GOTO 450
C
  411   WRITE(6,31)
        CALL FRINIT(FLOW,IER,5)
        GOTO 450
C
  412   WRITE(6,32)
        CALL FRINIT(DHOLE,IER,5)
        GOTO 450
C
  413   WRITE(6,33)
        CALL FRINIT(DWATER,IER,5)
        GOTO 450
C
  414   WRITE(6,34)
        CALL FRINIT(OMEGA,IER,5)
        GOTO 450
C
  415   WRITE(6,35)
        CALL FRINIT(PH,IER,5)
        GOTO 450
C
  416   WRITE(6,36)
        CALL FRINIT(TDS,IER,5)
        GOTO 450
C
  417   WRITE(6,37)
        CALL FRINIT(TCO3,IER,5)
        MTCO3=TCO3/61017.
        GOTO 450
C
  418   WRITE(6,38)
        CALL FRINIT(CL,IER,5)
        MCL=CL/35453.
        GOTO 450
C
  419   WRITE(6,39)
        CALL FRINIT(SO4,IER,5)
        MSO4=SO4/96036.
        GOTO 450
C
  420   WRITE(6,40)
        CALL FRINIT(SIO2,IER,5)
        GOTO 450
C
  421   WRITE(6,41)
        CALL FRINIT(CA,IER,5)
        MCA=CA/40080.
```

29/62

```
         GOTO 450
C
  422    WRITE(6,42)
         CALL FRINIT(MG,IER,5)
         MMG=MG/24305.
         GOTO 450
C
  423    WRITE(6,43)
         CALL FRINIT(K,IER,5)
         MK=K/39098.
         GOTO 450
C
  424    WRITE(6,44)
         CALL FRINIT(NA,IER,5)
         MNA=NA/22990.
         GOTO 450
C
  425    WRITE(6,45)
         CALL FRINIT(FE,IER,5)
         MFE=FE/55847.
         GOTO 450
C
  426    WRITE(6,46)
         CALL FRINIT(AL,IER,5)
         MAL=AL/26981.54
         GOTO 450
C
  427    WRITE(6,47)
         CALL FRINIT(NO3,IER,5)
         MNO3=NO3/62004.9
         GOTO 450
C
  428    WRITE(6,48)
         CALL FRINIT(F,IER,5)
         MF=F/18998.4
C
  450 CONTINUE
C
C
      GOTO 400
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C TRANSFER DATA TO "WB" AND INITIALISE SEARCH RESULTS TO .FALSE.
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  500 CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,10*')
      IREC=MAXNOS
      CALL POSITN(2,IREC)
      WRITE(2) ALINE
      CALL SYSCOM(I,'AT WB,11*')
C
C INITIALISE RESULTS OF SEARCHES 1-10 TO .FALSE.  AND
C  TRANSFER DATA TO "WBLOG"
      CALL SYSCOM(I,'AC WBLOG,4*')
      CALL SYSCOM(I,'AT WBLOG,10*')
      DO 550 I=1,10
  550 FOUND(I)=.FALSE.
      CALL POSITN(4,IREC)
      WRITE(4) FOUND
      CALL SYSCOM(I,'AT WBLOG,11*')
C
      WRITE(6,51) MAXNOS
   51 FORMAT('NEW RECORD(NUMBER',I5,') HAS BEEN TRANSFERRED TO "WB"'//)
C UPDATE MAXNOS,NUMBER OF RECORDS IN "WB"
      MAXNOS=MAXNOS+1
C RETURN TO BEGINNING OF INPUT PROCEDURE FOR ANOTHER SAMPLE
      GOTO 100
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C SAVE MAXNOS AND PMAXNS IN APPROPRIATE DATA-FILES
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  900 IF ( PMAXNS.EQ.MAXNOS ) GOTO 999
C ( IF PMAXNS=MAXNOS, THEN NO RECORDS HAVE BEEN ADDED IN THE UPDATE )
      CALL SYSCOM(I,'AC WBD1,1*')
      CALL SYSCOM(I,'AT WBD1,0*')
      WRITE(1,1) MAXNOS
      CALL SYSCOM(I,'WF 1*')
      CALL SYSCOM(I,'AT WBD1,1*')
      CALL SYSCOM(I,'CL*')
C
      CALL SYSCOM(I,'AC WBD2,7*')
      CALL SYSCOM(I,'AT WBD2,0*')
      WRITE(7,1) MAXNOS
      CALL SYSCOM(I,'WF 7*')
      CALL SYSCOM(I,'AT WBD2,1*')
      CALL SYSCOM(I,'CL*')
C
      CALL SYSCOM(I,'AC WBD3,3*')
      CALL SYSCOM(I,'AT WBD3,0*')
      WRITE(3,1) PMAXNS
      CALL SYSCOM(I,'WF 3*')
      CALL SYSCOM(I,'AT WBD3,1*')
      CALL SYSCOM(I,'CL*')
C
  999 WRITE(6,52) MAXNOS
   52 FORMAT('"ADTOWB": END OF RUN'/
     *'"WB",THE WATER-ANALYSIS DATA-FILE,NOW CONTAINS',I5,' RECORDS')
      STOP
      END
```

```
$TITL WBS2
$DBUG
C
C SOURCE FILE FOR "CRCTWB"
C
C "CRCTWB" CORRECTS THE DATA FILE "WB"
C CORRECTIONS ARE MADE FROM THE VDU
C
C THE INPUT DATA FOR EACH RECORD ARE DISPLAYED ON THE VDU
C FOR CHECKING BEFORE EACH TRANSFER OF CORRECTIONS IS MADE
C
C LOGICAL UNIT ASSIGNMENTS:
C 01-WBD1    -STORES MAXNOS (NUMBER OF RECORDS IN "WB")
C 02-WB     -RANDOM ACCESS DATA FILE
C 05-VDU    -INPUT
C 06-VDU    -PROMPTS
C 08-99     -CORE BUFFER
C
C SUBROUTINES: 0
C FORLIB ROUTINES:  SYSCOM, POSITN, INFREE, NUMIN, FRINIT
C
      DOUBLE PRECISION
     *                LTNNOS, REFNOS, REF
      REAL
     *    LOCLTY(4), REGION(4),
     *    OMEGA, PH, TCO3, CL, SO4, SIO2, CA, MG, K, NA, TDS, FE, AL, NO3, F,
     *    MTCO3, MCL, MSO4, MSIO2, MCA, MMG, MK, MNA, MFE, MAL, MNO3, MF,
     *    FLOW, DHOLE, DWATER
      INTEGER*2
     *         WBNOS, TYPE, QUADNO, SQUARE, EAST, NORTH,
     *         NROCK, ROCK1, ROCK2, ROCK3, ROCK4, ROCKRL,
     *         CODE1, CODE2, CODE3, ANALRL, LOCRL,
     *         ECOORD, NCOORD,
     *         E100KM, N100KM, CHARA, CHARM, CHARN,
     *         CHARB, CHARC, CHARD, CHARE, CHARF, CHARG, CHARK,
     *         CHARL, CHARP, CHARQ, CHARR, CHARS, CHART, BLANK,
     *         MAXNOS, PMAXNS, IREC, ALINE(109),
     *         MOREIP, DATAOK, YES, NO,
     *         CHANGE(28), COUNT, NCHNGE
C
      LOGICAL
     *       FOUND(10), IER, OUTSYD, EWRONG, NSRONG, RECHEQ, AWRONG, LWRONG
C
      EQUIVALENCE
     *(ALINE(1), LTNNOS), (ALINE(5), REFNOS), (ALINE(9), REF),
     *(ALINE(13), WBNOS), (ALINE(14), TYPE), (ALINE(15), QUADNO),
     *(ALINE(16), SQUARE), (ALINE(17), EAST), (ALINE(18), NORTH),
     *(ALINE(19), ECOORD), (ALINE(20), NCOORD),
     *(ALINE(21), LOCLTY(1)), (ALINE(29), REGION(1)),
     *(ALINE(37), NROCK), (ALINE(38), ROCK1), (ALINE(39), ROCK2),
     *(ALINE(40), ROCK3), (ALINE(41), ROCK4), (ALINE(42), ROCKRL),
     *(ALINE(43), CODE1), (ALINE(44), CODE2), (ALINE(45), CODE3),
     *(ALINE(46), ANALRL), (ALINE(47), LOCRL),
     *(ALINE(48), PH), (ALINE(50), TCO3), (ALINE(52), CL),
     *(ALINE(54), SO4), (ALINE(56), SIO2), (ALINE(58), CA), (ALINE(60), MG),
     *(ALINE(62),K),(ALINE(64),NA),(ALINE(66),TDS),
     *(ALINE(68),FE),(ALINE(70),AL),(ALINE(72),NO3),(ALINE(74),F),
     *(ALINE(76),OMEGA),(ALINE(78),EH),(ALINE(80),MTCO3),(ALINE(82),CL),
     *(ALINE(84),MSO4),(ALINE(86),MSIO2),(ALINE(88),MCA),
     *(ALINE(90),MMG),(ALINE(92),MK),(ALINE(94),MNA),
     *(ALINE(96),MFE),(ALINE(98),MAL),(ALINE(100),MNO3),(ALINE(102),F),
     *(ALINE(104),FLOW),(ALINE(106),DHOLE),(ALINE(108),DWATER)
C
C
      DATA YES,NO/1HY,1HN/
      DATA CHARA,CHARB,CHARC,CHARD,CHARE,CHARF,CHARG,CHARK,
     *     CHARL,CHARM,CHARN,CHARP,CHARQ,CHARR,CHARS,CHART/
     *1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HK,1HL,1HM,1HN,1HP,1HQ,1HR,1HS,1HT/
      DATA BLANK/1H /
C
C LTNNOS    LAUNCESTON WATER ANALYSIS NUMBER
C REFNOS    REFERENCE NUMBER IN DATA SOURCE
C REF       SOURCE OF DATA
C WBNOS     "WB" DATAFILE RECORD NUMBER
C TYPE      SAMPLE TYPE(SPRING,B/HOLE,DAM,RIVER,PRECIPITATION,LEACHATE)
C QUADNO    1:50 000 QUADRANGLE NUMBER
C SQUARE    100 000 M SQUARE DESIGNATION
C EAST      EASTING (10KM-1KM-100M)
C NORTH     NORTHING(10KM-1KM-100M)
C ECOORD    EASTING (100KM-10KM-1KM-100M)
C NCOORD    NORTHING(100KM-10KM-1KM-100M)
C LOCLTY    LOCALITY OF SAMPLE
C REGION    REGION
C NROCK     NUMBER OF MAJOR ROCK TYPES ASSOCIATED WITH SAMPLE
C ROCK1     ROCK TYPE NUMBER 1
C ROCK2     ROCK TYPE NUMBER 2
C ROCK3     ROCK TYPE NUMBER 3
C ROCK4     ROCK TYPE NUMBER 4
C ROCKRL    RELIABILITY OF ROCK-TYPE DATA
C CODE1     UNSPECIFIED CODE 1.
C CODE2     UNSPECIFIED CODE 2.
C CODE3     UNSPECIFIED CODE 3.
C ANALRL    RELIABILITY OF ANALYSIS
C LOCRL     RELIABILITY OF LOCATION DATA
C PH        PH OF SAMPLE
C TCO3      TOTAL DISSOLVED CARBONATE
C CL        CHLORIDE (PPM)
C SO4       SULPHATE (PPM)
C SIO2      SILICA (PPM)
C CA        CALCIUM (PPM)
C MG        MAGNESIUM (PPM)
C K         POTASSIUM (PPM)
C NA        SODIUM (PPM)
C TDS       TOTAL DISSOLVED SOLIDS (PPM)
C FE        IRON (PPM)
C AL        ALUMINIUM (PPM)
C NO3       NITRATE (PPM)
C F         FLUORIDE (PPM)
C OMEGA     CONDUCTIVITY ( MICRO-SIEMENS/CM  AT 20 C )
```

