File

D. B. Simons

SEDIMENTATION STUDY OF THE YAZOO RIVER BASIN

USER'S MANUAL FOR THE YAZOO DATA STORAGE AND RETRIEVAL SYSTEM
VOLUME II

CONTRACT NO. DACW 38-76-C-0193

Prepared for

U. S. ARMY CORPS OF ENGINEERS

VICKSBURG DISTRICT

Vicksburg, Mississippi

Prepared by

Civil Engineering Department
Engineering Research Center
Colorado State University
Fort Collins, Colorado

D. B. Simons
R. M. Li
N. Duong

October 1978

CER78-79DBS-RML-ND20

SEDIMENTATION STUDY OF THE YAZOO RIVER BASIN

USER'S MANUAL FOR THE YAZOO DATA STORAGE AND RETRIEVAL SYSTEM

VOLUME II

CONTRACT NO. DACW 38-76-C-0193

Prepared for

U. S. ARMY CORPS OF ENGINEERS

VICKSBURG DISTRICT

Vicksburg, Mississippi

Prepared by

Civil Engineering Department
Engineering Research Center
Colorado State University
Fort Collins, Colorado

D. B. Simons
R. M. Li
N. Duong

October 1978

CER78-79DBS-RML-ND20

AUTHORIZATION

This user's manual and the Yazoo Data Storage and Retrieval System (YAZDB) was devised for the U.S. Army Corps of Engineers, Vicksburg District, Lower Mississippi Division, under Modification No. P00001, Supplemental Agreement to Contract No. DACW38-76-C-0193. Larry Banks was the authorized Project Manager for the Vicksburg District, and Daryl B. Simons and Ruh-Ming Li were the Principal Investigators for Colorado State University. The purpose of this manual is to assist personnel in the utilization of the YAZDB.

ACKNOWLEDGMENTS

Several people assisted in preparing Volume II of the User's Manual for the Yazoo Data Storage and Retrieval System. The authors extend their thanks to Larry Banks, U. S. Army Corps of Engineers, for his cooperation throughout the entire Yazoo Study.

In addition, the authors wish to thank Glenn Brown for reviewing this manual; and Laurie Ernst and Gary Edelen for providing valuable comments during the preparation of the manual. Appreciation is also extended to Annette Ward, Tammy McFall and Kittie Hook for editing the manual.

YAZOO DATA STORAGE AND RETRIEVAL SYSTEM

VOLUME II

This manual describes the Yazoo Data Storage and Retrieval System (YAZBD). This system was developed as part of the Phase I - Sedimentation Study of the Yazoo River Basin. The system enables users with varying amounts of computer experience to efficiently access, retrieve, store, and analyze large amounts of hydraulic and hydrological data from the Yazoo Basin. Volume I of this manual is designed for general users. It concentrates on accessing, retrieving, and analyzing data.

Volume II is intended for the system analysts and/or data managers. Included are details of program execution, input and output, data updating, flowcharts, variable definitions, and hardware and software requirements.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The Yazoo Data Storage and Retrieval System (YAZDB) manages stage-discharge, discharge, river stage, sediment, channel cross-section, precipitation, control structure, and reservoir data. Eventually the program can be expanded to include watershed information and water quality data as well as detailed reservoir and control structures information.

The objectives of developing the YAZDB is to: 1) design an efficient data system that will store and retrieve data to analyze the evolution of the basin system; 2) expedite the daily duties of the U.S. Army Corps of Engineers; 3) provide a system that can be utilized by persons not proficient with the computer; and 4) develop a system with a flexible structure to enable improvements or expansions without major modifications.

Volume II of this manual was prepared to assist systems analysts and/or data managers in maintaining and upgrading YAZDB. The manual is organized in six sections. General system information is contained in SectionII; SectionIII describes the purpose and use of the command language; program coordination for data management is explained in Section IV; data retrieval and processing operations are described in Sections V and VI, respectively; Section VII covers the information display and plotting requirements, and data updating operations are covered in Section VIII. The appendices contain information on data formats, flowcharts of program operation, definitions of variables, and a program listing.

II. GENERAL INFORMATION

YAZDB consists of two main elements as shown in Figure 1 . The data bank stores all the data files and the data base management system, YAZDB, which allows users to retrieve and process the data to obtain useful information.

The data bank consists of magnetic tapes that store all the data files of the various categories. These tapes are mounted on the computer system whenever is required. The only data file stored permanently in the computer is the river system descriptive data file (DESCRIP) that contains information regarding the status of various data categories at all the gaging stations in the Yazoo Basin along with a node system describing the river system network. If a data category is used frequently, a permanent file can be created from the corresponding data tape to minimize expense and accelerate data retrieval and processing.

The YAZDB consists of four overlay programs and a library of utility subroutines to manage the data. This overlay structure and modular concept aims at increasing the flexibility of the data management system in changing component elements and reducing computer memory requirements. The general structure of the YAZDB is given in Figure 2.

DATA STORAGE

To reduce computer time, and facilitate data updating the Yazoo data base is designed to use ten different data files. The structure of the Yazoo data base is shown in Figure 3. The first file which should always be on the system, DESCRIP, contains the status or inventory of the data contained on the nine other files along with the numbers

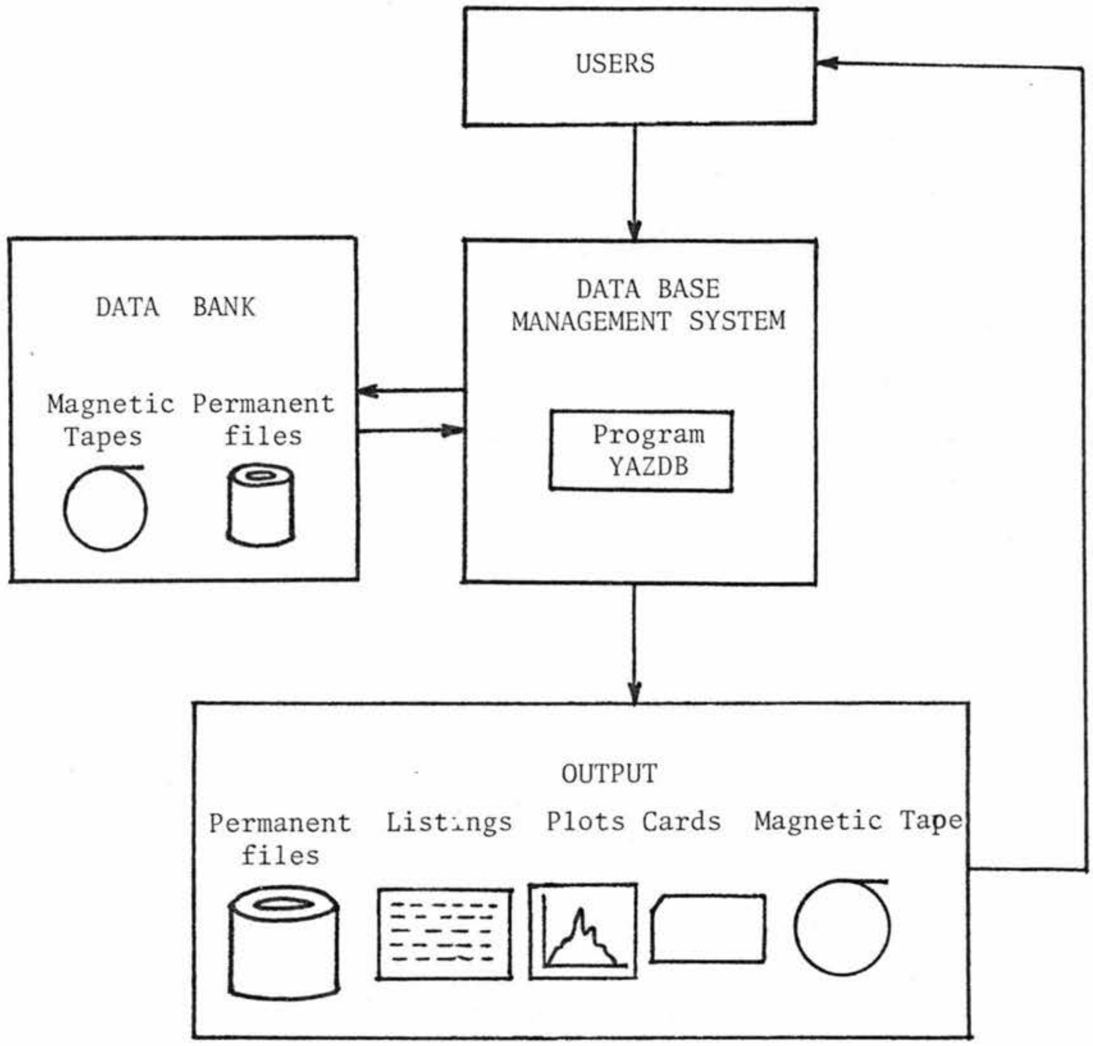


Figure 1. CONCEPTUAL DESIGN OF THE YAZOO DATA STORAGE & RETRIEVAL SYSTEM.

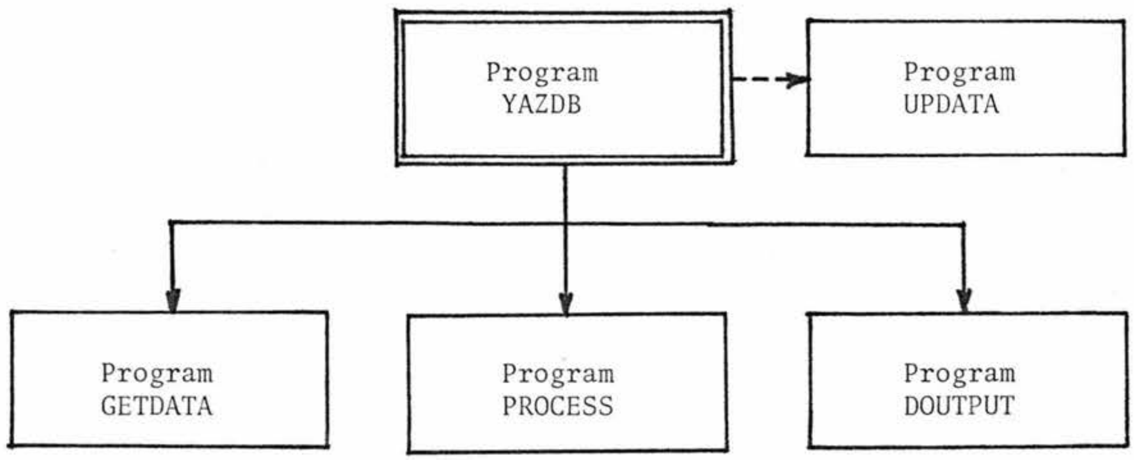
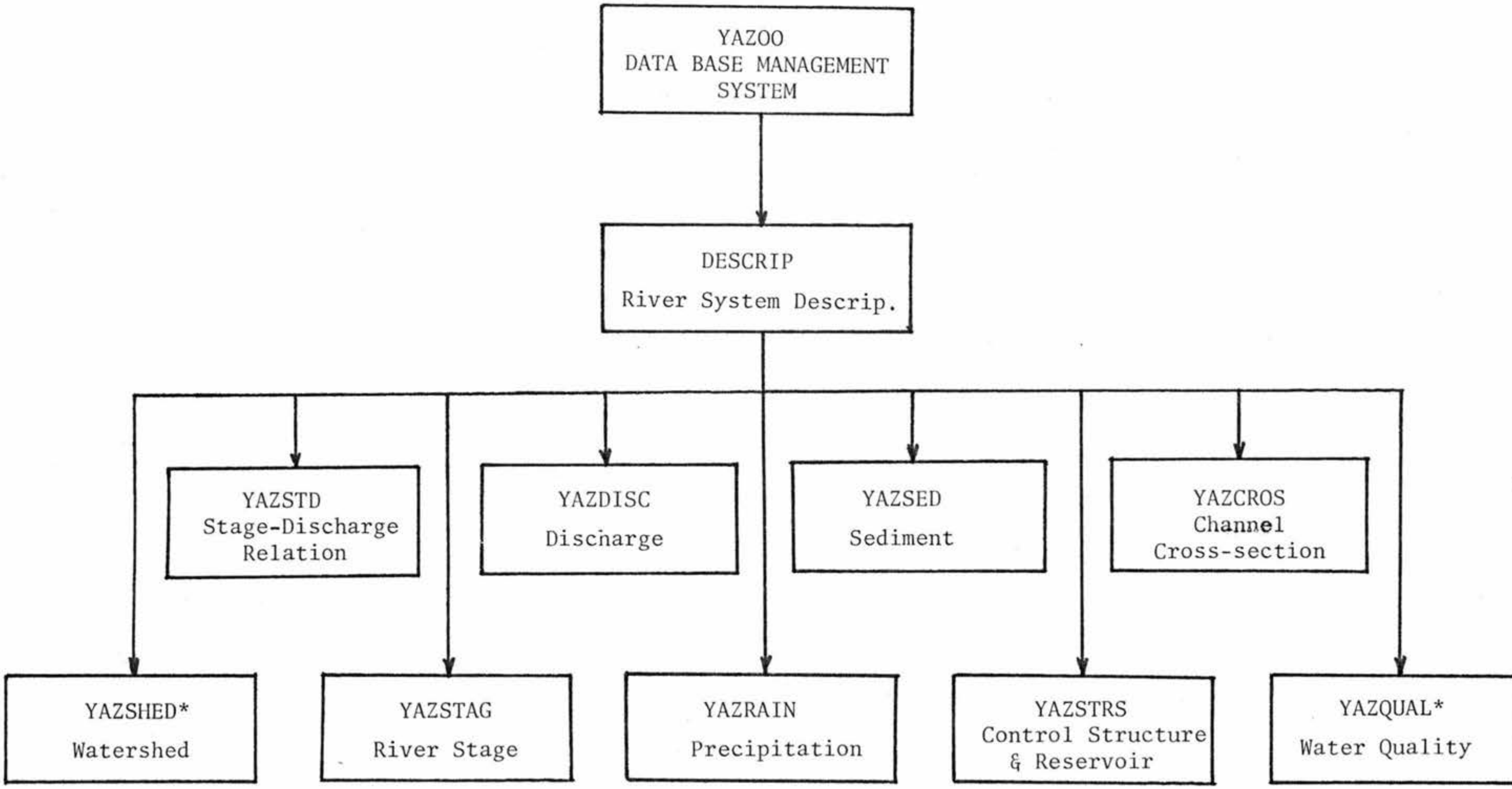


Figure 2. STRUCTURE OF THE YAZOO DATA MANAGEMENT



* Not presently in system

Figure 3. STRUCTURE OF THE YAZOO DATA BASE.

of the magnetic tapes where the data is kept. The last nine files contain the actual data. The different data is divided among the nine by data category. These files need not be maintained on the system at all times. The files can easily be created from the magnetic tapes when needed.

DATA RETRIEVAL

Retrieval of data is accomplished through the sequential access of keys based on a node system developed according to the structure of the Yazoo River system. Retrieval is executed through a simple command language described in Volume I.

Several data specification options are available in the YAZDB. These different options make the program more convenient for the user. The different options, which are presented in Table 1, include three levels of information, the general gaging station information and data status, the data value, and all information available. The four types of location information are basin, river, segment of a river and gaging station. There are five alternative ways of locating a gaging station, which are station name, station number, geographical coordinates, river name (or node) with a river-mile. There are six types of time specification available; calendar year, water year, date, calendar year to calendar year, water year to water year, and from date to date.

DATA PROCESSING

Several data processing options are available in the YAZDB. They include regression analysis, frequency distribution analysis, determination of minimum and maximum, calculation of sample basic statistics, channel

TABLE 1. DATA SPECIFICATION OPTIONS FOR YAZDB.

LEVEL OF INFORMATION	LOCATION	GAGING STATION LOCATION	TIME
Data Status	Basin	Station Name	Calendar Year
Data Value	River	Station Number	Water Year
Gaging Station Information and data value	River Segment	Geographical Coordinates	Date
	Gaging Station	River Name and River-mile	From Cal. Yr (X) to (Y)
		Node Number and River-mile	From Wat. Yr (X) to (Y)
			From Date (X) to (Y)

deformation analysis, sediment concentration analysis, cumulative rainfall analysis and reservoir and control structure surveys.

INFORMATION DISPLAY

Two modes of information display are offered in the YAZDB. The user can select to output the retrieved information on a Tektronix screen by using a Textronix 4010A01, or on paper through a 130 line printer.

DATA UPDATING

Since data file creation has to go through many verification steps, it is assummed that only minor changes are needed to be made in the various data files from time to time. The updating program described in this manual is designed to satisfy this demand.

III. COMMAND LANGUAGE

COMMAND LANGUAGE SPECIFICATIONS

A user-oriented command language was developed to help users understand the computer programming language that retrieves and processes a desired data set from the Yazoo Data Bank. Each data retrieval and processing operation required the use of a command group consisting of two or more command statements. A command group always begins with a GET command and is terminated by a PROCESS command or an OUTPUT command. The structure of each type of command statement is described below.

GET command (GET, < data category >< information level >< sub-level >)

Data category consists of:

ALL (categories)

STAGE-DISCHARGE

SUSPENDED SEDIMENT

BED MATERIAL

CROSS SECTION

CONTROL STRUCTURE

RESERVOIR

DISCHARGE

RIVER STAGE

PRECIPITATION

Information level can be:

STATUS

DATA

ALL INFORMATION

Information sub-levels are available for reservoir and precipitation data only.

For reservoir data:

RULE CURVE

CAPACITY CURVE

SPILLWAY CURVE

For precipitation data:

HOURLY

DAILY

LOCATION command (LOCATION, < location type >< attribute >)

Location type consists of

ALL (locations)

BASIN

RIVER

SEGMENT (of a river)

STATION (gaging station)

Location attributes are used only to specify the location of a station or to delimit a segment of a river. For a station, the following attributes are available:

LOCATED AT/NR (station name)

NUMBER (station number)

COORDINATES (latitude, longitude)

ON (river name)/AT (river mile)

NODE (node number)/AT (river mile)

For a segment of a river, the following attributes are available:

(river name)/FROM (river mile)/TO (river mile)

(node number)/FROM (river mile)/TO (river mile)

TIME command (TIME, < time-period specifications >)

Time period specifications used in the YAZDB are:

ALL (time periods)

YEAR (xxxx)

DATE (DYMOYR)

FROM YEAR (xxxx) TO (xxxx)

FROM DATE (DYMOYR) TO (DYMOYR)

WATER YEAR (xxxx)

FROM WATER YEAR (xxxx) TO (xxxx)

PROCESS command (PROCESS, < output type >< processing type >)

Output type can be one of the following:

LIST

PLOT

DISPLAY (list and plot)

SAVE

In this version of the YAZDB (version 1), the following processing types are implemented:

CUM FREQUENCY (list/plot cumulative frequency CDF)

HISTOGRAM (list/plot relative frequency histogram PDF)

FREQUENCY ANALYSIS (list range, mid-range, PDF, CDF)

THALWEG LEVEL (list/plot river bed profile for a given date)

CHANGING STAGE FOR Q = (a given discharge in CFS) (list/plot the yearly average values of river stage for a given Q)

CUM RAINFALL (list/plot daily cumulative rainfall)

REGRESSION ANALYSIS (list results of a curve fitting to a power function)

DISCHARGE HYDROGRAPH (list/plot a generated discharge hydrograph from a given stage hydrograph based on stage-discharge relationship)

STAGE HYDROGRAPH (list/plot a generated stage hydrograph from a given discharge hydrograph based on stage-discharge relationship)

MIN VALUE (list minimum value)

MAX VALUE (list maximum value)

MIN-MAX (list minimum and maximum values)

BASIC STATISTICS (list minimum, maximum, sample mean and standard deviation)

OUTPUT command (< output type >)

COMMAND LANGUAGE OPERATIONS

The GET command is used to specify the data category type, information level and sub-level desired; the LOCATION command specifies the location type and location attribute, and the TIME command specifies the time period desired. These three command statements define the desired data set to be retrieved from the Yazoo Data Bank and are used to activate the operations of various subprograms in the program GETDATA.

The PROCESS command is used to specify the type of data-processing operation and the type of output desired. This command directs the operations of various subprograms in the program PROCESS. The OUTPUT command is used to specify the type of output desired for displaying the data retrieval results. This command directs the operations of various subprograms in program GETDATA to list information or activate the plotting subprograms PPLOTS and DPLOTS in the program DOUTPUT. The information is displayed by a line printer or on a Textronix screen.

IV. PROGRAM COORDINATION FOR DATA MANAGEMENT

The YAZDB is an overlay computer program package, written in FORTRAN IV language. The main overlay, OVERLAY (0,0), consists of program YAZDB and all utility subroutines which are commonly used by other programs and subprograms in the package. The overall coordination of operations related to data retrieval, processing and display is the main task of the program YAZDB. Its structure is summarized by a flowchart in Figure 4. The following is a comprehensive description of programs and subprograms in the main overlay, including their interrelationships and internal operations. Detailed explanations of data retrieval, processing, display, and updating operations will be covered in Sections V, VI, VII, and VIII, respectively.

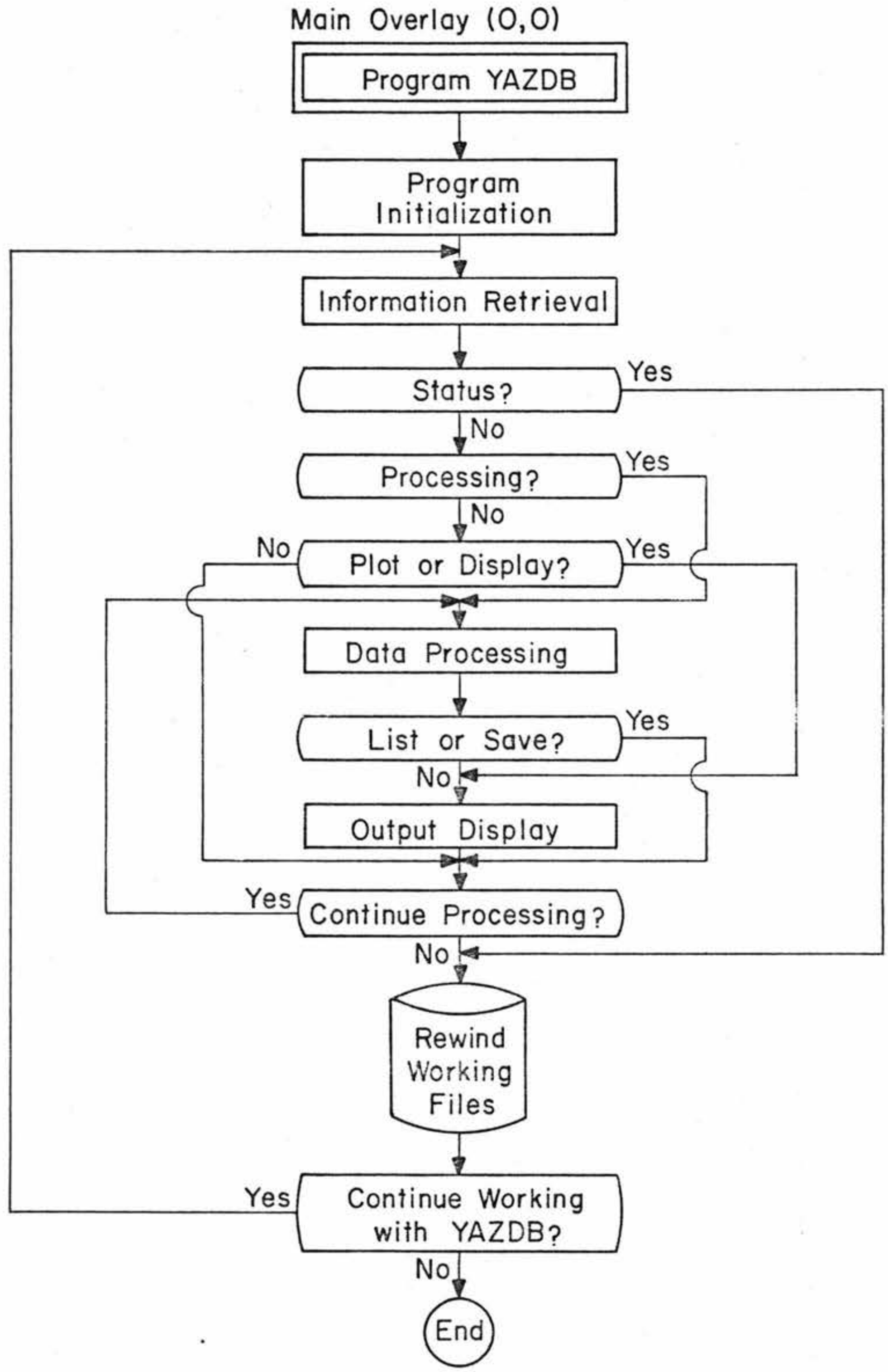


Figure 4. FLOWCHART OF THE MAIN OVERLAY OF THE YAZDB.

Program YAZDB

PROGRAM YAZDB
(INPUT=64, OUTPUT=64, TAPE5=INPUT, TAPE6=OUTPUT, TAPE1=64, TAPE2=64,
TAPE3=64, TAPE11=64, TAPE12=64, TAPE13=64, TAPE14=64, TAPE15=64,
TAPE16=64, TAPE17=64, TAPE18=64)

This program constitutes the main OVERLAY of the YAZDB package. It coordinates all the data management operations, ranging from data retrieval and processing to information display.

Program(s) and Subprogram(s) Called By This Program:

SUBROUTINE INITT

SUBROUTINE OUTFLAG

SUBROUTINE PROFLAG

PROGRAM GETDATA

PROGRAM PROCESS

PROGRAM DOUTPUT

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	working array NTEMP	ICOM, ICOMD
/ELEMT/	none	LEVEL, DF, JCOMWD
/LOCATE/	not currently used here	
/TIME/	not currently used here	
/WORK/	working variable NX working array ITEMP	IN, W, WF1, WF2, WF3
/OUTPUT/	none	IOUT, X, Y, MTIT, IPR ITER, NRECORD
/PLOT/	none	MO
/RIVSED/	not currently used here	
/RIVCRO/	not currently used here	
/RIVSTR/	not currently used here	

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/RIVRES/	not currently used here	
/STADIS/	none	NDAY
/SUB/	none	NNAME, NNODE, NSUB

Declarations:

```
INTEGER W, WF1, WF2, WF3, DF, STNA, STNO, TYPE, YEAR, DATE, CONC,
        CTYP, PCEN1, PCEN2, PCEN3, PCEN4, PCEN5, PCEN6, SNAME,
        SUBLEVL, XTIT, YTIT
REAL MWSL, MXEL, MOAVG, MOMIN, MOMAX
```

Input:

Name	Description
none	

Output:

Name	Description
ICOM	identifiable part of a command statement
ICOMD	index for helping in setting-up command statements (ICOMD=YES)
LEVEL	information level index
DF	data file
JCOMWD	array containing a command statement
IN	input file
W	output file
WF1, WF2, WF3	working files
IOUT	output type index
IPR	data processing index
X	array containing abscissa of points to be plotted
Y	array containing ordinates of points to be plotted

Name	Description
MTIT	array containing title of the graph
ITEK	Tektronix plotting index (ITEK=YES)
NRECORD	total number of records retrieved
MO	array containing months of the year
NDAY	array containing numbers of days in a month for a year
NNAME	array containing names of rivers in the system
NNODE	array containing node numbers in the system
NSUB	total number of rivers and tributaries in the Yazoo basin

Operation(s):

1. Initialize all working arrays and indices.
2. Write headings and print-out instructions if the user wishes to be helped (ICOMD=YES).
3. Call program GETDATA to activate data retrieval operations.
4. If display of the retrieval information is desired, call program DOUTPUT.
5. If data processing operations are desired, call program PROCESS.
6. For multiple processing of the retrieval data set, subroutines OUTFLAG and PROFLAG are called to assign the proper values to output type and data processing type indices, respectively, before going back to call program PROCESS.
7. Print ending message if no further data management operations are desired.

Subroutine RANGE

SUBROUTINE RANGE (X, JI, KN, XHIGH, IPMAX, XLOW, IPMIN)

This subroutine gets the maximum and minimum values for a given subset X of a data set S.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STDSTA

SUBROUTINE STDVAL

SUBROUTINE CDFPDF

SUBROUTINE PPLOTS

SUBROUTINE DPLOTS

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

SUBROUTINE XMAX

SUBROUTINE XMIN

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

DIMENSION X(1)

Input:

Name	Description
X	a given subset of values
JI	starting position in a set S
KN	ending position in a set S

Output:

Name	Description
XHIGH	maximum value in subset X

Name	Description
IPMAX	position of the maximum in set S
XLOW	minimum value in subset X
IPMIN	position of the minimum in set S

Operation(s):

1. Call subroutine XMAX to get XHIGH and IPMAX.
2. Call subroutine XMIN to get XLOW and IPMIN.

Subroutine XMAX

SUBROUTINE XMAX (X, JI, KN, XHIGH, IPMAX)

This subroutine finds the maximum XHIGH and its position IPMAX in a given subset X of a data set S.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RANGE

SUBROUTINE YXMAX

SUBROUTINE DSTAT

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

DIMENSION X(1)

Input:

Name	Description
X	a given subset of values
JI	starting position in a set S
KN	ending position in a set S

Output:

Name	Description
XHIGH	maximum value in subset X
IPMAX	position of the maximum in Set S

Operation(s):

1. Compare values of elements in subset X to find the maximum.
2. Return this value and its position.

Subroutine XMIN

SUBROUTINE XMIN (X, JI, KN, XLOW, IPMIN)

This subroutine finds the minimum XLOW and its position IPMIN in a given subset X of a data set S.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RANGE

SUBROUTINE YXMIN

SUBROUTINE DSTAT

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

DIMENSION X(1)

Input:

Name	Description
X	a given subset of values
JI	starting position in a set S
KN	ending position in a set S

Output:

Name	Description
XLOW	minimum value in subset X
IPMIN	position of the minimum in Set S

Operation(s):

1. Compare values of elements in subset X to find the minimum.
2. Return this value and its position.

Subroutine YXMAX

SUBROUTINE YXMAX (YXHIGH, MYEAR, IPYMAX)

This subroutine finds the maximum value and its position in NYEAR records.

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

SUBROUTINE DREAD

SUBROUTINE XMAX

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, SUBLEVL	none
/WORK/	WF1	none
/OUTPUT/	X, NPTS, NYEAR	NDATA

Declarations:

INTEGER WF1, DF

Input:

Name	Description
WF1	working file
ICATEG	data category index
SUBLEVL	information sublevel
X	a given set of data
NPTS	number of points in the set X
NYEAR	total number of years retrieved

Output:

Name	Description
YXHIGH	maximum value in NYEAR records

Name	Description
MYEAR	year having the maximum
IPYMAX	position of the maximum in year MYEAR
NDATA	total number of points considered

Operation(s):

1. Read-in data from file WF1 by calling subroutine DREAD.
2. Find the maximum and its position in each yearly data set by calling subroutine XMAX.
3. Sort the maximum in NYEAR records and its position by comparing the yearly maxima obtained.

Subroutine YXMIN

SUBROUTINE YXMIN (YXLOW, MYEAR, IPYMIN)

This subroutine finds the minimum value and its position in NYEAR records.

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

SUBROUTINE DREAD

SUBROUTINE XMIN

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, SUBLEVL	none
/WORK/	WF1	none
/OUTPUT/	X, NPTS, NYEAR	NDATA

Declarations:

INTEGER WF1, DF

Input:

Name	Description
WF1	working file
ICATEG	data category index
SUBLEVL	information sublevel
X	a given set of data
NPTS	number of points in the set X
NYEAR	total number of years retrieved

Output:

Name	Description
YXLOW	minimum value in NYEAR records

Name	Description
MYEAR	year having the minimum
IPYMIN	position of the minimum in year MYEAR
NDATA	total number of points considered

Operation(s):

1. Read-in data from file WF1 by calling subroutine DREAD.
2. Find the minimum and its position in each yearly data set by calling subroutine XMIN.
3. Sort the minimum in NYEAR records and its position by comparing the yearly minima obtained.

Subroutine DSTAT

SUBROUTINE DSTAT (YXLOW, YXHIGH, XMEAN, XSTDV)

This subroutine calculates the basic statistics of a given set X containing NPTS data points.

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

SUBROUTINE DREAD

SUBROUTINE XMIN

SUBROUTINE XMAX

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, SUBLEVL	none
/WORK/	WF1	none
/OUTPUT/	X, NPTS, NDATA, NYEAR	none

Declarations:

INTEGER WF1

Input:

Name	Description
WF1	working file
X	a given set of values
NPTS	number of points in the set
NDATA	total number of data points considered
NYEAR	total number of years considered
ICATEG	data category index
SUBLEVL	information sublevel

Output:

Name	Description
YXLOW	minimum value in the set X
YXHIGH	maximum value in the set X
XMEAN	sample mean
XSTDV	sample standard deviation

Operation(s):

1. Read-in data from file WF1 by calling subroutine DREAD.
2. Find the minimum and the maximum elements by calling subroutines XMIN and XMAX, respectively.
3. Calculate the sample mean and standard deviation.

Subroutine DCONVRT

SUBROUTINE DCONVRT (JYEAR, NDT, IMO, IDAY)

This subroutine converts the order of a given date in a year into a date for display.

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

SUBROUTINE LEAPYR

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	NDAY	none

Declarations:

none

Input:

Name	Description
JYEAR	year of a given data set
NDT	order of a given date in JYEAR

Output:

Name	Description
IMO	month of the data element considered
IDAY	day of the data element considered

Operation(s):

1. Check whether JYEAR is a leap year by calling subroutine LEAPYR.
2. Find the corresponding month and day of the order of the given date and return the results.

Subroutine LEAPYR

SUBROUTINE LEAPYR (YEAR, LEAP)

This subroutine identifies whether a given year is a leap year or not.

Subprogram(s) Calling This Subroutine:

SUBROUTINE DCONVRT

SUBROUTINE DORDER

SUBROUTINE STDSTA

SUBROUTINE RSTD

SUBROUTINE PRESTA

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

INTEGER YEAR

Input:

Name	Description
YEAR	year to be identified

Output:

Name	Description
LEAP	leap year index (LEAP=1)

Operation(s):

1. Check whether the given year is divisible by 4. If this is the case, then LEAP=1.

Subroutine PROFLAG

SUBROUTINE PROFLAG (W, IPRO, IPR)

This subroutine sets flags for data processing operations.

Subprogram(s) Calling This Subroutine:

PROGRAM YAZDB

PROGRAM GETDATA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

none

Input:

Name	Description
W	output file
IPRO	identifiable part of the PROCESS command

Output:

Name	Description
IPR	data processing type index

Operation(s):

1. Set values for the data processing type index (IPR).

Subroutine OUTFLAG

SUBROUTINE OUTFLAG (W, IOUTP, IOUT)

This subroutine sets flags for information outputting operations.

Subprogram(s) Calling This Subroutine:

PROGRAM YAZDB

PROGRAM GETDATA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

none

Input:

Name	Description
W	output file
IOUTP	identifiable part of the OUTPUT command

Output:

Name	Description
IOUT	output type index

Operation(s):

1. Set values for the output type index (IOUT).

Subroutine DREAD

SUBROUTINE DREAD (IFILE, ICATEG, SUBLEVL)

This subroutine reads data from file IFILE.

Subprogram(s) Calling This Subroutine:

SUBROUTINE YXMIN

SUBROUTINE YXMAX

SUBROUTINE DSTAT

SUBROUTINE PPLOTS

SUBROUTINE DPLOTS

PROGRAM PROCESS

Subprogram(s) Called by This Subroutine:

SUBROUTINE SWITCH

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/RIVSED/	none	CONC, XSECT, SIZE1, PCEN1, SIZE2, PCEN2, SIZE3, PCEN3, SIZE4, PCEN4, SIZE5, PCEN5, SIZE6, PCEN6
/OUTPUT/	IOUT, IPR, ITEK	X, Y, YEAR, NPTS, IDATE

Declarations:

INTEGER YEAR, SUBLEVL, CONC, PCEN1, PCEN2, PCEN3, PCEN4, PCEN5, PCEN6

Input:

Name	Description
IFILE	working file
ICATEG	data category index
SUBLEVL	information sublevel
IOUT	output type index

Name	Description
IPR	data processing type index
ITEK	Tektronix plotting index

Output:

Name	Description
X	array containing data read from IFILE
Y	array containing data read from IFILE
YEAR	year of the data set
NPTS	number of points in the data set
IDATE	date of the data set
CONC	sediment concentration (ppm)
XSECT	transversal distance in a cross-section
SIZE1	sediment size = .062 mm
PCEN1	cumulative concentration, in percent, for sediment size, SIZE1
SIZE2	sediment size = .125 mm
PCEN2	cumulative concentration, in percent, for sediment size, SIZE2
SIZE3	sediment size = .250 mm
PCEN3	cumulative concentration, in percent, for sediment size, SIZE3
SIZE4	sediment size = .500 mm
PCEN4	cumulative concentration, in percent, for sediment size, SIZE4
SIZE5	sediment size = 1.0 mm
PCEN5	cumulative concentration, in percent, for sediment size, SIZE5
SIZE6	sediment size = 2.0 mm
PCEN6	cumulative concentration, in percent, for sediment size, SIZE6

Operation(s):

1. Read data from file IFILE according to formats specified by data category index (ICATEG).

Subroutine DWRITE

SUBROUTINE DWRITE (IFILE, ICATEG, SUBLEVL)

This subroutine writes data to file IFILE.