```
C EH       OXIDATION POTENTIAL ( MILLI-VOLTS)
C MTCO3    MOLARITY OF TOTAL DISSOLVED CARBONATE
C MCL      MOLARITY OF CHLORIDE
C MSO4     MOLARITY OF SULPHATE
C MSIO2    MOLARITY OF SILICA
C MCA      MOLARITY OF CALCIUM
C MMG      MOLARITY OF MAGNESIUM
C MK       MOLARITY OF POTASSIUM
C MNA      MOLARITY OF SODIUM
C MFE      MOLARITY OF DISSOLVED IRON
C MAL      MOLARITY OF DISSOLVED ALUMINIUM
C MNO3     MOLARITY OF NITRATE
C MF       MOLARITY OF FLUORIDE
C FLOW     FLOW IN GALLONS/HOUR(SPRINGS,BOREHOLES)
C DHOLE    DEPTH OF BORE-HOLE ( METRES )
C DWATER   DEPTH TO WATER ( METRES )
C
      CALL SYSCOM(I,'AS 0520*')
      CALL SYSCOM(I,'AS 0620*')
      CALL SYSCOM(I,'AS 0899*')
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C READ NUMBER OF RECORDS IN "WB"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      CALL SYSCOM(I,'AC WBD1,1*')
      READ(1,1) MAXNOS
    1 FORMAT(I5)
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C DETERMINE NUMBER OF RECORD TO BE ALTERED ( IF ANY)
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  100 WRITE(6,2)
    2 FORMAT('CRCTWB: CORRECTS DATA IN "WB"'/
     *       'DO YOU WISH TO CORRECT A RECORD?(Y/N)')
      READ(5,3) MOREIP
    3 FORMAT(A1)
      IF ( MOREIP.NE.YES .AND.  MOREIP.NE.NO ) GOTO 100
      IF ( MOREIP.EQ.NO ) GOTO 999
C
  140 WRITE(6,4)
    4 FORMAT('RECORD NUMBER OF SAMPLE TO BE CORRECTED?')
      CALL INFREE(IREC,IER,5)
      IF ( IREC.GE.MAXNOS) WRITE(6,5) IREC
    5 FORMAT('RECORD NUMBER',I5,' DOES NOT EXIST IN THE DATAFILE "WB"'/
C (THE HIGHEST RECORD NUMBER IN "WB" IS MAXNOS-1)
      IF ( IREC.GE.MAXNOS ) GOTO 140
      IF ( IER ) GOTO 140
C
C %%%%%%%%%%%%%%%%%%%%%%%%
C READ IN RECORD FROM "WB"
C %%%%%%%%%%%%%%%%%%%%%%%%
      CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,11*')
      CALL POSITN(2,IREC)
      READ(2) ALINE
```

```
C
C %%%%%%%%%%
C CHECK DATA
C %%%%%%%%%%
  400 WRITE(6,6) WBNOS,LTNNOS,REF,REFNOS,QUADNO,TDS,
     *           SQUARE,EAST,NORTH,TCO3,LOCLTY,CL,REGION,SO4,
     *           NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,SIO2,
     *           CODE1,CODE2,CODE3,CA,ANALRL,LOCRL,MG,
     *           TYPE,K,FLOW,NA,DHOLE,FE,DWATER,AL,OMEGA,NO3,PH,F
    6 FORMAT(
     *'STORED DATA FOR THE WATER SAMPLE NUMBER,'I5//
     *'(1)LAUNCESTON ANALYSIS NUMBER   ',A6/
     *'(2)DATA SOURCE,IDENTIFICATION IN SOURCE:',A6,'/',A8//
     *'(3)QUADRANGLE NUMBER:',18X,I2,10X,'(16)TDS (PPM):',F7.1/
     *'(4)AMG 100M GRID REFERENCE:',4X,A2,X,I3,'/',I3,10X,
     *                    '(17)TCO3(PPM):',F7.1/
     *'(5)LOCALITY:',13X,4A4,10X,'(18)CL  (PPM):',F7.1/
     *'(6)REGION:',15X,4A4,10X,'(19)SO4 (PPM):',F7.1/
     *'(7)ROCK TYPE DATA:',8X,I1,X,4(A2,X),A1,10X,'(20)SIO2(PPM):',F7.1/
     *'(8)CODES:',26X,3(X,A1),10X,'(21)CA  (PPM):',F7.1/
     *'(9)DATA RELIABILITY:',17X,2(X,A1),10X,'(22)MG  (PPM):',F7.1/
     *'(10)TYPE OF SAMPLE:',21X,A1,10X,'(23)K   (PPM):',F7.1/
     *'(11)FLOW(G/H):',21X,F6.0,10X,'(24)NA  (PPM):',F7.1/
     *'(12)DEPTH OF BOREHOLE(M):',10X,F6.1,10X,'(25)FE  (PPM):',F7.1/
     *'(13)DEPTH TO WATER(M):',13X,F6.1,10X,'(26)AL  (PPM):',F7.1/
     *'(14)CONDUCTIVITY (uS/cm):',9X,F7.1,10X,
     *                       '(27)NO3 (PPM):'F7.1/
     *'(15)PH:',30X,F4.1,10X,'(28)F   (PPM):',F7.1//
     *'IF THE DATA IS OK, THEN TYPE IN A ZERO: "0"'/
     *'IF YOU WISH TO CHANGE SOME DATA'/
     *'TYPE IN THE NUMBER OF PARAMETER(GROUPS) YOU WISH TO CORRECT')
      CALL INFREE(NCHNGE,IER,5)
C
C IF NO CHANGES REQUIRED THEN CORRECTION PROCEDURE IS SKIPPED
      IF ( NCHNGE.EQ.0 ) GOTO 500
C
C %%%%%%%%%%%%%%%%%%%%
C CORRECTION PROCEDURE
C %%%%%%%%%%%%%%%%%%%%
      WRITE(6,7)
    7 FORMAT(
     *'TYPE IN THE NUMBERS OF THE PARAMETER GROUPS YOU WISH TO CHANGE'/
     *'(EG 15,16,21 TO CHANGE PH,TDS AND CA)')
C
      CALL NUMIN(CHANGE,NCHNGE,IER,5,COUNT)
C
      DO 150 I=1,COUNT
         NVAR=CHANGE(I)
         GOTO (101,102,103,104,105,106,107,108,109,
     *      110,111,112,113,114,115,116,117,118,119,
     *      120,121,122,123,124,125,126,127,128),NVAR
C
  101    WRITE(6,8)
    8    FORMAT(/,'TYPE IN LAUNCESTON ANALYSIS NUMBER'/
     *            'IF NONE,PRESS THE "RETURN" KEY'/
     *          '******')
        READ(5,9) LTNNOS
    9   FORMAT(A6)
        GOTO 150
C
  102   WRITE(6,10)
   10   FORMAT(/,'TYPE IN DATA SOURCE AND IDENTIFICATION IN SOURCE'/
     *          '****** ********')
        READ(5,11) REF,REFNOS
   11   FORMAT(A6,X,A8)
        GOTO 150
C
  103   WRITE(6,12)
   12   FORMAT(/,'TYPE IN QUADRANGLE NUMBER'/'**')
        READ(5,13) QUADNO
   13   FORMAT(I2)
        IF ( QUADNO.LT.0 .OR.  QUADNO.GT.97 ) WRITE(6,14)
   14   FORMAT('THIS QUADRANGLE NUMBER DOES NOT EXIST')
        IF ( QUADNO.LT.0 .OR.  QUADNO.GT.97 ) GOTO 103
        GOTO 150
C
  104   WRITE(6,15)
   15   FORMAT(/,'TYPE IN AMG 100M GRID REFERENCE:SQUARE,E,N'/
     *          '** ***/***')
        READ(5,16) SQUARE,EAST,NORTH
   16   FORMAT(A2,X,I3,X,I3)
        WRITE(8,17) SQUARE
   17   FORMAT(A2)
        READ(8,18) E100KM,N100KM
   18   FORMAT(2A1)
C CHECK THAT LOCATION IS INSIDE TASMANIA
        OUTSYD=.FALSE.
        EWRONG= E100KM.NE.CHARB .AND.  E100KM.NE.CHARC
     *    .AND.  E100KM.NE.CHARD .AND.  E100KM.NE.CHARE
     *    .AND.  E100KM.NE.CHARF
        NSRONG= N100KM.NE.CHARM .AND.  N100KM.NE.CHARN
     *    .AND.  N100KM.NE.CHARP .AND.  N100KM.NE.CHARQ
     *    .AND.  N100KM.NE.CHARR .AND.  N100KM.NE.CHARS
        OUTSYD= EWRONG .OR.  NSRONG
        IF ( OUTSYD ) WRITE(6,19)
   19   FORMAT('THIS GRID COORDINATE IS NOT IN TASMANIA')
        IF ( OUTSYD ) GOTO 104
C CONVERT 100KM SQUARE DESIGNATION TO 100KM PART OF E,N, COORDINATES
        ECOORD=EAST + (((E100KM-CHARA)/256) + 1)*1000
        IF ( N100KM.GT.CHARN ) N100KM=N100KM-256
C("O" IS NOT USED AS SQUARE DESIGNATION LETTER)
        NCOORD=NORTH + (((N100KM-CHARM)/256) + 1)*1000
        GOTO 150
C
  105   WRITE(6,20)
   20   FORMAT(/,'TYPE IN LOCALITY'/'****************')
        READ(5,21) LOCLTY
   21   FORMAT(4A4)
        GOTO 150
```