Subprogram(s) Calling This Subroutine:

SUBROUTINE HYDSTA

SUBROUTINE SEDDAT

SUBROUTINE RIVGEO

SUBROUTINE GEOSTA

SUBROUTINE RIVRES

SUBROUTINE STADISC

SUBROUTINE STDSTA

SUBROUTINE PRESTA

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/RIVSED/	CONC, XSECT, SIZE1, PCEN1, SIZE2, PCEN2, SIZE3, PCEN3, SIZE4, PCEN4, SIZE5, PCEN5, SIZE6, PCEN6	none
/STADIS/	STAG, DISC, HVALUE	none
/OUTPUT/	X, Y, YEAR, NPTS, IDATE	none

Declarations:

INTEGER YEAR, CONC, PCEN1, PCEN2, PCEN3, PCEN4, PCEN5, PCEN6, SUBLEVL

Input:

Name	Description
IFILE	working file
ICATEG	data category index

Name	Description
SUBLEVL	information sublevel
CONC	sediment concentration (ppm)
XSECT	transversal distance in a cross-section
SIZE1	sediment size = .062 mm
PCEN1	cumulative concentration, in percent, for sediment size, SIZE1
SIZE2	sediment size = .125 mm
PCEN2	cumulative concentration, in percent, for sediment size, SIZE2
SIZE3	sediment size = .250 mm
PCEN3	cumulative concentration, in percent, for sediment size, SIZE3
SIZE4	sediment size = .500 mm
PCEN4	cumulative concentration, in percent, for sediment size, SIZE4
SIZE5	sediment size = 1.0 mm
PCEN5	cumulative concentration, in percent, for sediment size, SIZE5
SIZE6	sediment size = 2.0 mm
PCEN6	cumulative concentration, in percent, for sediment size, SIZE6
STAG	array containing stage values
DISC	array containing discharge values
HVALUE	array containing hourly rainfall in a year
X	array containing data to be written to file IFILE
Y	array containing data to be written to file IFILE
YEAR	year of the data set

Name	Description
NPTS	number of points in the data set
IDATE	date of the data set

Output:

Name	Description
none	

Operation(s):

1. Write data to file IFILE according to formats specified by data category index (ICATEG).

Subroutine MULPLOT

SUBROUTINE MULPLOT (ISYMBOL, IPLOT, MOST, XL, SH, YL, YH)

This subroutine plots multiple curves using a line printer.

Subprogram(s) Calling This Subroutine:

SUBROUTINE CDFPDF

SUBROUTINE PPLOTS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/WORK/	W, IYEAR	none
/OUTPUT/	X, Y, XTIT, YTIT, NPTS, working arrays IPOSX, IPOSY	none
/PLOT/	MO, XUNIT, YUNIT, LINES IOPT	none

Declarations:

```
INTEGER W, XTIT, YTIT, XUNIT, YUNIT
DIMENSION ZX(12), IGRAPH(111), IPOINTS(3)
```

Input:

Name	Description
W	output file
ISYMBOL	symbol used for graph plotting
IPLOT	graph number
MOST	maximum number of graphs to be plotted (MOST=3)
XL	minimum value of abscissa desired
XH	maximum value of abscissa desired
YL	minimum value of ordinate desired
YH	maximum value of ordinate desired

Name	Description
IYEAR	array containing years to be printed
X	array containing abscissa of points to be plotted
Y	array containing ordinates of points to be plotted
XTIT	title to be printed on X-axis
YTIT	title to be printed on Y-axis
NPTS	total number of points to be plotted
MO	array containing months of the year
XUNIT	unit to be printed on X-axis
YUNIT	unit to be printed on Y-axis
LINES	total number of lines to be used by the line printer
IOPT	plotting option, i.e., = 0: regular plot = 1: X-axis will be labeled with alphabetic names = 2: X-axis will plot time in ascending order

Output:

Name	Description
none	

Operation(s):

1. Calculate scale parameters XSCALE and YSCALE.
2. Store plotting positions in working arrays IPOSX and IPOSY.
3. Check plotting option.
4. If IOPT=2, write title on X-axis.
5. Begin plotting and, at the same time, write title on Y-axis.
6. If IOPT≠2, write title on X-axis.

V. DATA RETRIEVAL

All data retrieval operations are carried out in OVERLAY (1,0) which consists of the program GETDATA and related subroutines. Program GETDATA reads and executes the PROCESS, LOCATION, and TIME commands. From PROCESS command, it uses subroutine SETFLAG to determine the data category to be assessed. Then, with information in LOCATION and TIME commands, interpreted through subroutines SETLOC and SETTIM, it directs the retrieval of the correct information element(s) in the data file. The retrieval operation actually occurs at each retrieval component dealing with one particular data category. At these components, the retrieved information is also reformatted and stored in a working file (WF1), ready for processing or display. These latter functions will be explained in detail in Sections VI and VII.

The data retrieval process can be summarized by the flowchart in Figure 5 and a comprehensive description of the program GETDATA and related subroutines follows.

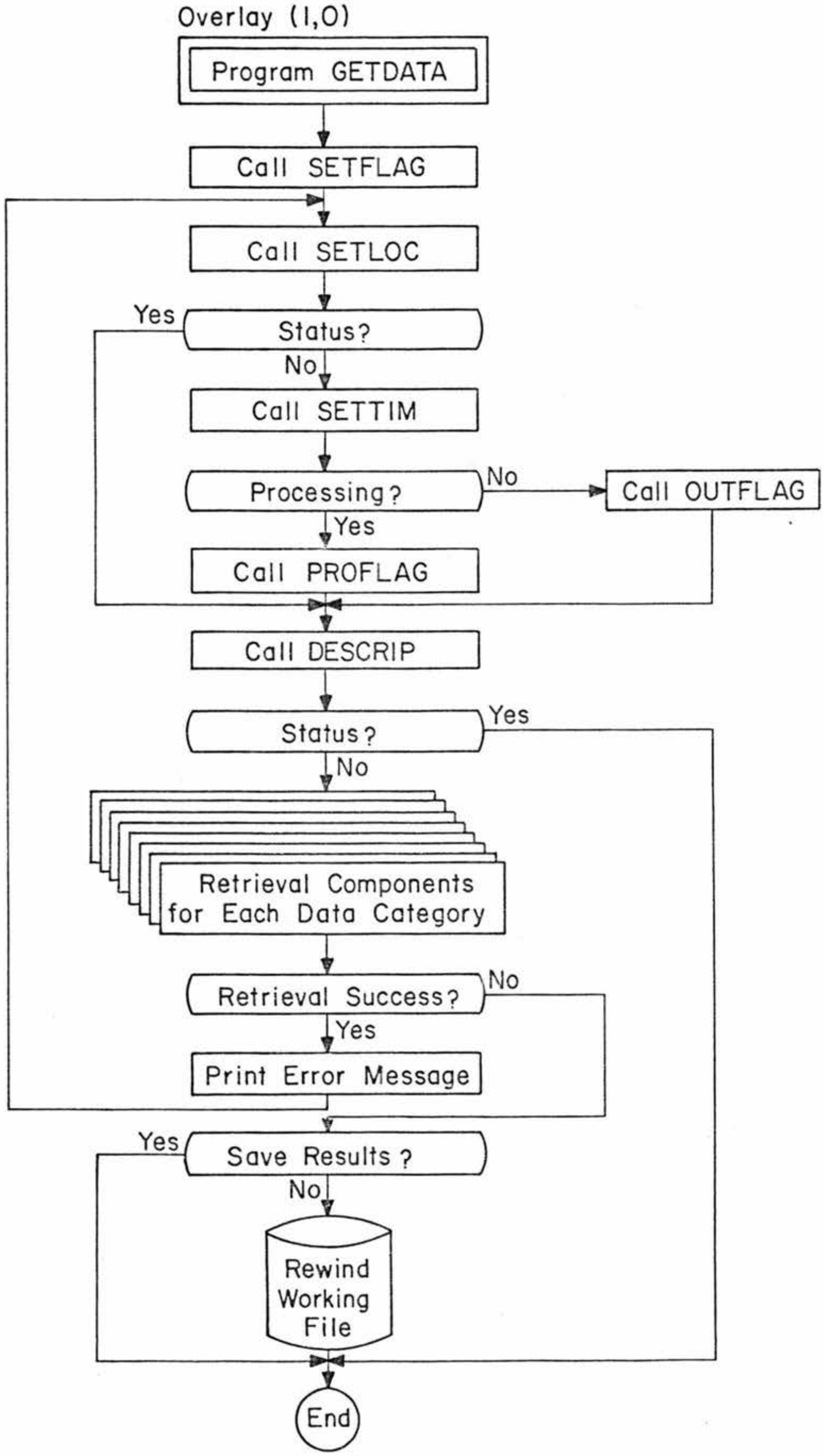


Figure 5. FLOWCHART OF DATA RETRIEVAL PROCESS

Program GETDATA

PROGRAM GETDATA

This program interprets all command statements and coordinates the sequential operation of various data retrieval subroutines.

Program(s) Calling This Program:

PROGRAM YAZDB

Subprogram(s) Called by This Program:

SUBROUTINE SETFLAG

SUBROUTINE SETLOC

SUBROUTINE SETTIM

SUBROUTINE OUTFLAG

SUBROUTINE PROFLAG

SUBROUTINE DESCRIP

SUBROUTINE RIVSYS

SUBROUTINE RIVHYD

SUBROUTINE RIVSED

SUBROUTINE RIVGEO

SUBROUTINE RIVSTR

SUBROUTINE RIVRES

SUBROUTINE STADISC

SUBROUTINE PRECIP

Common Blocks:

Block Name	Variables Obtained From Common Block	Variable Placed In Common Block
/INFORM/	none	ICOM,ICOMD
/ELEMT/	working array JCOMWD	ICATEG,LEVEL,SUBLEVL,DF

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/LOCATE/	None	LFLAG
/TIME/	None	JFLAG
/WORK/	IN, W, WF1	ISET, ILOC, ITIM, IRETRIV, IPRO, IOUTP, ICAT, ISUBL, NYR
/OUTPUT/	None	IOUT, IPR, NYEAR
/PLOT/	Not currently used here	
/RIVSED/	Not currently used here	
/RIVCRO/	Not currently used here	
/RIVRES/	Not currently used here	
/STADIS/	Not currently used here	
/SUB/	Not currently used here	

Declarations:

INTEGER W, WF1, DF

Input:

Name	Description
IN	input file
W	output file
WF1	working file

Output:

Name	Description
ICOM	command type
ICOMD	YES-NO command
ICATEG	data category index
LEVEL	information level index
SUBLEVEL	information sublevel
DF	data file

Name	Description
LFLAG	location type index
JFLAG	time-period type index
ISET	flag for setting data category index
ILOC	flag for setting location type index
ITIM	flag for setting time-period type index
IRETRIV	flag for data retrieval operation
IPRO	identifiable part of process command
IOUTP	identifiable part of output command
ICAT	array containing data category types
ISUBL	array containing information sublevels
NYR	array containing numbers of years having records
IOUT	output type index
IPR	data processing type index
NYEAR	total number of years to be retrieved

Operation (s):

1. Assign value to data category index by calling subroutine SETFLAG.
2. Assign value to location type index by calling subroutine SETLOC.
3. Assign value to time-period type index by calling subroutine SETTIM.
4. Identify data processing command by calling subroutine PROFLAG.
5. Identify output command by calling subroutine OUTFLAG.
6. Get the desired retrieval keys by calling subroutine DESCRIP.
7. Activate the corresponding retrieval component to get the desired data set.

Subroutine SETFLAG

SUBROUTINE SETFLAG

This subroutine sets flags for data categories and information levels and sublevels.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFØRM/	working arrays JTEMP, KTEMP	none
/ELEMT/	none	ICATEG, LEVEL, SUBLEVL
/WØRK/	W, working arrays IY, IZ, ITEMP working variables NX, NY, NZ	ISET

Declarations:

INTEGER W, SUBLEVL

Input:

Name	Description
IY	array containing closures of the GET-command

Output:

Name	Description
ICATEG	flag for data category identification, i.e., = 1: all (categories) = 2: stage-discharge = 3: suspended sediment = 4: bed material = 5: cross-section = 6: control structure = 7: reservoir = 8: discharge = 9: river stage = 10: precipitation

Name	Description
LEVEL	flag for information level, i.e., = 1: status = 2: data = 3: all information
SUBLEVL	flag for information sub-level, i.e., for reservoir data: = all = spillway curve = capacity curve = rule curve for precipitation data: = hourly = daily
ISET	index for sucess or failure in the identification of the GET-command, i.e., = 1: success = 0: failure

Operation(s):

1. Decode the information contained in the GET-command.

2. Assign value(s) to ICATEG, LEVEL, and SUBLEVL.

Subroutine SETLØC

SUBROUTINE SETLØC

This subroutine sets flags for data locations.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFØRM/	working arrays JTEMP, KTEMP, LTEMP	none
/LØCATE/	none	LFLAG, INODE, INAME, NAMST, NUMST, ILAT, ILONG, DIST1, DIST2
/WØRK/	W, working arrays IY, ITEMP working variables, NX, NY, NZ	ILØC

Declarations:

INTEGER W

Input:

Name	Description
IY	array containing closures of the LOCATION-command

Output:

Name	Description
LFLAG	flag for data location identification, i.e., = 1: all (locations) = 2: basin = 3: river = 4: segment (of a river) = 5: station located at/nr (station name) = 6: station number = 7: station coordinates (latitude, longitude) = 8: station on (river name)/at (river-mile) = 9: station node (node number)/at (river-mile)

Name	Description
ILØC	index for success or failure in the identification of the LOCATION-command, i.e., = 1: success = 0: failure
INODE	node number of the desired river
INAME	name of the desired river
NAMST	name of the desired gaging station
NUMST	station number of the desired gaging station
ILAT	latitude of the desired gaging station
ILONG	longitude of the desired gaging station
DIST1	desired starting river-mile
DIST2	desired ending river-mile

Operation(s):

1. Decode the information contained in the LOCATION-command.
2. Assign value(s) to LFLAG.
3. Get the desired information for INAME, NAMST, NUMST, ILAT, ILONG, INODE, DIST1, and DIST2.

Subroutine SETTIM

SUBROUTINE SETTIM

This subroutine sets flag for time-period

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

Subprogram(s) Called By This Subroutine:

SUBROUTINE DØRDER

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFØRM/	working arrays JTEMP, KTEMP	none
/TIME/	none	JFLAG, JSTART, JEND, ND1, JYR1, ND2, JYR2
/WØRK/	W, working arrays IY, IZ, ITEMP working variables NX, NY	ITIM

Declarations:

INTEGER W

Input:

Name	Description
IY	array containing closures of the TIME-command

Output:

Name	Description
JFLAG	flag for time-period identification, i.e., = 1: all (time-periods) = 2: year (....) = 3: date (DYMOYR) = 4: from year (....) to (....) = 5: from date (DYMOYR) to (DYMOYR) = 6: water year (....) = 7: from water year (....) to (....)

Name	Description
ITIM	index for sucess or failure in the identification of the TIME-command, i.e., = 1: success = 0: failure
JSTART	desired starting year or date, alphanumeric
JEND	desired ending year or date, alphanumeric
JYR1	desired starting year (····), integer
JYR2	desired ending year (····), integer
ND1	order of the desired starting date in a given year
ND2	order of the desired ending date in a given year

Operation(s):

1. Decode the information contained in the TIME-command.
2. Assign value(s) to JFLAG.
3. Get the desired information for JSTART, JEND, JYR1, JYR2, ND1, and ND2.

Subroutine DDATE

SUBROUTINE DDATE (JDATE, IDD)

This subroutine checks the date of a data element with the desired time-period.

Subprogram(s) Calling This Subroutine:

SUBROUTINE HYDSTA

SUBROUTINE GEOSTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/TIME/	JSTART, JEND	none

Declarations:

none

Input:

Name	Description
JDATE	date of a given data element
JSTART	desired starting date
JEND	desired ending date

Output:

Name	Description
IDD	index for acceptance or rejection of a given date, i.e., = 1: accepted = 0: rejected

Operation(s):

1. Check the date of a data element with the desired time-period and return the value of IDD.

Subroutine DYEAR

SUBROUTINE DYEAR (IYR, JYD)

Description:

This subroutine checks the year of a data set with the desired time-period.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVHYD

SUBROUTINE HYDSTA

SUBROUTINE SEDSTA

SUBROUTINE RIVGEO

SUBROUTINE STDSTA

SUBROUTINE PRESTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/TIME/	JSTART, JEND	none

Declarations:

none

Input:

Name	Description
IYR	a given year
JSTART	desired starting year
JEND	desired ending year

Output:

Name	Description
JYD	index for acceptance or rejection of a given year, i.e.,

Name	Description
JYD	= 1: accepted = 0: rejected

Operation(s):

1. Check the year of a data set with the desired time-period and return the value of JYD.

Subroutine DESCRIP

SUBROUTINE DESCRIP

This subroutine retrieves data status and get key(s) for data retrieval.

Subprogram(s) Calling This Subroutine:

PRØGRAM GETDATA

Subprogram(s) Called By This Subroutine:

SUBROUTINE STNID

SUBROUTINE HEADING

SUBROUTINE SUBSYS

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFØRM/	working variables ICOM, JTEMP	none
/ELEMT/	ICATEG, LEVEL, DF	none
/LØCATE/	LFLAG, INAME, INODE, NAMST, NUMST, ILAT, ILONG, DIST1, DIST2	none
/TIME/	working variable JNAME	none
/WORK/	W, working variables NX, ITEMP	IRETRIV, STNA, STNO, DIST, LATI, LONG, TYPE NYRS, NODE, NAME,
/SUB/	NSUB	KEY, NNAME

Declarations:

INTEGER W, DF, STNA, STNØ, TYPE

Input:

Name	Description
W	output file
DF	data file

Name	Description
ICATEG	data category index
LEVEL	information level index
LFLAG	location index
INAME	name of the desired river
INØDE	node number of the desired river
NAMST	desired gaging station name
NUMST	desired gaging station number
ILAT	desired gaging station latitude
ILONG	desired gaging station longitude
DIST1	starting river-mile
DIST2	ending river-mile
NSUB	total number of subsystems in the Yazoo river basin system (39)

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process, i.e., = 1: success = 0: failure
STNA	station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
TYPE	data type
NYRS	number of years having records
NODE	river node
NAME	array containing river name

Name	Description
KEY	retrieval key
NNAME	array containing all desired river names

Operation(s):

1. Read file DESCRIP through tape 11 (DF=11).
2. Check location index, LFLAG, and switch the retrieval of data status to various components.
3. Find the correct station location by calling subroutine STNID.
4. Switch to the desired data category.
5. Print status heading by calling subroutine HEADING.
6. Print data status and store retrieval keys by calling subroutine SUBSYS.

Subroutine HEADING

SUBROUTINE HEADING

This subroutine writes headings for displaying data status.

Subprogram(s) Calling This Subroutine:

SUBROUTINE DESCRIP

Subprogram(s) Called By This Subroutine:

none

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG	none
/WØRK/	W	none

Declarations:

INTEGER W

Input:

Name	Description
W	output file
ICATEG	data category index

Output:

Name	Description
none	

Operation(s):

1. Write heading on file W according to data category index.

Subroutine SUBSYS

SUBROUTINE SUBSYS

This subroutine displays data status.

Subprogram(s) Calling This Subroutine:

SUBROUTINE DESCRIP

Subprogram(s) Called by This Subroutine:

none

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, LEVEL, DF	none
/SUB/	KEY	NKEYS, IKEY
/LOCATE/	LFLAG, DIST1, DIST2	none
/WORK/	W, NODE, STNO, DIST, LATI, LONG, STNA, NYRS, TYPE	IYEAR, GELEV, NCR
/OUTPUT/		DATE

Declarations:

INTEGER W, DF, STNA, STNØ, TYPE

Input:

Name	Description
W	output file
DF	data file
ICATEG	data category index
LEVEL	information level index
LFLAG	location index
DIST1	starting river-mile
DIST2	ending river-mile
KEY	retrieval key

Name	Description
NODE	river node
STNO	station number
DIST	river-mile
LATI	latitude
LONG	longitude
STNA	station name
NYRS	number of years having records
TYPE	data type

Output:

Name	Description
NKEYS	total number of retrieval keys
IKEY	array containing retrieval keys
IYEAR	array containing years having records
GELEV	mean elevation, in FT, above MSL
NCR	number of dates having records
DATE	array containing dates having records

Operation(s):

1. Display data status on file W according to data category index.

Subroutine RIVSYS

SUBROUTINE RIVSYS

This subroutine is used to retrieve data of all categories.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

Subprogram(s) Called By This Subroutine:

SUBROUTINE RIVHYD

SUBROUTINE RIVSED

SUBROUTINE RIVGEO

SUBROUTINE RIVSTR

SUBROUTINE RIVRES

SUBROUTINE STADISC

SUBROUTINE PRECIP

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFØRM/	not currently used here	
/ELEMT/	not currently used here	
/LOCATE/	not currently used here	
/TIME/	not currently used here	
/WORK/	not currently used here	
/OUTPUT/	not currently used here	
/PLOT/	not currently used here	
/RIVSED/	not currently used here	
/RIVCRØ/	not currently used here	
/RIVSTR/	not currently used here	
/RIVRES/	not currently used here	
/STADIS/	not currently used here	

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/SUB/	not currently used here	

Declarations:

INTEGER W, WF1, WF2, WF3, DF, STNA, STNO, TYPE, YEAR, DATE, CONC, CTYP, PCEN1, PCEN2, PCEN3, PCEN4, PCEN5, PCEN6, SNAME, SUBLEVL, XTIT, YTIT
REAL MWSL, MXEL, MØAVG, MØMIN, MØMAX

Input:

Name	Description
none	

Output:

Name	Description
none	

Operation(s):

1. Call the other subroutines to retrieve data of all categories.

Subroutine RIVHYD

SUBROUTINE RIVHYD

This subroutine manages stage-discharge relationship data.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE WSTD

SUBROUTINE DYEAR

SUBROUTINE HYDDAT

SUBROUTINE HYDSTA

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	LEVEL, DF	none
/LOCATE/	LFLAG	none
/TIME/	JFLAG	none
/WORK/	W, NAME working arrays IY, ITEMP	IRETRIV, STNA, STNO, DIST, LATI, LONG, GAGO, TYPE, NYRS, SNAME
/ØUTPUT/	none	YEAR, NPTS, DATE, NRECORD
/STADIS/	none	DISC, STAG
/PLØT/	not currently used here	
/SUB/	not currently used here	

Declarations:

INTEGER W, DF, STNA, STNØ, TYPE, YEAR, DATE, SNAME

Input:

Name	Description
W	output file
DF	data file
LEVEL	level of information
LFLAG	location index
JFLAG	time-period index
NAME	array containing river name

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
STNA	station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
GAGO	gage -zero elevation (in FT.)
TYPE	data type
NYRS	number of years having records
SNAME	array containing complete station name for display
YEAR	year of the data set
NPTS	number of data points in the set
DATE	array containing dates
NRECORD	number of retrieved records
DISC	array containing discharge data
STAG	array containing corresponding river stage data

Operation(s):

1. Read file YAZSTD through tape 12 (DF=12).
2. Check the correctness of the data file by calling subroutine CATEG.
3. If data for a particular gaging station is desired, call subroutine HYDSTA.
4. Otherwise, call subroutine WSTD to write headings.
5. Read data year by year and call subroutine DYEAR to check for the desired year.
6. Call subroutine HYDDAT to display stage-discharge data.

Subroutine HYDSTA

SUBROUTINE HYDSTA

This subroutine retrieves stage-discharge data for one particular gaging station.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVHYD

Subprogram(s) Called By This Subroutine:

SUBROUTINE CHKEY

SUBROUTINE WSTD

SUBROUTINE HYDDAT

SUBROUTINE DWRITE

SUBROUTINE DYEAR

SUBROUTINE DDATE

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	not currently used here	
/TIME/	JFLAG, JSTART, JEND	none
/WORK/	W, WF1, NAME working arrays IY, ITEMP	STNA, STNO, DIST, LATI, LONG, GAGO, TYPE, NYRS, SNAME
/OUTPUT/	IOUT, IPR	MTIT, XTIT, YTIT, YEAR, NPTS, NYEAR, DATE, NRECORD
/PLOT/	none	XUNIT, YUNIT, TUNIT
/STADIS/	none	STAG, DISC
/SUB/	NKEYS	KEY

Declarations:

```
INTEGER W, WF1, DF, STNA, STNO, TYPE, YEAR, DATE, SNAME, SUBLEVL,
        XTIT, YTIT, XUNIT, YUNIT, TUNIT
```

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
JFLAG	time-period index
JSTART	starting time-period
JEND	ending time-period
IOUT	output-type index, i.e., = 1: list = 2: plot = 3: display (list and plot) = 4: save
IPR	data processing type index
NKEYS	number of keys to be retrieved

Output:

Name	Description
STNA	array containing gaging station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude

Name	Description
GAGO	gage-zero elevation (in FT.)
TYPE	data type
NYRS	number of years having records
SNAME	array containing complete station name for display
MTIT	array containing title of the graph
XTIT	title on the X-axis
YTIT	title on the Y-axis
XUNIT	unit to be printed on the X-axis
YUNIT	unit to be printed on the Y-axis
TUNIT	time-unit to be printed below the graph title
YEAR	year of the data set
NPTS	number of data points in the set
NYEAR	total number of years to be retrieved
DATE	array containing dates
NRECORD	number of records retrieved
STAG	array containing river stage data
DISC	array containing corresponding discharge data
KEY	retrieval key

Operation(s):

1. Read stage-discharge data from file DF.
2. Check for the desired location by calling subroutine CHKEY.
3. Write headings by calling subroutine WSTD.
4. Retrieve data according to the selected time-period.
5. Check the desired year (by calling subroutine DYEAR) or the desired date (by calling subroutine DDATE).
6. Display stage-discharge data by calling subroutine HYDDAT
7. Write the retrieved data set on file WF1 by calling subroutine DWRITE.

Subroutine CHKEY

SUBROUTINE CHKEY (IDENT)

This subroutine checks retrieval key(s).

Subprogram(s) Calling This Subroutine:

SUBROUTINE HYDSTA

SUBROUTINE SEDSTA

SUBROUTINE GEOSTA

SUBROUTINE RIVSTR

SUBROUTINE RIVRES

SUBROUTINE STDSTA

SUBROUTINE PRESTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/SUB/	IKEY, NKEYS, KEY	none

Declarations:

none

Input:

Name	Description
IKEY	array containing retrieval keys
NKEYS	total number of retrieval keys
KEY	a key used to identify a gaging station

Output:

Name	Description
IDENT	key identification index, i.e., = 1: correct key = 0: incorrect key

Operation(s):

1. Compare the identification key of a gaging station with the set of desired keys and return a value for the key identification index.

Subroutine WSTD

SUBROUTINE WSTD

This subroutine is used to write headings for displaying stage/discharge or sediment data.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVHYD

SUBROUTINE HYDSTA

SUBROUTINE RIVSED

SUBROUTINE SEDSTA

SUBROUTINE STADISC

SUBROUTINE STDSTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/WORK/	SNAME, STNO, DIST, LATI, LONG, GAGO, TYPE, NYRS, W	none

Declarations:

INTEGER W, SNAME, STNO, TYPE

Input:

Name	Description
W	output file
SNAME	array containing gaging station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude

Name	Description
GAGO	gage -zero elevation (in FT.)
TYPE	data type
NYRS	number of years having records

Output:

Name	Description
none	

Operation(s):

1. Write headings for displaying stage/discharge or sediment data on file W.

Subroutine HYDDAT

SUBROUTINE HYDDAT (W, LEVEL)

This subroutine is used to display stage-discharge data.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVHYD

SUBROUTINE HYDSTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	STAG, DISC	none
/OUTPUT/	YEAR, NPTS, DATE	none

Declarations:

INTEGER W

Input:

Name	Description
W	output file
LEVEL	information level
STAG	array containing river stage data
DISC	array containing corresponding discharge data
DATE	array containing corresponding dates
YEAR	year of the data set
NPTS	number of data points in the year

Output:

Name	Description
none	

Operation(s):

1. Write stage-discharge data on file W.

Subroutine RIVSED

SUBROUTINE RIVSED

This subroutine is used to retrieve sediment data.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE WSTD

SUBROUTINE SEDDAT

SUBROUTINE SEDSTA

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	not currently used here	
/ELEMT/	ICATEG, LEVEL, DF	none
/LOCATE/	LFLAG	none
/TIME/	not currently used here	
/WORK/	W, WF1, NAME working arrays IY, ITEMP	IRETRIV, STNA, STNO, DIST, LATI, LONG, GAGO, SNAME, NYRS, TYPE
/OUTPUT/	IOUT	YEAR, NPTS, IDATE, NRECORD
/PLOT/	not currently used here	
/RIVSED/	not currently used here	
/SUB/	not currently used here	

Declarations:

INTEGER W, WF1, DF, STNA, STNO, TYPE, YEAR, SNAME

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
LFLAG	location-type index
NAME	array containing river name
IOUT	output-type index

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
STNA	array containing gaging station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
GAGO	gage -zero elevation (in FT.)
SNAME	array containing complete station name for display
NYRS	number of years having records
TYPE	data type
YEAR	year of the data set
NPTS	number of data points in the set
IDATE	date of a particular data point
NRECORD	total number of records retrieved

Operation(s):

1. Read sediment data from file YAZSED through tape 15(DF=15).

2. Check the correctness of the data file by calling subroutine CATEG.

3. If sediment data for a particular station is desired, call subroutine SEDSTA.

4. Otherwise, write headings by calling subroutine WSTD.

5. Write sediment data on output file W and store the retrieved information on file WF1 by calling subroutine SEDDAT.

Subroutine SEDSTA

SUBROUTINE SEDSTA

This subroutine retrieves sediment data for a particular gaging station.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVSED

Subprogram(s) Called By This Subroutine:

SUBROUTINE CHKEY

SUBROUTINE WSTD

SUBROUTINE SEDDAT

SUBROUTINE DYEAR

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	not currently used here	
/ELEMT/	ICATEG, LEVEL, DF	none
/LOCATE/	not currently used here	
/TIME/	JFLAG	none
/WORK/	W, WF1, NAME working variable NX working arrays IY, ITEMP	STNA, STNO, DIST, LATI, LONG, GAGO, SNAME, NYRS, TYPE
/OUTPUT/	IOUT	MTIT, XTIT, YTIT, YEAR, NPTS, NYEAR, IDAT, NRECORD
/PLOT/	not currently used here	
/RIVSED/	not currently used here	
/SUB/	NKEYS	KEY

Declarations:

INTEGER W, WF1, DF, STNA, STNO, TYPE, YEAR, SNAME, XTIT, YTIT

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
JFLAG	time-period type index
NAME	array containing river name
IOUT	output-type index
NKEYS	number of retrieval keys

Output:

Name	Description
STNA	array containing gaging station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
GAGO	gage -zero elevation (in FT.)
SNAME	array containing complete station name for display
NYRS	number of years having records
TYPE	data type
MTIT	array containing title of the graph
XTIT	title to be printed on the X-axis
YTIT	title to be printed on the Y-axis
YEAR	year of the data set
NPTS	number of data points in the set

Name	Description
NYEAR	total number of years retrieved
IDATE	date of a particular data point
NRECORD	total number of records retrieved
KEY	retrieval key

Operation(s):

1. Read sediment data from file DF.
2. Check for the desired station by calling subroutine CHKEY.
3. Write headings by calling subroutine WSTD.
4. Check for the desired year by calling subroutine DYEAR.
5. Write sediment data on output file W and store the retrieved information on file WF1 by calling subroutine SEDDAT.

Subroutine SEDDAT

SUBROUTINE SEDDAT (W, DF, IDENT, IOUT, NPTS, ICATEG, LEVEL, WF1, IDATE, NRECORD)

This subroutine reads and writes sediment data.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVSED

SUBROUTINE SEDSTA

Subprogram(s) Called By This Subroutine:

SUBROUTINE DWRITE

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/RIVSED/	none	CONC, XSECT, SIZE1, SIZE2, SIZE3, SIZE4, SIZE5, SIZE6, PCEN1, PCEN2, PCEN3, PCEN4, PCEN5, PCEN6

Declarations:

INTEGER W, WF1, DF, CONC, PCEN1, PCEN2, PCEN3, PCEN4, PCEN5, PCEN6

Input:

Name	Description
W	output file
DF	data file
IDENT	station identification index
IOUT	output-type index
NPTS	number of data points in the set
ICATEG	data-category index
LEVEL	information level index
WF1	working file

Output:

Name	Description
IDATE	date of a particular data point
NRECORD	total number of records retrieved
CONC	sediment concentration (ppm)
XSECT	transversal distance in a cross-section
SIZE1	sediment size = .062 mm
SIZE2	sediment size = .125 mm
SIZE3	sediment size = .250 mm
SIZE4	sediment size = .500 mm
SIZE5	sediment size = 1.0 mm
SIZE6	sediment size = 2.0 mm
PCEN1	cumulative concentration, in percent, for sediment size, SIZE1
PCEN2	cumulative concentration, in percent, for sediment size, SIZE2
PCEN3	cumulative concentration, in percent, for sediment size, SIZE3
PCEN4	cumulative concentration, in percent, for sediment size, SIZE4
PCEN5	cumulative concentration, in percent, for sediment size, SIZE5
PCEN6	cumulative concentration, in percent, for sediment size, SIZE6

Operation(s):

1. Write sediment data on file W.
2. Store the retrieved information on file WF1 by calling subroutine DWRITE.

Subroutine RIVGEO

SUBROUTINE RIVGEO

This subroutine is used to retrieve cross-section data.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE NAMNOD

SUBROUTINE DYEAR

SUBROUTINE GEODAT

SUBROUTINE DWRITE

SUBROUTINE GEOSTA

Common Blocks:

Block Name	Variables Obtained From Common Block	Variable Placed In Common Block
/INFORM/	working array KTEMP	none
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	LFLAG	none
/TIME/	JFLAG	none
/WORK/	W, WF1 working variable NZ working array IY	IRETRIV
/OUTPUT/	IOUT, IPR	X, Y, YEAR, NPTS, IDATE, NRECORD
/PLOT/	not currently used here	
/RIVCRO/	none	XDIS, YELEV
/SUB/	none	KEY

Declarations:

INTEGER W, WF1, DF, YEAR

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
LFLAG	location type index
JFLAG	time-period type index
IOUT	output type index
IPR	data processing type index

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
X	array containing transversal distances from a reference line
Y	array containing cross-section elevations
YEAR	year of the data set
NPTS	number of points on cross-section
IDATE	date of a cross-section
NRECORD	total number of records retrieved
XDIS	array containing transversal distances from a reference line

Name	Description
YELEV	array containing cross-section elevations
KEY	retrieval key

Operation(s):

1. Read cross-section data from file YAZCROS through tape 16 (DF=16).
2. Check the correctness of the data file by calling subroutine CATEG.
3. If cross-section data at a location is desired, call subroutine GEOSTA.
4. Read key and write headings on output file W by calling subroutine NAMNOD.
5. Check the desired year by calling subroutine DYEAR.
6. Write cross-section data on output file W by calling subroutine GEODAT.
7. Store the retrieved information on file WF1 by calling subroutine DWRITE.