```
C
  106    WRITE(6,22)
   22    FORMAT(/,'TYPE IN REGION'/'****************')
         READ(5,21) REGION
         GOTO 150
C
  107    WRITE(6,23)
   23    FORMAT(/,'TYPE IN ROCK DATA:NUMBER OF ROCK TYPES'/
     *          '                  ROCK TYPES 1,2,3,4'/
     *          '                  DATA RELIABILITY'/
     *          '* ** ** ** ** *')
         READ(5,24) NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL
   24    FORMAT(I1,X,4(A2,X),A1)
C CHECK IF RECOGNISABLE ROCK RELIABILITY CODE HAS BEEN ENTERED
         RECHEQ= .FALSE.
         RECHEQ= ROCKRL.NE.CHARD .AND.  ROCKRL.NE.CHARG
     *     .AND.  ROCKRL.NE.CHARM .AND.  ROCKRL.NE.BLANK
         IF ( RECHEQ ) WRITE(6,25)
   25    FORMAT('UNRECOGNISABLE CODE FOR ROCK DATA RELIABILITY'/
     *          'PLEASE REENTER DATA (D/G/M/ )')
         IF ( RECHEQ ) GOTO 107
         GOTO 150
C
  108    WRITE(6,26)
   26    FORMAT(/,'TYPE IN CODES(1,2,3)'/'* * *')
         READ(5,27) CODE1,CODE2,CODE3
   27    FORMAT(A1,X,A1,X,A1)
         GOTO 150
C
  109    WRITE(6,28)
   28    FORMAT(/,'TYPE IN RELIABILITY DATA:ANALYSIS,LOCATION'/'* *')
         READ(5,29) ANALRL,LOCRL
   29    FORMAT(A1,X,A1)
C CHECK IF RECOGNISABLE RELIABILITY CODES HAVE BEEN ENTERED
         RECHEQ=.FALSE.
         AWRONG= ANALRL.NE.CHARR .AND.  ANALRL.NE.CHARD
     *     .AND.  ANALRL.NE.CHARE .AND.  ANALRL.NE.CHARP
     *     .AND.  ANALRL.NE.CHARC .AND.  ANALRL.NE.CHART
     *     .AND.  ANALRL.NE.BLANK
         LWRONG=LOCRL.NE.CHARR .AND.  LOCRL.NE.CHARK
     *    .AND.  LOCRL.NE.CHART .AND.  LOCRL.NE.BLANK
         RECHEQ= AWRONG .OR.  LWRONG
         IF ( RECHEQ ) WRITE(6,30)
   30    FORMAT('UNRECOGNISABLE RELIABILITY CODE'/
     *          'PLEASE REENTER DATA(R/D/E/P/C/T/ ,R/K/T/ )')
         IF ( RECHEQ ) GOTO 109
         GOTO 150
C
  110    WRITE(6,31)
   31    FORMAT(/,'TYPE IN THE TYPE OF WATER SAMPLE'/'*')
         READ(5,32) TYPE
   32    FORMAT(A1)
C CHECK THAT RECOGNISABLE TYPE CODE HS BEEN ENTERED
         RECHEQ=.FALSE.
         RECHEQ=TYPE.NE.CHARB .AND.  TYPE.NE.CHARS .AND.  TYPE.NE.CHARR
     *   . AND.  TYPE. NE. CHARP . AND.  TYPE. NE. CHARL . AND.  TYPE. NE. BLANK
     *   . AND.  TYPE. NE. CHARD
         IF ( RECHEQ ) WRITE(6,33)
   33    FORMAT('UNRECOGNISABLE CODE FOR TYPE OF WATER SAMPLE'
     *          'PLEASE D/REENTED/R DATA (B/S/D/R/P/L/ )')
         IF ( RECHEQ ) GOTO 110
         GOTO 150
C
  111    WRITE(6,34)
   34    FORMAT(/,'TYPE IN FLOW(GALLONS/HOUR)')
         CALL FRINIT(FLOW,IER,5)
         GOTO 150
C
  112    WRITE(6,35)
   35    FORMAT(/,'TYPE IN DEPTH OF BOREHOLE(METRES)')
         CALL FRINIT(DHOLE,IER,5)
         GOTO 150
C
  113    WRITE(6,36)
   36    FORMAT(/,'TYPE IN DEPTH TO WATER(METRES)')
         CALL FRINIT(DWATER,IER,5)
         GOTO 150
C
  114    WRITE(6,37)
   37    FORMAT(/,'TYPE IN CONDUCTIVITY(uS/cm)')
         CALL FRINIT(OMEGA,IER,5)
         GOTO 150
C
  115    WRITE(6,38)
   38    FORMAT(/,'TYPE IN PH')
         CALL FRINIT(PH,IER,5)
         GOTO 150
C
  116    WRITE(6,39)
   39    FORMAT(/,'TYPE IN SALINITY AS PPM TDS')
         CALL FRINIT(TDS,IER,5)
         GOTO 150
C
  117    WRITE(6,40)
   40    FORMAT(/,'TYPE IN TOTAL CARBONATE(PPM):')
         CALL FRINIT(TCO3,IER,5)
         MTCO3=TCO3/61017.
         GOTO 150
C
  118    WRITE(6,41)
   41    FORMAT(/,'TYPE IN CHLORIDE(PPM)')
         CALL FRINIT(CL,IER,5)
         MCL=CL/35453.
         GOTO 150
C
  119    WRITE(6,42)
   42    FORMAT(/,'TYPE IN SULPHATE(PPM)')
         CALL FRINIT(SO4,IER,5)
         MSO4=SO4/96036.
         GOTO 150
```

```
C
  120     WRITE(6,43)
   43     FORMAT(/,'TYPE IN SILICA(PPM)')
          CALL FRINIT(SIO2,IER,5)
          GOTO 150
C
  121     WRITE(6,44)
   44     FORMAT(/,'TYPE IN CALCIUM(PPM)')
          CALL FRINIT(CA,IER,5)
          MCA=CA/40080.
          GOTO 150
C
  122     WRITE(6,45)
   45     FORMAT(/,'TYPE IN MAGNESIUM(PPM)')
          CALL FRINIT(MG,IER,5)
          MMG=MG/24305.
          GOTO 150
C
  123     WRITE(6,46)
   46     FORMAT(/,'TYPE IN POTASSIUM(PPM)')
          CALL FRINIT(K,IER,5)
          MK=K/39098.
          GOTO 150
C
  124     WRITE(6,47)
   47     FORMAT(/,'TYPE IN SODIUM(PPM)')
          CALL FRINIT(NA,IER,5)
          MNA=NA/22990.
          GOTO 150
C
  125     WRITE(6,48)
   48     FORMAT(/,'TYPE IN IRON(PPM)')
          CALL FRINIT(FE,IER,5)
          MFE=FE/55847.
          GOTO 150
C
  126     WRITE(6,49)
   49     FORMAT(/,'TYPE IN ALUMINIUM')
          CALL FRINIT(AL,IER,5)
          MAL=AL/26981.54
          GOTO 150
C
  127     WRITE(6,50)
   50     FORMAT(/,'TYPE IN NITRATE')
          CALL FRINIT(NO3,IER,5)
          MNO3=NO3/62004.9
          GOTO 150
C
  128     WRITE(6,51)
   51     FORMAT(/,'TYPE IN FLUORIDE(PPM)')
          CALL FRINIT(F,IER,5)
          MF=F/18998.4
C
  150 CONTINUE
```

```
C
C
      GOTO 400
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C TRANSFER CORRECTED RECORD TO "WB"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  500 CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,10*')
      CALL POSITN(2,IREC)
      WRITE(2) ALINE
      CALL SYSCOM(I,'AT WB,11*')
      WRITE(6,52) IREC
   52 FORMAT('RECORD NUMBER',I5,' HAS BEEN CORRECTED'/)
C RETURN TO BEGINNING OF INPUT PROCEDURE FOR ANOTHER SAMPLE
      GOTO 100
C
C
  999 WRITE(6,53)
   53 FORMAT('"CRCTWB:  END OF RUN')
      END
C
```

40/62

```
$TITL WBS3
$DBUG
C
C SOURCE FILE FOR "RITEWB"
C
C "RITEWB" READS WATER SAMPLE DATA FROM "WB" OR "FNDWB" AS SPECIFIED
C AND OUTPUTS THIS INFORMATION IN READABLE CODE TO THE PRINTER
C
C
C "WB" IS THE MAIN WATER-ANALYSIS DATA-FILE
C "FNDWB" IS A FILE OF RECORDS FOUND BY A SPECIFIED SEARCH AND
C  TRANSFERRED ( USING THE PROGRAMME MODULE "SRCHWB" )
C
C LOGICAL UNIT ASSIGNMENTS:
C 01-WBD1  -STORES MAXNOS, NUMBER OF RECORDS IN "WB"
C 02-WB    -RANDOM ACCESS DATA FILE
C   -FNDWB -RANDOM ACCESS FILE OF FOUND RECORDS
C 03-WBD3  -STORES PMAXNS, NUMBER OF RECORDS IN "WB" BEFORE LAST UPDATE
C 04-PRINTER-OUTPUT
C 05-VDU   -INPUT
C 06-VDU   -PROMPTS
C 07-WBD5  -STORES FMAXNS NUMBER OF RECORDS IN "FNDWB"
C 08-99    -CORE BUFFER
C
C SUBROUTINES: 0
C FORLIB ROUTINES:  SYSCOM,POSITN,INFREE,NUMIN
      DOUBLE PRECISION
     *       LTNNOS,REFNOS,REF,TTYPE,QTITLE(97,2),
     *       TYPEB,TYPED,TYPEL,TYPEP,TYPER,TYPES,
     *       AARL(6),ARLR(6),ARLD(6),ARLT(6),ARLP(6),ARLC(6),ARLE(6),
     *       RRRL(6),RRLL(6),RRLM(6),
     *       RLUSP(6)
      REAL
     *    LOCLTY(4),REGION(4),
     *    OMEGA,EH,PH,TCO3,CL,SO4,SIO2,CA,MG,K,NA,TDS,FE,AL,NO3,F,
     *    MTCO3,MCL,MSO4,MSIO2,MCA,MMG,MK,MNA,MFE,MAL,MNO3,MF,
     *    FLOW,FLOWLS,DHOLE,DWATER
      INTEGER*2
     *         WBNOS,TYPE,QUADNO,SQUARE,EAST,NORTH,
     *         NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,
     *         CODE1,CODE2,CODE3,ANALRL,LOCRL,
     *         OPTION,START,FINISH,FMAXNS,YES,NO,COUNT,
     *         NREC,RECNOS(50),PSTYLE,FULL,PROOF,
     *         ECOORD,NCOORD,
     *         MAXNOS,PMAXNS,RECORD,I,IREC,ALINE(109),
     *         CHARB,CHARL,CHARP,CHARR,CHARS,
     *         CHARD,CHART,CHARC,CHARE,CHARM,BLANK
C
      LOGICAL
     *       IER
C
      EQUIVALENCE
     *(ALINE(1),LTNNOS),(ALINE(5),REFNOS),(ALINE(9),REF),
     *(ALINE(13),WBNOS),(ALINE(14),TYPE),(ALINE(15),QUADNO),
     *(ALINE(16),SQUARE),(ALINE(17),EAST),(ALINE(18),NORTH),
```

41/62