Permeameter No. 3 Q $ CK ARE STANDARDIZED TO 60.0 DEGREE F

TEST	TIME(HR)	Q(CC/MIN)	CK(CM/SEC)	PRESSURE DROP (CM)			
1	2.00	508.82	.00522661	24.60	21.10	15.00	9.00
2	5.50	442.74	.00447510	25.00	21.10	15.50	9.30
3	8.30	400.73	.00405039	25.00	21.40	15.10	9.00
4	13.80	388.61	.00392797	25.00	21.70	15.30	9.00
5	18.50	355.31	.00357703	25.10	21.50	14.70	8.50
6	25.20	323.98	.00324871	25.20	22.10	14.70	8.50
7	49.10	340.94	.00347384	24.80	21.20	13.60	7.80
8	72.80	335.27	.00340243	24.90	22.00	13.30	7.40
9	97.80	335.27	.00337532	25.10	21.80	14.00	7.90
10	121.50	249.76	.00253464	24.90	21.60	11.90	7.10
11	145.50	216.40	.00218733	25.00	21.50	11.60	6.00
12	169.20	277.65	.00278413	25.20	21.80	12.00	5.50
13	193.50	117.68	.00122378	24.30	23.10	7.60	3.70
14	242.00	147.04	.00145713	25.50	23.20	7.00	3.50
15	269.00	112.23	.00112090	25.30	22.70	6.70	2.40
16	313.00	193.71	.00197378	24.80	20.10	12.90	5.90
17	337.30	82.85	.00085104	24.60	21.00	8.70	3.80
18	361.30	70.24	.00070434	25.20	21.60	7.50	3.40
19	385.50	69.78	.00069968	25.20	20.40	9.90	4.40
20	433.60	57.09	.00057705	25.00	20.70	11.30	4.20
21	479.40	33.58	.00034217	24.80	21.40	9.60	3.70
22	505.20	46.90	.00047031	25.20	22.20	8.50	3.40
23	529.70	36.66	.00037203	24.90	21.80	6.80	3.00
24	557.00	16.46	.00016568	25.10	22.10	3.80	1.40
25	648.80	16.90	.00017150	24.90	23.50	5.50	1.50
26	672.80	30.28	.00030008	25.50	22.40	5.30	1.90
27	720.60	28.45	.00029827	24.10	22.50	3.50	1.50
28	793.00	63.82	.00065031	24.80	24.10	11.60	2.60
29	816.50	17.06	.00017458	24.70	24.40	4.40	2.10
30	863.40	23.99	.00024055	25.20	25.10	3.20	.90
31	888.00	31.57	.00031914	25.00	23.00	8.40	2.30
32	984.00	16.36	.00016535	25.00	24.10	4.40	1.40
33	1056.00	15.07	.00015172	25.10	24.30	4.80	.90

Permeameter No. 4 Q $ CK ARE STANDARDIZED TO 60.0 DEGREE F

TEST	TIME(HR)	Q(CC/MIN)	CK(CM/SEC)	PRESSURE DROP (CM)			
1	2.00	568.72	.00572549	25.10	20.10	14.30	7.50
2	5.50	546.66	.00554767	24.90	19.80	13.90	7.20
3	8.30	511.49	.00517000	25.00	20.60	14.10	7.40
4	13.80	465.86	.00465290	25.30	21.20	14.30	7.70
5	18.50	421.42	.00425961	25.00	20.70	14.20	7.60
6	25.20	388.04	.00390651	25.10	20.70	14.00	7.50
7	49.10	375.11	.00376139	25.20	21.20	13.80	7.00
8	72.80	280.24	.00284395	24.90	20.30	13.00	5.90
9	97.80	287.86	.00287510	25.30	18.70	13.60	6.50
10	121.50	259.07	.00263976	24.80	17.10	11.00	4.80
11	145.50	264.92	.00267768	25.00	19.80	9.50	4.30
12	169.20	171.55	.00174799	24.80	19.30	8.30	3.20
13	193.50	131.32	.00134887	24.60	18.90	5.00	2.00
14	242.00	115.32	.00115636	25.20	16.60	4.40	1.90
15	269.00	85.58	.00089358	24.20	17.40	7.90	3.00
16	313.00	274.48	.00275237	25.20	19.10	11.60	5.60
17	337.30	112.86	.00112720	25.30	19.70	7.30	3.00
18	361.30	66.30	.00065954	25.40	19.90	5.80	2.10
19	385.50	77.44	.00079220	24.70	17.80	7.80	2.80
20	433.60	82.08	.00083968	24.70	16.90	9.00	4.10
21	479.40	37.68	.00038233	24.90	13.70	6.00	2.30
22	505.20	25.41	.00025278	25.40	10.70	4.70	.90
23	529.70	33.27	.00033233	25.30	10.30	4.50	1.40
24	557.00	24.98	.00031401	20.10	10.40	4.30	1.40
25	648.80	22.09	.00022603	24.70	16.60	4.90	1.80
26	672.80	34.20	.00034986	24.70	19.80	5.30	1.60
27	720.60	14.82	.00014916	25.10	20.00	5.20	1.70
28	793.00	69.47	.00069381	25.30	19.20	8.10	2.70
29	816.50	21.93	.00022076	25.10	17.60	5.60	1.30
30	863.40	25.56	.00025831	25.00	19.40	4.10	.90
31	888.00	41.96	.00041098	25.80	21.80	8.80	1.60
32	984.00	14.79	.00015073	24.80	22.40	8.20	2.00
33	1056.00	16.68	.00020968	20.10	17.00	4.80	1.30

Subroutine GEOSTA

SUBROUTINE GEOSTA

This subroutine retrieves cross-section data for a particular location along the river channel.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVGEO

Subprogram(s) Called By This Subroutine:

SUBROUTINE NAMNOD

SUBROUTINE CHKEY

SUBROUTINE DDATE

SUBROUTINE GEODAT

SUBROUTINE DWRITE

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	working array KTEMP	none
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	LFLAG, DIST1, DIST2	none
/TIME/	JFLAG	none
/WORK/	W, WF1, WF2, DIST, NAME	SNAME, NCR
/OUTPUT/	IOUT, IPR	X, Y, MTIT, XTIT, YTIT, NPTS, NYEAR, IDATE, NRECORD
/PLOT/	none	XUNIT, YUNIT, TUNIT
/RIVCRO/	none	XDIS, YELEV
/SUB/	NKEYS, KEY	none

Declarations:

INTEGER W, WF1, WF2, DF, SNAME, SUBLEVL, XTIT, YTIT, XUNIT, YUNIT, TUNIT

Input:

Name	Description
W	output file
WF1, WF2	working files
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
LFLAG	location-type index
DIST	river-mile
DIST1	starting river-mile
DIST2	ending river-mile
JFLAG	time-period type index
NAME	array containing river name
IØUT	output type index
IPR	data processing type index
NKEYS	total number of retrieval keys
KEY	retrieval key

Output:

Name	Description
SNAME	array containing complete station name for display
NCR	number of cross-sections retrieved
X	array containing transversal distances from a reference line

Name	Description
Y	array containing cross-section elevations
MTIT	array containing title of the graph
XTIT	title to be printed on the X-axis
YTIT	title to be printed on the Y-axis
NPTS	total number of data points
NYEAR	total number of years retrieved
IDATE	date of a cross-section
NRECORD	total number of records retrieved
XUNIT	unit to be printed on the X-axis
YUNIT	unit to be printed on the Y-axis
TUNIT	time-unit to be printed below the graph title
XDIS	array containing transversal distances from a reference line
YELEV	array containing cross-section elevations

Operation(s):

1. Read cross-section data from file DF.
2. Read key and write headings on output file W by calling subroutine NAMNOD.
3. Check for the desired location by calling subroutine CHKEY.
4. Check for the desired date by calling subroutine DDATE.
5. Write cross-section data on output file W by calling subroutine GEODAT.
6. Store the retrieved information on file WF1 by calling subroutine DWRITE.

Subroutine GEODAT

SUBROUTINE GEODAT (W, LEVEL, IOUT, IPR, NRECORD, IDATE, NPTS)

This subroutine writes channel cross-section data.

Subprogram(s) Calling This Subroutine:

SUBROUTINE RIVGEO

SUBROUTINE GEOSTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/RIVCRO/	XDIS, YELEV	none

Declarations:

INTEGER W

Input:

Name	Description
W	output file
LEVEL	information level index
IOUT	output type index
IPR	data processing type index
IDATE	date of a data element
NPTS	number of points in a data set
XDIS	array containing transversal distances from a reference line
YELEV	array containing cross-section elevations

Output:

Name	Description
NRECORD	total number of records retrieved

Operation(s):

1. Write channel cross-section data on output file W.

Subroutine RIVSTR

SUBROUTINE RIVSTR

This subroutine is used to retrieve river control structure data.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE CHKEY

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	not currently used here	
/ELEMT/	LEVEL, DF	none
/LOCATE/	not currently used here	
/TIME/	not currently used here	
/WORK/	W, working array IY	IRETRIV, STNA, STNO, DIST, LATI, LONG, SNAME
/OUTPUT/	IOUT	YEAR NPTS, NRECORD
/RIVSTR/		STAGE, AREA
/SUB/	NKEYS	KEY

Declarations:

INTEGER W, DF, STNA, STNO, CTYP

Input:

Name	Description
W	output file
DF	data file
LEVEL	information level index

Name	Description
IOUT	output type index
NKEYS	total number of retrieval keys

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
STNA	array containing station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
SNAME	array containing complete station name for display
YEAR	year of the data set
NPTS	number of points in a data set
NRECORD	total number of records retrieved
STAGE	array containing stage values
AREA	array containing flow areas
KEY	retrieval key

Operation(s):

1. Read control structure data from file YAZSTRS through tape 18 (DF=18).

2. Check the correctness of the data file by calling subroutine CATEG.

3. Check for the desired location by calling subroutine CHKEY.

4. Write headings and the retrieved information on output file W.

Subroutine RIVRES

SUBROUTINE RIVRES

This subroutine is used to retrieve reservoir information.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE CHKEY

SUBROUTINE DWRITE

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	working array JTEMP	none
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	not currently used here	
/TIME/	not currently used here	
/WORK/	W, WF1 working array IY	IRETRIV, NODE, DIST, LATI, LONG, NAME
/OUTPUT/	IOUT	X, Y, MTIT, XTIT, YTIT, YEAR, NPTS, NYEAR, DATE, FDATE, NRECORD
/PLOT/	none	XUNIT, YUNIT, TUNIT
/RIVRES/	none	GAHT, NRCP, SPIQ, NSCP, QVOL, NCCP, NXG, NXS, NXQ
/SUB/	NKEYS	KEY

Declarations:

```
INTEGER W, WF1, DF, YEAR, DATE, SUBLEVL, XTIT, YTIT
REAL MWSL, MXEL
```

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
IOUT	output type index
NKEYS	total number of retrieval keys

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
NODE	river node
DIST	river-mile
LATI	station latitude
LONG	station longitude
NAME	array containing reservoir name
X	array containing values of either dates, SPIQ, or QVOL
Y	array containing gate height values
MTIT	array containing title of the graph
XTIT	title to be put on X-axis
YTIT	title to be put on Y-axis
YEAR	year of the data set
NPTS	number of points in the data set
NYEAR	total number of years retrieved

Name	Description
DATE	array containing retrieved dates
FDATE	array containing floating values of retrieved dates
NRECORD	total number of records retrieved
XUNIT	unit to be printed on X-axis
YUNIT	unit to be printed on Y-axis
TUNIT	time-unit to be printed below graph title
GAHT	array containing gate height values
NRCP	number of data points on reservoir rule curve
SPIQ	array containing spillway discharge values (in CFS)
NSCP	number of data points on reservoir spillway curve
QVOL	array containing reservoir water volume, in cubic feet
NCCP	number of data points on reservoir capacity curve
NXG	working array for gate height data
NXS	working array for spillway discharge data
NXQ	working array for reservoir water volume data
KEY	retrieval key

Operation(s):

1. Read reservoir data from file YAZSTRS through tape 18 (DF=18).
2. Check the correctness of the data file by calling subroutine CATEG.
3. Check for the desired location by calling subroutine CHKEY.
4. Write headings for displaying reservoir data.
5. Read and write "rule curve" data if SUBLEVL = RULE CURVE.
6. Read and write "spillway curve" data if SUBLEVL = SPILLWAY CURVE.

7. Read and write "capacity curve" data if SUBLEVL = CAPACITY CURVE.

8. Write the retrieved information on file WF1 by calling subroutine DWRITE.

Subroutine STADISC

SUBROUTINE STADISC

This subroutine manages river stage/discharge data.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE WSTD

SUBROUTINE RSTD

SUBROUTINE STDVAL

SUBROUTINE DWRITE

SUBROUTINE WSTDS

SUBROUTINE STDSTA

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	LFLAG	none
/TIME/	not currently used here	
/WORK/	W, WF1, NAME working array IY	IRETRIV, STNA, STNO, DIST, LATI, LONG, GAGO, SNAME, NYRS, TYPE
/OUTPUT/	IOUT, IPR	NPTS, NRECORD
/PLOT/	not currently used here	
/STADIS/	not currently used here	
/SUB/	not currently used here	

Declarations:

```
INTEGER W, WF1, DF, STNA, STNO, TYPE, SNAME, SUBLEVL
REAL MOAVG, MOMIN, MOMAX
```

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
LFLAG	location type index
NAME	array containing river name
IOUT	output type index
IPR	data processing type index

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
STNA	array containing station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
GAGO	gage -zero elevation (in FT.)
SNAME	array containing complete station name for display
NYRS	number of years having records
TYPE	data type

Name	Description
NPTS	number of points in a data set
NRECORD	total number of records retrieved

Operation(s):

1. Check data category index. If ICATEG = 8 (discharge), read discharge data from file YAZDISC through tape 14 (DF=14). If ICATEG = 9 (river stage), read river stage data from file YAZSTAG through tape 13 (DF=13).

2. Check the correctness of the data file by calling subroutine CATEG.

3. If data for a particular location is desired, call subroutine STDSTA. Otherwise, write headings for information display by calling subroutine WSTD.

4. Continue reading stage/discharge data from file DF by calling subroutine RSTD.

5. Process the information to get monthly mean, minimum and maximum values by calling subroutine STDVAL.

6. Write the retrieved information on file WF1 by calling subroutine DWRITE.

7. Display stage/discharge information with summary statistics on output file W by calling subroutine WSTDS.

Subroutine STDSTA

SUBROUTINE STDSTA

This subroutine retrieves river stage/discharge data for one particular gaging station.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STADISC

Subprogram(s) Called by This Subroutine:

SUBROUTINE CHKEY

SUBROUTINE WSTD

SUBROUTINE RSTD

SUBROUTINE STDVAL

SUBROUTINE DWRITE

SUBROUTINE WSTDS

SUBROUTINE DYEAR

SUBROUTINE RANGE

SUBROUTINE LEAPYR

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	not currently used here	
/TIME/	JFLAG, JSTART, JEND, ND1 JYR1, ND2, JYR2	none
/WORK/	W, WF1, NAME working variable NX working arrays IY, ITEMP	STNA, STNO, DIST, LATI, LONG, GAGO, SNAME, NYRS, TYPE, LEAP
/OUTPUT/	IOUT, IPR	X, Y, XTIT, YTIT, YEAR NPTS, NYEAR, NRECORD
/PLOT/	MØ	XUNIT, YUNIT, TUNIT

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	none	MOAVG, MOMIN, MOMAX, DVALUE, NDAY, FVALUE
/SUB/	NKEYS	KEY

Declarations:

```
INTEGER W, WF1, DF, STNA, STNO, TYPE, YEAR, SNAME, SUBLEVL, XTIT,
        YTIT, XUNIT, YUNIT, TUNIT
REAL MOAVG, MOMIN, MOMAX
```

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
JFLAG	time-period type index
JSTART	starting time desired
JEND	ending time desired
ND1	order of a starting date in a given year
JYR1	corresponding year for a given starting date
ND2	order of an ending date in a given year
JYR2	corresponding year for a given ending date
NAME	array containing river name
IOUT	output type index
IPR	data processing type index

Name	Description
MØ	array containing months of the year
NKEYS	total number of retrieval keys

Output:

Name	Description
STNA	array containing station name
STNO	station number
DIST	river-mile
LATI	station latitude
LONG	station longitude
GAGO	gage -zero elevation (in FT.)
SNAME	array containing complete station name for display
NYRS	number of years having records
TYPE	data type
LEAP	leap year index
X	array containing stage/discharge data
Y	array containing stage/discharge data
XTIT	title to be printed on X-axis
YTIT	title to be printed on Y-axis
YEAR	year of the data set
NPTS	number of points in a data set
NYEAR	total number of years retrieved
NRECORD	total number of records retrieved
XUNIT	unit to be printed on X-axis
YUNIT	unit to be printed on Y-axis
TUNIT	time-unit to be printed below the graph title

Name	Description
MØAVG	array containing monthly average stage/discharge value
MØMIN	array containing monthly minimum stage/discharge value
MØMAX	array containing monthly maximum stage/discharge value
DVALUE	array containing daily values in a year
NDAY	array containing numbers of days in a month for a year
FVALUE	array containing daily values in a month
KEY	retrieval key

Operation(s):

1. Read stage/discharge data from file DF.
2. Check for the desired location by calling subroutine CHKEY.
3. Write headings for displaying stage/discharge information by calling subroutine WSTD.
4. Continue reading stage/discharge data by calling subroutine RSTD.
5. Begin data retrieval process according to time-period type index.
6. Process the information to get monthly mean, minimum and maximum values by calling subroutine STDVAL.
7. Write the retrieved information on file WF1 by calling subroutine DWRITE.
8. Display stage/discharge information with summary statistics on output file W by calling subroutine WSTDS.

Subroutine RSTD

SUBROUTINE RSTD (DF)

This subroutine reads stage/discharge data.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STADISC

SUBROUTINE STDSTA

Subprogram(s) Called By This Subroutine:

SUBROUTINE LEAPYR

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	NDAY	DVALUE
/WORK/	working variable NX working arrays IY, ITEMP	LEAP
/OUTPUT/	none	X, YEAR, NPTS

Declarations:

INTEGER DF, YEAR

Input:

Name	Description
DF	data file
NDAY	array containing numbers of days in a month for a year

Output:

Name	Description
DVALUE	array containing daily values in a year
LEAP	leap year index
X	array containing stage/discharge values
YEAR	year of the data set
NPTS	number of points in a data set

Operation(s):

1. Check whether the given year is a leap year or not by calling subroutine LEAPYR.

2. Read the whole year of daily stage/discharge values and store in the array DVALUE.

Subroutine STDVAL

SUBROUTINE STDVAL (W, ICATEG, LEVEL, DISTOT, FLOWH, MAXMO, MAXDAY, FLOWL, MINMO, MINDAY, NMON, MSTART)

This subroutine retrieves and processes stage/discharge data.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STADISC

SUBROUTINE STDSTA

Subprogram(s) Called By This Subroutine:

SUBROUTINE RANGE

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	DVALUE, NDAY	MOAVG, MOMIN, MOMAX FVALUE
/PLOT/	MO	none
/OUTPUT/	YEAR	X, NPTS

Declarations:

```
INTEGER W
REAL MOAVG, MOMIN, MOMAX
```

Input:

Name	Description
W	output file
ICATEG	data category index
LEVEL	information level index
NMON	number of months
MSTART	starting month
NDAY	array containing numbers of days in a month for a year
MO	array containing months of the year

Name	Description
YEAR	year of the data set
DVALUE	array containing daily values in a year

Output:

Name	Description
DISTOT	yearly total
FLOWH	maximum discharge/stage in a year
MAXMO	month having maximum discharge/stage
MAXDAY	day having maximum discharge/stage
FLOWL	minimum discharge/stage in a year
MINMO	month having minimum discharge/stage
MINDAY	day having minimum discharge/stage
MOAVG	array containing monthly average discharge/stage
MOMIN	array containing monthly minimum discharge/stage
MOMAX	array containing monthly maximum discharge/stage
FVALUE	array containing daily values in a month
X	array containing stage/discharge values
NPTS	number of points in a data set

Operation(s):

1. Write table headings if display option is desired.
2. Process stage/discharge data to get MINMO, MINDAY, FLOWL, MAXMO, MAXDAY, FLOWH and DISTOT.
3. Call subroutine RANGE to get MOMIN and MOMAX.

Subroutine WSTDS

SUBROUTINE WSTDS (W, ICATEG, IOUT, NPTS, DISTOT, FLOWH, MAXMO, MAXDAY, FLOWL, MINMO, MINDAY)

This subroutine displays stage/discharge information with summary statistics.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STADISC

SUBROUTINE STDSTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	MOAVG, MOMIN, MOMAX, DVALUE	none
/PLOT/	MO	

Declarations:

```
INTEGER W
REAL MOAVG, MOMIN, MOMAX
```

Input:

Name	Description
W	output file
ICATEG	data category index
IOUT	output type index
NPTS	number of points in a data set
DISTOT	yearly total
FLOWH	maximum stage/discharge in a year
MAXMO	month having maximum stage/discharge
MAXDAY	day having maximum stage/discharge
FLOWL	minimum stage/discharge in a year

Name	Description
MINMO	month having minimum stage/discharge
MINDAY	day having minimum stage/discharge
MOAVG	array containing monthly average stage/discharge
MOMIN	array containing monthly minimum stage/discharge
NOMAX	array containing monthly maximum stage/discharge
DVALUE	array containing daily values in a year
MO	array containing months of the year

Output:

Name	Description
none	

Operation(s):

1. Write daily stage/discharge values on output file W, in a table format.
2. Write summary statistics of stage/discharge data for that year.

Subroutine PRECIP

SUBROUTINE PRECIP

This subroutine manages precipitation data.

Subprogram(s) Calling This Subroutine:

PROGRAM GETDATA

SUBROUTINE RIVSYS

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE PREDAT

SUBROUTINE PRESTA

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, LEVEL, DF	none
/LOCATE/	LFLAG	none
/TIME/	not currently used here	
/WORK/	W, NAME working array IY	IRETRIV, STNA, STNO, LATI, LONG, GELEV, NYRS, TYPE, SNAME
/OUTPUT/	none	NRECORD
/PLOT/	MO	none
/STADIS/	none	JVAL
/SUB/	not currently used here	

Declarations:

```
INTEGER W, DF, STNA, STNO, TYPE, SNAME, SUBLEVL
REAL MOAVG, MOMIN, MOMAX
```

Input:

Name	Description
W	output file
DF	data file

Name	Description
ICATEG	data category index
LEVEL	information level index
LFLAG	location type index
NAME	basin name
MO	array containing months of the year

Output:

Name	Description
IRETRIV	index indicating success or failure of the data retrieval process
STNA	array containing station name
STNO	station number
LATI	station latitude
LONG	station longitude
GELEV	station mean elevation (in FT.) above MSL
NYRS	number of years having records
TYPE	data type
SNAME	array containing complete station name for display
NRECORD	total number of records retrieved
JVAL	array containing hourly values for a day

Operation(s):

1. Read precipitation data from file YAZRAIN through tape 17 (DF=17).
2. Check the correctness of the data file by calling subroutine CATEG.
3. If data for a particular station is desired, call subroutine PRESTA.
4. Write headings for displaying precipitation information on output file W by calling subroutine PREDAT.
5. Calculate daily total from hourly data.
6. Write hourly data and daily total value on output file W.
7. Return value of retrieval index (IRETRIV).

Subroutine PRESTA

SUBROUTINE PRESTA

This subroutine retrieves precipitation data for one particular station.

Subprogram(s) Calling This Subroutine:

SUBROUTINE PRECIP

Subprogram(s) Called By This Subroutine:

SUBROUTINE CHKEY

SUBROUTINE PREDAT

SUBROUTINE LEAPYR

SUBROUTINE DORDER

SUBROUTINE DWRITE

SUBROUTINE WHRAIN

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, LEVEL, SUBLEVL, DF	none
/LOCATE/	not currently used here	
/TIME/	JFLAG, JSTART, JEND, ND1, JYR1, ND2, JYR2	none
/WORK/	W, WF1, working variable NX working array IY	STNA, SNAME, STNO, LATI, LONG, TYPE, NYRS, GELEV, LEAP
/OUTPUT/	IOUT	X, MTIT, XTIT, YTIT, YEAR, NPTS, NYEAR NRECORD
/PLOT/	MO	XUNIT, YUNIT, TUNIT
/STADIS/	NDAY	MOAVG, DVALUE, FVALUE, JVAL, HVALUE
/SUB/	NKEYS	KEY

Declarations:

```
INTEGER W, WF1, DF, STNA, STNO, TYPE, YEAR, SNAME, SUBLEVL, GELEV,
        XTIT, YTIT, XUNIT, YUNIT, TUNIT
REAL MOAVG
```

Input:

Name	Description
W	output file
WF1	working file
DF	data file
ICATEG	data category index
LEVEL	information level index
SUBLEVL	information sublevel
JFLAG	time-period type index
JSTART	starting time desired
JEND	ending time desired
ND1	order of the starting date in a given year
JYR1	corresponding year for a given starting date
ND2	order of the ending date in a given year
JYR2	corresponding year for a given ending date
IOUT	output type index
MO	array containing months of the year
NDAY	array containing numbers of days in a month for a year
NKEYS	total number of retrieval keys

Output:

Name	Description
STNA	array containing station name
SNAME	array containing complete station name for display

Name	Description
STNO	station number
LATI	station latitude
LONG	station longitude
TYPE	data type
NYRS	number of years having records
GELEV	station mean elevation (in FT.) above MSL
LEAP	leap year index
X	array containing daily precipitation values
MTIT	array containing title of the graph
XTIT	title to be printed on X-axis
YTIT	title to be printed on Y-axis
YEAR	year of the data set
NPTS	number of points in a data set
NYEAR	total number of years retrieved
NRECORD	total number of records retrieved
XUNIT	unit to be printed on X-axis
YUNIT	unit to be printed on Y-axis
TUNIT	time-unit to be printed below the graph title
MOAVG	array containing monthly totals
DVALUE	array containing daily values in a year
FVALUE	array containing daily values in a month
JVAL	array containing hourly values in a day
HVALUE	array containing hourly values in a year
KEY	retrieval key

Operation(s):

1. Read precipitation data from file DF.

2. Check for the desired location by calling subroutine CHKEY.

3. Write headings for displaying precipitation information on output file W by calling subroutine PREDAT.

4. Check for leap year by calling subroutine LEAPYR.

5. Retrieve precipitation data according to time-period type index.

6. Write the retrieved information on file WF1 by calling subroutine DWRITE.

7. If hourly values are desired for display, call subroutine WHRAIN. Otherwise write daily values and monthly summary on output file W.

Subroutine PREDAT

SUBROUTINE PREDAT

This subroutine writes headings for displaying precipitation information.

Subprogram(s) Calling This Subroutine:

SUBROUTINE PRECIP

SUBROUTINE PRESTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/WORK/	W, SNAME, STNO, LATI, LONG, GELEV, TYPE, NYRS	none

Declarations:

INTEGER W, TYPE, GELEV

Input:

Name	Description
W	output file
SNAME	array containing complete station name for display
STNO	station number
LATI	station latitude
LONG	station longitude
GELEV	station mean elevation (in FT.) above MSL
TYPE	data type
NYRS	number of years having records

Output:

Name	Description
none	

Operation(s):

1. Write headings for displaying precipitation information on output file W.

Subroutine WHRAIN

SUBROUTINE WHRAIN (I, W, IYR, NPTS)

This subroutine writes hourly values and monthly total for a month.

Subprogram(s) Calling This Subroutine:

SUBROUTINE PRESTA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/STADIS/	MOAVG, NDAY, HVALUE	none
/PLOT/	MO	none

Declarations:

INTEGER W

Input:

Name	Description
I	index for month I
W	output file
IYR	year of the data set
NPTS	number of points in the data set
MOAVG	array containing monthly totals
NDAY	array containing numbers of days in a month for a year
HVALUE	array containing hourly values in a year
MO	array containing months of the year

Output:

Name	Description
none	

Operation(s):

1. Write hourly values and monthly total for month I on output file W.

VI. DATA PROCESSING

The task of data processing operations is to extract useful information from the raw data which are stored in the data bank. Thus data retrieval and processing are two closely related operations. The first retrieves the desired data set and prepares it for further manipulation by the second.

Data processing operations are carried out by various processing components and coordinated by the program PROCESS. All these program and subprograms constitute OVERLAY (2,0) of the YAZDB package. A flowchart of the overall structure of this overlay is presented in Figure 6. The following is a comprehensive description of the program PROCESS and related subroutines. Flowcharts of various data processing operations are also given in Appendis B.

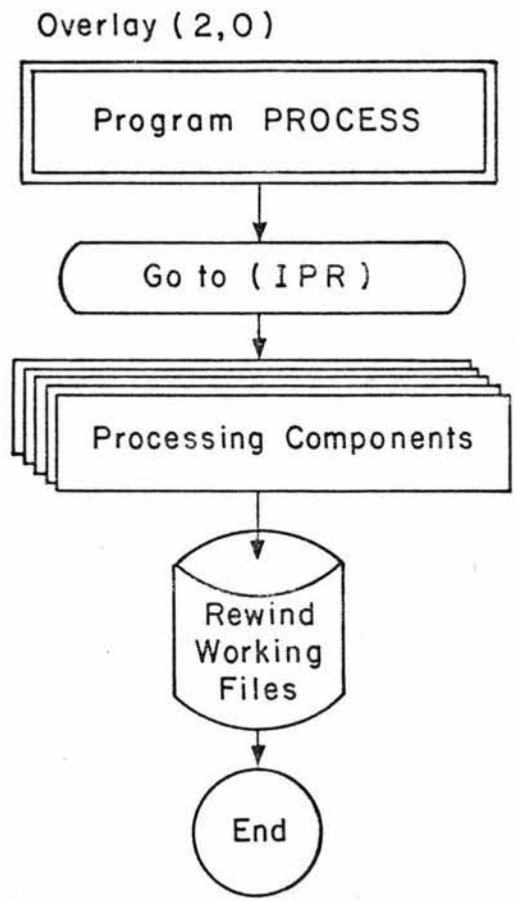


Figure 6. FLOWCHART FOR DATA PROCESSING.

Program PROCESS

PROGRAM PROCESS

This program executes data processing commands and coordinates the operations of related subprograms.

Program(s) Calling This Program;

PROGRAM YAZDB

Subprogram(s) Called By This Program:

SUBROUTINE DREAD

SUBROUTINE RANGE

SUBROUTINE CDFPDF

SUBROUTINE XMIN

SUBROUTINE PINV

SUBROUTINE DWRITE

SUBROUTINE LEAPYR

SUBROUTINE SWITCH

SUBROUTINE PLSQ

SUBROUTINE PEVAL

SUBROUTINE YXMIN

SUBROUTINE YXMAX

SUBROUTINE DCONVRT

SUBROUTINE DSTAT

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, SUBLEVL	none
/INFORM/	not currently used here	
/TIME/	JFLAG	none

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/WORK/	W, WF1, WF2, WF3, DIST GAGO, FIXQ, IYEAR, NCR, LEAP working variables NX,NY, NZ, IZ	none
/OUTPUT/	IOUT, YEAR, NPTS, NYEAR NDATA, IPR	X, Y, MTIT, XTIT, YTIT, CA, CB
/PLOT/	MO	XUNIT, YUNIT, TUNIT
/STADIS/	NDAY	DVALUE
/RIVCRO/	XDIS, YELEV	

Declarations

INTEGER W, WF1, WF2, WF3, YEAR, SUBLEVL, XTIT, YTIT, XUNIT, YUNIT, TUNIT

Input:

Name	Description
W	output file
WF1, WF2, WF3	working files
ICATEG	data category index
SUBLEVL	information sublevel
JFLAG	time-period type index
DIST	river-mile
GAGO	gage -zero elevation (in FT.)
FIXQ	a given discharge (in CFS.)
IYEAR	array containing retrieved years
NCR	number of retrieved cross-sections
LEAP	leap year index
IOUT	output type index
YEAR	year of a data set
NPTS	number of points in a data set

Name	Description
NYEAR	total number of years retrieved
NDATA	total number of data points
IPR	data processing type index
MO	array containing months of the year
NDAY	array containing numbers of days in a month for a year
XDIS	array containing transversal distances from a reference line
YELEV	array containing cross-section elevations

Output:

Name	Description
X	array containing abscissa of points to be plotted
Y	array containing ordinates of points to be plotted
MTIT	array containing title of the graph
XTIT	title to be printed on X-axis
YTIT	title to be printed on Y-axis
XUNIT	unit to be printed on X-axis
YUNIT	unit to be printed on Y-axis
TUNIT	time-unit to be printed below the graph title
CA	arrays containing regression coefficients in a power function curve fitting
CB	arrays containing regression coefficients in a power function curve fitting
DVALUE	array containing daily cumulative rainfall in a year

Operation(s):

1. Activate data processing operations according to the value of IPR index.

2. For frequency analysis, call subroutine DREAD first to get the data from working file WF1, then call subroutines RANGE and CDFPDF to get the desired information.

3. For thalweg level computation, use subroutine XMIN to get the lowest point XLOW in each cross-section, then store DIST and XLOW in arrays X and Y respectively.

4. To investigate the changes in river stage for a given discharge FIXQ over years, a regression analysis of stage-discharge data is carried out first. In this operation, subroutine PLSQ is called to get the coefficients A1 and B1 of a power function giving the best fit to the given data set of each year. Values of A1 and B1 are then stored in the two arrays CA and CB. An inverse operation is then carried out by subroutine PINV to get a value of the stage for a given FIXQ for each year, based on each pair of values of CA and CB. The various values of the stage over the years are stored in the array Y and the corresponding years are saved in the array X.

5. To get cumulative daily rainfall from daily rainfall data, each year of records is read-in first by subroutine DREAD, a summation operation is then carried out, and finally the result is saved in file WF2 by calling subroutine DWRITE. Additional operations are also provided to display the result on output file W.

6. For regression analysis, subroutines DREAD and PLSQ are called successively to read-in stage-discharge data and perform a power function curve fitting operation for each year.

7. To generate discharge hydrograph from stage hydrograph, subroutines DREAD and PEVAL are called to successively read-in the stage data from file WF1 and solve the power function which were obtained previously through a regression analysis operation performed on the corresponding stage-discharge data set. Subroutine DWRITE is then called to save the result on file WF3. An exchange of data files is finally performed to have the result stored on file WF1.

8. To generate stage hydrograph from discharge hydrograph, similar operations as described in (7) are used. But in this case, subroutine PEVAL is replaced by subroutine PINV to solve the inverse problem.

9. To find minimum/maximum value for a given data set, subroutine YXMIN or YXMAX is called and subroutine DCONVRT is used to convert an order in a year into a date before displaying the result on output file W.

10. To calculate the basic statistics of a given data set, subroutine DSTAT is called.

Subroutine CDFPDF

SUBROUTINE CDFPDF (MULT, IPLT, XMAX, XMIN)

This subroutine computes a discrete probability density function and its corresponding cumulative density function for a given sample X of size NDATA. The result will be printed or plotted according to value of IPLT index.

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/WORK/	W, WF1	none
/OUTPUT/	IOUT, MTIT, XTIT, IPR, NPTS, NYEAR, NDATA	YTIT
/PLOT/	LINES	XUNIT, YUNIT, TUNIT

Declarations:

```
INTEGER W, WF1, XTIT, YTIT, XUNIT, YUNIT, TUNIT
DIMENSION PDFX(50), CDFX(50), V(50), OFEQ(50), IXT(8), IYT(8)
```

Input:

Name	Description
W	output file
WF1	working file
IOUT	output type index
IPR	data processing type index
MTIT	array containing title of the graph
XTIT	title to be printed on X-axis
NPTS	number of points in a data set

Name	Description
NYEAR	total number of years used in the analysis
NDATA	total number of data points
LINES	number of lines used to plot by a line printer
MULT	year number
IPLT	plotting index
XMAX	maximum value in the data set
XMIN	minimum value in the data set

Output:

Name	Description
YTIT	title to be printed on Y-axis
XUNIT	unit to be printed on X-axis
YUNIT	unit to be printed on Y-axis
TUNIT	time-unit to be printed below the graph title

Operation(s):

1. Calculate the observed frequency and store the result in array OFEQ.
2. Calculate the discrete cumulative frequency function and the relative frequency histogram, and store the results in arrays CDFX and PDFX respectively.
3. Prepare for plotting by plotting subroutines.

Subroutine PLSQ

SUBROUTINE PLSQ(A1, B1, RC, SBAR)

This subroutine conducts a least-squares curve fitting of power function Y = A1*(X**B1).

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/OUTPUT/	X, Y, NPTS	none

Declarations:

none

Input:

Name	Description
X	array containing values of the independent variable
Y	array containing values of the dependent variable
NPTS	total number of points

Output:

Name	Description
A1, B1	regression coefficients obtained for the power function
RC	correlation coefficient
SBAR	standard error

Operation(s):

1. Check for non-null values of X and Y and take logarithms of all values obtained.

2. Fit a linear equation to the transformed data: get B1 and log (A1). Thus $A1=e^{\log(A1)}$.

3. Calculate the correlation coefficient RC and the standard error SBAR.

Subroutine PEVAL

SUBROUTINE PEVAL (XX, YY, A1, B1, ANEW, BNEW)

This subroutine evaluates YY from XX by the power function

YY = A1*(XX**B1)

and return the coefficients of the inverse function

XX = ANEW*(YY**BNEW).