```
     *(ALINE(19), ECOORD), (ALINE(20), NCOORD),
     *(ALINE(21), LOCLTY), (ALINE(29), REGION),
     *(ALINE(37), NROCK), (ALINE(38), ROCK1), (ALINE(39), ROCK2),
     *(ALINE(40), ROCK3), (ALINE(41), ROCK4), (ALINE(42), ROCKRL),
     *(ALINE(43), CODE1), (ALINE(44), CODE2), (ALINE(45), CODE3),
     *(ALINE(46), ANALRL), (ALINE(47), LOCRL),
     *(ALINE(48), PH), (ALINE(50), TCO3), (ALINE(52), CL),
     *(ALINE(54), SO4), (ALINE(56), SIO2), (ALINE(58), CA), (ALINE(60), MG),
     *(ALINE(62), K), (ALINE(64), NA), (ALINE(66), TDS),
     *(ALINE(68), FE), (ALINE(70), AL), (ALINE(72), NO3), (ALINE(74), F),
     *(ALINE(76), OMEGA), (ALINE(78), EH),
     *(ALINE(80), MTCO3), (ALINE(82), MCL), (ALINE(84), MSO4),
     *(ALINE(86), MSIO2), (ALINE(88), MCA), (ALINE(90), MMG), (ALINE(92), MK),
     *(ALINE(94), MNA), (ALINE(96), MFE), (ALINE(98), MAL), (ALINE(100), NO3),
     *(ALINE(102), MF),
     *(ALINE(104), FLOW), (ALINE(106), DHOLE), (ALINE(108), DWATER)
C
      DATA YES, NO/1HY, 1HN/
      DATA FULL, PROOF/1HF, 1HP/
      DATA CHARB, CHARL, CHARP, CHARR, CHARS/1HB, 1HL, 1HP, 1HR, 1HS/
      DATA CHARD, CHART, CHARC, CHARE, CHARM/1HD, 1HT, 1HC, 1HE, 1HM/
      DATA BLANK/1H /
      DATA TYPEB, TYPED, TYPEL, TYPEP, TYPER, TYPES/
     *8HBOREHOLE, 8HDAM     , 8HLEACHATE, 8HRAINFALL, 8HRIVER   , 8HSPRING  /
      DATA ((QTITLE(I, J), J=1, 2), I=1, 54)/
     *8HCURRIE  , 8H        , 8HNARACOOP, 8HA       , 8HCURRIE  , 8H        ,
     *8HNARACOOP, 8HA       , 8HCURRIE  , 8H        , 8HNARACOOP, 8HA       ,
     *8HN. W. FLIN, 8HDERS  , 8HN. E. FLIN, 8HDERS  , 8HPEARSHAP, 8HE       ,
     *8HGRASSY  , 8H        , 8HGREEN IS, 8HLAND    , 8HS. E. FLIN, 8HDERS  ,
     *8HTHREE HU, 8HMMOCK   , 8HCHAPPELL, 8H ISLAND , 8HCAPE BAR, 8HREN IS. ,
     *8HCAPE GRI, 8HM       , 8HHIGHFIEL, 8HD       , 8H        , 8H        ,
     *8HBANKS ST, 8HRAIT    , 8HWOOLNORT, 8HH       , 8HSMITHTON, 8H        ,
     *8HTABLE CA, 8HPE      , 8HNOLAND B, 8HAY      , 8HBOOBYALL, 8HA       ,
     *8HEDDYSTON, 8HE       , 8HBLUFF PO, 8HINT     , 8HTROWUTTA, 8H        ,
     *8HBURNIE  , 8H        , 8HDEVONPOR, 8HT       , 8HBEACONSF, 8HIELD    ,
     *8HPIPERS R, 8HIVER    , 8HRINGAROO, 8HMA      , 8HBLUE TIE, 8HR       ,
     *8HBALFOUR , 8H        , 8HMAGNET  , 8H        , 8HST VALEN, 8HTINES   ,
     *8HSHEFFIEL, 8HD       , 8HFRANKFOR, 8HD       , 8HLAUNCEST, 8HTON     ,
     *8HALBERTON, 8H        , 8HST HELEN, 8HS       , 8HPIEMAN H, 8HEAD     ,
     *8HCORINNA , 8H        , 8HMACKINTO, 8HSH      , 8HMIDDLESE, 8HX
     *8HQUAMBY  , 8H        , 8HLONGFORD, 8H        , 8HBEN LOMO, 8HND
     *8HST MARYS, 8H        , 8HZEEHAN  , 8H        , 8HMURCHISO, 8HN
     *8HDU CANE , 8H        , 8HGREAT LA, 8HKE      , 8HLAKE RIV, 8HER
      DATA ((QTITLE(I, J), J=1, 2), I=55, 97)/
     *8HSNOW HIL, 8HL       , 8HBICHENO , 8H        , 8HSTRAHAN , 8H
     *8HLYELL   , 8H        , 8HST CLAIR, 8H        , 8HLAKE ECH, 8HO
     *8HINTERLAK, 8HEN      , 8HTOOMS   , 8H        , 8HSWANSEA , 8H
     *8HMACQUARI, 8HE H/BOUR, 8HPILLINGE, 8HR       , 8HKING WIL, 8HLIAM
     *8HOUSE    , 8H        , 8HOATLANDS, 8H        , 8HSWANSTON, 8H
     *8HSCHOUTEN, 8H        , 8HPT HIBBS, 8H        , 8HGORDON  , 8H
     *8HHUNTLEY , 8H        , 8HELLENDAL, 8HE       , 8HBRIGHTON, 8H
     *8HBUCKLAND, 8H        , 8HMARIA   , 8H        , 8HMONTGOME, 8HRY
     *8HROCKY PO, 8HINT     , 8HPEDDER  , 8H        , 8HSTYX    , 8H
     *8HHOBART  , 8H        , 8HSORELL  , 8H        , 8H        , 8H
     *8HDE WITT , 8H        , 8HARTHUR  , 8H        , 8HPICTON  , 8H
     *8HKINGBORO,8HUGH     ,8HTASMAN  ,8H        ,8H        ,8H        ,
     *8HDAVEY   ,8H        ,8HBATHURST,8H        ,8HADAMSON ,8H        ,
     *8HDOVER   ,8H        ,8H        ,8H        ,8HS.W. CAP,8HE       ,
     *8HS.E. CAP,8HE       /
C
      DATA (ARLR(I),I=1,6)/
     *8HCOMPLETE,8H        ,8H        ,8H        ,8H        ,8H        /
      DATA (ARLD(I),I=1,6)/
     *8HCOMPLETE,8H:SOME MA,8HJOR COMP,8HONENTS <,8HDETECTIO,8HN LIMITS/
      DATA (ARLT(I),I=1,6)/
     *8HCOMPLETE,8H:ANION A,8HND CATIO,8HN TOTAL ,8H DO NOT ,8HMATCH   /
      DATA (ARLP(I),I=1,6)/
     *8HPARTIAL ,8HONLY    ,8H        ,8H        ,8H        ,8H        /
      DATA (ARLC(I),I=1,6)/
     *8HTDS FROM,8HCONDUCTI,8HVITY MEA,8HSUREMENT,8HS ONLY  ,8H        /
      DATA (ARLE(I),I=1,6)/
     *8HTDS ONLY,8H FROM QU,8HALITATIV,8HE ESTIMA,8HTE OF SA,8HLINITY  /
C
      DATA (RRLL(I),I=1,6)/
     *8HDATA FRO,8HM BORE-H,8HOLE LOG ,8HOR GEOLO,8HGIST'S R,8HEPORTetc/
      DATA (RRLM(I),I=1,6)/
     *8HDATA GUE,8HSSED FRO,8HM GEOLOG,8HICAL MAP,8H        ,8H        /
      DATA (RLUSP(I),I=1,6)/
     *8HUNSPECIF,8HIED     ,8H        ,8H        ,8H        ,8H        /
C
      CALL SYSCOM(I,'AS 0422,0520,0620,0899*')
C
   50 WRITE(6,1)
    1 FORMAT(
     *'RITEWB: WRITES OUT WATER-ANALYSIS RECORDS'//
     *'THREE OPTIONS ARE AVAILABLE TO YOU:'/
     *' 1.PRINTOUT ALL THE RECORDS IN THE WATER-ANALYSIS DATA-FILE'/
     *' 2.PRINTOUT ALL THE RECORDS ADDED TO THE WATER-ANALYSIS'/
     *'        DATA-FILE IN THE LATEST UPDATE'/
     *' 3.PRINTOUT THE RECORDS FOUND BY A SEARCH AND TRANSFERRED'/
     *'        TO THE DATA-FILE "FNDWB"'/
     *' 4.PRINTOUT A NUMBER OF INDIVIDUAL RECORDS'/
     *'        WHOSE RECORD NUMBERS WILL BE SPECIFIED'//
     *'WHICH OPTION DO YOU REQUIRE?(1/2/3/4)')
      READ(5,2) OPTION
    2 FORMAT(I1)
C
      IF ( OPTION.LT.1 .OR. OPTION.GT.4 ) GOTO 50
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C READ NUMBER OF RECORDS IN VARIOUS FILES
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      CALL SYSCOM(I,'AC WBD1,1*')
      READ(1,3) MAXNOS
      CALL SYSCOM(I,'AC WBD3,3*')
      READ(3,3) PMAXNS
      CALL SYSCOM(I,'AC WBD5,7*')
      READ(7,3) FMAXNS
    3 FORMAT(I5)
```

```
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C ASSIGN VALUES TO "START" AND "FINISH" RECORD NUMBERS
C FOR THE RECORD-WRITING DO-LOOP
C AND ACTIVATE APPROPRIATE RANDOM ACCESS DATA-FILE
C ACCORDING TO THE OPTION REQUIRED
      GOTO (100,200,300,400),OPTION
C
C FOR COMPLETE PRINTOUT OF "WB"
  100 START=1
      FINISH=MAXNOS
      WRITE(6,4) MAXNOS
    4 FORMAT('COMPLETE PRINTOUT OF',I5,' RECORDS IN "WB"')
      CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,11*')
      GOTO 500
C
C FOR PRINTOUT OF LATEST UPDATE OF "WB"
  200 START=PMAXNS+1
      FINISH=MAXNOS
      WRITE(6,5) PMAXNS,MAXNOS
    5 FORMAT('PRINTOUT OF UPDATE',I5,' TO ',I5)
      CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,11*')
      GOTO 500
C
C FOR PRINTOUT OF RECORDS FOUND BY SEARCH AND TRANSFERRED TO "FNDWB"
  300 START=1
      FINISH=FMAXNS
      IF ( FINISH.EQ.0 ) WRITE(6,6)
    6 FORMAT('NO RECORDS FOUND BY SEARCH.NO PRINTOUT')
      IF ( FINISH.EQ.0 ) GOTO 999
      WRITE(6,7) FMAXNS
    7 FORMAT('PRINTOUT OF',I5,' FOUND RECORDS')
      CALL SYSCOM(I,'AC FNDWB,2*')
      CALL SYSCOM(I,'AT FNDWB,11*')
      GOTO 500
C
C FOR PRINTOUT OF A NUMBER OF INDIVIDUAL RECORDS
  400 WRITE(6,8)
    8 FORMAT('HOW MANY RECORDS DO YOU WISH PRINTED?')
      CALL INFREE(NREC,IER,5)
C CHECK THAT THE NUMBER OR RECORDS REQUESTED IS +VE AND =<50
      IF ( NREC.GT.50 ) WRITE(6,9)
    9 FORMAT('SORRY-YOU CANNOT PRINT MORE THAN 50 RECORDS THIS WAY')
      IF ( NREC.GT.50 ) GOTO 400
      IF ( NREC.LT.0 ) WRITE(6,10)
   10 FORMAT('INPUT OF NEGATIVE NUMBER OF REQUIRED RECORDS'/
     *       'WILL CAUSE SEVERE HICCOUGHS IN THE SYSTEM'/
     *       'PLEASE TYPE IN AGAIN')
      IF ( NREC.LT.0 ) GOTO 400
C IF 0 RECORDS REQUIRED THEN SKIP PRINT PROCEDURES
      IF ( NREC.EQ.0 ) GOTO 900
      WRITE(6,11) NREC
   11 FORMAT('TYPE IN THE NUMBERS OF THE',I3,' RECORDS REQUIRED')
```