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

none

Input:

Name	Description
XX	a given value of the independent variable
A1, B1	a set of coefficients of the power function for a given year

Output:

Name	Description
YY	calculated value of the dependent variable
ANEW, BNEW	a set of coefficients of the inverse function

Operation(s):

1. Evaluate YY from XX by the relation: YY = A1*(XX**B1).
2. Calculate ANEW AND BNEW:

BNEW = 1./B1 , ANEW = (1./A1)**BNEW.

Subroutine PINV

SUBROUTINE PINV(XX, YY, A1, B1, ANEW, BNEW)

This subroutine evaluates XX from YY by the inverse of the power function YY = A1*(XX**B1).

Subprogram(s) Calling This Subroutine:

PROGRAM PROCESS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

none

Input:

Name	Description
YY	a given value of the "dependent" variable
A1, B1	a set of coefficients of a given power function

Output:

Name	Description
XX	calculated value of the "independent" variable
ANEW, BNEW	a set of coefficients of the inverse function

Operation(s):

1. Calculate ANEW and BNEW:

   BNEW = 1./B1 , ANEW = (1./A1)**BNEW.

2. Calculate XX from YY by the relation:

   XX = ANEW*(YY**BNEW).

VII. INFORMATION DISPLAY

The Yazoo Data Storage and Retrieval System has options for two types of plotting. One option allows plotting on Tektronix graphics terminals while the second allows for page printer plots. The user can select between two options on retrieval operations. When processing is done on a data management job, only the Tektronix option is available.

The Tektronix plotting option uses the Tektronix 4010A01 Plot-10 Level 1 and the 4010A02 Plot-10 Advanced Graphing II Level 1 plotting packages. These packages were written by Tektronix for their graphics terminals and are maintained on user libraries at the Boeing Computer Center, Seattle, Washington. The library must be linked to the base program if any Tektronix plotting is desired. To obtain graphs with the best possible appearance, the program uses those Plot 10 subroutines that are designed for a Tektronix model 4014-1 terminal with the Enhanced Graphics option installed. Plotting on other Tektronix terminals will still be possible, but graph quality will be somewhat less. A users manual describing the individual subroutines of the plotting package may be obtained from the Boeing Computer Center Consulting Group.

The page plot option uses program internal subroutines, therefore no external libraries are required. The only hardware required for the page plotting is a 130 line printer.

The information display component constitues OVERLAY (3,0) of the YAZDB package. Figure 7 presents a flowchart of the overall structure of this component. Following is a comprehensive description of program DOUTPUT and related subroutines, excluding those already described by the Boeing Computer Manual.

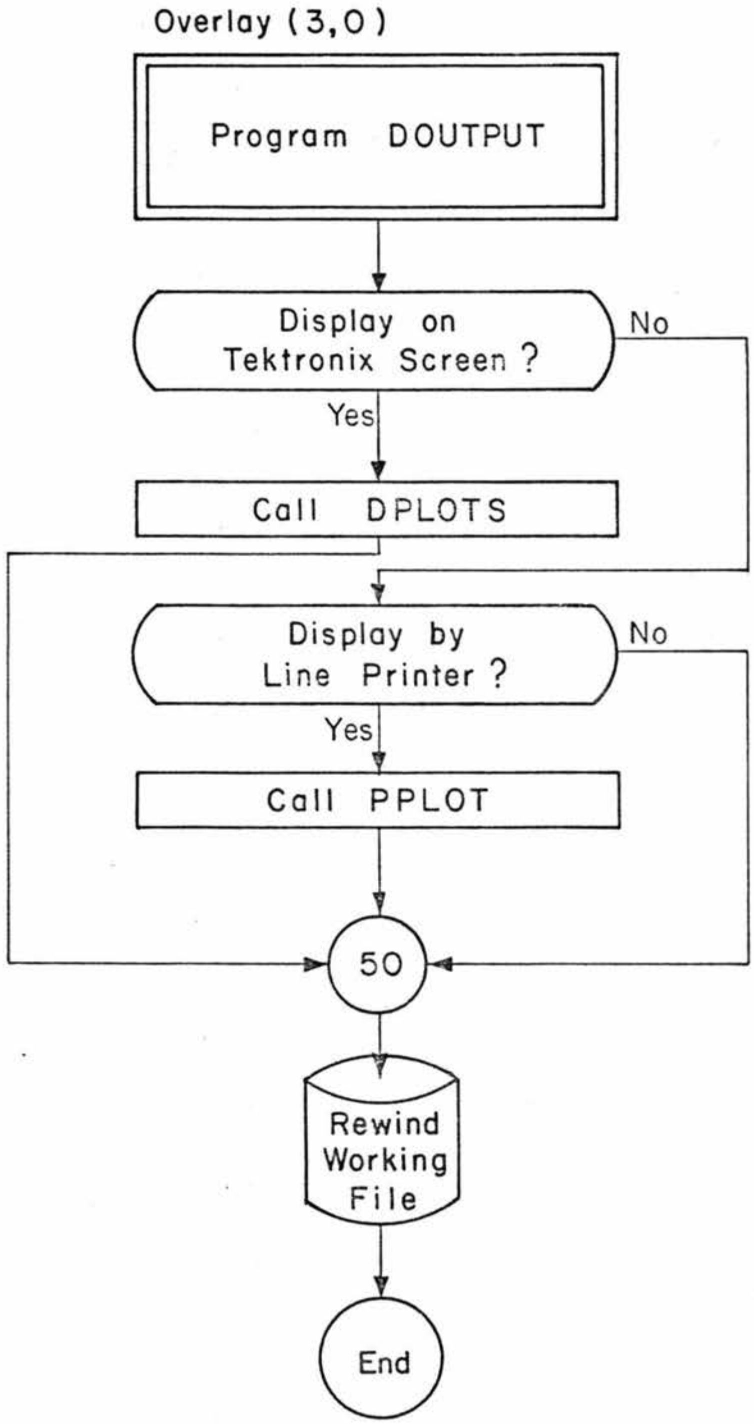


Figure 7. FLOWCHART FOR INFORMATION DISPLAY.

Program DOUTPUT

PROGRAM DOUTPUT

This program activates the operations of two plotting routines: DPLOTS for displaying the result on a Tektronix screen, and PPLOTS for displaying the result by a line printer.

Program(s) Calling This Program:

PROGRAM YAZDB

Subroutine(s) Called By This Program:

SUBROUTINE PPLOTS

SUBROUTINE DPLOTS

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	working variable ICOM	none
/ELEMT/	not currently used here	
/WORK/	W, WF1	none
/OUTPUT/	none	ITEK
/PLOT/	not currently used here	
/RIVCRO/	not currently used here	
/STADIS/	not currently used here	

Declarations:

INTEGER W, WF1

Input:

Name	Description
W	output file
WF1	working file

Output:

Name	Description
ITEK	Tektronix plotting flag (ITEK=YES)

Operation(s):

1. Flag plotting operation to one of the two subroutines: DPLOTS or PPLOTS.

Subroutine PPLOTS

SUBROUTINE PPLOTS

This subroutine displays the information by a line printer.

Subprogram(s) Calling This Subroutine:

PROGRAM DOUTPUT

Subprogram(s) Called By This Subroutine:

SUBROUTINE DREAD

SUBROUTINE RANGE

SUBROUTINE AXISTL

SUBROUTINE SWITCH

SUBROUTINE PLTITL

SUBROUTINE MULPLOT

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	not currently used here	
/ELEMT/	ICATEG, SUBLEVL	none
/WORK/	W, WF1, GAGO	none
/OUTPUT/	X, Y, MTIT, NPTS, NYEAR IPR	none
/PLOT/	not currently used here	
/RIVCRO/	not currently used here	
/STADIS/	not currently used here	

Declarations:

INTEGER W, WF1

Input:

Name	Description
W	output file
WF1	working file

Name	Description
ICATEG	data category index
SUBLEVL	information sublevel
GAGO	gage -zero elevation (in FT.)
X	array containing abscissa of data points to be plotted
Y	array containing ordinates of data points to be plotted
MTIT	array containing title of the graph
NPTS	total number of points to be plotted
NYEAR	total number of years of data to be plotted
IPR	data processing type index

Output:

Name	Description
MOST	maximum number of graphs (limited to 3)
ISYMBOL	symbol used for each graph

Operation(s):

1. Read-in data from file WF1 by calling subroutine DREAD.
2. Call subroutine RANGE to determine lower and upper bounds on the values of X and Y for automatic frame scaling.
3. Call subroutine AXISTL to define units of measure on the axes.
4. Call subroutine PLTITL to write title of the graph.
5. Call subroutine MULPLOT to plot the desired curves.

Subroutine AXISTL

SUBROUTINE AXISTL (ICATEG)

This subroutine defines units of measure on the axes.

Subprogram(s) Calling This Subroutine:

SUBROUTINE PPLOTS

Subprogram(s) Called By This Subroutine:

none

Common Block:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/PLOT/	none	LINES, XUNIT, YUNIT IOPT

Declarations:

none

Input:

Name	Description
ICATEG	data category index

Output:

Name	Description
LINES	number of lines to be plotted by a line printer
XUNIT	unit on the X-axis
YUNIT	unit on the Y-axis
IOPT	plotting option

Operation(s):

1. Define and return the values of the variables LINES, XUNIT, YUNIT, IOPT.

Subroutine PLTITL

SUBROUTINE PLTITL (ICATEG, IPLOT, ISYMBOL)

This subroutine writes title of the graph.

Subprogram(s) Calling This Subroutine:

SUBROUTINE PPLOTS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Commom Block
/WORK/	W	none
/OUTPUT/	IPR, YEAR, IDATE, NPTS	none

Declarations:

INTEGER W, YEAR

Input:

Name	Description
ICATEG	data category index
IPLOT	plot number
ISYMBOL	symbol used for plotting
W	output file
YEAR	year of the data set
IDATE	date of the data set
NPTS	number of points in the data set

Output:

Name	Description
none	

Operation(s):

1. Write title of the graph on output file W.

Subroutine DPLOTS

SUBROUTINE DPLOTS

This subroutine coordinates plotting operations on a Tektronix screen.

Subprogram(s) Calling This Subroutine:

PROGRAM DOUTPUT

Subprogram(s) Called by This Subroutine:

SUBROUTINE DREAD

SUBROUTINE RANGE

SUBROUTINE ERASE

SUBROUTINE BINITT

SUBROUTINE TERM

SUBROUTINE SLIMX

SUBROUTINE SLIMY

SUBROUTINE TOUTPT

SUBROUTINE GETINFO

SUBROUTINE TOASCII

SUBROUTINE CHRSIZ

SUBROUTINE MOVABS

SUBROUTINE HLABEL

SUBROUTINE VLABEL

SUBROUTINE NPTS

SUBROUTINE DLIMX

SUBROUTINE DLIMY

SUBROUTINE CHECK

SUBROUTINE DSPLAY

SUBROUTINE DRWABS

SUBROUTINE CPLOT

SUBROUTINE TINPUT

SUBROUTINE AHMODE

SUBROUTINE TSEND

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/INFORM/	not currently used here	
/ELEMT/	ICATEG, SUBLEVL	none
/WORK/	W, WF1, GAGO	none
/OUTPUT/	X, Y, MTIT, XTIT, YTIT YEAR, NPTS, NYEAR, IDATE, IPR	none
/PLOT/	XUNIT, YUNIT, TUNIT	none
/RIVCRO/	not currently used here	
/STADIS/	not currently used here	

Declarations:

```
INTEGER W, WF1, YEAR, SUBLEVL, XTIT, YTIT, XUNIT, YUNIT, TUNIT
DIMENSION HEADER (10), IOUT (100), IDASH (5)
```

Input:

Name	Description
ICATEG	data category index
SUBLEVL	information sublevel
W	output file
WF1	working file
GAGØ	gage -zero elevation (in FT.)
X	array containing abscissa of data points to be plotted
Y	array containing ordinates of data points to be plotted

Name	Description
MTIT	array containing title of the graph
XTIT	title to be printed on X-axis
YTIT	title to be printed on Y-axis
YEAR	year of the data set
NPTS	total number of points to be plotted
NYEAR	total number of years to be plotted
IDATE	date of the data set
IPR	data processing type index
XUNIT	unit on the X-axis
YUNIT	unit on the Y-axis
TUNIT	time-unit to be printed below the graph title

Output:

Name	Description
none	

Operation(s):

1. Read-in data from file WF1 by calling subroutine DREAD.
2. Call subroutine RANGE to determine lower and upper bounds on the values of X and Y for automatic frame scaling.
3. Read-in headings by calling subroutine GETINFO.
4. Set printer positions and character sizes for plotting horizontal and vertical graph titles.
5. Plot the first graph and write vertical and horizontal title, and headings.
6. Plot subsequent graphs.

Subroutine GETINFO

SUBROUTINE GETINFO (IPR, HEADER)

This subroutine provides headings for Tektronix display.

Subprogram(s) Calling This Subroutine:

SUBROUTINE DPLOTS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	ICATEG, SUBLEVL	none
/WORK/	FIXQ	none

Declarations:

```
INTEGER SUBLEVL
DIMENSION HEADER(3)
```

Input:

Name	Description
IPR	data processing type index
ICATEG	data category index
SUBLEVL	information sublevel
FIXQ	a given discharge value (in CFS.)

Output:

Name	Description
HEADER	array containing headings of the graph

Operation(s):

1. Define headings for various graph plottings and return the results.

Subroutine TOASCII

SUBROUTINE TOASCII (IARY, NCHAR, NEW)

This subroutine provides suitable character code for display on Tektronix screen.

Subprogram(s) Calling This Subroutine:

SUBROUTINE DPLOTS

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
none		

Declarations:

```
INTEGER X
DIMENSION ITABL(52), IARY(10), NEW(NCHAR)
```

Input:

Name	Description
IARY	array of characters to be displayed
NCHAR	number of characters to be converted

Output:

Name	Description
NEW	array of new character codes

Operation(s):

1. Convert titles of the graph to suitable codes for display and return the results.

VIII. DATA UPDATING

Data updating operations consist of the deletion of some existing information elements in a data file, additions of some new data values, or changes in the contents of some data sets. The Cyber 175 computer system operated by the Boeing Company already has a very powerful text editor to help data managers carry out the above operations. However, the development of a data updating component for the quick editing of some specific information elements in the various data files of the Yazoo data base is still needed. This component consists of the program UPDATA and related subroutines, each of which deals with one particular data category. The overall structure of the data updating program package is presented in Figure 8. To start the data updating operations, the procedure file -YAZUP is used. The response of the computer will be as follows:

```
                                        YAZOO DATA UPDATING PACKAGE
TO USE, FOLLOW THE INSTRUCTIONS

KEY-IN THE COMMAND WORD "UPDATE" WITH THE DESIRED DATA CATEGORY TO BE UPDATED
```

The user then keys in the desired data category (i.e., data file name) to be updated and follows the instructions appearing on the Tektronix screen. Following are some examples to illustrate the various features of the program UPDATA.

EXAMPLE 1: Correct a river name in the Descriptive Data File.
(Change LT TALLAHATCHIE RIVER to TALLAHATCHIE RIVER)

(a) Interactive Inputs:

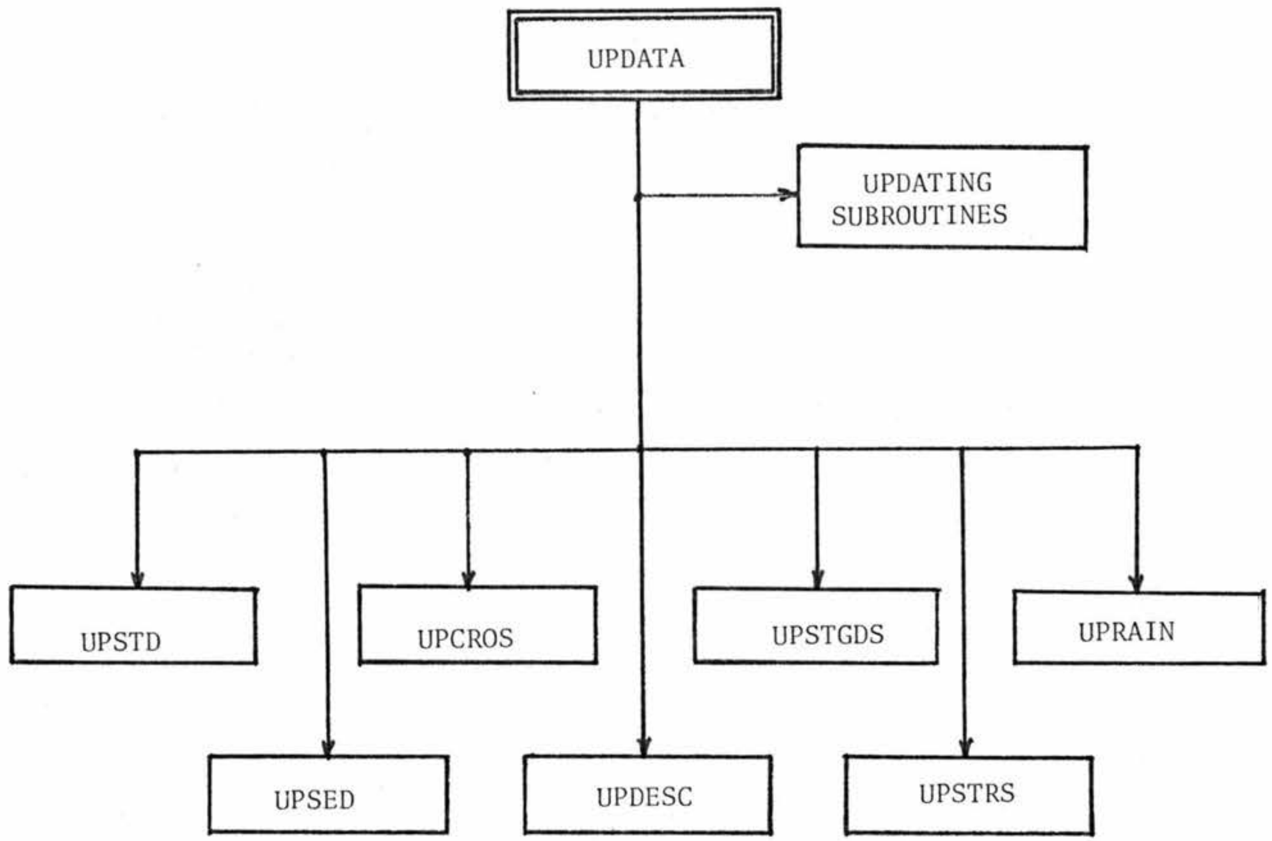


Figure 8. OVERALL STRUCTURE OF THE DATA UPDATING PROGRAM

```
I>UPDATE,RIVER DESCRIPTION

 KEY-IN LOCATION COMMAND

I>LOCATION,RIVER,LT TALLAHATCHIE RIVER
RIVLT TALLAHATCHIE RIVER                                                     102
 EDIT ANY INFORMATION ON THIS CARD IMAGE (0-NO, 1-YES)
I>1
 SELECT ONE OF THE FOLLOWING OPTIONS:
     1--DELETE
     2--ADD
     3--CHANGE
I>3
 ENTER THE INFORMATION ELEMENT NUMBER TO BE UPDATED
     1--RIVER CODE
     2--RIVER NAME
     3--RIVER NODE
     4--ALL
I>2
 ENTER THE DESIRED RIVER NAME (3A10)
I>TALLAHATCHIE RIVER
 DO YOU WISH TO
     1--RE-EDIT
     2--GO ON
     3--STOP
I>3

 DO YOU WISH TO CONTINUE UPDATING THE INFORMATION IN    THIS DATA BANK (0-NO,1-YES)
I>0
      .822 CP SECONDS EXECUTION TIME
C>-UPDESC
```

(b) <u>Result</u>:

```
RIVTALLAHATCHIE RIVER                                                        102
```

EXAMPLE 2: Correct a gaging station name in the Stage-Discharge Data File.(Change AT BELZONI to NR BELZONI by editing the whole location description card).

(a) <u>Interactive Inputs</u>:

```
I>UPDATE,STAGE DISCHARGE

 KEY-IN LOCATION COMMAND

I>LOCATION,STATION LOCATED AT BELZONI
IST4  AT BELZONI                     353  116.1033 10 02 90 29 35  76.023INTER 10

 EDIT ANY INFORMATION ON THIS CARD IMAGE (0-NO, 1-YES)
I>1

 SELECT ONE OF THE FOLLOWING:
     1--DELETE
     2--ADD
     3--CHANGE

I>3
```

```
 THE WHOLE CARD IMAGE (0-NO, 1-YES)
I>1

 ENTER THE CONTENT OF THE CARD IMAGE

I>IST4  NR BELZONI                          353  116.1033 10 02 90 29 35  76.023INTER 10

 DO YOU WISH TO
     1--RE-EDIT
     2--GO ON
     3--STOP
I>3

 DO YOU WISH TO CONTINUE UPDATING THE INFORMATION IN    THIS DATA BANK (0-NO,1-YES)
I>0
     1.017 CP SECONDS EXECUTION TIME
C>-UPSTD
```

(b) Result:

```
IST4  NR BELZONI                          353  116.1033 10 02 90 29 35  76.023INTER 10
```

EXAMPLE 3: Correct a gaging station name in the Stage-Discharge Data File. (Change NR BELZONI to AT BELZONI by replacing only the location name).

(a) Interactive Inputs:

```
I>UPDATE,STAGE DISCHARGE

 KEY-IN LOCATION COMMAND

I>LOCATION,STATION NUMBER 353
IST4  NR BELZONI                          353  116.1033 10 02 90 29 35  76.023INTER 10

 EDIT ANY INFORMATION ON THIS CARD IMAGE (0-NO, 1-YES)
I>1

 SELECT ONE OF THE FOLLOWING:
     1--DELETE
     2--ADD
     3--CHANGE

I>3

 THE WHOLE CARD IMAGE (0-NO, 1-YES)
I>0

 ENTER NUMBER OF INFORMATION ELEMENT TO BE UPDATED
     1--STATION NAME
     2--STATION NUMBER
     3--RIVER MILE
     4--LATITUDE
     5--LONGITUDE
     6--GAGE-ZERO
     7--DATA TYPE
     8--NUMBER OF YEARS HAVING RECORDS

I>1
```

```
 ENTER STATION NAME (3A10)
I>AT BELZONI
IST4  AT BELZONI                     353  116.1033 10 02 90 29 35  76.023INTER 10

 DO YOU WISH TO
     1--RE-EDIT
     2--GO ON
     3--STOP
I>3

 DO YOU WISH TO CONTINUE UPDATING THE INFORMATION IN    THIS DATA BANK (0-NO,1-YES)
I>0
     1.042 CP SECONDS EXECUTION TIME
C>-UPSTD
```

(b) Result:

```
IST4  AT BELZONI                     353  116.1033 10 02 90 29 35  76.023INTER 10
```

EXAMPLE 4: Add a new cross-section to the one specified in the Cross-section Data File.

(a) Interactive Inputs:

```
I>UPDATE,CROSS-SECTION

 KEY-IN LOCATION COMMAND

I>LOCATION,STATION ON YAZOO RIVER
 AT I>3.2

EDIT ANY PERMANENT RANGE INFORMATION (0-NO, 1-YES)
I>0

EDIT ANY CROSS SECTIONS (0-NO, 1-YES)
I>1

SELECT ONE OF THE FOLLOWING:
      1--DELETE
      2--ADD
      3--CHANGE
      4--GO ON
I>2

ENTER DATE OF CROSS SECTION (MM, DD, YY)
I>8,30,62

ADD NEW CROSS SECTION AFTER  8 30 62 (0-NO, 1-YES)
I>1

ENTER DATE:  MO,DAY,YEAR
I>99,99,99

ENTER COMMENT(47 CHAR OR LESS)
I>FICTIOUS CROSS-SECTION

ENTER NUMBER OF POINTS
I>5

ENTER    5 PAIRS OF POINTS (HORIZONTAL, VERTICAL), ONE PAIR PER LINE
```

```
I>200,10
I>220,15
I>225,18
I>232,16
I>250,11

CROSS SECTION ADDED

SELECT ONE OF THE FOLLOWING:
     1--DELETE
     2--ADD
     3--CHANGE
     4--GO ON
I>4

SELECT ONE OF THE FOLLOWING:
     1--NEXT PERMANENT RANGE
     2--NEXT RIVER
     3--EDITING COMPLETED
I>3

 DO YOU WISH TO CONTINUE UPDATING THE INFORMATION IN THIS DATA BANK (0-NO,1-YES)
I>0
      .100 CP SECONDS EXECUTION TIME
C>-UPCROS
```

(b) <u>Result</u>:

```
CS 16    0.00     8 30 62  CORPS OF ENG.X-SECT.PLOTS
PT          -710.  80.0 -690.  84.0 -670.  81.0 -633.  61.0 -580.  43.0 -522.  32.0
PT          -496.  37.0 -480.  35.0 -405.  34.0 -141.  48.0  -85.  56.0  -20.  70.0
PT             0.  71.0   50.  79.0  127.  79.0  270.  86.0
CS  5    0.00    99 99 99  FICTIOUS CROSS-SECTION
PT           200.  10.0  220.  15.0  225.  18.0  232.  16.0  250.  11.0
CS  9    0.00     8 11 72  CORPS OF ENG.X-SECT.PLOTS
PT          -650.  63.0 -620.  58.0 -570.  44.0 -520.  36.0 -475.  34.0 -430.  35.0
PT          -230.  50.0 -137.  55.0  -85.  62.0
```

Following is the detail description of each component subroutine in the program UPDATA.

Program UPDATA

```
PROGRAM UPDATA (INPUT=64, OUTPUT=64, TAPE5=INPUT, TAPE6=OUTPUT, TAPE1=64,
                TAPE2=64, TAPE3=64, TAPE11=64, TAPE12=64, TAPE13=64,
                TAPE14=74, TAPE15=64, TAPE15=64, TAPE17=64, TAPE18=64)
```

This program coordinates the operations of all updating subroutines and subprograms.

Program(s) and Subprogram(s) Called By This Program:

SUBROUTINE SETFLAG

SUBROUTINE UPDESC

SUBROUTINE UPSTD

SUBROUTINE UPSED

SUBROUTINE UPCROS

SUBROUTINE UPSTRS

SUBROUTINE UPSTGDS

SUBROUTINE UPRAIN

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	working variable NX	IN, OUT, WF1, WF2, WF3, DF, ICATEG, NDAY
/WORK/	not currently used here	
/CROS/	not currently used here	

Declarations:

INTEGER OUT, DF, WF1, WF2, WF3, DATE, STNA, STNO, YEAR, XN, ZN, GAGO

Input:

Name	Description
none	

Output:

Name	Description
IN	input file
OUT	output file

Name	Description
WF1,WF2,WF3	working files
DF	data file
ICATEG	data category index
NDAY	array containing numbers of days in a month for a year

Operation(s):

1. Identify data category to be updated by calling subroutine SETFLAG.
2. Activate the desired updating subroutine according to the value of the data categroy index (ICATEG).

Subroutine SETFLAG

SUBROUTINE SETFLAG (IN, OUT, ICATEG, DF)

This subroutine sets flags for data category identification.

Subprogram(s) Calling this Subroutine:

PROGRAM UPDATA

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Command Block
none		

Declarations:

INTEGER DF, OUT

Input:

Name	Description
IN	input file
OUT	output file

Output:

Name	Description
ICATEG	data category index
DF	data file

Operation(s):

1. Read-in UPDATE command.
2. Set values for parameters ICATEG and DF.

Subroutine CATEG

SUBROUTINE CATEG

This subroutine identifies the desired data category.

Subprogram(s) Calling This Subroutine:

SUBROUTINE UPSTD

SUBROUTINE UPSED

SUBROUTINE UPCROS

SUBROUTINE UPSTRS

SUBROUTINE UPSTGDS

SUBROUTINE UPRAIN

SUBROUTINE UPDESC

Subprogram(s) Called By This Subroutine:

none

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	OUT, DF, WF1,ICATEG	INFO
	working variable NX	

Declarations:

INTEGER OUT, DF, WF1

Input:

Name	Description
OUT	output file
DF	data file
WF1	working file
ICATEG	data category index

Output:

Name	Description
INFO	array containing data category name

Operation(s):

1. Identify data category file and return the result.

Subroutine LEAPYR

SUBROUTINE LEAPYR (YEAR, LEAP)

This subroutine identifies wheather a given year is a leap year or not.

Subprogram(s) Calling this Subroutine:

SUBROUTINE UPSTGDS

Subprogram(s) Called by this Subroutine:

None

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
None		

Declarations:

INTEGER YEAR

Input

Name	Description
YEAR	Year to be identified

Output

Name	Description
LEAP	Leap year index (LEAP = 1)

Operation(s)

1. Check whether the given year is divisible by 4. If this is the case, then LEAP = 1.

Subroutine STAHEAD

SUBROUTINE STAHEAT (IDENT, IREAD)

This subroutine updates station information card.

Subprogram(s) Calling This Subroutine:

SUBROUTINE UPSTD

SUBROUTINE UPSED

SUBROUTINE UPSTGDS

Subprogram(s) Called By This Subroutine:

SUBROUTINE UPCARD

SUBROUTINE PTHEAD

Common Blocks:

Block Name	Variables Obrained From Common Block	Variables Placed In Common Block
/ELEMT/	IN, OUT, WF1, ICATEG, ISTN, Working variable NX, Working arrays INFO, LY	None
/WORK/	None	STNA, STNO, IDIST, LAT, LONG, GAGO, ITYPE, NYRS

Declarations:

INTEGER OUT, DF, WF1, STNA, STNO, GAGO, DATE, YEAR

Input:

Name	Description
IN	Input File
OUT	Output File
WF1	Working File
ICATEG	Data Category Index
ISTN	Station Identification Key

Output:

Name	Description
IDENT	Identification Index
IREAD	Card Reading Index
STNA	Array Containing Station Name
IDIST	River Mile
LAT	Station Latitude
LONG	Station Longitude
GAGO	Guage-zero Elevation (In Feet)
ITYPE	Data Type
NYRS	Number of Years having Records

Operations:

1. Decode and identify each element in the station information card
2. Check for updating option desired
3. If updating the whole card is desired, call subroutine UPCARD
4. If updating one information element is desired, call subroutine PTHEAD

Subroutine UPCARD

SUBROUTINE UPCARD (IOP, IN, WF1, IREAD, IY)

This subroutine updates the whole card image.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STAHEAD

SUBROUTINE UPSTD

SUBROUTINE UPSTGDS

Subprogram(s) Called By This Subroutine:

None

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
None		

Declarations:

INTEGER WF1

DIMENSION IY (8)

Input:

Name	Description
IOP	Updating Option Selected
IN	Input File
WF1	Working File

Output:

Name	Description
IREAD	Card Reading index
IY	Array Containing all information in the card image

Operation(s):

1. Read-in all information in a crad image
2. Update the information and write the new card image on file WF1.

Subroutine PTHEAD

SUBROUTINE PTHEAD

This subroutint updates one particular information element in the station information card.

Subprogram(s) Calling This Subroutine:

SUBROUTINE STAHEAD

Subprogram(s) Called By This Subroutine:

None

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	IN, OUT	none
/WORK/	none	STNA, STNO, IDIST, LAT, LONG, GAGO, ITYPE, NYRS

Declarations:

INTEGER OUT, STNA, STNO, GAGO

Input:

Name	Description
IN	Input File
OUT	Output File

Output:

Name	Description
STNA	Array containing station name
STNO	Station Number
IDIST	River Mile
LAT	Station Latitude
LONG	Station Longitude
GAGO	Guage-zero elevation (in feet)

ITYPE	Data type
NYRS	Number of years having records

Operations:

1. Read number of information elements to be updated (IANS)
2. Update the desired information element and return the result.

Subroutine UPSTD

SUBROUTINE UPSTD

This subroutine coordinates the updating operations related to stage-discharge data.

Subprogram(s) Calling This Subroutine:

PROGRAM UPDATA

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE STAHEAD

SUBROUTINE UPCARD

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT /	IN, OUT, DF,WF1,	IYEAR, NDATA, YEAR
	Working Array ITEMP	INFO, DATE, IY, ISTN
/WORK/	none	XN, ZN

Declarations:

INTEGER OUT, DF, WF1. DATE, STNA, STNO, XN, ZN, GAGO, YEAR

Input:

Name	Description
IN	Input File
OUT	Output File
DF	Data File
WF1	Working File

Output:

Name	Description
IYEAR	Year of the data set
NDATA	Number of data points in the set

YEAR	Desired year
INFO	Array containing information to be updated
DATA	Date of the data set
ISTN	Station indentifcation key
IY	Array containing all information in the card image
XN	Array containing updated values of river stage
ZN	Array containing updated values of discharge

Operation(s):

1. Identify data category file by calling subroutine CATEG.
2. Read in the desired station name and year of record to be updated
3. Check the station information card (IST card) by calling subroutine STAHEAD
4. Check the data card (HST card)
5. If updating the whole card image is desired, call subroutine UPCARD

Subroutine UPSTGDS

SUBROUTINE UPSTGDS

This subrotuine coordinates the updating operations related to stage or discharge hydrograph data.

Subprogram(s) Calling This Subroutine:

PROGRAM UPDATA

Subprogram(s) Called By this Subroutine:

SUBROUTINE CATEG

SUBROUTINE STAHEAD

SUBROUTINE LEAPYR

SUBROUTINE UPCARD

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	IN, OUT, DF, WF1,	IYEAR, NDATA,YEAR,INFO,
	Working Array ITEMP	DATE, IY, ISTN
/WORK/	none	XN

Declarations:

INTEGER OUT, DF, WF1, DATE, STNA, STNO, XN, GAGO, YEAR

Input:

Name	Description
IN	Input file
OUT	Output File
DF	Data file
WF1	Working file

Output:

Name	Description
IYEAR	Year of the data set

NDATA	Number of data points in the set
YEAR	Desired year
INFO	Array containing information to be updated
DATE	Date of the data set
IY	Array containing all information in the card image
ISTN	Station identification key
XN	Array containing updated values of river stage or discharge

Operation(s):

1. Identify data category file by calling subroutine CATEG
2. Read-in the desired station anme and year of record to be up dated
3. Check the station information card (DST card for discharge, STA card for stage) by calling subroutine STAHEAD
4. Check the data card (DVL card for discharge, STV card for stage)
5. If updating the whole card image is desired, call subroutine UPCARD.

Subroutine UPCROS

SUBROUTINE UPCROS

This subroutine coordinates the updating operations related to channel corss-section data.

Subprogram(s) Calling This Subroutine:

PROGRAM UPDATA

Subprogram(s) Called By This Subroutine:

SUBROUTINE CATEG

SUBROUTINE PRED

Common Blocks:

Block Name	Variables Obtained From Common Block	Variables Placed In Common Block
/ELEMT/	IN, OUT, DF, WF1.	INFO
/WORK/	Working Arrays XN,ZN X, Z	MO, IDY, IYR
/CROS/	Working Arrays ICOM, JCOM	PR, NPTS, L, NCS, RM

Declarations:

none

Input:

Name	Description
IN	Input file
OUT	Output file
DF	Data file
WF1	Working file

Output:

Name	Description
INFO	Array Containing information to be updated

MO	Array containing months of all corss-section measurements at a location
IDY	Array containing days of all cross-section measurememts at a location
IYR	Array containing years of all cross-section measurements at a location
PR	Array containing river-mile of all cross-sections having the same PR
NPTS	Array containing numbers of data points of these cross-sections
L	Array containing latitude and longitude
NCS	Number of cross-section
RM	River mile

Operation(s):

1. Read-in river name and river-mile of the cross-section to be updated
2. Read-in the old cross-section data set from file DF
3. Find the desired river and PR in the file
4. Determine if user wishes to update PR card information. If yes is the answer, (IANS) , call subroutine PRED to update the PR card
5. Select updating option to edit cross-section data
6. Find the correct measurement date and update the desired data point in that cross-section
7. Update all cross-section information for new change.
8. Store the updated cross-section information in file WF1.