```
      CALL NUMIN(RECNOS,NREC,IER,5,COUNT)
C CHECK THAT ALL THE REQUIRED RECORD NUMBERS EXIST IN "WB"
      DO 450 I=1,NREC
         IF ( RECNOS(I).LT.0 .OR. RECNOS(I).GE.MAXNOS ) WRITE(6,12)
   12    FORMAT('YOU HAVE TYPED IN AN ILLEGAL RECORD NUMBER'/
     *          'OR AN INCORRECT NUMBER OF RECORD NUMBERS')
         IF ( RECNOS(I).LT.0 .OR. RECNOS(I).GE.MAXNOS ) GOTO 400
  450 CONTINUE
      WRITE(6,13) NREC
   13 FORMAT('PRINTOUT OF',I3,' RECORDS')
      START=1
      FINISH=NREC
      CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,11*')
C
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C DETERMINE TYPE OF OUTPUT REQUIRED
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  500 WRITE(6,14)
   14 FORMAT('DO YOU REQUIRE A FULL PRINTOUT OR A PROOF SHEET(F/P)?')
      READ(5,15) PSTYLE
   15 FORMAT(A1)
      IF ( PSTYLE.NE.FULL .AND. PSTYLE.NE.PROOF ) GOTO 500
C
      IF ( PSTYLE.EQ.PROOF ) GOTO 700
C
C %%%%%%%%%%%%%%%%%%%%%%%%%
C WRITE OUT RECORDS IN FULL
C %%%%%%%%%%%%%%%%%%%%%%%%%
      DO 600 RECORD=START,FINISH
         IF ( OPTION.NE.4 ) IREC=RECORD-1
         IF ( OPTION.EQ.4 ) IREC=RECNOS(RECORD)
         CALL POSITN(2,IREC)
         READ(2) ALINE
C
C CONVERT "TYPE" OUTPUT FROM CODE TO EXPLANATION
         IF (TYPE.EQ.CHARB) TTYPE=TYPEB
         IF (TYPE.EQ.CHARD) TTYPE=TYPED
         IF (TYPE.EQ.CHARS) TTYPE=TYPES
         IF (TYPE.EQ.CHARR) TTYPE=TYPER
         IF (TYPE.EQ.CHARP) TTYPE=TYPEP
         IF (TYPE.EQ.CHARL) TTYPE=TYPEL
C
C CONVERT "ANALRL" AND "ROCKRL" OUTPUT FROM CODE TO EXPLANATION
         DO 650 I=1,6
            IF (ANALRL.EQ.CHARR) AARL(I)=ARLR(I)
            IF (ANALRL.EQ.CHARD) AARL(I)=ARLD(I)
            IF (ANALRL.EQ.CHART) AARL(I)=ARLT(I)
            IF (ANALRL.EQ.CHARP) AARL(I)=ARLP(I)
            IF (ANALRL.EQ.CHARC) AARL(I)=ARLC(I)
            IF (ANALRL.EQ.CHARE) AARL(I)=ARLE(I)
            IF (ANALRL.EQ.BLANK) AARL(I)=RLUSP(I)
            IF (ROCKRL.EQ.BLANK) RRRL(I)=RLUSP(I)
            IF (ROCKRL.EQ.CHARL) RRRL(I)=RRLL(I)
            IF (ROCKRL. EQ. CHARM) RRRL(I)=RRLM(I)
  650    CONTINUE
C
         FLOWLS=FLOW*4. 5/3600
C
C WRITE A RECORD
         WRITE(4, 16)
     *             WBNOS, LTNNOS, REFNOS, REF,
     *             TTYPE,
     *             QUADNO, QTITLE(QUADNO, 1), QTITLE(QUADNO, 2),
     *             SQUARE, EAST, NORTH,
     *             LOCLTY, REGION,
     *             NROCK, ROCK1, ROCK2, ROCK3, ROCK4, RRRL,
     *             CODE1, CODE2, CODE3, LOCRL, AARL,
     *             OMEGA, PH,
     *             TCO3, CL, SO4, SIO2, CA, MG, K, NA, TDS,
     *             FE, AL, NO3, F,
     *             FLOW, FLOWLS,
     *             DHOLE, DWATER
   16   FORMAT(
     *   'IDENTIFICATION:'/
     *   15X, 'RECORD NOS(WB):', I5, 5X, ' LAUNCESTON WATER ANALYSIS NOS:'
     *   , A8, 5X, 'REFERENCE:', A8, '(', A6, ')', 5X/
     *   'TYPE:', A8/
     *   'QUADRANGLE NUMBER:', I2, '(', 2A8, ')', 5X, 'AMG REF:', A2, I3, '/', I3/
     *   'LOCALITY:', 4A4, 10X, 'REGION:', 4A4/
     *   'NUMBER OF ROCK TYPES ASSOCIATED WITH SAMPLE:', I2/
     *   'ROCK TYPE CODES:', 4(X, A2, 2X)/
     *   'RELIABILITY OF ROCK-TYPE DATA:', 6A8/
     *   'CODES:', 3(2X, A1, 2X), '       LOCATION RELIABILITY:', A1/
     *   'RELIABILITY OF CHEMICAL ANALYSIS:', 6A8/
     *    'CONDUCTIVITY=', F6. 1, 'uS/cm    PH=', F4. 1/
     *   10X, 'TOTAL CO3=', F7. 1, 'PPM'/
     *   10X, 'CHLORIDE =', F7. 1, 'PPM'/
     *   10X, 'SULPHATE =', F7. 1, 'PPM'/
     *   10X, 'SILICA   =', F7. 1, 'PPM'/
     *   10X, 'CALCIUM  =', F7. 1, 'PPM'/
     *   10X, 'MAGNESIUM=', F7. 1, 'PPM'/
     *   10X, 'POTASSIUM=', F7. 1, 'PPM'/
     *   10X, 'SODIUM   =', F7. 1, 'PPM'//
     *   'TOTAL DISSOLVED SALTS', F7. 1, 'PPM'/
     *   'FE=', F7. 1, 'AL=', F7. 1, 'NO3=', F7. 1, 'F=', F7. 1/
     *   'FLOW=', I4, 'GALLONS/HOUR     (', I4, 'LITRES/SECOND)'/
     *    'DEPTH OF BOREHOLE=', I4, 'M     DEPTH TO WATER=', I4, 'M'//////)
  600 CONTINUE
C
      GOTO 900
C
C %%%%%%%%%%%%%%%%%%%%%%
C WRITE OUT PROOF SHEET
C %%%%%%%%%%%%%%%%%%%%%%
  700 DO 800 RECORD=START, FINISH
         IF ( OPTION. NE. 4 ) IREC=RECORD-1
         IF ( OPTION. EQ. 4 ) IREC=RECNOS(RECORD)
         CALL POSITN(2, IREC)
         READ(2) ALINE
         WRITE(4,17) WBNOS,LTNNOS,REF,REFNOS,QUADNO,TDS,
     *              SQUARE,EAST,NORTH,TCO3,LOCLTY,CL,REGION,SO4,
     *              NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,SIO2,
     *              CODE1,CODE2,CODE3,CA,ANALRL,LOCRL,MG,
     *              TYPE,K,FLOW,NA,DHOLE,FE,DWATER,AL,OMEGA,NO3,PH,F
   17    FORMAT(
     *   'STORED DATA FOR THE WATER SAMPLE NUMBER,',I5//
     *   '(1)LAUNCESTON ANALYSIS NUMBER   ',A6/
     *   '(2)DATA SOURCE,IDENTIFICATION IN SOURCE:',A6,'/',A8//
     *   '(3)QUADRANGLE NUMBER:',10X,I2,10X,'(16)TDS (PPM):',F7.1/
     *   '(4)AMG 100M GRID REFERENCE:',4X,A2,X,I3,'/',I3,10X,
     *                     '(17)TCO3(PPM):',F7.1/
     *   '(5)LOCALITY:',13X,4A4,10X,'(18)CL  (PPM):',F7.1/
     *   '(6)REGION:',15X,4A4,10X,'(19)SO4 (PPM):',F7.1/
     *   '(7)ROCK TYPE DATA:',8X,I1,X,4(A2,X),A1,10X,
     *                     '(20)SIO2(PPM):',F7.1/
     *   '(8)CODES:',26X,3(X,A1),10X,'(21)CA  (PPM):',F7.1/
     *   '(9)DATA RELIABILITY:',17X,2(X,A1),10X,'(22)MG  (PPM):',F7.1/
     *   '(10)TYPE OF SAMPLE:',21X,A1,10X,'(23)K   (PPM):',F7.1/
     *   '(11)FLOW(G/H):',21X,F6.0,10X,'(24)NA  (PPM):',F7.1/
     *   '(12)DEPTH OF BOREHOLE(M):',10X,F6.1,10X,'(25)FE  (PPM):',F7.1
     *   '(13)DEPTH TO WATER(M):',13X,F6.1,10X,'(26)AL  (PPM):',F7.1/
     *   '(14)CONDUCTIVITY (uS/cm):',9X,F7.1,10X,'(27)NO3 (PPM):',F7.1/
     *   '(15)PH:',30X,F4.1,10X,'(28)F   (PPM):',F7.1//)
  800 CONTINUE
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%
C PRINT DESCRIPTION OF OUTPUT
C %%%%%%%%%%%%%%%%%%%%%%%%%%%
  900 IF (OPTION.EQ.1) WRITE(4,4) MAXNOS
      IF (OPTION.EQ.2) WRITE(4,5) PMAXNS,MAXNOS
      IF (OPTION.EQ.3) WRITE(4,7) FMAXNS
      IF (OPTION.EQ.4) WRITE(4,13) NREC
C
  999 STOP
      END
```