APPENDIX A

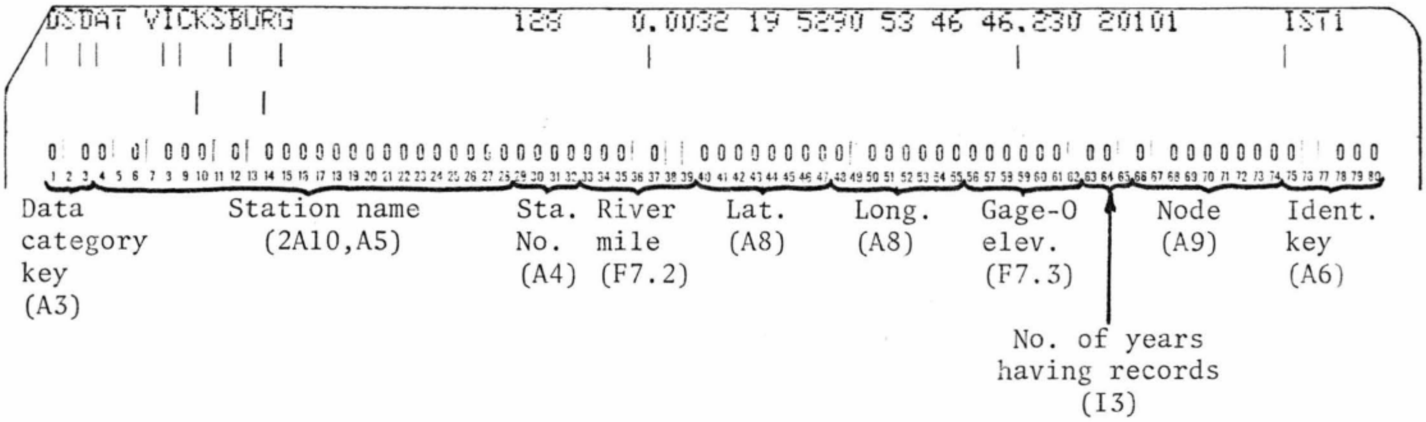
DATA FORMATS

DESCRIPTIVE DATA FILE (DESCRIP)

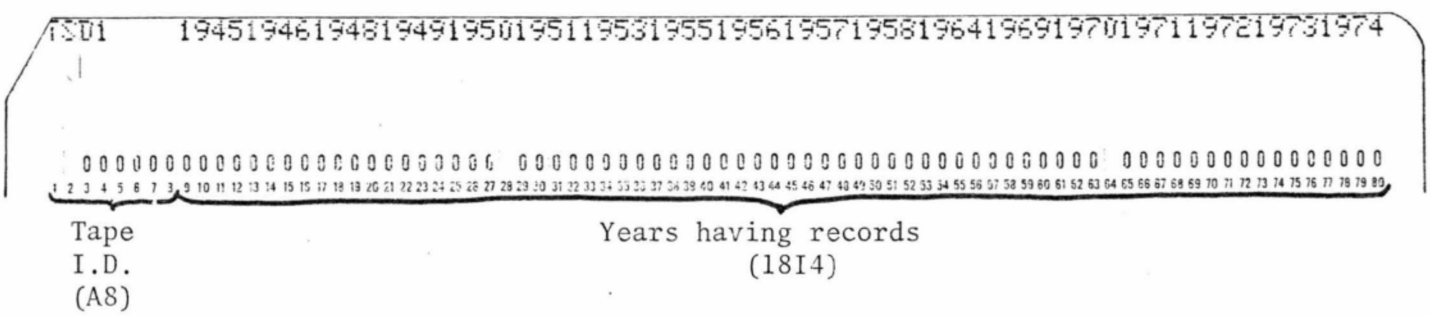
STAGE-DISCHARGE DESCRIPTIVE DATA FILE

EXAMPLE: Gaging station at Vicksburg.

CARD 1: (station description)

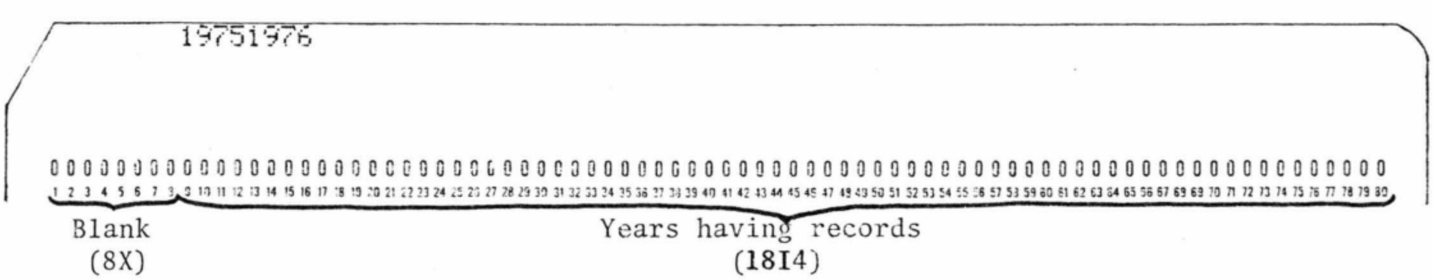


CARD 2: (year identification)



If the third card is required to input the years having records it should have the same format as the second card, except the tape identification key is no longer required, i.e.,

CARD 3:



RIVER STAGE DESCRIPTIVE DATA FILE

EXAMPLE: Gaging station at Belzoni.

CARD 1: (station description)

```
DSGAT BELZONI                 353  116.1033 10 0290 29 35 76.02  11101         STA1
```

Data category key (A3)	Station name (2A10,A5)	Sta. No. (A4)	River mile (F7.2)	Lat. (A8)	Long. (A8)	Gage-0 elev. (F7.3)	No. of years having records (I3)	Node (A9)	Ident. key (A6)

CARD 2: (year identification)

```
TSG1     196419651966196719681969197019711972197319 74
```

Tape I.D. (A8)	Years having records (18I4)

RIVER DISCHARGE DESCRIPTIVE DATA FILE

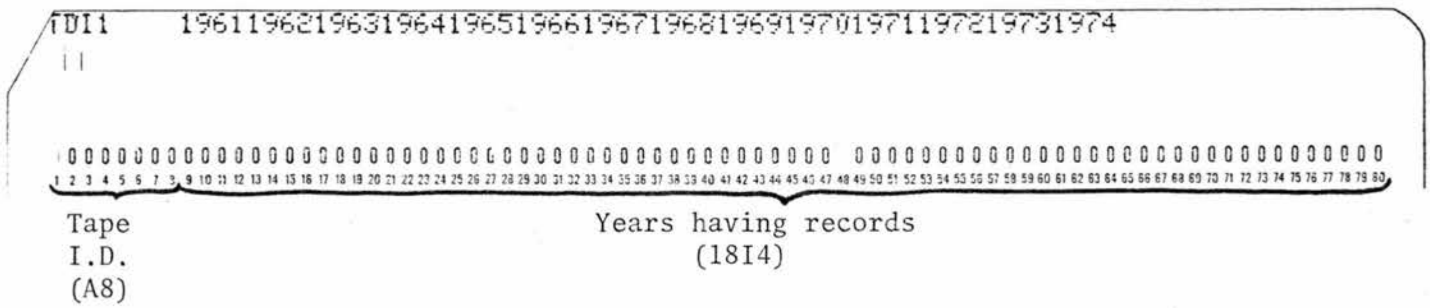
EXAMPLE: Gaging station at Greenwood.

CARD 1: (station description)

```
DDIAT GREENWOOD               129  166.0033 31 1790 11 03 92.067 14101         DST1
```

Data category key (A3)	Station name (2A10,A5)	Sta. No. (A4)	River mile (F7.2)	Lat. (A8)	Long. (A8)	Gage-0 elev. (F7.3)	No. of years having records (I3)	Node (A9)	Ident. key (A6)

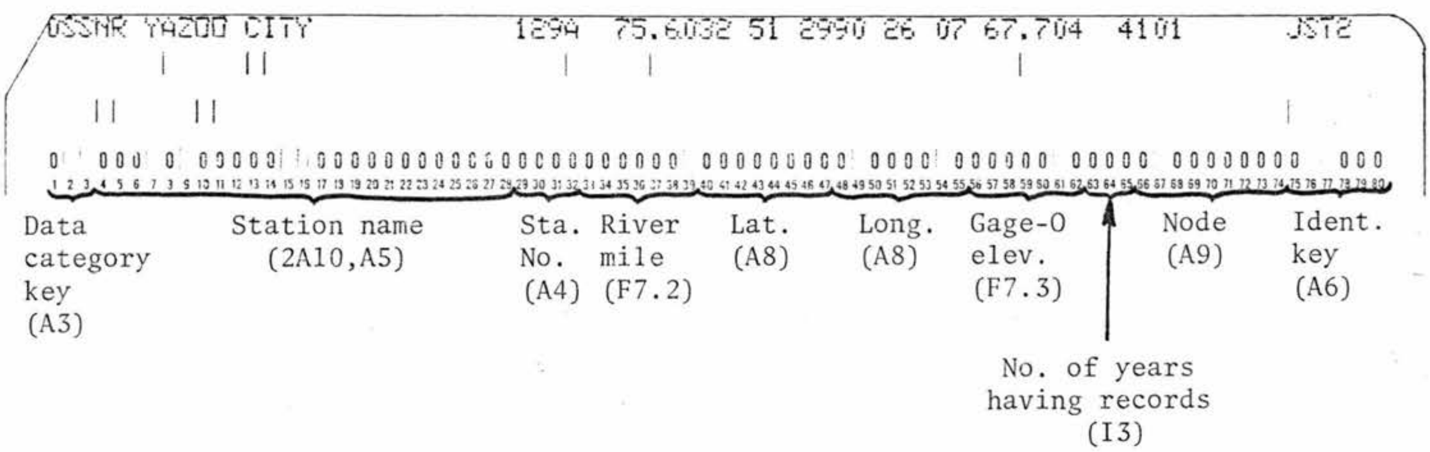
CARD 2: (year identification)



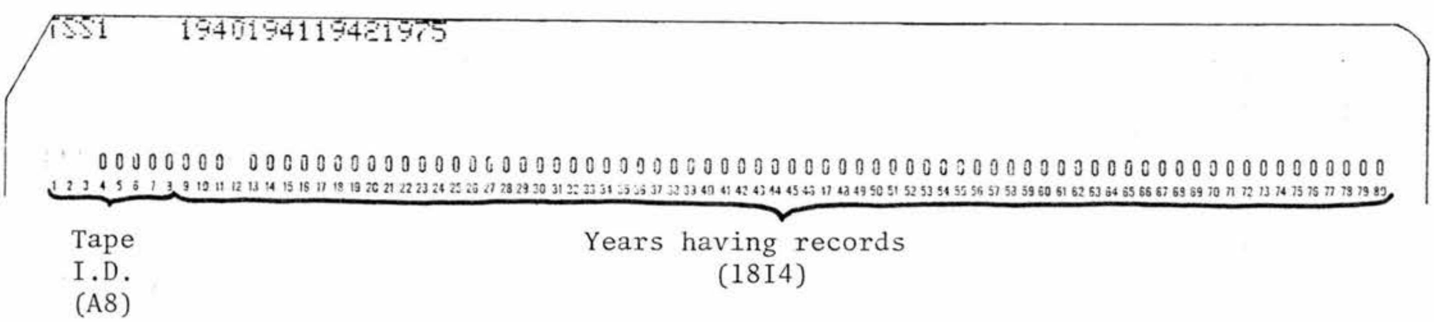
SUSPENDED SEDIMENT DESCRIPTIVE DATA FILE

EXAMPLE: Gaging station near Yazoo City.

CARD 1: (station description)



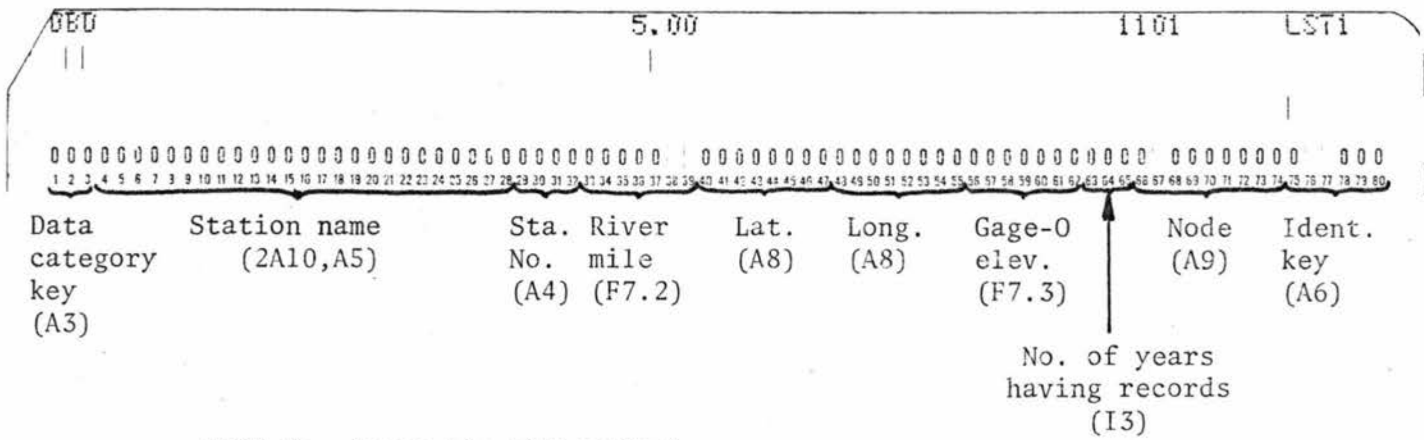
CARD 2: (year identification)



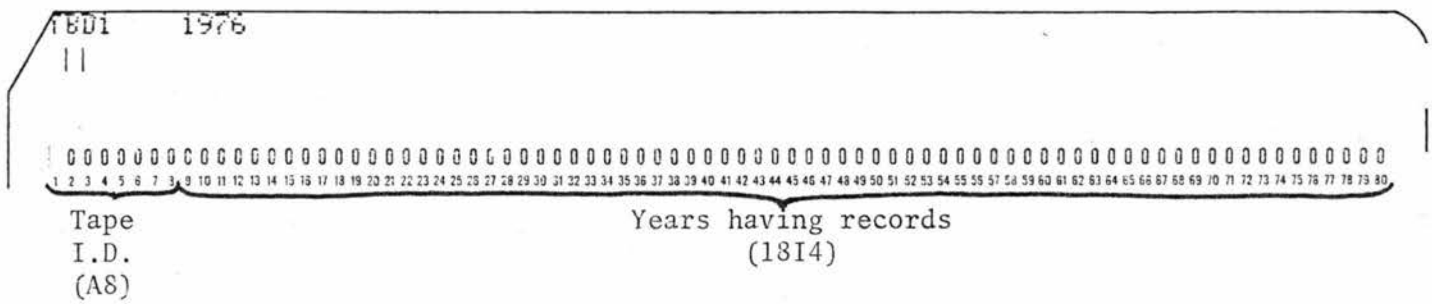
BED MATERIAL DESCRIPTIVE DATA FILE

EXAMPLE: Gaging station at river-mile 5.00 upstream from Vicksburg.

CARD 1:



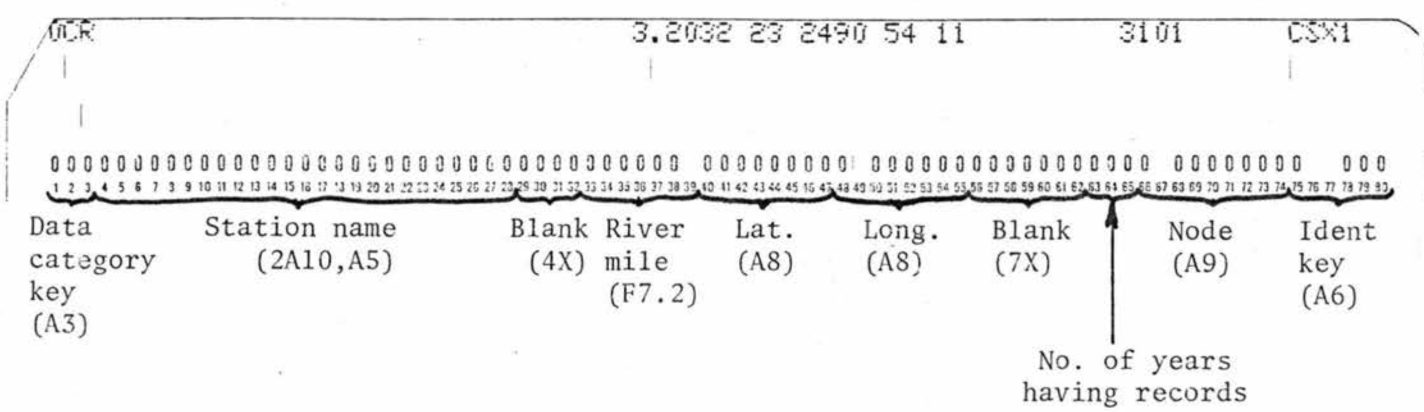
CARD 2: (year identification)



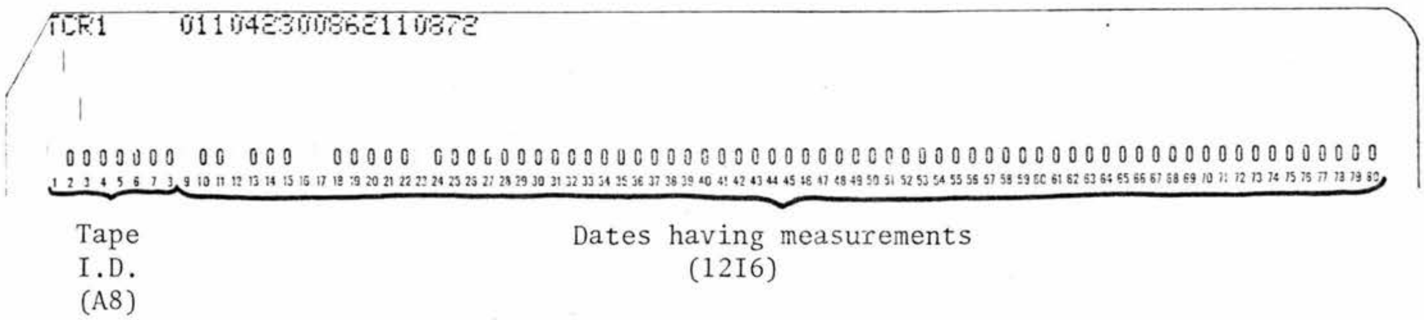
CHANNEL CROSS-SECTION DESCRIPTIVE DATA FILE

EXAMPLE: Gaging station at river mile 3.20 upstream from Vicksburg.

CARD 1:



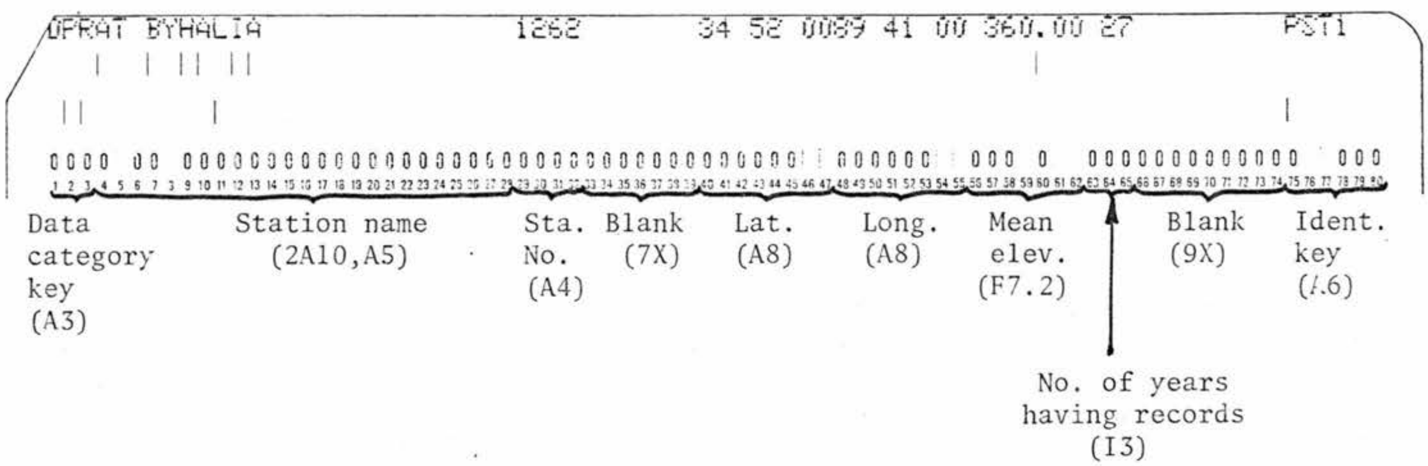
CARD 2: (date identification)



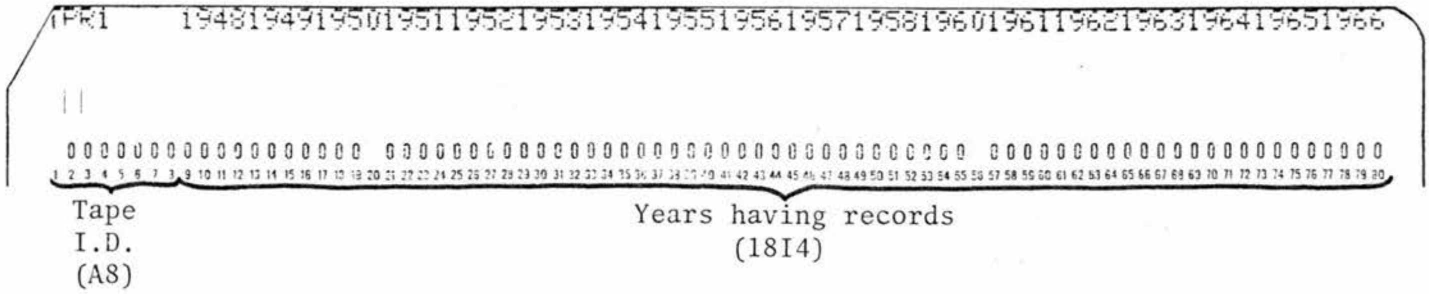
PRECIPITATION DESCRIPTIVE DATA FILE

EXAMPLE: Rainfall gaging station at Byhalia.

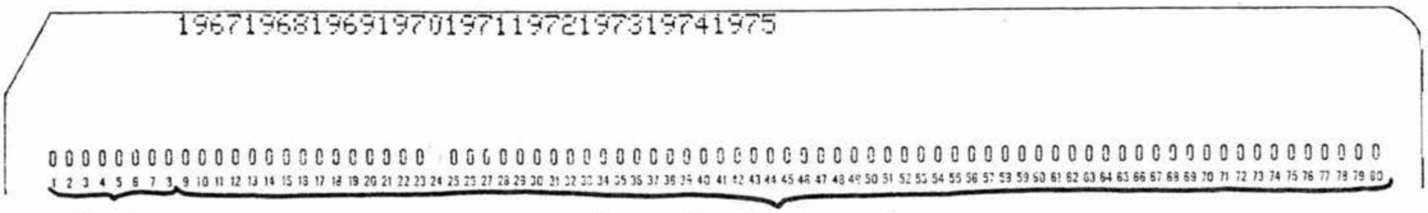
CARD 1: (station description)



CARD 2: (year identification)



CARD 3:



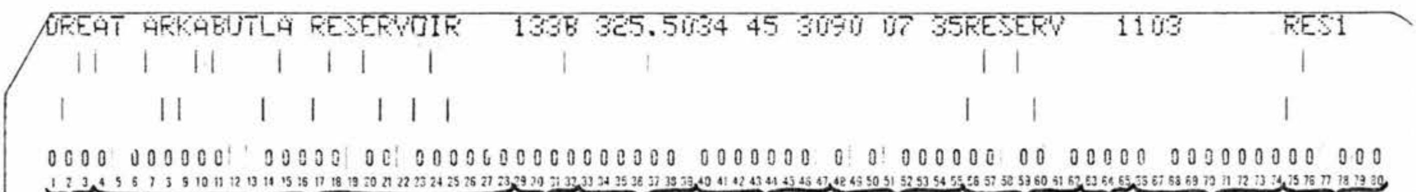
Blank (8X)

Years having records (18I4)

RESERVOIR DESCRIPTIVE DATA FILE

EXAMPLE: Gaging station at Arkabutla reservoir.

CARD 1: (station description)



Data category key (A3)

Station name (2A10,A5)

Sta. No. (A4)

River mile (F7.2)

Lat. (A8)

Long. (A8)

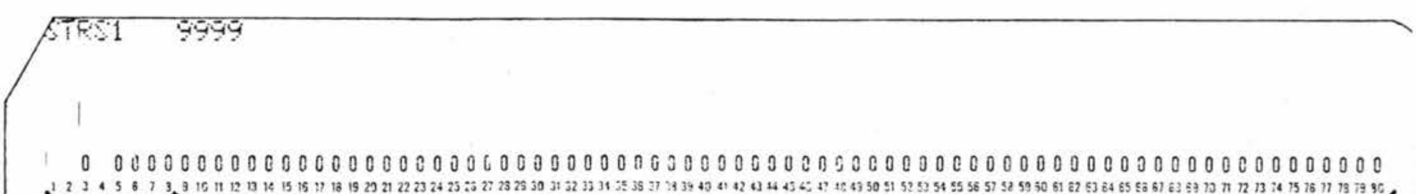
Structure type (A7)

No. of years having records (I3)

Node (A9)

Ident. key (A6)

CARD 2: (year identification)



Tape I.D. (A8)

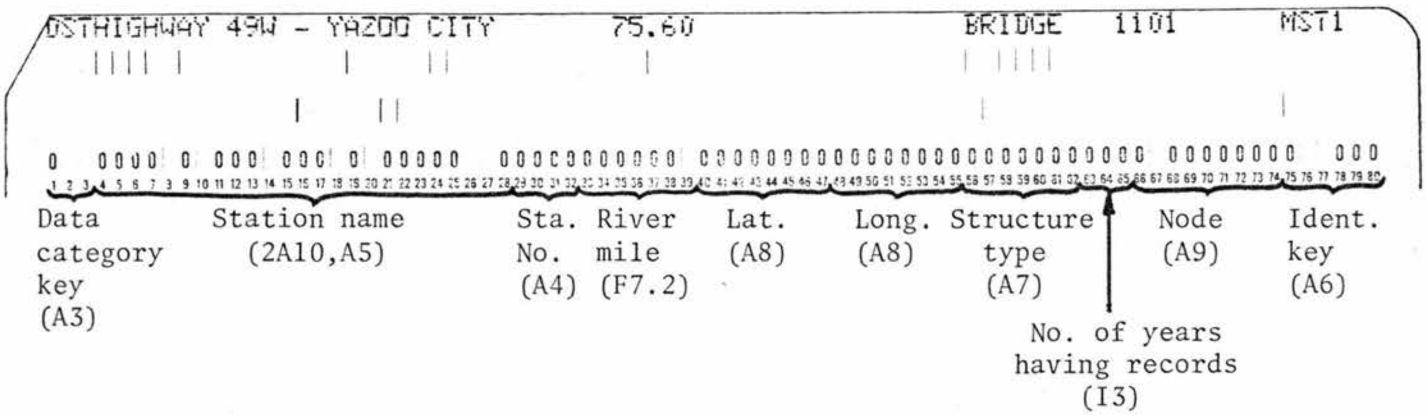
Years having records (18I4)

9999 stands for an unknown year.

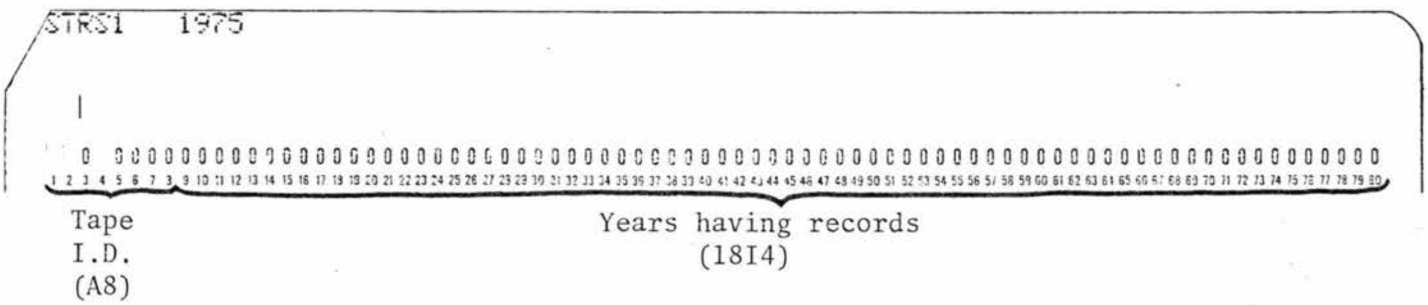
CONTROL STRUCTURE DESCRIPTIVE DATA FILE

EXAMPLE: Bridge located at Yazoo City on highway 49W.

CARD 1: (station description)



CARD 2: (year identification)

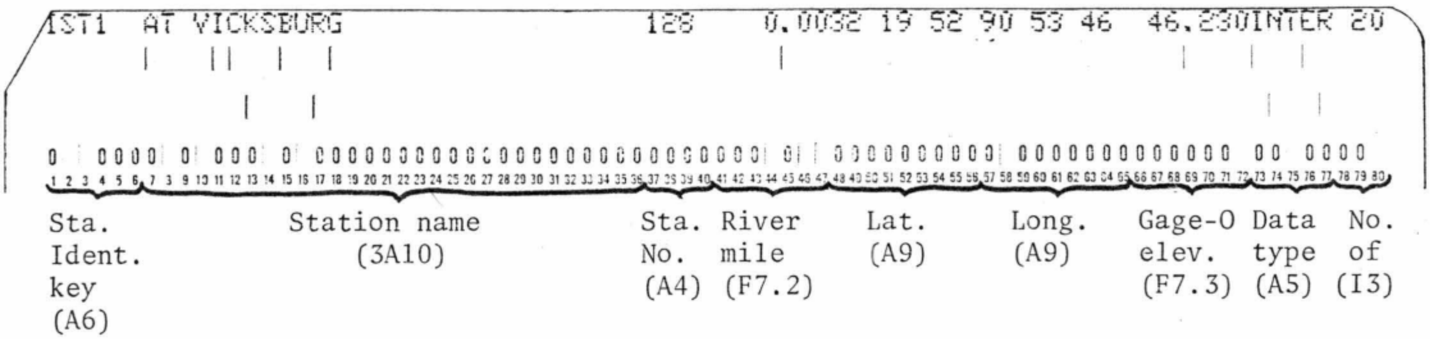


STAGE-DISCHARGE DATA FILE (YAZSTD)

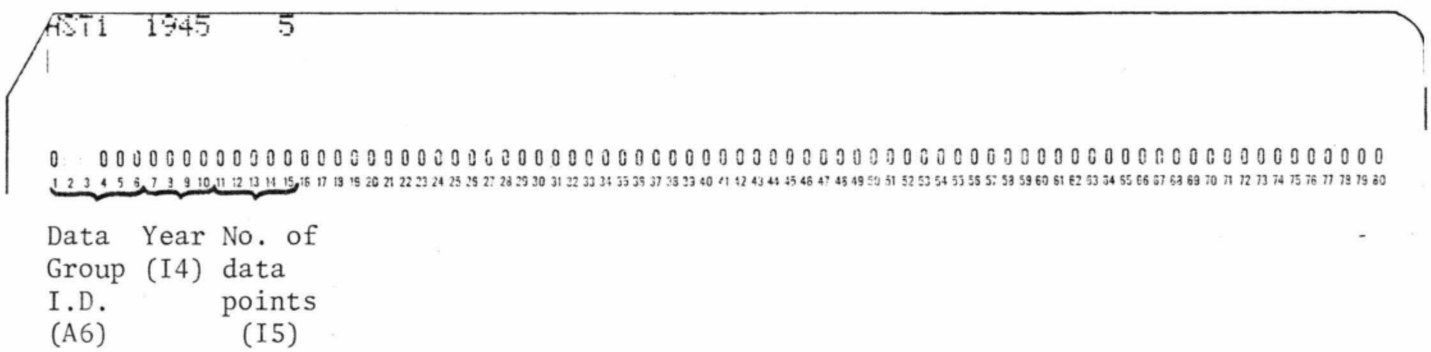
In the stage-discharge data file, each gaging station is headed by a card-image describing the station and followed by groups of data cards. Each group of data cards contains one year of stage-discharge data points and is headed by a year identification card. Each data point consists of three elements: the date (day--month--year), the stage (in feet, above gage-zero elevation), and the corresponding discharge (in CFS).

EXAMPLE: Stage-discharge relation of the Yazoo River at Vicksburg (1945).

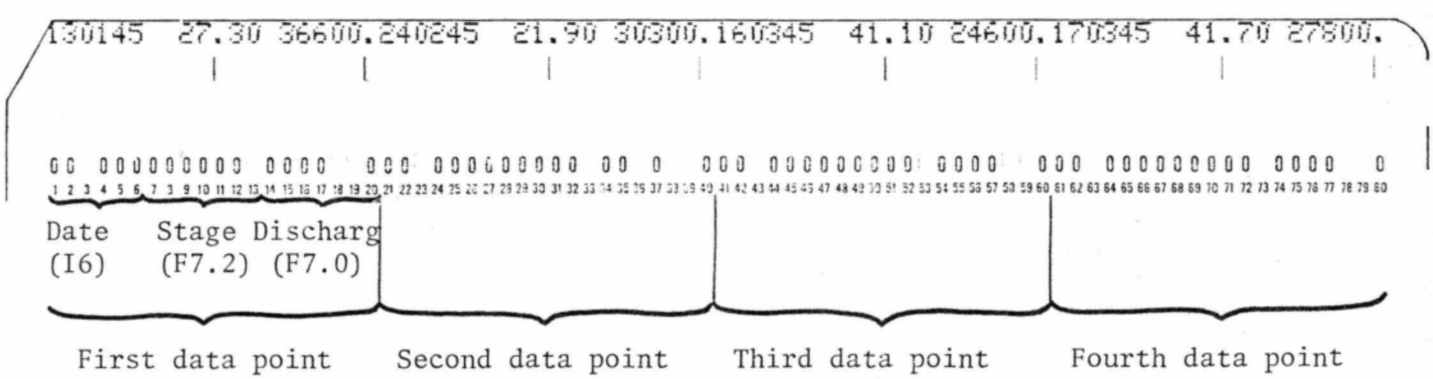
CARD 1: (station description)



CARD 2: (year identification)

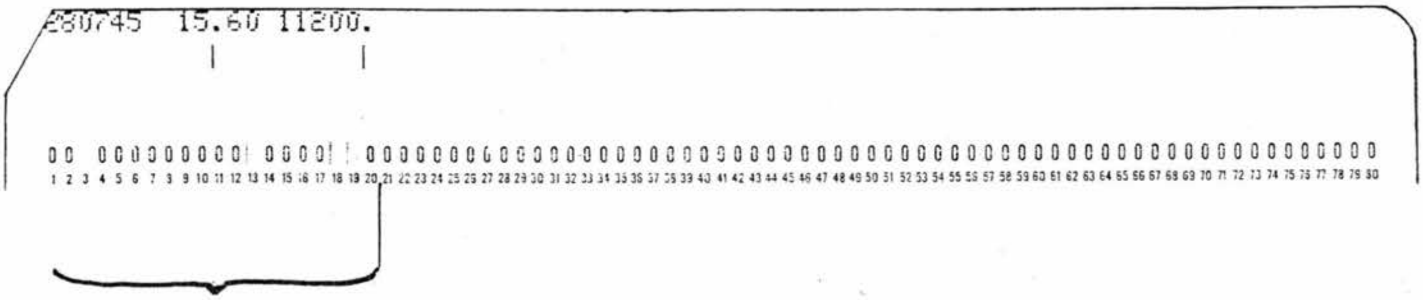


CARD 3: (data card)



If some additional cards are required to input stage-discharge data points, they should have the same format as used for the third card.i.e.,

CARD 4:

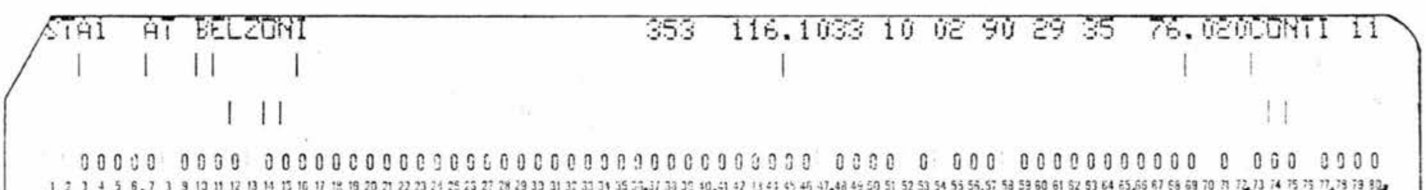


Fifth data point

STAGE-HYDROGRAPH DATA FILE (YAZSTAG)

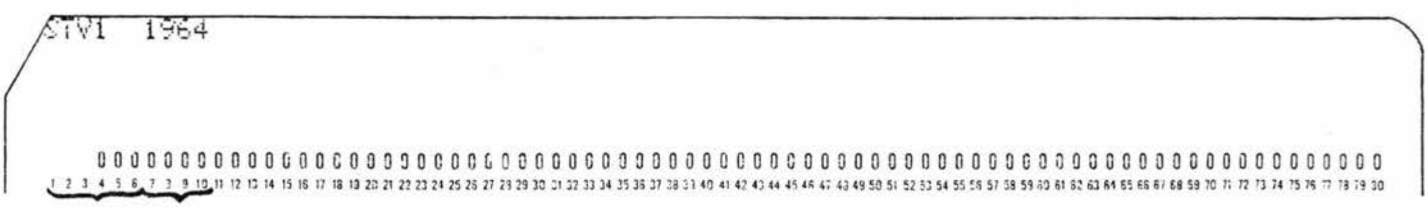
The station description card and the year identification card are similarly coded for the stage-discharge data file, but have different identification keys. The stage data are stored in a calendar year-block of 365 data points (or 366 data points for a leap year).