```
$TITL WBS4
$DBUG
C
C SOURCE FILE FOR "SRCHWB"
C
C "SRCHWB" SEARCHES THE WATER ANALYSIS DATA-FILE "WB"
C ACCORDING TO THE SEARCH PARAMETERS INPUT VIA THE VDU
C AND TRANSFERS THE FOUND RECORDS TO A RANDOM ACCESS FILE "FNDWB"
C
C LOGICAL UNIT ASSIGNMENTS:
C 01-WBLOG -RANDOM ACCESS FILE OF SEARCH RESULTS
C 02-WB    -RANDOM ACCESS DATA FILE
C 03-WBD1  -STORES MAXNOS
C   -WBD6  -STORES NFOUND(10),NUMBER OF RECORDS FOUND BY SEARCHES
C 04-PRINTER-OUTPUT
C 05-VDU   -PROMPTS AND INPUT
C 06-FNDWB -RANDOM ACCESS DATA FILE OF FOUND RECORDS
C 07-WBD5  -STORES FMAXNS(NUMBER OF RECORDS IN "FNDWB"
C 08-99    -CORE BUFFER
C
C SUBROUTINES : 6 (DSERCH,PREPLY,SORT,INCLUD,ORDER,CSERCH)
C FUNCTIONS   : 1 (INOUT)
C FORLIB ROUTINES:  SYSCOM,POSITN,FREEIN
C SUBROUTINES PREPLY,SORT.INCLUD AND ORDER; AND FUNCTION INOUT
C  HAVE BEEN TAKEN FROM THE FSTJDN(MINES DEPT.  PROGRAMME COLLECTION)
C
      DOUBLE PRECISION
     *               LTNNOS,REFNOS,REF
      REAL
     *    LOCLTY(4),REGION(4),
     *    OMEGA,EH,PH,TCO3,CL,SO4,SIO2,CA,MG,K,NA,TDS,FE,AL,NO3,F,
     *    MTCO3,MCL,MSO4,MSIO2,MCA,MMG,MK,MNA,
     *    FLOW,DHOLE,DWATER
      INTEGER*2
     *         WBNOS,TYPE,QUADNO,SQUARE,EAST,NORTH,
     *         NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,
     *         CODE1,CODE2,CODE3,ANALRL,LOCRL,
     *         ECOORD,NCOORD,
     *         ALINE(109),MAXNOS,RECORD,IREC,FREC,
     *         NFOUND(10),STRANS,
     *         SNOS,DUMMY,SCH,YES,NO,TOS,CMBNSR,DATBSR,TRNSFR
C
      LOGICAL FOUND(10)
C
      EQUIVALENCE
     *(ALINE(1),LTNNOS),(ALINE(5),REFNOS),(ALINE(9),REF),
     *(ALINE(13),WBNOS),(ALINE(14),TYPE),(ALINE(15),QUADNO),
     *(ALINE(16),SQUARE),(ALINE(17),EAST),(ALINE(18),NORTH),
     *(ALINE(19),ECOORD),(ALINE(20),NCOORD),
     *(ALINE(21),LOCLTY(1)),(ALINE(29),REGION(1)),
     *(ALINE(37),NROCK),(ALINE(38),ROCK1),(ALINE(39),ROCK2),
     *(ALINE(40),ROCK3),(ALINE(41),ROCK4),(ALINE(42),ROCKRL),
     *(ALINE(43),CODE1),(ALINE(44),CODE2),(ALINE(45),CODE3),
     *(ALINE(46),ANALRL),(ALINE(47),LOCRL),
     *(ALINE(48),PH),(ALINE(50),TCO3),(ALINE(52),CL),
     *(ALINE(54),SO4),(ALINE(56),SIO2),(ALINE(58),CA),(ALINE(60),MG),
     *(ALINE(62),K),(ALINE(64),NA),(ALINE(66),TDS),
     *(ALINE(68),FE),(ALINE(70),AL),(ALINE(72),NO3),(ALINE(74),F),
     *(ALINE(76),OMEGA),(ALINE(78),EH),(ALINE(80),MTCO3),
     *(ALINE(82),MCL),(ALINE(84),MSO4),(ALINE(86),MSIO2),
     *(ALINE(88),MCA),(ALINE(90),MMG),(ALINE(92),MK),
     *(ALINE(94),MNA),(ALINE(96),MFE),(ALINE(98),MAL),
     *(ALINE(100),MNO3),(ALINE(102),MF),
     *(ALINE(104),FLOW),(ALINE(106),DHOLE),(ALINE(108),DWATER)
C
      DATA YES,NO/1HY,1HN/
      DATA CMBNSR,DATBSR/1HC,1HD/
C
      CALL SYSCOM(I,'AS 0422,0520,0899*')
C
      WRITE(5,1)
    1 FORMAT(
     *'SRCHWB: SEARCHES THE WATER ANALYSIS DATA-BASE "WB"'/
     *' THE DATA-BASE MAY BE SEARCHED BY THE FOLLOWING PARAMETERS'/
     *'  QUADRANGLE NUMBER,REGION,LOCALITY,COORDINATES(AMG),'/
     *'  TYPE OF WATER SAMPLE,FLOW RATE,DEPTH,ROCK TYPE,SALINITY,'/
     *'  ANALYSIS RELIABILITY AND A NUMBER OF CODES(1,2,3)'//
     *' "PARAMETER" SEARCHES OF THE DATA-BASE MAY ALSO BE COMBINED'/
     *' UP TO 10 DIFFERENT SEARCHES MAY BE CARRIED OUT'/
     *' FOUND RECORDS MAY BE TRANSFERRED TO THE DATA-FILE "FNDWB"'/
     *' A PRINTOUT OF THESE RECORDS MAY BE OBTAINED BY RUNNING'/
     *'  "RITEWB" AFTER THE SEARCH IS COMPLETED'///
     *'AFTER READING INSTRUCTIONS,PRESS "RETURN" KEY TO START')
      READ(5,2) DUMMY
    2 FORMAT(A1)
C
      SNOS=0
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C READ MAXNOS,NUMBER OF RECORDS IN "WB"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      CALL SYSCOM(I,'AC WBD1,3*')
      READ(3,3) MAXNOS
    3 FORMAT(I5)
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C READ NFOUND(10), THE NUMBER OF RECORDS FOUND IN PREVIOUS SEARCHES
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      CALL SYSCOM(I,'AC WBD6,3*')
      CALL SYSCOM(I,'AT WBD6,1*')
      READ(3,4) (NFOUND(I),I=1,10)
    4 FORMAT(10I5)
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C DETERMINE IF SEARCH IS REQUIRED
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 1000 IF ( SNOS.EQ.10 ) GOTO 2000
      WRITE(5,6)
    6 FORMAT('DO YOU REQUIRE A SEARCH?(Y/N)')
      READ(5,2) SCH
      IF ( SCH.EQ.NO ) GOTO 2000
      IF ( SCH.NE.NO .AND.  SCH.NE.YES ) GOTO 1000
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C PRINT NUMBER OF RECORDS FOUND IN PREVIOUS SEARCHES
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      WRITE(5,5) (I,NFOUND(I),I=1,10)
    5 FORMAT(
     *'SEARCH  RECORDS'/
     *'NUMBER   FOUND '/
     *10(I6,3X,I5,/)/)
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%
C CONDUCT SEARCH IF REQUIRED
C %%%%%%%%%%%%%%%%%%%%%%%%%%
C UPDATE NUMBER OF SEARCH
      SNOS=SNOS+1
C DETERMINE IF SEARCH OF DATA-BASE OR COMBINATION OF PREVIOUS SEARCHES
C  IS REQUIRED
 1400 WRITE(5,7)
    7 FORMAT('DO YOU WISH TO COMBINE TWO OF THESE SEARCHES'/
     *'OR SEARCH THE DATA-BASE "WB"?'/
     *'TYPE "C" FOR COMBINATION SEARCH OR "D" FOR DATA-BASE SEARCH')
      READ(5,2) TOS
      IF ( TOS.NE.CMBNSR .AND.  TOS.NE.DATBSR ) GOTO 1400
C COMBINE PREVIOUS SEARCHES IF REQUIRED
      IF ( TOS.EQ.CMBNSR ) CALL CSERCH(SNOS,NFOUND,MAXNOS)
C SEARCH DATABASE IF REQUIRED
      IF ( TOS.EQ.DATBSR ) CALL DSERCH(SNOS,NFOUND,MAXNOS)
C PRINT NUMBER OF RECORDS FOUND BY SEARCH
      WRITE(5,8) NFOUND(SNOS),SNOS
    8 FORMAT(I5,' RECORDS SATISFY THE REQUIREMENTS OF SEARCH NUMBER',I5)
C RETURN FOR ANOTHER SEARCH
      GOTO 1000
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C UPDATE NFOUND(10), NUMBER OF RECORDS FOUND BY SEARCHES
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2000 CALL SYSCOM(I,'AC WBD6,3*')
      CALL SYSCOM(I,'AT WBD6,0*')
      WRITE(3,4) (NFOUND(I),I=1,10)
      CALL SYSCOM(I,'AT WBD6,1*')
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C DETERMINE IF TRANSFER OF RECORDS FOUND BY ONE SEARCH IS REQUIRED
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2200 WRITE(5,9)
    9 FORMAT('DO YOU WISH TO TRANSFER THE RECORDS FOUND BY ONE SEARCH'/
     *'TO THE DATA-FILE "FNDWB"?(Y/N)'/
     *'(IF REQUIRE FULL PRINTOUT OF FOUND RECORDS,TYPE "Y")')
      READ(5,2) TRNSFR
      IF ( TRNSFR.NE.NO .AND.  TRNSFR.NE.YES ) GOTO 2200
      IF ( TRNSFR.EQ.NO ) GOTO 3000
C
```

```
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C TRANSFER A SEARCH TO "FNDWB":STORE NUMBER OF RECORDS IN "WBD5"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2400 WRITE(5,10)
   10 FORMAT('WHICH SEARCH TO YOU WISH TO BE TRANSFERRED AND SAVED?')
      CALL INFREE(STRANS,IER,5)
      IF ( .NOT. (STRANS.GE.1 .AND. STRANS.LE.10) ) GOTO 2400
C
      CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,11*')
      CALL SYSCOM(I,'AC FNDWB,6*')
      CALL SYSCOM(I,'AT FNDWB,10*')
      CALL SYSCOM(I,'AC WBLOG,1*')
      CALL SYSCOM(I,'AT WBLOG,11*')
C
      FREC=0
C
      DO 2500 RECORD=1,MAXNOS
         IREC=RECORD-1
         CALL POSITN(1,IREC)
         READ(1) FOUND
         IF ( .NOT.FOUND(STRANS) ) GOTO 2500
            CALL POSITN(2,IREC)
            READ(2) ALINE
            CALL POSITN(6,FREC)
            WRITE(6) ALINE
            FREC=FREC+1
 2500 CONTINUE
C
      CALL SYSCOM(I,'AT FNDWB,11*')
C
C STORE NUMBER OF RECORDS FOUND AND TRANSFERRED
      CALL SYSCOM(I,'AC WBD5,7*')
      CALL SYSCOM(I,'AT WBD5,0*')
      WRITE(7,3) FREC
      CALL SYSCOM(I,'AT WBD5,1*')
C
      WRITE(5,11) STRANS
   11 FORMAT(/,'THE RECORDS FOUND BY SEARCH NUMBER',I5,/
     *         'TRANSFERRED TO THE FILE "FNDWB"'/
     *         '"RU RITEWB" TO OBTAIN A LISTING OF THESE RECORDS'/)
C
 3000 STOP
      END
C
C
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C SUBROUTINE DSERCH SEARCHES THE DATA-FILE "WB"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      SUBROUTINE DSERCH(SNOS,NFOUND,MAXNOS)
C
      DOUBLE PRECISION
     *                LTNNOS,REFNOS,REF
      REAL
     *    LOCLTY(4),REGION(4),
     *    OMEGA,EH,PH,TCO3,CL,SO4,SIO2,CA,MG,K,NA,TDS,FE,AL,NO3,F,
     *    MTCO3,MCL,MSO4,MSIO2,MCA,MMG,MK,MNA,FLOW,DHOLE,DWATER,
     *    SSREGN(4),SSLOCL(4),SSTDS,FMIN,FMAX,DPTMIN,DPTMAX,MINMAX(2),
     *    X(21),Y(21),XTMP,YTMP,YINTVL(21),RSLOPE(21),
     *    XMAX,XMIN,YMAX,YMIN
      INTEGER*2
     *         WBNOS,TYPE,QUADNO,SQUARE,EAST,NORTH,
     *         NROCK,ROCK1,ROCK2,ROCK3,ROCK4,ROCKRL,
     *         ECOORD,NCOORD,CODE1,CODE2,CODE3,ANALRL,
     *         SSQUAD,SSTYPE,SSANRL,SSROCK,SSCDE1,SSCDE2,SSCDE3,
     *         S1,SQ,SR,SL,SA,ST,SF,SD,SAR,SRT,SC,SC1,SC2,SC3,STDS,
     *         ALINE(109),MAXNOS,RECORD,IREC,SNOS,NFOUND(10),
     *         CESQ,CNSQ,CE,CN,CORDOK,CHARA,CHARM,CHARN,NCORD,
     *         INTVLS,SWATH(21,25),
     *         NO,YES
C
      LOGICAL IER,FOUND(10),INOUT,TFOUND(5000)
C
      EQUIVALENCE
     *(ALINE(1),LTNNOS),(ALINE(5),REFNOS),(ALINE(9),REF),
     *(ALINE(13),WBNOS),(ALINE(14),TYPE),(ALINE(15),QUADNO),
     *(ALINE(16),SQUARE),(ALINE(17),EAST),(ALINE(18),NORTH),
     *(ALINE(19),ECOORD),(ALINE(20),NCOORD),
     *(ALINE(21),LOCLTY(1)),(ALINE(29),REGION(1)),
     *(ALINE(37),NROCK),(ALINE(38),ROCK1),(ALINE(39),ROCK2),
     *(ALINE(40),ROCK3),(ALINE(41),ROCK4),(ALINE(42),ROCKRL),
     *(ALINE(43),CODE1),(ALINE(44),CODE2),(ALINE(45),CODE3),
     *(ALINE(46),ANALRL),(ALINE(47),LOCRL),
     *(ALINE(48),PH),(ALINE(50),TCO3),(ALINE(52),CL),
     *(ALINE(54),SO4),(ALINE(56),SIO2),(ALINE(58),CA),(ALINE(60),MG),
     *(ALINE(62),K),(ALINE(64),NA),(ALINE(66),TDS),
     *(ALINE(68),FE),(ALINE(70),AL),(ALINE(72),NO3),(ALINE(74),F),
     *(ALINE(76),OMEGA),(ALINE(78),EH),(ALINE(80),MTCO3),
     *(ALINE(82),MCL),(ALINE(84),MSO4),(ALINE(86),MSIO2),
     *(ALINE(88),MCA),(ALINE(90),MMG),(ALINE(92),MK),
     *(ALINE(94),MNA),(ALINE(96),MFE),(ALINE(98),MAL),
     *(ALINE(100),MNO3),(ALINE(102),MF),
     *(ALINE(104),FLOW),(ALINE(106),DHOLE),(ALINE(108),DWATER)
C
      DATA YES,NO/1HY,1HN/
      DATA CHARA,CHARM,CHARN/1HA,1HM,1HN/
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%
C DETERMINE SEARCH PARAMETERS
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%
C SET EQUAL TO NO ALL SEARCH REQUESTS THAT OTHERWISE
C MAY NOT BE SET IN THIS PROCEDURE
      SQ=NO
      SR=NO
      SL=NO
      SA=NO
      SC1=NO
      SC2=NO
      SC3=NO
```