EXAMPLE: Stage-hydrograph data of the Yazoo River at Belzoni (1964)
CARD 1: (station description)



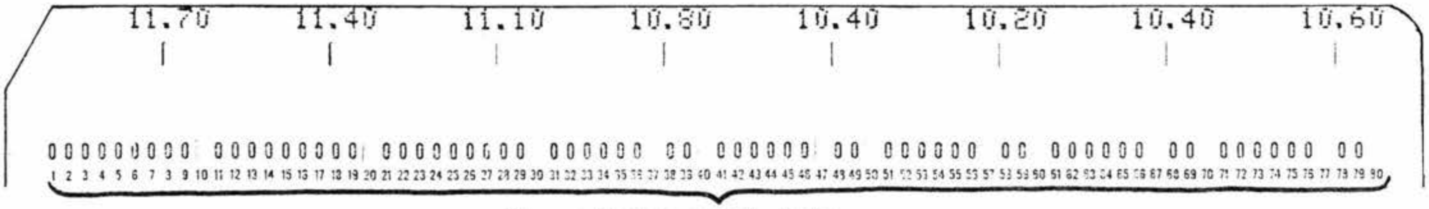
Sta. Ident. key (A6)	Station name (3A10)	Sta. No. (A4)	River mile (F7.2)	Lat. (A9)	Long. (A9)	Gage-0 elev. (F7.3)	Data type (A5)	No. of years (I3)

CARD 2: (year identification)



Data Group I.D. (A6)	Year (I4)

CARD 3: (or subsequent data cards)



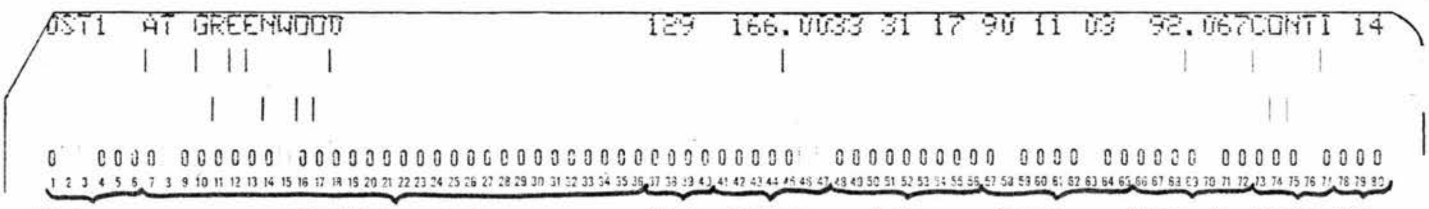
Stage-hydrograph data
(8F10.2)

DISCHARGE-HYDROGRAPH DATA FILE (YAZDISC)

The discharge-hydrograph data file is coded in a similar way to the stage-hydrograph data file but with different identification keys.

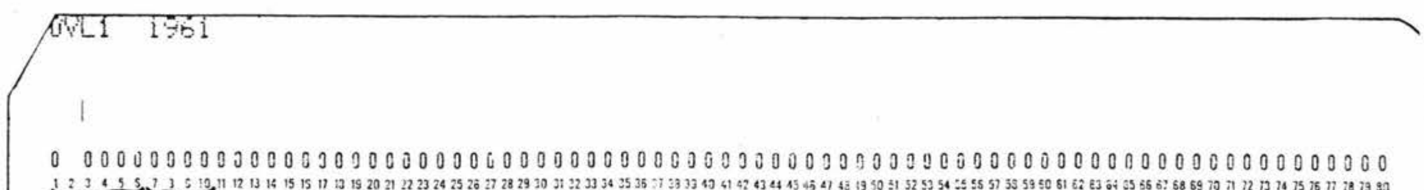
EXAMPLE: Discharge-hydrograph data for the Yazoo River at Greenwood (1961)

CARD 1: (station description)



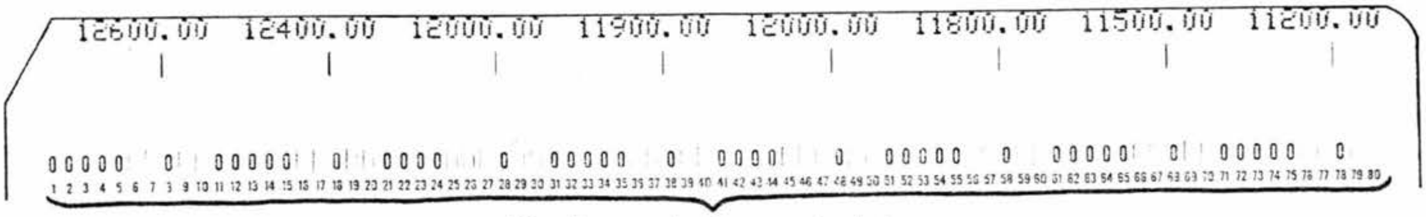
Sta. Ident. key (A6)	Station name (3A10)	Sta. No. (A4)	River mile (F7.2)	Lat. (A9)	Long. (A9)	Gage-0 elev. (F7.3)	Data type (A5)	No. of years (I3)

CARD 2: (year identification)



Data Group I.D. (A6)	year (I4)

CARD 3: (or subsequent data cards)



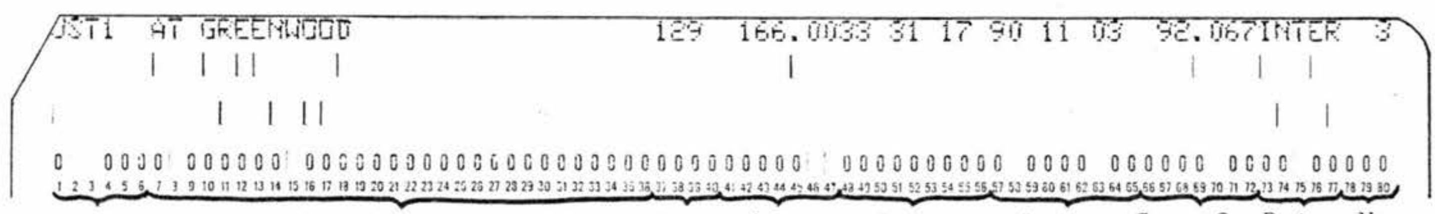
Discharge-hydrograph data
(8F10.2)

SEDIMENT DATA FILE (YAZSED)

The sediment data file contains two types of data: suspended sediment data and bed material data. Each type is stored differently but has the same general structure.

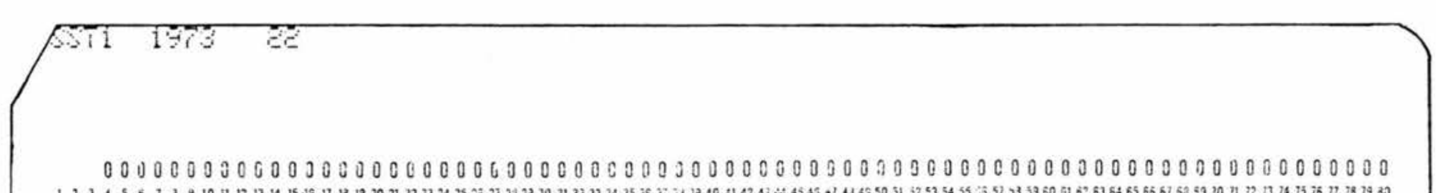
EXAMPLE: Suspended sediment data for the Yazoo River at Greenwood (1973)

CARD 1: (station description)



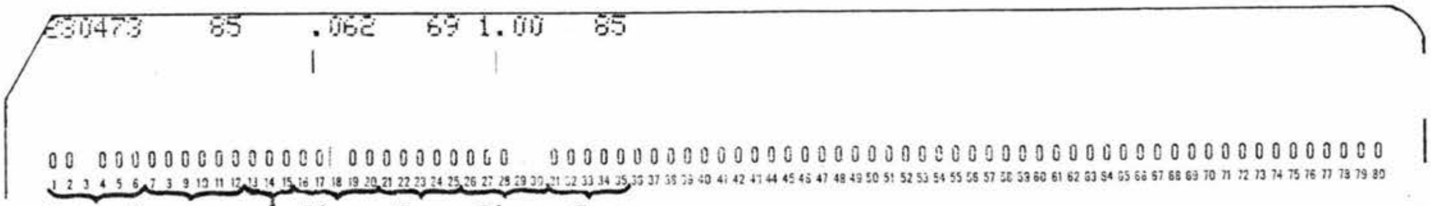
Sta. Ident. key (A6)	Station name (3A10)	Sta. No. (A4)	River mile (F7.2)	Lat. (A9)	Long. (A9)	Gage-0 elev. (F7.3)	Data type (A5)	No. of years (I3)

CARD 2: (year identification)



Data Group I.D. (A6)	Year (I4)	No. of data pts. (I5)

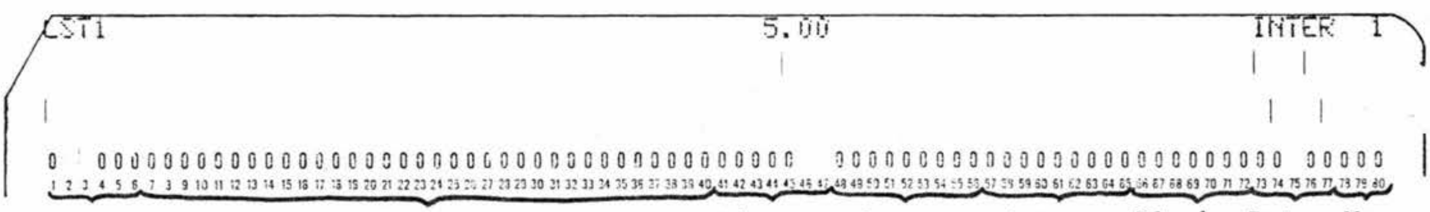
CARD 3: (or subsequent data cards)



Date (I6)	Concentration (I6)	Blank (3X)	Size #1 (F5.0)	Cum concentration (I5)	Size #2 (F5.0)	Cum concentration (I5)

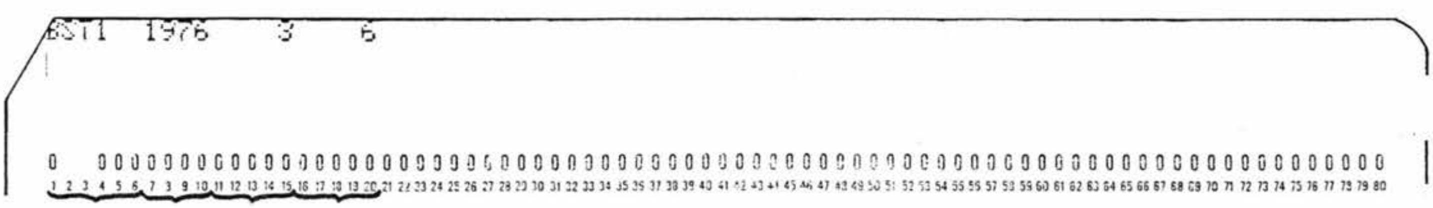
EXAMPLE: Bed material data for the Yazoo River at river mile 5.(1976).

CARD 1: (station description)



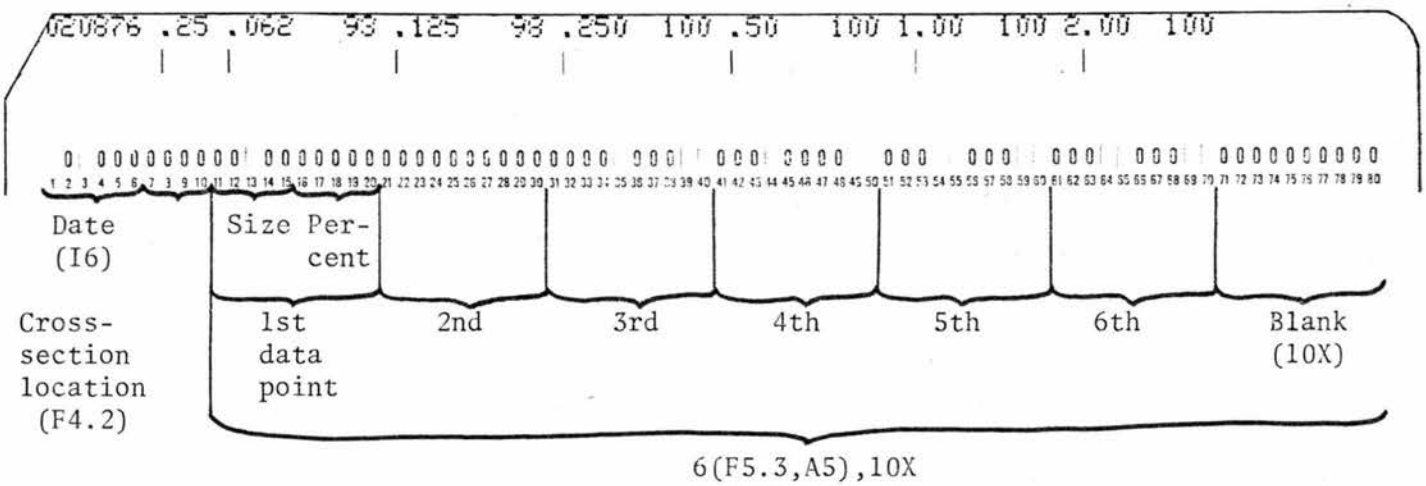
Sta. Ident. key (A6)	Blank (34X)	River mile (F7.2)	Lat. (A9)	Long. (A9)	Blank (7X)	Data type (A5)	No. of years (I3)

CARD 2: (year identification)



Data group I.D. (A6)	year (I4)	No. of data pts. (I5)	No. of particle sizes (I5)

CARD 3: (or subsequent data cards)

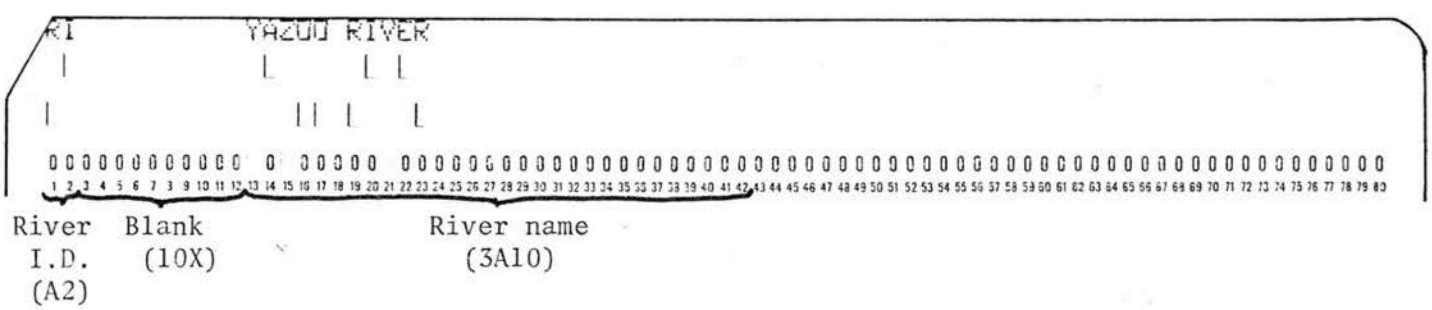


CHANNEL CROSS-SECTION DATA FILE (YAZCROS)

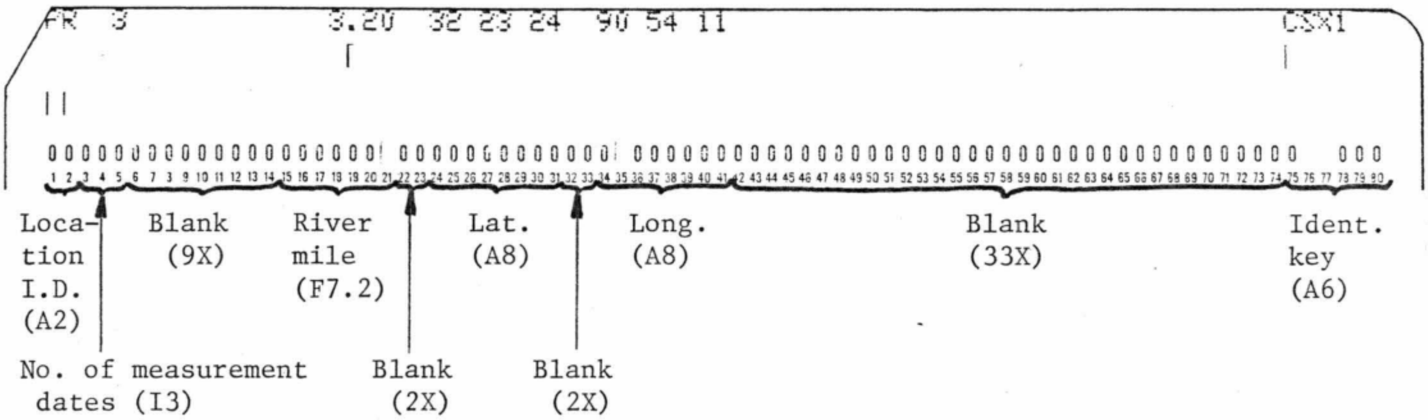
These types of data cards are used to code channel cross-section information. The RI-card is used to read all cross-sectional data of one particular river, the PR-card describes the location of the cross-section, and the CS-card identifies the data card related to the date.

EXAMPLE: Cross-section of the Yazoo River at river mile 3.2 (measured on October 1, 1942).

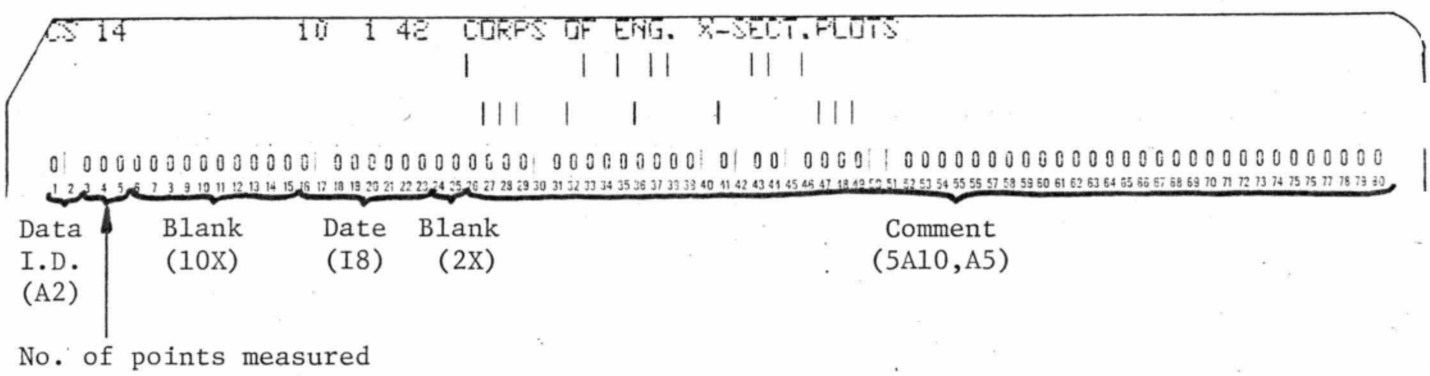
CARD 1: (river description)



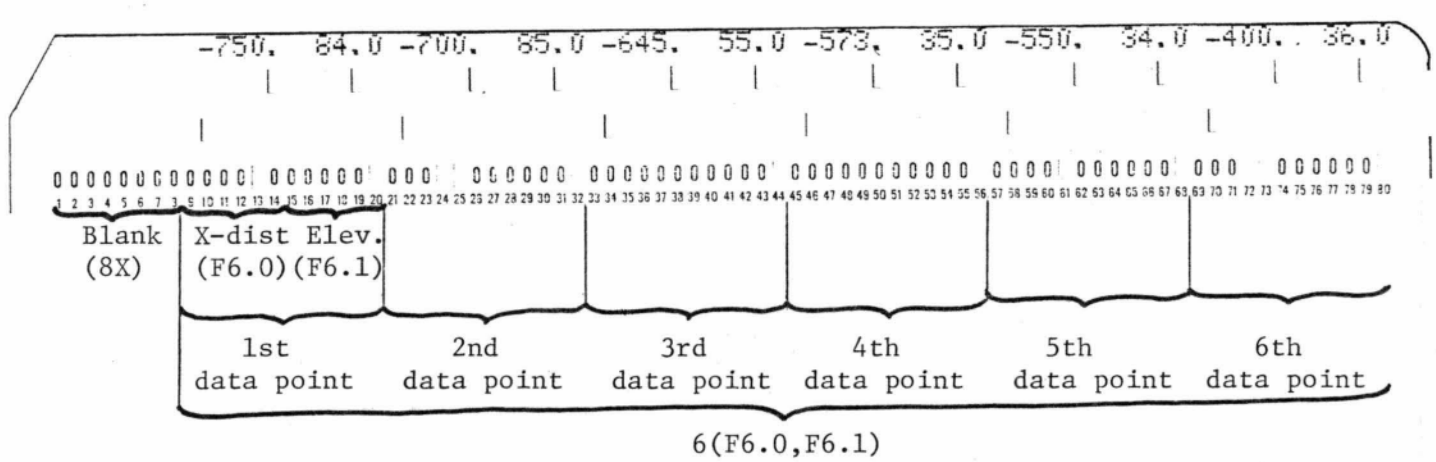
CARD 2: (location description)



CARD 3: (data identification)



CARD 4: (or subsequent data cards)

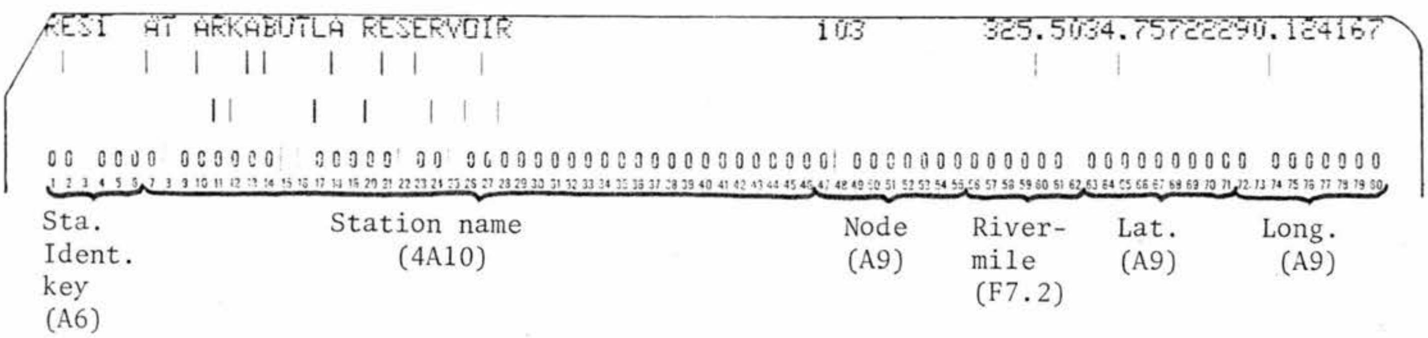


RIVER CONTROL STRUCTURE AND RESERVOIR DATA FILE (YAZSTRS)

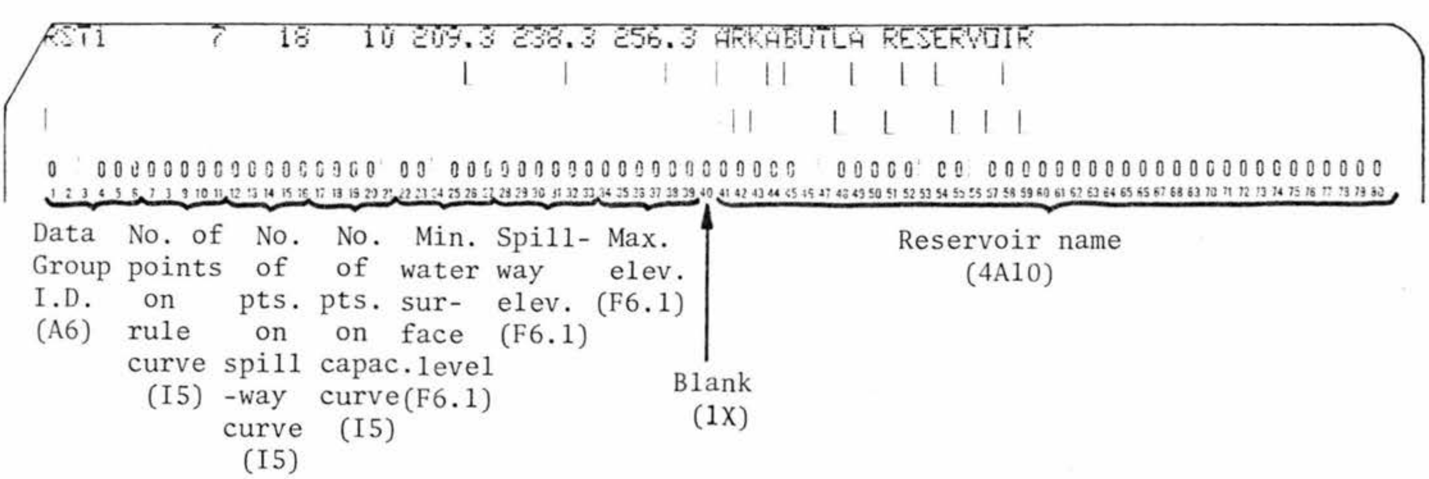
This file is the combination of two small data files: the River Control Structure Data File and the Reservoir Data File. Coding of the information in these two files is totally different. In the Reservoir Data File, each group of data cards belonging to the particular reservoir is headed by two cards: a location description card and a reservoir characteristic summary card.

EXAMPLE: Reservoir data of Arkabutla Reservoir.

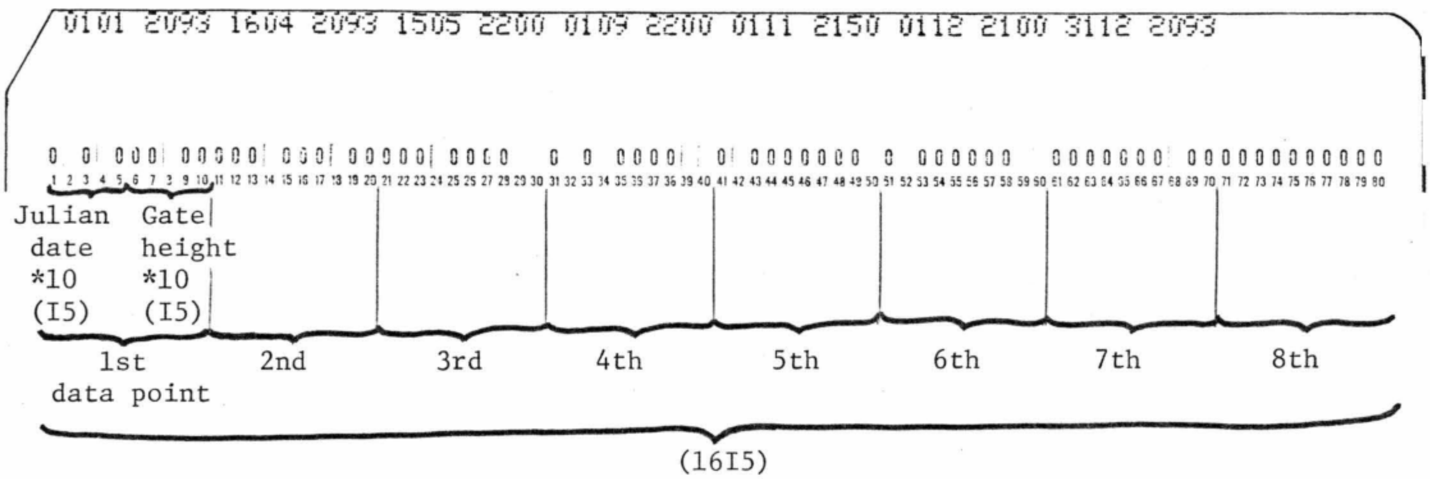
CARD 1: (location description)



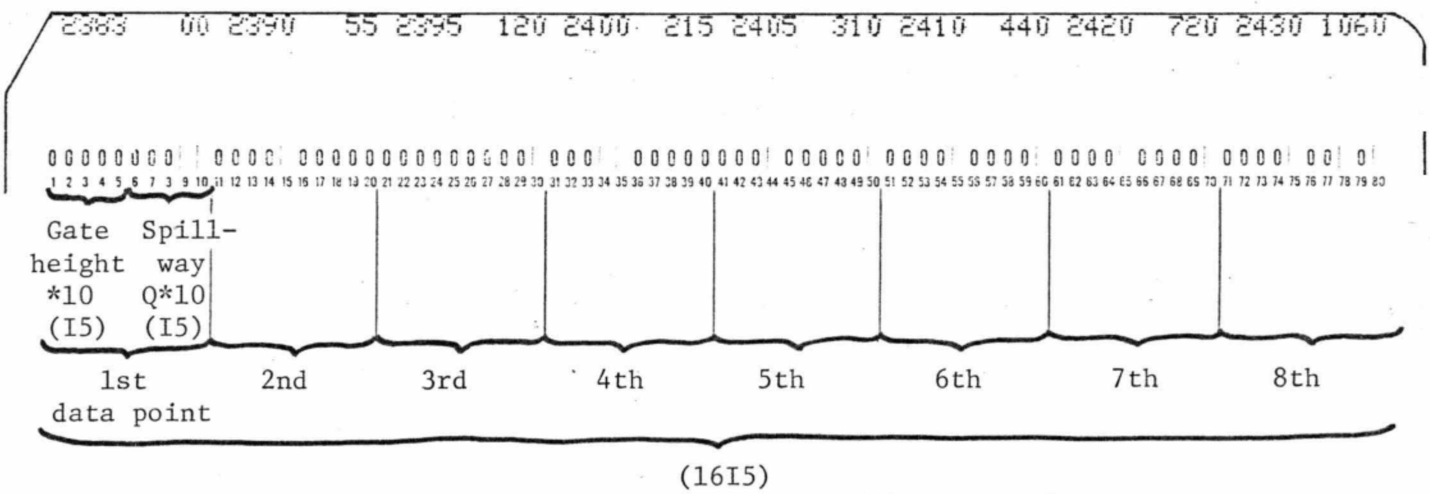
CARD 2: (reservoir characteristics summary)



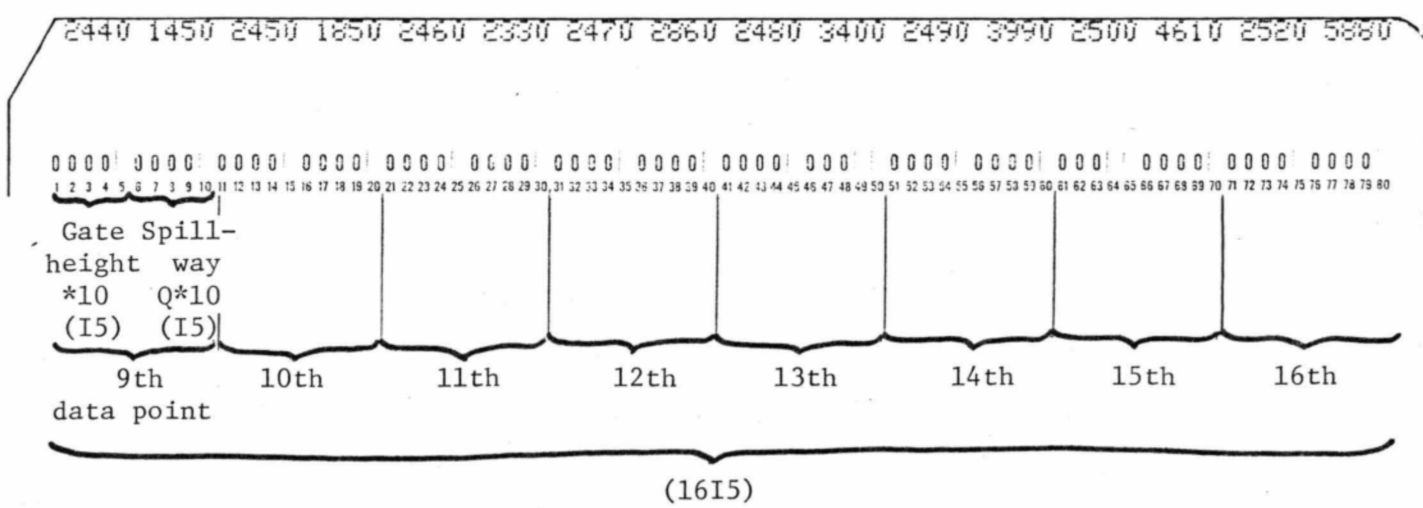
CARD 3: (reservoir operation data for rule curve)



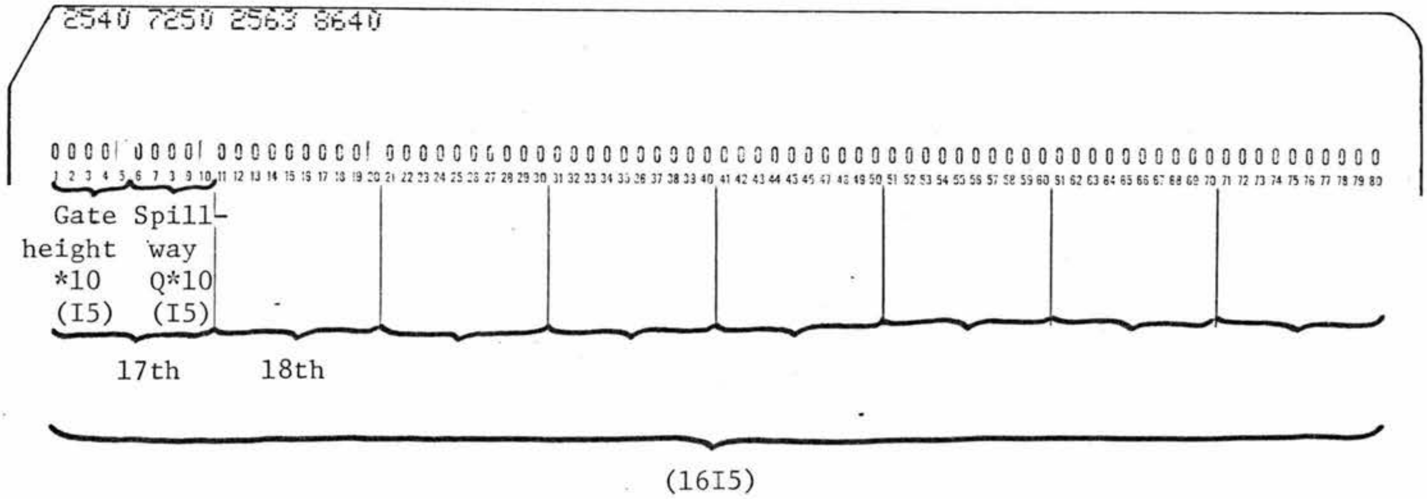
CARD 4: (reservoir operation data for spillway curve)