```
C
      WRITE(5,1)
    1 FORMAT('DO YOU WISH TO SEARCH BY LOCATION?(Y/N)')
      READ(5,2) S1
    2 FORMAT(A1)
      IF ( S1.NE.YES ) GOTO 200
         WRITE(5,3)
    3    FORMAT('DO YOU WISH TO SEARCH BY QUADRANGLE NUMBER?(Y/N)')
         READ(5,2) SQ
         IF ( SQ.NE.YES ) GOTO 110
            WRITE(5,4)
    4       FORMAT('TYPE IN QUADRANGLE NUMBER'/'**')
            READ(5,5) SSQUAD
    5       FORMAT(I2)
  110    WRITE(5,6)
    6    FORMAT('DO YOU WISH TO SEARCH BY REGION?(Y/N)')
         READ(5,2) SR
         IF ( SR.NE.YES ) GOTO 120
            WRITE(5,7)
    7       FORMAT('TYPE IN REGION NAME'/'****************')
            READ(5,8) SSREGN
    8       FORMAT(4A4)
  120    WRITE(5,9)
    9    FORMAT('DO YOU WISH TO SEARCH BY LOCALITY?(Y/N)')
         READ(5,2) SL
         IF ( SL.NE.YES ) GOTO 130
            WRITE(5,10)
   10       FORMAT('TYPE IN LOCALITY NAME'/'****************')
            READ(5,8) SSLOCL
C
C
  130    WRITE(5,11)
   11    FORMAT('DO YOU WISH TO SEARCH AN AREA DEFINED BY CORNERS'/
     *          'WHOSE EAST/NORTH COORDINATES YOU WILL SPECIFY?(Y/N)')
         READ(5,2) SA
         IF ( SA.NE.YES ) GOTO 200
  135       WRITE(5,13)
   13       FORMAT('NUMBER OF CORNERS DEFINING AREA?(MAXIMUM 20)')
            CALL INFREE(NCORD,IER,5)
            IF (IER) GOTO 135
            IF ( NCORD.GT.20 ) WRITE(5,14) NCORD
   14       FORMAT(I5,' IS GREATER THAN 20!!')
            IF ( NCORD.GT.20 ) GOTO 135
C SET MAX VALUES OF X,Y
            XMAX=-9.9E+9
            YMAX=XMAX
            XMIN=-XMAX
            YMIN=XMIN
C
            WRITE(5,15)
   15        FORMAT(
     *    'ENTER AMG 100M COORDINATES FOR EACH CORNER AS PROMPTED'/
     *    'COORDINATES MUST BE IN THE SAME FORM AS THE EXAMPLE BELOW'/
     *    'CQ156/329')
```

```
C
             DO 140 I=1,NCORD
                WRITE(5,81) I
   81           FORMAT('COORDINATES OF CORNER NUMBER ',I3)
                READ(5,82) CESQ,CNSQ,CE,CN
   82           FORMAT(A1,A1,I3,X,I3)
C CONVERT AMG100M REFERENCE TO X,Y COORDINATES IN METRES
                XTMP=(CE+(((CESQ-CHARA)/256)+1)*1000)*100.0
                IF ( CNSQ.GT.CHARN ) CNSQ=CNSQ-256
                YTMP=(CN+(((CNSQ-CHARM)/256)+1)*1000)*100.0
C DETERMINE MAX AND MIN VALUES OF COORDINATED AREA
                XMAX=AMAX1(XMAX,XTMP)
                XMIN=AMIN1(XMIN,XTMP)
                YMAX=AMAX1(YMAX,YTMP)
                YMIN=AMIN1(YMIN,YTMP)
                X(I)=XTMP
                Y(I)=YTMP
  140       CONTINUE
C
C CLOSE THE COORDINATED AREA IF NOT ALREADY DONE
            IF ( X(1).EQ.X(NCORD) .AND.  Y(1).EQ.Y(NCORD) ) GOTO 137
               NCORD=NCORD+1
               X(NCORD)=X(1)
               Y(NCORD)=Y(1)
C
C CHECK IF COORDINATES OK,PRINT OUT IF OK,OTHERWISE REENTER
  137       WRITE(5,83)
   83       FORMAT('COORDINATES OF THE SEARCH AREA AREA:')
            WRITE(5,84) (X(I),Y(I),I=1,NCORD)
   84       FORMAT(F7.1,2X,F7.1)
            WRITE(5,85)
   85       FORMAT('COORDINATES OK?(Y/N)')
            READ(5,2) CORDOK
        IF ( CORDOK.NE.YES ) GOTO 135
C
            NCORD=NCORD-1
C
C PREPARE POLYGON
            CALL PREPLY(X,Y,NCORD,YINTVL,INTVLS,SWATH,RSLOPE)
C
C
  200 WRITE(5,17)
   17 FORMAT('DO YOU WISH TO SEARCH BY SAMPLE TYPE?(Y/N)')
      READ(5,2) ST
      IF ( ST.NE.YES ) GOTO 300
         WRITE(5,18)
   18    FORMAT('TYPE IN SAMPLE TYPE'/'*')
         READ(5,2) SSTYPE
C
  300 WRITE(5,19)
   19 FORMAT('DO YOU WISH TO SEARCH BY FLOW RATE?(Y/N)')
      READ(5,2) SF
      IF ( SF.NE.YES ) GOTO 400
      WRITE(5,20)
   20 FORMAT('TYPE IN RANGE OF FLOW RATES REQUIRED(LOWEST FIRST)'/
     *'(NB FOR SAMPLES WHERE THERE IS NO FLOW DATA,FLOW IS SET TO 0)'/
     *'****.*-****.*')
      CALL FREEIN(MINMAX,2,IER,5,COUNT)
      FMIN=MINMAX(1)
      FMAX=MINMAX(2)
C
  400 WRITE(5,21)
   21 FORMAT('DO YOU WISH TO SEARCH BY  BOREHOLE DEPTH?(Y/N)')
      READ(5,2) SD
      IF ( SD.NE.YES ) GOTO 500
         WRITE(5,22)
   22    FORMAT('TYPE IN RANGE OF BOREHOLE DEPTHS(M) REQUIRED'/
     *           '****.*-****.*')
         CALL FREEIN(MINMAX,2,IER,5,COUNT)
         DPTMIN=MINMAX(1)
         DPTMAX=MINMAX(2)
C
  500 WRITE(5,23)
   23 FORMAT('DO YOU WISH TO SEARCH BY ANALYSIS RELIABILITY?(Y/N)')
      READ(5,2) SAR
      IF ( SAR.NE.YES ) GOTO 600
         WRITE(5,24)
   24    FORMAT('TYPE IN CODE FOR REQUIRED ANALYSIS RELIABILITY')
         READ(5,2) SSANRL
C
  600 WRITE(5,25)
   25 FORMAT('DO YOU WISH TO SEARCH BY ROCK TYPE?(Y/N)')
      READ(5,2) SRT
      IF ( SRT.NE.YES ) GOTO 700
         WRITE(5,96)
   96    FORMAT('TYPE IN CODE FOR REQUIRED ROCK TYPE'/'**')
         READ(5,26) SSROCK
   26    FORMAT(A2)
C
  700 WRITE(5,27)
   27 FORMAT('DO YOU WISH TO SEARCH BY ANY OF THE CODES (1,2,3)?(Y/N)')
      READ(5,2) SC
      IF ( SC.NE.YES ) GOTO 750
         WRITE(5,28)
   28    FORMAT('DO YOU WISH TO SEARCH BY CODE1?(Y/N)')
         READ(5,2) SC1
         IF ( SC1.NE.YES ) GOTO 710
            WRITE(5,29)
   29       FORMAT('TYPE IN SEARCH CODE1'/'*')
            READ(5,2) SSCDE1
  710    WRITE(5,30)
   30    FORMAT('DO YOU WISH TO SEARCH BY CODE2?(Y/N)')
         READ(5,2) SC2
         IF ( SC2.NE.YES ) GOTO 720
            WRITE(5,31)
   31       FORMAT('TYPE IN SEARCH CODE2'/'*')
            READ(5,2) SSCDE2
  720    WRITE(5,33)
   33    FORMAT('DO YOU WISH TO SEARCH BY CODE3?(Y/N)')
         READ(5,2) SC3
         IF ( SC3.NE.YES ) GOTO 750
            WRITE(5,34)
   34       FORMAT('TYPE IN SEARCH CODE3'/'*')
            READ(5,2) SSCDE3
  750    WRITE(5,35)
   35    FORMAT('DO YOU WISH TO SEARCH BY TDS?(Y/N)')
         READ(5,2) STDS
         IF ( STDS.NE.YES ) GOTO 800
            WRITE(5,36)
   36       FORMAT('TYPE IN RANGE OF SALINITIES REQUIRED'/
     *             '****.*-****.*')
            CALL FREEIN(MINMAX,2,IER,5,COUNT)
            TDSMIN=MINMAX(1)
            TDSMAX=MINMAX(2)
C
  800 CONTINUE
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%
C PRINT OUT SEARCH PARAMETERS
C %%%%%%%%%%%%%%%%%%%%%%%%%%%
      WRITE(5,37) SNOS
   37 FORMAT('SEARCH NUMBER',I5)
      IF ( SQ.EQ.YES ) WRITE(5,38) SSQUAD
   38 FORMAT('QUADRANGLE NUMBER:',I2)
      IF ( SR.EQ.YES ) WRITE(5,39) SSREGN
   39 FORMAT('REGION:',4A4)
      IF ( SL.EQ.YES ) WRITE(5,40) SSLOCL
   40 FORMAT('LOCALITY:',4A4)
      IF ( SA.EQ.YES ) WRITE(5,83)
      IF ( SA.EQ.YES ) WRITE(5,84) (X(I),Y(I),I=1,NCORD)
      IF ( ST.EQ.YES ) WRITE(5,43) SSTYPE
   43 FORMAT('TYPE OF SAMPLE:',A1)
      IF ( SF.EQ.YES ) WRITE(5,44) FMIN,FMAX
   44 FORMAT('FLOW RATE BETWEEN',F6.0,'AND',F6.0'G/H')
      IF ( SD.EQ.YES ) WRITE(5,45) DPTMIN,DPTMAX
   45 FORMAT('DEPTH BETWEEN',F6.1,'AND',F6.1,'METRES')
      IF ( SAR.EQ.YES ) WRITE(5,46) SSANRL
   46 FORMAT('ANALYSIS RELIABILITY:',A1)
      IF ( SRT.EQ.YES ) WRITE(5,47) SSROCK
   47 FORMAT('ROCK TYPE(CODE):',A2)
      IF ( SC1.EQ.YES ) WRITE(5,48) SSCDE1
   48 FORMAT('CODE1:',A1)
      IF ( SC2.EQ.YES ) WRITE(5,49) SSCDE2
   49 FORMAT('CODE2:',A1)
      IF ( SC3.EQ.YES ) WRITE(5,50) SSCDE3
   50 FORMAT('CODE3:',A1)
      IF ( STDS.EQ.YES ) WRITE(5,52) TDSMIN,TDSMAX
   52 FORMAT('SALINITY BETWEEN',F7.1,' AND',F7.1,' PPM')
C
C %%%%%%%%%%%
C SEARCH "WB"
C %%%%%%%%%%%
      CALL SYSCOM(I,'AC WBLOG,1*')
      CALL SYSCOM(I,'AT WBLOG,10*')
      CALL SYSCOM(I,'AC WB,2*')
      CALL SYSCOM(I,'AT WB,11*')
C
      NFOUND(SNOS)=0
C
      DO 1000 RECORD=1,MAXNOS
         IREC=RECORD-1
         CALL POSITN(2,IREC)
         READ(2) ALINE
C
         IF ( S1.NE.YES ) GOTO 1200
         IF ( SQ.NE.YES ) GOTO 1110
            IF ( SSQUAD.NE.QUADNO ) GOTO 1990
 1110    IF ( SR.NE.YES ) GOTO 1120
            IF ( SSREGN(1).NE.REGION(1)
     *       .OR.SSREGN(2).NE.REGION(2)
     *       .OR.SSREGN(3).NE.REGION(3)
     *       .OR.SSREGN(4).NE.REGION(4) )   GOTO 1990
 1120    IF ( SL.NE.YES ) GOTO 1130
            IF ( SSLOCL(1).NE.LOCLTY(1)
     *       .OR.SSLOCL(2).NE.LOCLTY(2)
     *       .OR.SSLOCL(3).NE.LOCLTY(3)
     *       .OR.SSLOCL(4).NE.LOCLTY(4) )  GOTO 1990
 1130    IF ( SA.NE.YES ) GOTO 1200
            XTMP=ECOORD*100.0
            YTMP=NCOORD*100.0
            IF ( XTMP.LE.XMIN .OR.  XTMP.GT.XMAX ) GOTO 1990
            IF ( YTMP.LE.YMIN .OR.  YTMP.GT.YMAX ) GOTO 1990
          IF (.NOT.INOUT(XTMP,YTMP,X,Y,YINTVL,INTVLS,SWATH,RSLOPE))
     *                                                     GOTO 1990
 1200    IF ( ST.NE.YES ) GOTO 1300
            IF ( SSTYPE.NE.TYPE ) GOTO 1990
 1300    IF ( SF.NE.YES ) GOTO 1400
            IF ( FLOW.LT.FMIN .OR.  FLOW.GT.FMAX ) GOTO 1990
 1400    IF ( SD.NE.YES ) GOTO 1500
            IF ( DHOLE.LT.DPTMIN .OR.  DHOLE.GT.DPTMAX ) GOTO 1990
 1500    IF ( SAR.NE.YES ) GOTO 1600
            IF ( SSANRL.NE.ANALRL ) GOTO 1990
 1600    IF ( SRT.NE.YES ) GOTO 1690
            IF ( SSROCK.NE.ROCK1 .AND.  SSROCK.NE.ROCK2
     *      .AND.  SSROCK.NE.ROCK3 .AND.  SSROCK.NE.ROCK4 ) GOTO 1990
 1690    IF ( SC.NE.YES ) GOTO 1730
 1700    IF ( SC1.NE.YES ) GOTO 1710
            IF ( SSCDE1.NE.CODE1 ) GOTO 1990
 1710    IF ( SC2.NE.YES ) GOTO 1720
            IF ( SSCDE2.NE.CODE2 ) GOTO 1990
 1720    IF ( SC3.NE.YES ) GOTO 1730
            IF ( SSCDE3.NE.CODE3 ) GOTO 1990
 1730    IF ( STDS.NE.YES ) GOTO 1800
            IF ( TDS.LT.TDSMIN .OR.  TDS.GT.TDSMAX ) GOTO 1990
C
 1800    TFOUND(IREC)=.TRUE.
         NFOUND(SNOS)=NFOUND(SNOS)+1
         GOTO 1000
 1990    TFOUND(IREC)=.FALSE.
 1000 CONTINUE
```