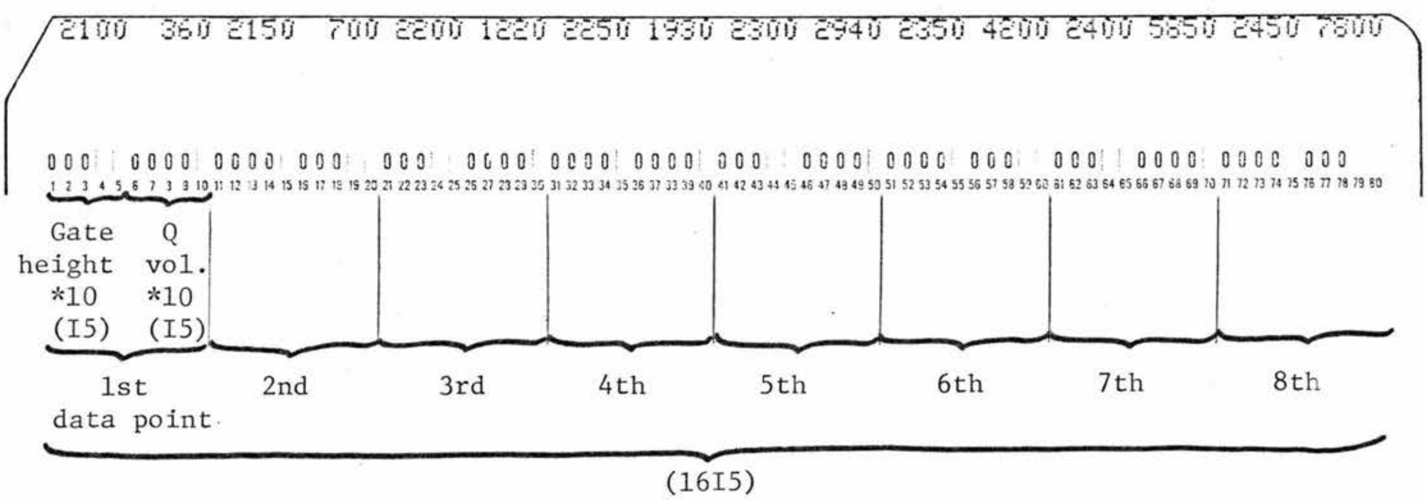
CARD 5: (reservoir operation data for spillway curve)



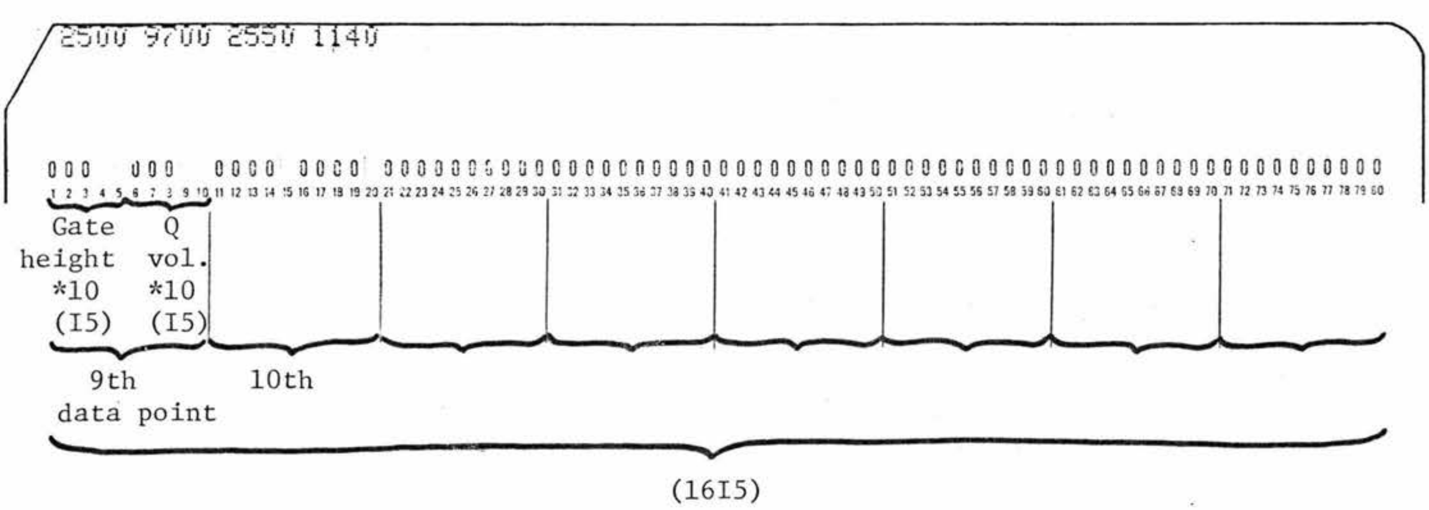
CARD 6: (reservoir operation data for spillway curve)



CARD 7: (reservoir operation data for capacity curve)



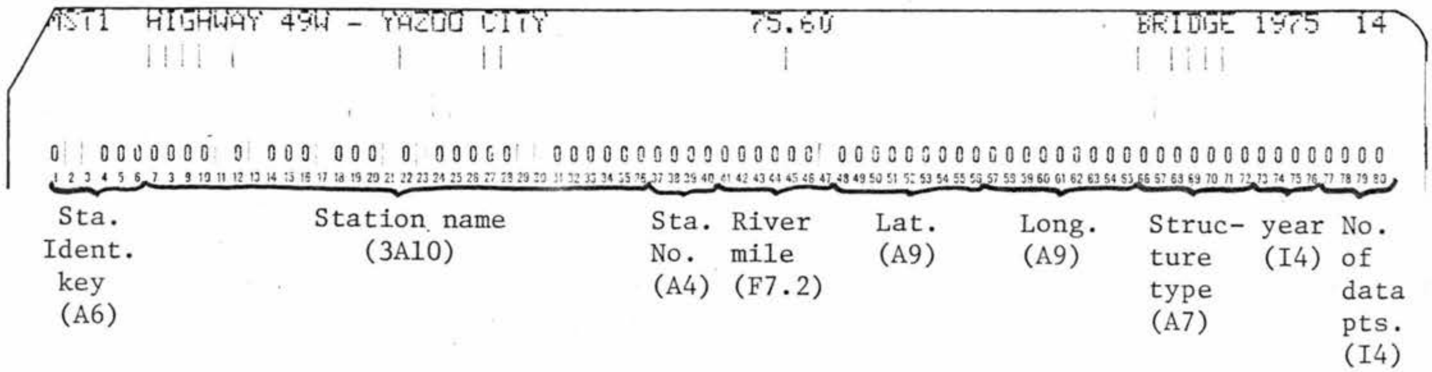
CARD 8: (reservoir operation data for capacity curve)



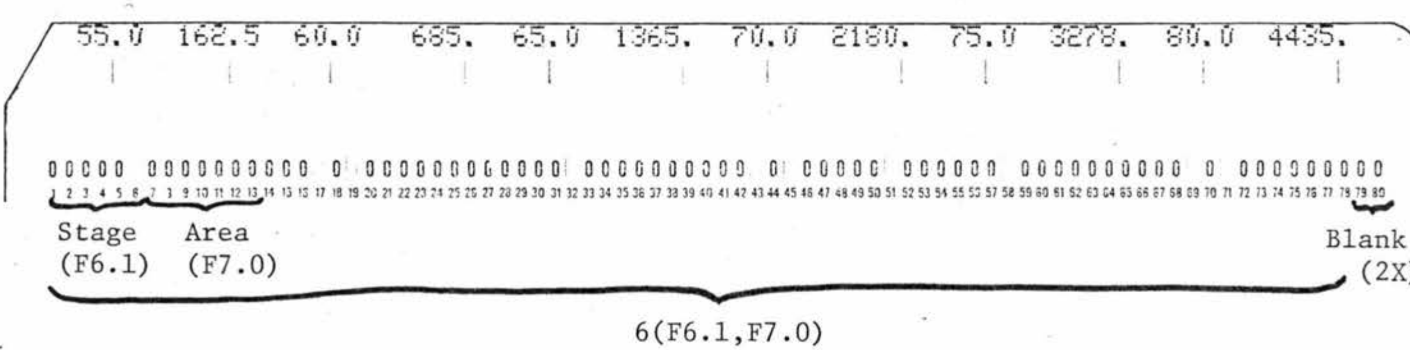
In the Control Structure Data File, each group of data cards belonging to the particular structure is headed by one location description card.

EXAMPLE: Bridge on the Yazoo River at river mile 75.60 (Highway 49W--Yazoo City).

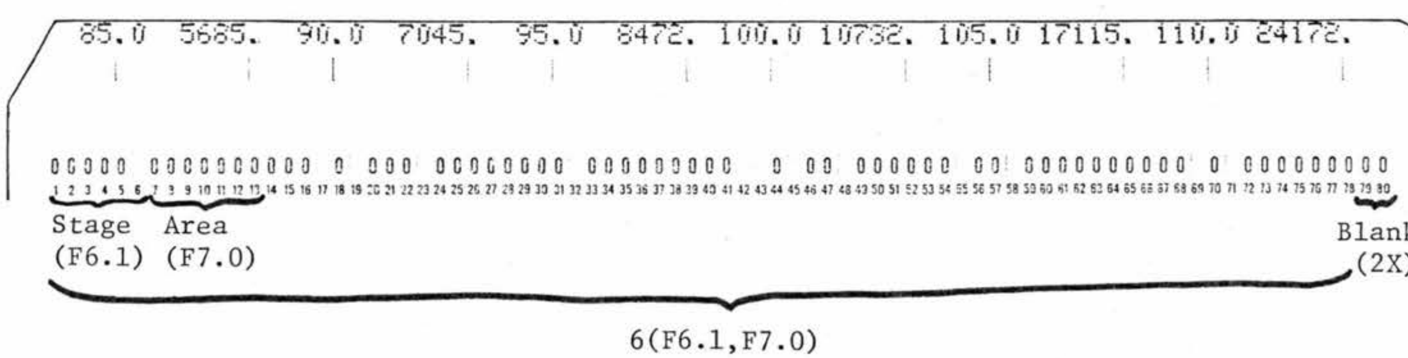
CARD 1: (location description)



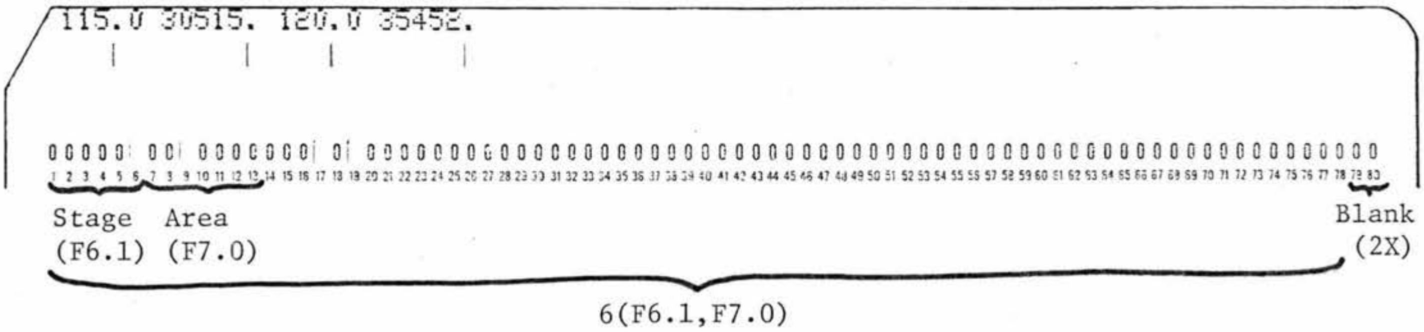
CARD 2: (or subsequent data cards)



CARD 3: (or subsequent data cards)



CARD 4: (or subsequent data cards)

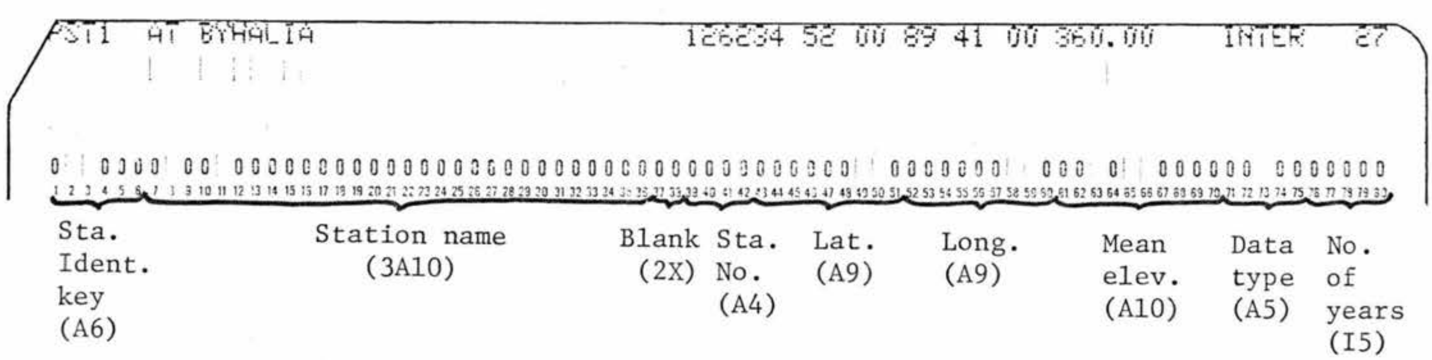


PRECIPITATION DATA FILE (YAZRAIN)

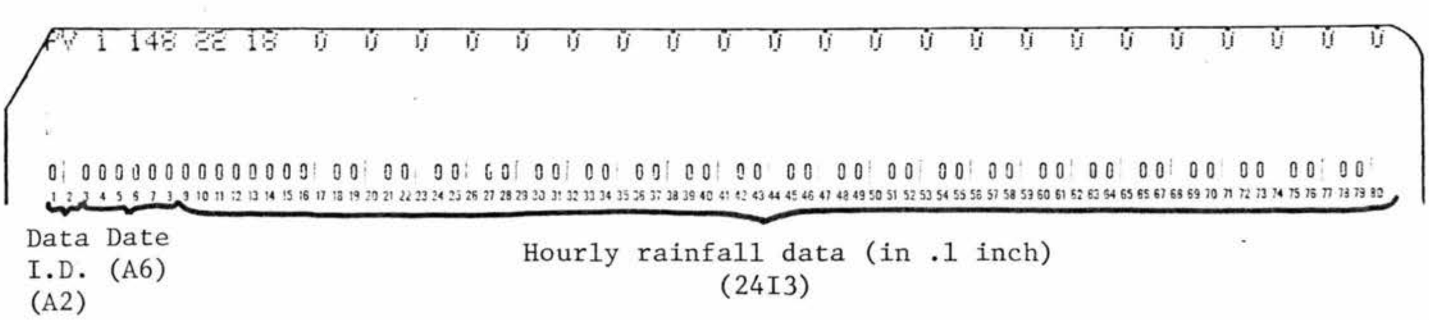
In the Precipitation Data File, each group of data cards belonging to one particular gaging station is headed by a location description card. All rainfall data are hourly values and coded as .1 inch. Each card represents one day having rainfall, therefore, it contains a date and 24 values.

EXAMPLE: Precipitation at Byhalia for one day (1948).

CARD 1: (location description)



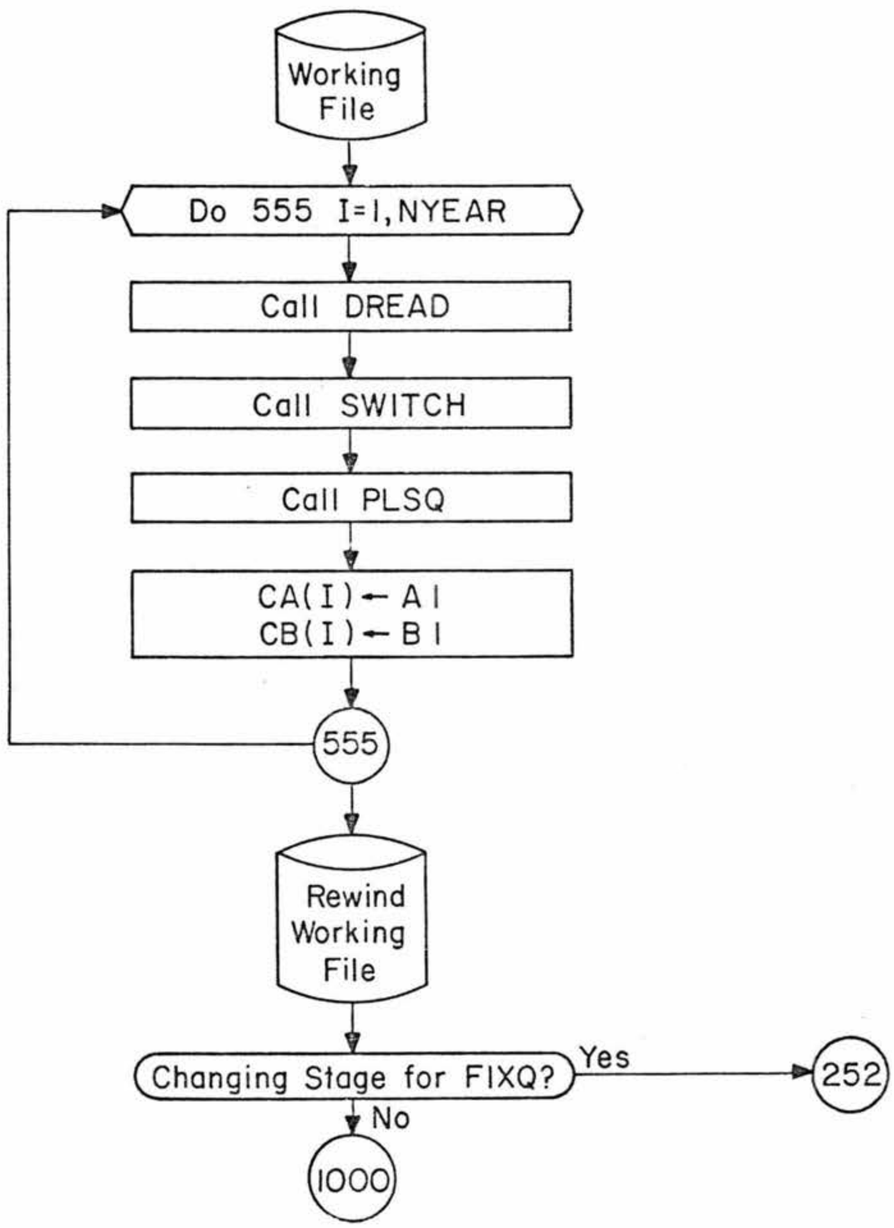
CARD 2: (or subsequent data cards)



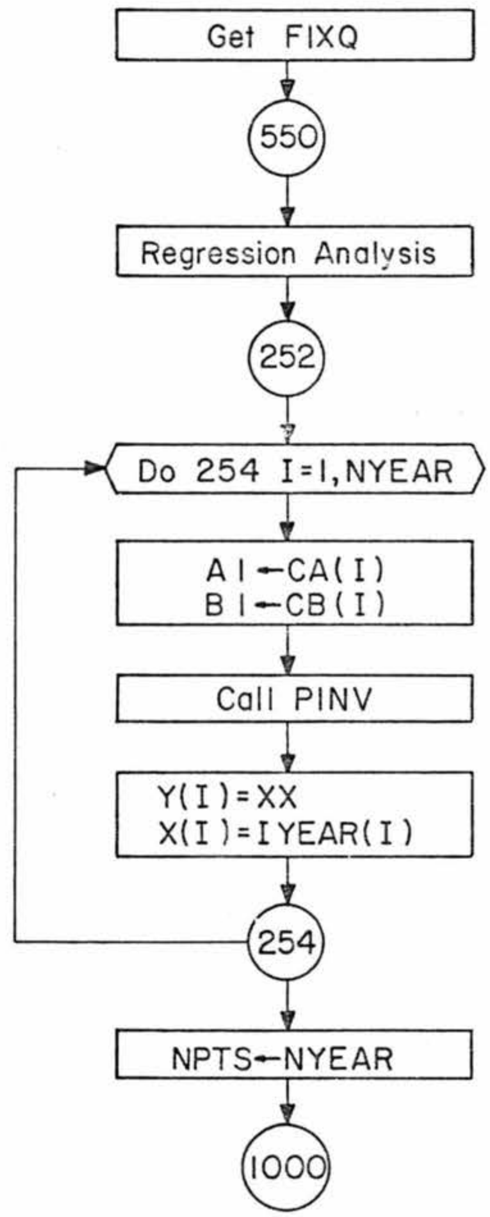
APPENDIX B

FLOWCHARTS OF PROGRAM OPERATION

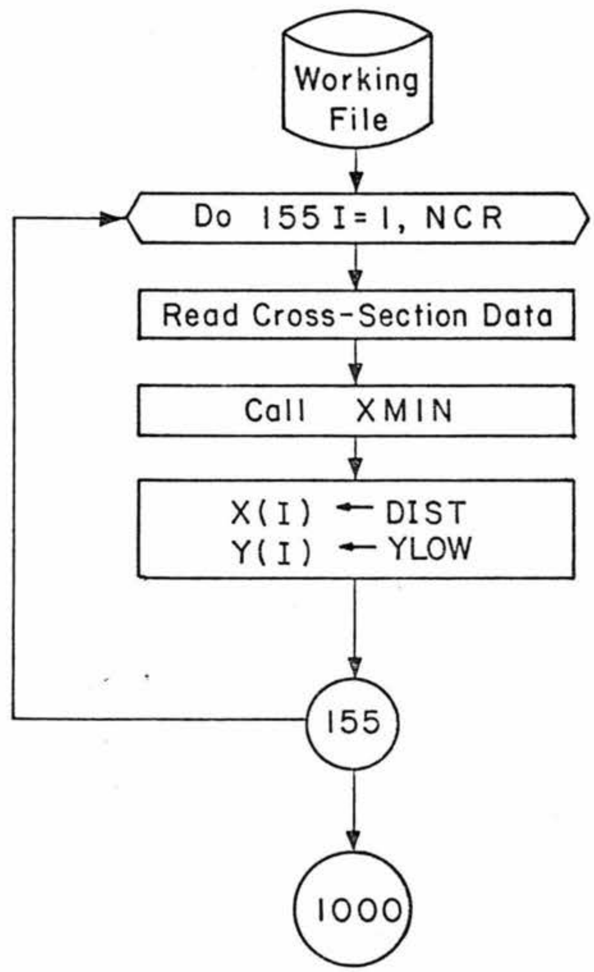
REGRESSION ANALYSIS



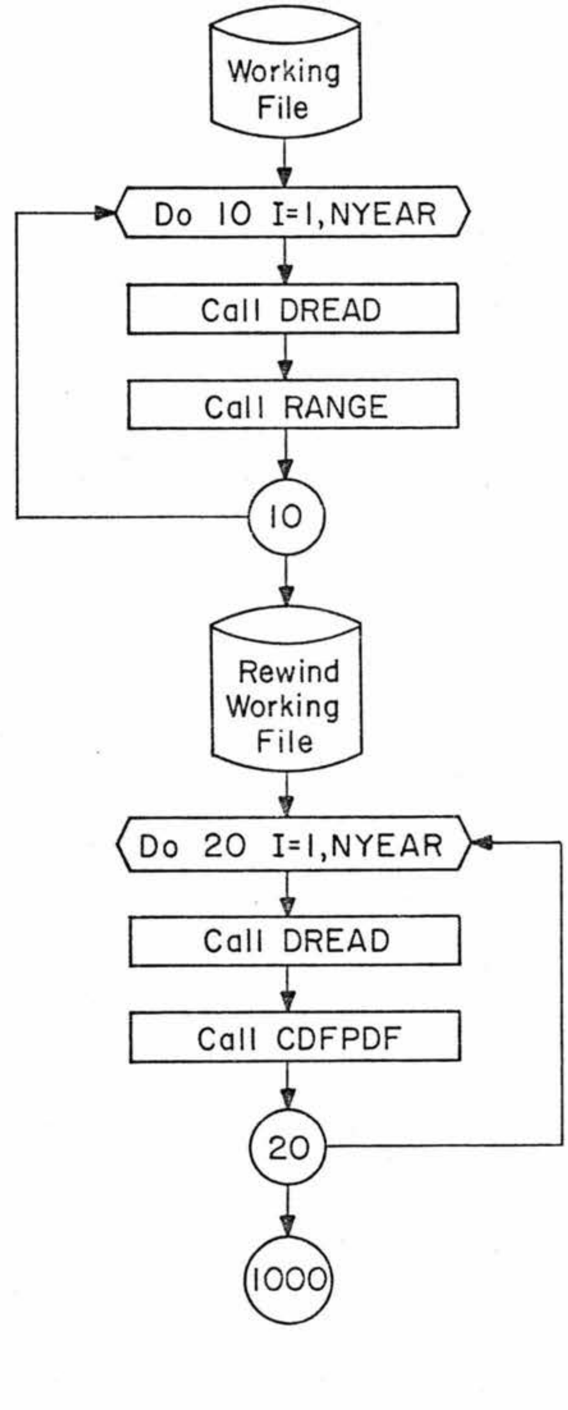
CHANGING STAGE FOR A GIVEN DISCHARGE (FIXQ)



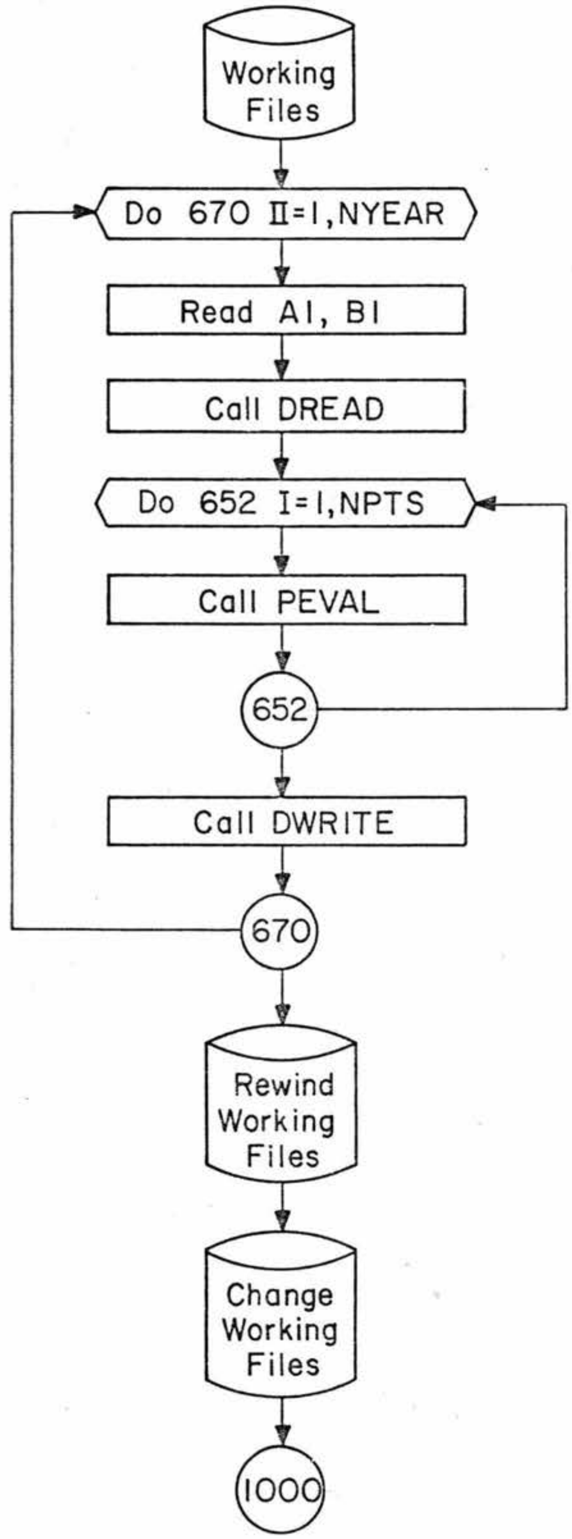
THALWEG LEVEL



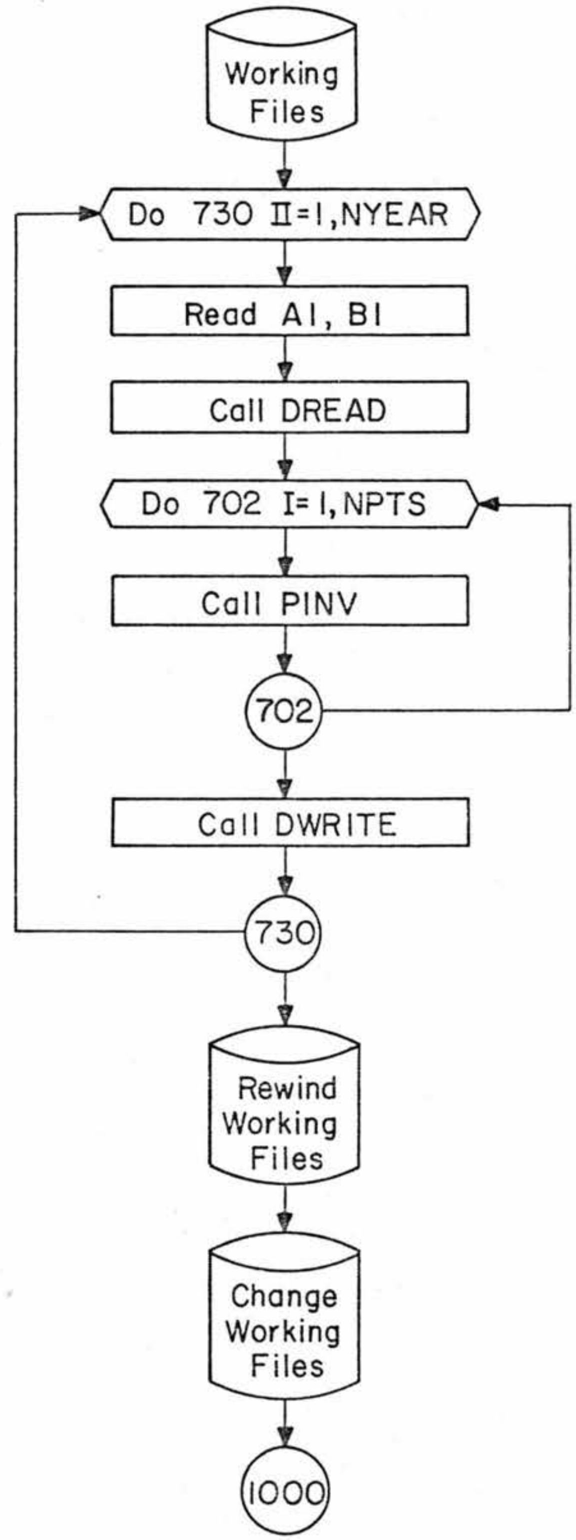
FREQUENCY ANALYSIS



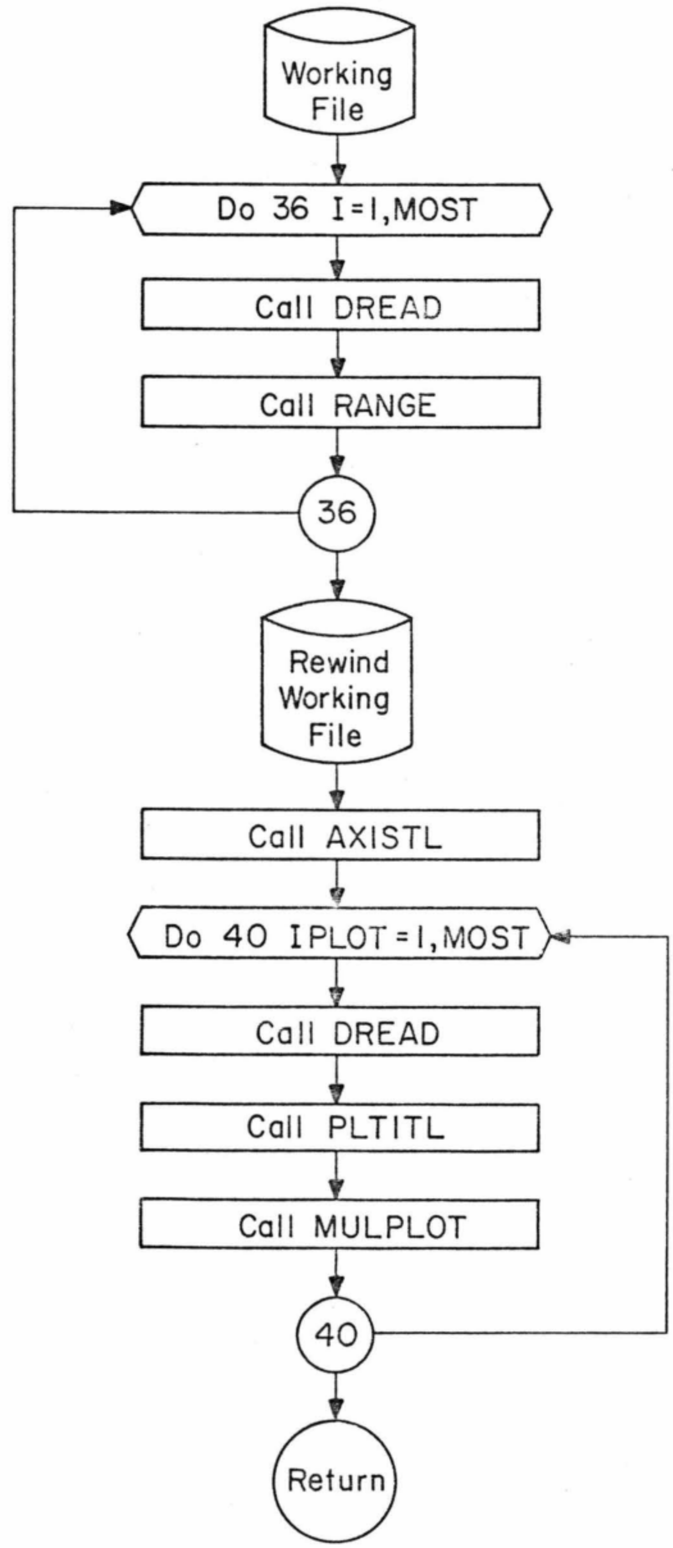
DISCHARGE HYDROGRAPH GENERATION



STAGE HYDROGRAPH GENERATION



INFORMATION DISPLAY BY A LINE PRINTER



APPENDIX C

DEFINITIONS OF VARIABLES

DEFINITIONS OF VARIABLES USED IN PROGRAM YAZDB

PARAMETER	DESCRIPTION
CA(150)	arrays containing regression coefficients in a power function curve fitting
CB(150)	arrays containing regression coefficients in a power function curve fitting
CØNC	sediment concentration (ppm)
DATE(150)	array containing dates (integer) in a given data set
DF	data file
DISC(150)	array containing discharge values
DIST	river-mile, from downstream node, in the data file
DIST 1	starting river-mile desired
DIST 2	ending river-mile desired
DVALUE(12,31)	array containing daily values in a year
FDATE(150)	array containing dates (floating) in a given data set
FIXQ	a given discharge value, in CFS
FVALUE(31)	array containing daily values in a month
GAGØ	gage-zero elevation
GAHT(50)	array containing gate height values
GELEV	mean basin elevation, ft above MSL
GVØL(50)	array containing reservoir water volume, in cubic ft
HVALUE(366,25)	array containing hourly values in a year
ICAT(10)	array containing data category indices
ICATEG	output, flag for data category identification

PARAMETER	DESCRIPTION
ICØM	identifiable parts of a command statement
ICØMD	identifiable parts of a command statement
IDATE	a given date (DYMØYR)
IFILE	a file name
IKEY(300)	array containing keys to be used in data retrieval
ILAT	location latitude desired
ILØC	checking parameter for location identification
ILØNG	location longitude desired
IN	input file
INAME	identification part of river name in the data file
INØDE	river node desired
IØUT	output, flag for output type selection
IØUTP	input, output type desired
IPR	output, flag for data processing operation
IPRØ	input, processing operation desired
IRETRIV	checking parameter for data retrieval operation
ISET	checking parameter for set flag operation
ISUBL(10)	array containing sublevels of information
ITEK	flag index for Tektronix plotting
ITEMP(7)	working arrays
ITIM	checking parameter for time-period identification
IY(8)	working arrays

PARAMETER	DESCRIPTION
lYEAR(100)	array containing years of data to be retrieved
IZ(2)	working arrays
JCØMWD(4)	array containing identifiable words in a statement
JEND	ending time desired
JFLAG	output, flag for time period type selection
JNAME	river name desired
JSTART	starting time desired
JTEMP(6)	working arrays
JVAL(24)	array containing hourly values in a day
JYR1	corresponding year for a given starting date
JYR2	corresponding year for a given ending date
KEY	a retrieval key
KKEY	number of retrieved keys
KTEMP(5)	working arrays
LATI	location latitude in the data file
LEAP	index for leap year (LEAP=1)
LEVEL	level of information desired
LFLAG	output, flag for location type selection
LINES	number of lines to be printed by the line printer
LØNG	location longitude in the data file
LTEMP(4)	working arrays
LTIT(8)	title of the data category file
MØAVG(12)	array containing monthly average values