```
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C TRANSFER SEARCH RESULTS (T/F) TO "WBLOG"
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      DO 2000 RECORD=1,MAXNOS
         IREC=RECORD-1
         CALL POSITN(1,IREC)
         READ(1) FOUND
         FOUND(SNOS)=TFOUND(IREC)
         CALL POSITN(1,IREC)
         WRITE(1) FOUND
 2000 CONTINUE
C
      CALL SYSCOM(I,'AT WBLOG,11*')
C
      RETURN
      END
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C FSTJDN : COLLECTION OF SUBROUTINES AND FUNCTIONS FROM MINES
C          DEPARTMENT PROGRAMME COLLECTION
C DIMENSIONS OF X,Y,YINTVL,RSLOPE,SWATH HAVE BEEN MODIFIED
C SUBROUTINES: PREPLY,SORT,INCLUD,ORDER
C FUNCTION   : INOUT
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
$TITL  FASTJORDAN - POINT IN POLYGON
C PROGRAM COLLECTION FSTJDN
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(21,25)
      REAL X(21),Y(21),YINTVL(21),RSLOPE(21)
      IMPLICIT INTEGER*2 (I-N)
      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(5,401)
  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(21),YINTVL(21),YSORT(21)
      INTEGER*2 UPPER
      IMPLICIT INTEGER*2 (I-N)
      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(21,25),POINTR
      IMPLICIT INTEGER*2 (I-N)
      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(21),Y(21),YINTVL(21),RSLOPE(21),XINTSC(25)
      INTEGER*2 SWATH(21,25),POINTR,SEGNO,UPPER
      IMPLICIT INTEGER*2 (I-N)
      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(5,301) 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(21),Y(21),YINTVL(21),RSLOPE(21)
      INTEGER*2 SWATH(21,25),SEGNO,INTVLS
      IMPLICIT INTEGER*2 (I-N)
      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
C
C
C
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C SUBROUTINE CSERCH COMBINES PREVIOUS SEARCHES
C %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      SUBROUTINE CSERCH(SNOS,NFOUND,MAXNOS)
C
      INTEGER*2
     *          SCHA,SCHB,KEY,UNION,ITSCTN,MAXNOS,IREC,
     *          RECORD,NFOUND(10),SNOS
      LOGICAL
     *        IER,UFOUND,IFOUND,FOUND(10)
C
      DATA UNION,ITSCTN/1HU,1HI/
   10 WRITE(5,1)
    1 FORMAT(
     *'YOU MAY COMBINE ANY TWO PREVIOUS SEARCHES BY EITHER'/
     *' 1.SEARCHING FOR RECORDS FOUND BY EITHER SEARCH(UNION)'/
     *' 2.SEARCHING FOR RECORDS FOUND BY BOTH SEARCHES(INTERSECTION)'//
     *'TYPE IN THE NUMBER OF THE FIRST SEARCH'/'*')
      CALL INFREE(SCHA,IER,5)
      IF ( .NOT. (SCHA.GE.1 .AND. SCHA.LE.10) ) GOTO 10
      WRITE(5,2)
    2 FORMAT('TYPE IN THE NUMBER OF THE SECOND SEARCH'/'*')
      CALL INFREE(SCHB,IER,5)
      IF ( .NOT. (SCHB.GE.1 .AND. SCHB.LE.10) ) GOTO 10
   20 WRITE(5,3)
    3 FORMAT(
     *'WHAT TYPE OF COMBINATION OF THESE SEARCHES IS REQUIRED?(U/I)')
      READ(5,4) KEY
    4 FORMAT(A1)
      IF ( KEY.NE.UNION .AND. KEY.NE.ITSCTN ) GOTO 20
C
      NFOUND(SNOS)=0
C
      CALL SYSCOM(I,'AC WBLOG,1*')
      CALL SYSCOM(I,'AT WBLOG,10*')
C
      DO 100 RECORD=1,MAXNOS
         IREC=RECORD-1
         CALL POSITN(1,IREC)
         READ (1) FOUND
         UFOUND= KEY.EQ.UNION .AND. ( FOUND(SCHA) .OR. FOUND(SCHB) )
         IFOUND= KEY.EQ.ITSCTN .AND. FOUND(SCHA) .AND. FOUND(SCHB)
         FOUND(SNOS)= UFOUND .OR. IFOUND
         CALL POSITN(1,IREC)
         WRITE(1) FOUND
         IF ( FOUND(SNOS) ) NFOUND(SNOS)=NFOUND(SNOS) + 1
  100 CONTINUE
C
      CALL SYSCOM(I,'AT WBLOG,11*')
C
      RETURN
      END
```

MAINTENANCE AND SYSTEM EXTENSIONS

At the time of writing (October 1982), several hundred records have been entered into the WBASE system; this number can readily be increased by increasing the number of cylinders assigned to the random access files WB, WBLOG and FNDWB. Each additional cylinder will increase the storage capacity by about 40 records. The number of records that can be searched is limited by the size of the array, TFOUND, of logical variables in the subroutine DSERCH (source file WBS3).

The files WB and WBLOG should be copied to a second disc from time to time; the frequency of this operation will depend on how often these files are updated.

Listings of the regions and localities input should be kept up to date. In addition, an up to date list of rock type and data source codes should be kept available.

One of the most valuable features of this system is the facility to transfer complete records of water samples, found by a particular search, to a random access file, FNDWB. This facility means that data is available in a form readily useable by analytical and plotting programmes.

A programme to plot binary and ternary diagrams of analysed chemical components (for example Na-Ca-Mg, HCO_3-Cl-SO_4) would be a useful extension to the system. Already existing on disc are programmes to calculate ionic ratios (WCALC) and to calculate equilibrium concentrations of chemical species in a water of known composition (SAQEQ7); these programmes would require only minor mofidications to be tailored to the WBASE system.

WCALC may be used to give an indication of possible water uses (for example, with respect to sodium hazard). The calculation of chemical species distribution (SAQEQ7) enables a more accurate assessment of water quality to be made, and is important for any regional study of the origins of and variations in groundwater chemistry. The programme SAQEQ7 is necessarily rather simple but more detailed aqueous speciation programmes are available elsewhere in Australia (for example, Monash University). Implementation of modifications of these programmes would be an important extension to the WBASE system.

CONCLUSIONS

The water analysis data-base system, WBASE, implemented on the Geological Survey computer provides a readily accessed storage for the water data collected by the Department.

The relatively quick and easy searching of the data-base and the facility for transferring records found by a particular search to a random access file (for rapid plotting and analysis) will save a considerable amount of time and money.

However, to be of most value, it is essential that all the water data held by the Department should be entered into the system as soon as possible.

[27 October 1982]