PARAMETER	DESCRIPTION
MØMIN(12)	array containing monthly minimum values
MØMAX(12)	array containing monthly maximum values
MØ(12)	array containing months of the year
MØST	maximum number of graphs to be plotted
MTIT(8)	title of a graph
NAME(4)	river name in the data file
NAMST	station name desired
NCCP	number of data points on reservoir capacity curve
NCR	number of cross-sections retrieved for a given location
NCRØS	number of cross-sections for a given location in the data file
ND1	order of a starting date in a given year
ND2	order of an ending date in a given year
NDATA	number of data points
NDAY(12)	array containing numbers of days in a month
NKEYS	number of keys to be used in data retrieved
NNAME(50,3)	array containing station names to be retrieved
NNØDE(50)	array containing river nodes to be retrieved
NØDE	river node in the data file
NPTS	number of points in the retrieved data set
NRCP	number of data points on reservoir rule curve
NRECORD	number of retrieved records

PARAMETER	DESCRIPTION
NSCP	number of data points on reservoir spillway curve
NSUB	number of rivers and tributaries in Yazoo basin
NTEMP(2)	working arrays
NUMST	station number desired
NWØRD	number of words in a statement
NX	working parameters
NXG(50)	working array for gate height data
NXQ(50)	working array for reservoir water volume data
NXS(50)	working array for spillway discharge data
NY	working parameters
NYEAR	number of years of data to be retrieved
NYR(10)	array containing numbers of years to be retrieved
NYRS	number of years having records
NZ	working parameters
PCEN1	cumulative concentration, in percent, for sediment size, SIZE1
PCEN2	cumulative concentration, in percent, for sediment size, SIZE2
PCEN3	cumulative concentration, in percent, for sediment size, SIZE3
PCEN4	cumulative concentration, in percent, for sediment size, SIZE4
PCEN5	cumulative concentration, in percent, for sediment size, SIZE5
PCEN6	cumulative concentration, in percent, for sediment size, SIZE6

PARAMETER	DESCRIPTION
SIZE1	sediment size = .062 mm
SIZE2	sediment size = .125 mm
SIZE3	sediment size = .250 mm
SIZE4	sediment size = .500 mm
SIZE5	sediment size = 1.0 mm
SIZE6	sediment size = 2.0 mm
SNAME(4)	location name
SPIQ(50)	array containing spillway discharge values, in CFS
STAG(150)	array containing river stage values
STNA(3)	station name in the data file
STNØ	station number in the data file
SUBLEVL	sub-level of information desired
TEMP(366)	arrays containing retrieved data
TUNIT	time-unit to be printed below the graph title
TYPE	data type
W	output file
WF1	working file, also used to save the results
WF2	working file
WF3	working file
X(366)	arrays containing retrieved data
XDIS(100)	array containing transversal distances from a reference line
XSECT	transversal distance in a cross-section
XTIT	title on the X-axis
XUNIT	unit to be printed on X-axis

PARAMETER	DESCRIPTION
Y(366)	arrays containing retrieved data
YEAR	year of a data set in a data file
YELEV(100)	array containing cross-section elevations
YTIT	title on the Y-axis
YUNIT	unit to be printed on Y-axis

APPENDIX D

PROGRAM LISTING

PROGRAM YAZDB

```
      OVERLAY(TEST,0,0)
      PROGRAM YAZDB
     1(INPUT=64,OUTPUT=64,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=64,TAPE2=64,
     2TAPE3=64,TAPE11=64,TAPE12=64,TAPE13=64,TAPE14=64,TAPE15=64,
     3TAPE16=64,TAPE17=64,TAPE18=64)
C
C     ************************************************************
C     *                                                          *
C     *              YAZOO DATA BASE MANAGEMENT                  *
C     *                       PACKAGE                            *
C     *                   -- VERSION 1 --                        *
C     *                                                          *
C     ************************************************************
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,CTYP,
     1PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SNAME,SUBLEVL,XTIT,YTIT
      REAL MWSL,MXEL,MOAVG,MOMIN,MOMAX
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/RIVSTR/STAGE(50),AREA(50)
      COMMON/RIVRES/GAHT(50),NRCP,SPIQ(50),NSCP,QVOL(50),NCCP,NXG(50),
     1NXS(50),NXQ(50)
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
C
C   ---   INPUT INFORMATION AND PROGRAM INITIALIZATION ---
C
      DATA (NDAY(I),I=1,12)/31,28,31,30,31,30,31,31,30,31,30,31/
      DATA (MO(I),I=1,12)/3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,3HJUL,
     13HAUG,3HSEP,3HOCT,3HNOV,3HDEC/
      IN=5   $   W=6   $   WF1=1   $   WF2=2   $   WF3=3
      NTIME=0
      NSUB=39
      CALL INITT(30)
      DF=11
      DO 45 I=1,NSUB
      READ(DF,40) (NNAME(I,J),J=1,3),NNODE(I)
   40 FORMAT(3X,3A10,32X,A9,6X)
   45 CONTINUE
  500 CONTINUE
      REWIND DF
      REWIND WF1
      REWIND WF2
      REWIND WF3
      IOUT=IPR=0
```

```
      ITEK=3H
      DO 10 I=1,366
      X(I)=Y(I)=0.
   10 CONTINUE
      DO 20 I=1,8
   20 MTIT(I)=10H
C
C   ---  SWITCH CONTROL TO VARIOUS OVERLAYS FOR DATA MANAGEMENT  ---
C
      IF(NTIME.GT.0) GO TO 30
      WRITE(W,1000)
 1000 FORMAT(//,26X,*YAZOO RIVER SYSTEM DATA BANK*/,* TO USE, ANSWER THE
     1 QUESTIONS*/)
   30 WRITE(W,1001)
 1001 FORMAT(* DO YOU WISH TO BE HELPED IN SETTING UP THE INPUT COMMAND
     1STATEMENTS*/)
      READ(IN,2000) ICOMD
 2000 FORMAT(A3)
      IF(ICOMD.EQ.3HNO ) GO TO 50
      WRITE(W,1002)
 1002 FORMAT(/,* ALL DATA RETRIEVALS BEGIN WITH "GET" COMMAND*/,
     1* DATA CATEGORIES IN THE YAZOO RIVER SYSTEM DATA BANK CONSIST OF*/
     2,20X,*ALL*/,20X,*STAGE-DISCHARGE*/,20X,*DISCHARGE */,20X,*RIVER ST
     3AGE*/,20X,*SUSPENDED SEDIMENT*/,20X,*BED MATERIAL*/,20X,*CROSS-SEC
     4TION*/,20X,*CONTROL STRUCTURE */,20X,*RESERVOIR */,20X,*RESERVOIR
     5DATA ,ALL*/,35X,*,SPILLWAY CURVE*/,35X,*,CAPACITY CURVE */
     6,35X,*,RULE CURVE */,20X,*PRECIPITATION */,15X,*AND  PRECIPITATION
     7 DATA ,HOURLY*/,39X,*,DAILY*/* EACH DATA ELEMENT CAN BE RETRIEVED
     8ACCORDING TO ONE OF THE FOLLOWING TYPES OF*/,* INFORMATION*/
     9,20X,*STATUS*/,20X,*DATA*/,15X,*OR   ALL INFORMATION*)
      WRITE(W,1102)
 1102 FORMAT(* KEY-IN THE COMMAND WORD "GET" WITH THE APPROPRIATE DATA C
     1ATEGORY AND TYPE OF */,* INFORMATION DESIRED: EXAMPLE: GET,RIVER S
     2TAGE,DATA*/)
C
C   ---    START DATA RETRIEVAL PROCESS    ---
C
   50 CALL OVERLAY(4HTEST,1,0,0)
      IF(LEVEL.EQ.1) GO TO 220
      IF(IPR.NE.0) GO TO 60
      IF(IOUT.EQ.2.OR.IOUT.EQ.3) GO TO 70
      GO TO 75
C
C   ---    START DATA PROCESSING    ---
C
   60 CALL OVERLAY(4HTEST,2,0,0)
      IF(IOUT.EQ.0.OR.IOUT.EQ.1) GO TO 75
      IF(IOUT.EQ.4) GO TO 220
C
C   ---    START DATA PRESENTATION    ---
C
   70 CALL OVERLAY(4HTEST,3,0,0)
      ITEK=3H
   75 WRITE(W,1010)
 1010 FORMAT(/,* DO YOU WISH TO CONTINUE PROCESSING THE RETRIEVED DATA S
     1ET*/)
      READ(IN,2000) ICOM
      IF(ICOM.EQ.3HNO ) GO TO 220
      REWIND WF1
      REWIND WF2
```

```
      REWIND WF3
      READ(IN,1011) IOUTP,(ITEMP(I),I=1,7)
 1011 FORMAT(8X,A3,6A10,A9)
      CALL OUTFLAG(IOUTP,IOUT)
      IF(IOUT.EQ.3) GO TO 92
      IF(IOUT.EQ.5) GO TO 94
      DECODE(69,90,ITEMP) NX,IPRO,(JCOMWD(I),I=1,4),(NTEMP(J),J=1,2)
   90 FORMAT(A2,6A10,A7)
      GO TO 96
   92 DECODE(69,93,ITEMP) NX,IPRO,(JCOMWD(I),I=1,4),(NTEMP(J),J=1,2)
   93 FORMAT(A5,6A10,A4)
      GO TO 96
   94 DECODE(69,95,ITEMP) NX,IPRO,(JCOMWD(I),I=1,4),(NTEMP(J),J=1,2)
   95 FORMAT(A3,6A10,A6)
   96 CALL PROFLAG(IPRO,IPR)
      GO TO 60
  220 WRITE(W,230)
  230 FORMAT(/,* DO YOU WISH TO CONTINUE WORKING WITH DATA IN THIS DATA
     1BASE*/)
      READ(IN,2000) ICOM
      IF(ICOM.EQ.3HNO ) GO TO 80
      NTIME=NTIME+1
      GO TO 500
   80 WRITE(W,85)
   85 FORMAT(5X,*END OF THE DATA MANAGEMENT JOB*)
C
      STOP
      END
```

SUBROUTINE MULPLOT (ISYMBOL, IPLOT, MOST, XL, XH, YL, YH)

```
      SUBROUTINE MULPLOT(ISYMBOL,IPLOT,MOST,XL,XH,YL,YH)
C
C            PLOT MULTIPLE CURVES ON PAPER
C
      INTEGER W,WF1,WF2,WF3,YEAR,DATE,STNA,STNO,XTIT,YTIT,XUNIT,
     1YUNIT,SNAME
      DIMENSION ZX(12),IGRAPH(111),IPOINTS(3)
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
C
C
      IF(LINES.GT.366) GO TO 999
      DECODE(10,3,YTIT) (IA(I),I=1,10)
    3 FORMAT(10A1)
      IPOINTS(IPLOT)=ISYMBOL
      IF(IPLOT.GT.1) GO TO 5
      IBORDER=1HI
      IBLANK=1H
      XSCALE=(XH-XL)*(1./110.)
      A=LINES-1
      YSCALE=(YH-YL)/A
      DO 200 I=1,366
      DO 200 J=1,3
      IPOSX(I,J)=IPOSY(I,J)=0
  200 CONTINUE
    5 DO 15 I=1,NPTS
      IF(XH-X(I)) 15,11,11
   11 IF(X(I)-XL) 15,12,12
   12 IF(YH-Y(I)) 15,13,13
   13 IF(Y(I)-YL) 15,14,14
   14 JX=(X(I)-XL)/XSCALE+1.5
      JY=(Y(I)-YL)/YSCALE+.5
      JY=LINES-JY
      IPOSX(I,IPLOT)=JX
      IPOSY(I,IPLOT)=JY
   15 CONTINUE
      IF(IPLOT-MOST) 20,30,20
   20 RETURN
   30 DO 40 K=1,12
      ZX(K)=10.*FLOAT(K-1)*XSCALE+XL
   40 CONTINUE
C          IF IOPT=0, REGULAR PLOT
C                 =1, X-AXIS WILL BE LABELED WITH ALPHABETIC NAMES
C                 =2, X-AXIS WILL PLOT TIME IN ASCENDING ORDER
      MID=LINES/2
      MID5=MID-5
      MID10=MID5+9
      IF(IOPT.NE.2) GO TO 41
      WRITE(W,8) XTIT
      WRITE(W,7)(ZX(K),K=1,12),XUNIT
      WRITE(W,72)
      GO TO 43
   41 WRITE(W,2) YUNIT
   43 IF(IOPT.EQ.2) GO TO 42
      YES=YH+YSCALE
      GO TO 44
```

```
   42 YES=YL-YSCALE
      K=LINES+1
   44 DO 70 JY=1,LINES
      IF(IOPT.EQ.2) GO TO 46
      YES=YES-YSCALE
      K=JY
      GO TO 48
   46 YES=YES+YSCALE
      K=K-1
   48 ZY=YES
      IGRAPH(1)=IGRAPH(111)=IBORDER
      DO 50 I=2,110
   50 IGRAPH(I)=IBLANK
      DO 60 J=1,MOST
      DO 60 I=1,NPTS
      IF(IPOSY(I,J).NE.JY) GO TO 60
      JX=IPOSX(I,J)
      IF(JX.EQ.0) GO TO 60
      IGRAPH(JX)=IPOINTS(J)
   60 CONTINUE
      IF((JY.LT.MID5).OR.(JY.GT.MID10)) GO TO 62
      J=JY-MID5+1
      WRITE(W,78) IA(J),ZY,(IGRAPH(I),I=1,111)
      GO TO 70
   62 WRITE(W,4) ZY,(IGRAPH(I),I=1,111)
   70 CONTINUE
      IF(IOPT.EQ.2) GO TO 74
      WRITE(W,6)
      IF(IOPT.EQ.0) WRITE(W,7)(7X(K),K=1,12),XUNIT
      IF(IOPT.EQ.1) GO TO 80
      WRITE(W,8) XTIT
      RETURN
   80 WRITE(W,9)(MO(K),K=1,12),(IYEAR(K),K=1,12)
      RETURN
   74 WRITE(W,76) YUNIT
      RETURN
C
C  --- FORMAT STATEMENTS ---
C
    2 FORMAT(7X,A5,2X,11(10HI---------),1HI)
    4 FORMAT(1H ,2X,F9.2,2X,111A1)
    6 FORMAT(1H ,13X,11(10HI---------),1HI)
    7 FORMAT(7X,12(2X,F8.2),A5)
    8 FORMAT(/,60X,A10/)
    9 FORMAT(1H ,6X,12(7X,A3),/,6X,12(6X,I4))
   78 FORMAT(1H ,1X,A1,F9.2,2X,111A1)
   72 FORMAT(14X,11(10HI---------),1HI)
   76 FORMAT(1H ,6X,A5,2X,11(10HI---------),1HI)
  999 PRINT 100
  100 FORMAT(1H ,5X,*DIMENSION IS NOT ENOUGH TO SET LINES*)
      RETURN
C
      END
```

SUBROUTINE RANGE (X, J1, KN, XHIGH, IPMAX, XLOW, IPMIN)

```
      SUBROUTINE RANGE(X,J1,KN,XHIGH,IPMAX,XLOW,IPMIN)
C        GET MAX AND MIN VALUES OF A GIVEN SET X.
      DIMENSION X(1)
      CALL XMAX(X,J1,KN,XHIGH,IPMAX)
      CALL XMIN(X,J1,KN,XLOW,IPMIN)
      RETURN
      END
```

SUBROUTINE XMAX (X, JI, KN, XHIGH, IPMAX)

```
      SUBROUTINE XMAX(X,JI,KN,XHIGH,IPMAX)
C          FIND THE MAXIMUM VALUE AND ITS POSITION IN A GIVEN SET X
      DIMENSION X(1)
      J2=JI+1   $   XHIGH=X(JI)   $   IPMAX=JI
      DO 10 I=J2,KN
      IF(XHIGH-X(I)) 5,10,10
    5 XHIGH=X(I)   $   IPMAX=I
   10 CONTINUE
      RETURN
      END
```

SUBROUTINE XMIN (X, JI, KN, XLOW, IPMIN)

```
      SUBROUTINE XMIN(X,JI,KN,XLOW,IPMIN)
C           FIND THE MINIMUM VALUE AND ITS POSITION IN A GIVEN SET X
      DIMENSION X(1)
      J2=JI+1    $    XLOW=X(JI)    $    IPMIN=JI
      DO 10 I=J2,KN
      IF(XLOW-X(I)) 10,10,5
    5 XLOW=X(I)    $    IPMIN=I
   10 CONTINUE
      RETURN
      END
```

SUBROUTINE YXMAX (YXHIGH, MYEAR, IPYMAX)

```
      SUBROUTINE YXMAX(YXHIGH,MYEAR,IPYMAX)
C
C     -  SEARCH FOR THE MAXIMUM VALUE AND ITS POSITION IN NYEAR RECORDS
C
      INTEGER W,WF1,WF2,WF3,DF
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRC,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      NDATA=0   $   YXHIGH=0.   $   IPYMAX=0
      DO 100 I=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      NDATA=NDATA+NPTS
      CALL XMAX(X,1,NPTS,XHIGH,IPMAX)
      IF(XHIGH.LE.YXHIGH) GO TO 100
      YXHIGH=XHIGH   $   MYEAR=I   $   IPYMAX=IPMAX
  100 CONTINUE
      REWIND WF1
C
      RETURN
      END
```

SUBROUTINE YXMIN (YXLOW, MYEAR, IPYMIN)

```
      SUBROUTINE YXMIN(YXLOW,MYEAR,IPYMIN)
C         SEARCH FOR THE MINIMUM VALUE AND ITS POSITION IN NYEAR RECORDS
      INTEGER W,WF1,WF2,WF3,DF
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      NDATA=0    $   YXLOW=100000000.    $    IPYMIN=0
      DO 100 I=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      NDATA=NDATA+NPTS
      CALL XMIN(X,1,NPTS,XLOW,IPMIN)
      IF(XLOW.GE.YXLOW) GO TO 100
      YXLOW=XLOW    $    MYEAR=I    $    IPYMIN=IPMIN
  100 CONTINUE
      REWIND WF1
      RETURN
      END
```

SUBROUTINE DCONVRT (JYEAR, NDT, IMO, IDAY)

```
      SUBROUTINE DCONVRT(JYEAR,NDT,IMO,IDAY)
C
C          CONVERT AN ORDER IN A YEAR INTO A DATE
C
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
C
      CALL LEAPYR(JYEAR,LEAP)
      IF(LEAP.EQ.1) NND=366
      IF(LEAP.EQ.0) NND=365
      IF(NDT.LE.NND) GO TO 5
      JYEAR=JYEAR+1    $    NDT=NDT-NND
    5 CALL LEAPYR(JYEAR,LEAP)
      IF(LEAP.EQ.1) NDAY(2)=29
      IF(LEAP.EQ.0) NDAY(2)=28
      DO 10 I=1,12
      IMO=I
      ND=NDAY(I)
      IF(ND.GE.NDT) GO TO 20
      NDT=NDT-ND
   10 CONTINUE
   20 IDAY=NDT
      RETURN
      END
```

SUBROUTINE LEAPYR (YEAR, LEAP)

```
      SUBROUTINE LEAPYR(YEAR,LEAP)
C
C          IDENTIFY WHETHER A GIVEN YEAR IS A LEAP YEAR, LEAP=1
C
      INTEGER YEAR
C
      LEAP=0
      IF((YEAR.EQ.1900).OR.(YEAR.EQ.2000)) GO TO 10
      MYR=YEAR/4
      IDIF=YEAR-(4*MYR)
      IF(IDIF.EQ.0) LEAP=1
   10 RETURN
      END
```

SUBROUTINE DSTAT (YXLOW, YXHIGH, XMEAN, XSTDV)

```
      SUBROUTINE DSTAT(YXLOW,YXHIGH,XMEAN,XSTDV)
C           CALCULATE THE BASIC STATISTICS OF A GIVEN SET X
      INTEGER W,WF1,WF2,WF3,DF
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IFR,IPROC,FDATE(150),ITEK
C
      YXLOW=100000000.   $   YXHIGH=0.
      NDATA=0
      SUM=SSQ=0.
      DO 100 J=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      NDATA=NDATA+NPTS
      CALL XMIN(X,1,NPTS,XLOW,IPMIN)
      IF(XLOW.LT.YXLOW) YXLOW=XLOW
      CALL XMAX(X,1,NPTS,XHIGH,IPMAX)
      IF(XHIGH.GT.YXHIGH) YXHIGH=XHIGH
      DO 10 I=1,NPTS
      SUM=SUM+X(I)
   10 SSQ=SSQ+X(I)*X(I)
  100 CONTINUE
      REWIND WF1
      FN=FLOAT(NDATA)    $    FN1=FN-1.
      XMEAN=SUM/FN
      SSQ=SSQ/FN
      XSTDV=SQRT((SSQ-XMEAN*XMEAN)*(FN/FN1))
      RETURN
      END
```

SUBROUTINE DREAD (IFILE, ICATEG, SUBLEVL)

```
      SUBROUTINE DREAD(IFILE,ICATEG,SUBLEVL)
C        READ DATA FROM FILE IFILE
      INTEGER YEAR,CONC,PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SUBLEVL
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      GO TO(100,20,30,40,50,100,20,80,86,90),ICATEG
   20 READ(IFILE,22) YEAR,NPTS
   22 FORMAT(2I4)
      READ(IFILE,24)(Y(I),X(I),I=1,NPTS)
   24 FORMAT(F7.2,F7.0)
      RETURN
   30 READ(IFILE,32) IDATE,CONC,SIZE1,PCEN1,SIZE2,PCEN2
   32 FORMAT(2I6,F5.3,I5,F5.2,I5)
      RETURN
   40 READ(IFILE,42) IDATE,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3
     1,SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
   42 FORMAT(I6,F4.2,6(F5.3,A5))
      RETURN
   50 READ(IFILE,52) IDATE,NPTS
   52 FORMAT(I6,I4)
      READ(IFILE,54)(X(I),Y(I),I=1,NPTS)
   54 FORMAT(F6.0,F6.1)
      RETURN
   80 READ(IFILE,22) YEAR,NPTS
      READ(IFILE,83)(X(I),I=1,NPTS)
   83 FORMAT(F7.0)
      IF(IOUT.EQ.1) RETURN
      IF(IPR.EQ.1.OR.IPR.EQ.2) RETURN
      DO 85 I=1,NPTS
   85 Y(I)=FLOAT(I)
      IF(ITEK.EQ.3HYES) GO TO 84
      RETURN
   86 READ(IFILE,22) YEAR,NPTS
      READ(IFILE,88)(X(I),I=1,NPTS)
   88 FORMAT(F7.2)
      IF(IOUT.EQ.1) RETURN
      IF(IPR.EQ.1.OR.IPR.EQ.2) RETURN
      DO 89 I=1,NPTS
   89 Y(I)=FLOAT(I)
      IF(ITEK.NE.3HYES) RETURN
   84 CALL SWITCH
      RETURN
   90 READ(IFILE,22) YEAR,NPTS
      IF(SUBLEVL.EQ.5HDAILY) GO TO 92
      NPTS=NPTS*24
   92 READ(IFILE,94)(X(I),I=1,NPTS)
   94 FORMAT(F5.2)
      IF(IOUT.EQ.1) RETURN
      IF(IPR.EQ.1.OR.IPR.EQ.2) RETURN
      DO 96 I=1,NPTS
   96 Y(I)=FLOAT(I)
      IF(ITEK.EQ.3HYES) GO TO 84
  100 RETURN
      END
```

SUBROUTINE DWRITE (IFILE, ICATEG, SUBLEVL)

```
      SUBROUTINE DWRITE(IFILE,ICATEG,SUBLEVL)
C         WRITE DATA ON IFILE
      INTEGER YEAR,CONC,PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SUBLEVL,
     1W,WF1,WF2,WF3,STNO,STNA,SNAME,TYPE
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/RIVSTR/STAGE(50),AREA(50)
      COMMON/RIVRES/GAHT(50),NRCP,SPIQ(50),NSCP,QVOL(50),NCCP,NXG(50),
     1NXS(50),NXQ(50)
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITER
C
      GO TO(100,20,30,40,50,100,70,80,86,90),ICATEG
   20 WRITE(IFILE,22) YEAR,NPTS
   22 FORMAT(2I4)
      WRITE(IFILE,24)(STAG(I),DISC(I),I=1,NPTS)
   24 FORMAT(F7.2,F7.0)
      RETURN
   30 WRITE(IFILE,32) IDATE,CONC,SIZE1,PCEN1,SIZE2,PCEN2
   32 FORMAT(2I6,F5.3,I5,F5.2,I5)
      RETURN
   40 WRITE(IFILE,42) IDATE,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3
     1,SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
   42 FORMAT(I6,F4.2,6(F5.3,A5))
      RETURN
   50 WRITE(IFILE,52) IDATE,NPTS
   52 FORMAT(I6,I4)
      WRITE(IFILE,54)(X(I),Y(I),I=1,NPTS)
   54 FORMAT(F6.0,F6.1)
      RETURN
   70 WRITE(IFILE,22) YEAR,NPTS
      WRITE(IFILE,24)(Y(I),X(I),I=1,NPTS)
      RETURN
   80 WRITE(IFILE,22) YEAR,NPTS
      WRITE(IFILE,84)(X(I),I=1,NPTS)
   84 FORMAT(F7.0)
      RETURN
   86 WRITE(IFILE,22) YEAR,NPTS
      WRITE(IFILE,88)(X(I),I=1,NPTS)
   88 FORMAT(F7.2)
      RETURN
   90 WRITE(IFILE,22) YEAR,NPTS
      IF(SUBLEVL.EQ.5HDAILY) GO TO 92
      DO 91 I=1,NPTS
      DO 91 J=1,24
      WRITE(IFILE,94) HVALUE(I,J)
   91 CONTINUE
      GO TO 100
   92 WRITE(IFILE,94)(X(I),I=1,NPTS)
   94 FORMAT(F5.2)
  100 RETURN
      END
```

SUBROUTINE PROFLAG (IPRO, IPR)

```
      SUBROUTINE PROFLAG(IPRO,IPR)
C
C        GET DATA PROCESSING TYPE
C
C
      IF(IPRO.EQ.10HCUM FREQUE) GO TO 310
      IF(IPRO.EQ.10HHISTOGRAM ) GO TO 320
      IF(IPRO.EQ.10HTHALWEG LE) GO TO 330
      IF(IPRO.EQ.10HCHANGING S) GO TO 340
      IF(IPRO.EQ.10HCUM RAINFA) GO TO 350
      IF(IPRO.EQ.10HREGRESSION) GO TO 360
      IF(IPRO.EQ.10HCORRELATIO) GO TO 370
      IF(IPRO.EQ.10HDISCHARGE ) GO TO 380
      IF(IPRO.EQ.10HSTAGE HYDR) GO TO 390
      IF(IPRO.EQ.10HMIN VALUE ) GO TO 400
      IF(IPRO.EQ.10HMAX VALUE ) GO TO 410
      IF(IPRO.EQ.10HBASIC STAT) GO TO 420
      IF(IPRO.EQ.10HMIN-MAX   ) GO TO 430
      IF(IPRO.EQ.10HFREQUENCY ) GO TO 440
      GO TO 300
  310 IPR=1    $   GO TO 28
  320 IPR=2    $   GO TO 28
  330 IPR=3    $   GO TO 28
  340 IPR=4    $   GO TO 28
  350 IPR=5    $   GO TO 28
  360 IPR=6    $   GO TO 28
  370 IPR=7    $   GO TO 28
  380 IPR=8    $   GO TO 28
  390 IPR=9    $   GO TO 28
  400 IPR=10    $   GO TO 28
  410 IPR=11    $   GO TO 28
  420 IPR=12    $   GO TO 28
  430 IPR=13    $   GO TO 28
  440 IPR=14
   28 RETURN
  300 WRITE(6,305)
  305 FORMAT(1H0,5X,*ERROR IN THE DATA PROCESSING CARD --- OPTION SELECT
     1ED IS NOT AVAILABLE AT*/,6X,*THE PRESENT TIME*)
C
      STOP
      END
```

SUBROUTINE OUTFLAG (IOUTP, IOUT)

```
      SUBROUTINE OUTFLAG(IOUTP,IOUT)
C
C        GET OUTPUT TYPE
C
      IF(IOUTP.EQ.3HLIS) IOUT=1
      IF(IOUTP.EQ.3HPLO) IOUT=2
      IF(IOUTP.EQ.3HDIS) IOUT=3
      IF(IOUTP.EQ.3HSAV) IOUT=4
      IF(IOUTP.EQ.3HPUN) IOUT=5
      RETURN
C
      END
```

SUBROUTINE SWITCH

```
      SUBROUTINE SWITCH
C
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      DO 10 I=1,NPTS
      XX=X(I)
      X(I)=Y(I)
      Y(I)=XX
   10 CONTINUE
      RETURN
      END
```

PROGRAM GETDATA

```
      OVERLAY(TEST,1,0)
      PROGRAM GETDATA
C
C        GET THE DESIRED INFORMATION ELEMENTS
C
       INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,CTYP,
     1PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SNAME,SUBLEVL,XTIT,YTIT
      REAL MWSL,MXEL,MOAVG,MOMIN,MOMAX
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/RIVSTR/STAGE(50),AREA(50)
      COMMON/RIVRES/GAHT(50),NRCP,SPIQ(50),NSCP,QVOL(50),NCCP,NXG(50),
     1NXS(50),NXQ(50)
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
C   ---  GET DATA CATEGORY AND INFORMATION LEVEL  ---
    2 READ(IN,10) ICOM,(IY(I),I=1,8)
   10 FORMAT(A3,7A10,A7)
      IF(ICOM.EQ.3HEND) STOP
      IF(ICOM.NE.3HGET) GO TO 250
    5 CALL SETFLAG
      IF(ISET.EQ.0) GO TO 12
      IF(ICOMD.EQ.3HNO ) GO TO 12
      WRITE(W,1003)
 1003 FORMAT(/,* DATA LOCATION CONSISTS OF THE FOLLOWING TYPES*/,20X*ALL
     1*/,20X,*BASIN*/,20X,*RIVER*/,20X,*SEGMENT OF A RIVER*/,16X,*OR  ST
     2ATION*/,* BY USING THE FOLLOWING COMMANDS*/,10X,*BASIN,(RIVER NAME
     3)  --- FOR A BASIN*/,10X,*RIVER,(RIVER NAME)  ---  FOR A RIVER*/
     4,10X,*SEGMENT,(RIVER NAME),FROM (RM1) TO (RM2)  --- FOR A SEGMENT
     5OF A RIVER*/,10X,*SEGMENT,(RIVER NAME),NODE (X),FROM (XRM) TO (YRM
     6)*/,10X,*STATION,LOCATED (AT/NR LOCATION NAME)  --- FOR A LOCATION
     7*/,10X,*STATION,NUMBER(STATION NUMBER)  --- FOR A GAGING STATION*/
     8,10X,*STATION,COORDINATES(LAT,LONG)  --- FOR A GEOGRAPHIC LOCATION
     9 IN THE */,45X,*BASIN*)
      WRITE(W,1004)
 1004 FORMAT(10X,*STATION,ON(RIVER NAME) AT (RM)  --- FOR A STATION ALON
     1G A RIVER  */,10X,*STATION,NODE(NODE NUMBER) AT (RM)  --- FOR A ST
     2ATION ALONG A RIVER *//,* KEY-IN THE COMMAND WORD "LOCATION" WITH
     3THE APPROPRIATE COMMAND STATEMENTS AS*/,* DESCRIBED ABOVE; EXAMPLE
     4: LOCATION,STATION COORDINATES 33 10 02,90 29 35*/)
   12 READ(IN,10) ICOM,(IY(I),I=1,8)
      IF(ICOM.EQ.3HEND) STOP
      IF(ISET.EQ.0) GO TO 5
      IF(ICOM.EQ.3HLOC) GO TO 14
      IF(ICOM.EQ.3HTIM) GO TO 16
```

```
      GO TO 200
C   ---  GET LOCATION INFORMATION LEVEL  ---
   14 CALL SETLOC
      IF(ILOC.EQ.0) GO TO 12
      IF(LEVEL.EQ.1) GO TO 402
      IF(ICOMD.EQ.3HNO ) GO TO 12
      WRITE(W,1005)
 1005 FORMAT(/,* TIME-PERIOD CAN BE OF THE FOLLOWING TYPES*/,20X,*ALL */
     1,20X,*YEAR (X) */,20X,*DATE (X) */,20X,*FROM YEAR (X) TO (Y)*/,20X
     2,*FROM DATE (X) TO (Y) */,20X,*WATER YEAR */,20X,*FROM WATER YEAR
     3(X) TO (Y)*/,* KEY-IN THE COMMAND WORD "TIME" WITH THE APPROPRIATE
     4 TIME-PERIOD AS DESCRIBED*/,* ABOVE; EXAMPLE: TIME,YEAR 1974*/)
      GO TO 12
C   ---  GET TIME PERIOD  ---
   16 CALL SETTIM
      IF(ITIM.EQ.0) GO TO 12
      IOUTP=3H          $      IPROC=5H            $   IPRO=10H
      DO 15 I=1,4
   15 JCOMWD(I)=10H
      IF(ICOMD.EQ.3HNO ) GO TO 17
      WRITE(W,1007)
 1007 FORMAT(/,* OUTPUT OPTIONS CONSIST OF THE FOLLOWING TYPES*/,20X,
     1*LIST*/,20X,*PLOT*/,20X,*DISPLAY*/,20X,*SAVE*/,* OR IF YOU WANT TO
     2 PROCESS THE DATA INSTEAD, THE FOLLOWING OPTIONS ARE AVAILABLE */
     3,* AT THE PRESENT TIME*/,20X,*COM FREQUENCY */,20X,*HISTOGRAM */
     4,20X,*FREQUENCY ANALYSIS*/,20X,*MIN VALUE*/,20X,*MAX VALUE*
     5/,20X,*MIN-MAX */,20X,*BASIC STATISTICS */,20X,
     6*REGRESSION ANALYSIS*/,20X,*COM RAINFALL */,20X,*STAGE HYDROGRAPH*
     7/,20X,*DISCHARGE HYDROGRAPH */,20X,*THALWEG LEVEL */,15X,*AND  CHA
     8NGING STAGE FOR Q=(A GIVEN DISCHARGE IN CFS)*/)
      WRITE(W,1006)
 1006 FORMAT(* KEY-IN THE SELECTED OUTPUT OPTION OR THE COMMAND WORD "PR
     1OCESS" WITH THE*/,* SELECTED OUTPUT OPTION AND THE DESIRED PROCESS
     2ING OPERATION; */,* EXAMPLE: PROCESS,LIST,MIN VALUE*/)
   17 READ(IN,10) IOUTP,(IY(I),I=1,8)
      IF(IOUTP.EQ.3HPRO) GO TO 18
      IPR=0
      CALL OUTFLAG(IOUTP,IOUT)
      GO TO 400
   18 DECODE(77,19,IY) NX,ICOM ,(ITEMP(I),I=1,7)
   19 FORMAT(A5,A3,6A10,A9)
      CALL OUTFLAG(ICOM,IOUT)
      IF(IOUT.EQ.3) GO TO 24
      IF(IOUT.EQ.5) GO TO 26
      DECODE(69,22,ITEMP) NX,IPRO,(JCOMWD(I),I=1,4),(IZ(J),J=1,2)
   22 FORMAT(A2,A10,4A10,A10,A7)
      GO TO 28
   24 DECODE(69,25,ITEMP) NX,IPRO,(JCOMWD(I),I=1,4),(IZ(J),J=1,2)
   25 FORMAT(A5,A10,4A10,A10,A4)
      GO TO 28
   26 DECODE(69,27,ITEMP) NX,IPRO,(JCOMWD(I),I=1,4),(IZ(J),J=1,2)
   27 FORMAT(A3,A10,4A10,A10,A6)
   28 CALL PROFLAG(IPRO,IPR)
  400 IF(LEVEL.GT.1.AND.LFLAG.EQ.1) GO TO 410
  402 CALL DESCRIP
      REWIND DF
      IF(IRETRIV.EQ.0) GO TO 140
      IF(LEVEL.EQ.1) GO TO 300
C   --- START THE RETRIEVAL PROCESS ---
  410 CONTINUE
```

```
      GO TO(20,30,40,40,60,70,80,85,85,95),ICATEG
   20 CALL RIVSYS
      GO TO 100
   30 CALL RIVHYD
      GO TO 100
   40 CALL RIVSED
      GO TO 100
   60 CALL RIVGEO
      GO TO 100
   70 CALL RIVSTR
      GO TO 100
   80 CALL RIVRES
      GO TO 100
   85 CALL STADISC
      GO TO 100
   95 CALL PRECIP
  100 IF(IRETRIV.EQ.0) GO TO 140
      IF(IOUT.EQ.4) GO TO 300
      ENDFILE WF1
      REWIND WF1
      GO TO 300
  140 WRITE(W,356)
  356 FORMAT(/,* COULD NOT FIND THE DESIRED DATA SET  ---  CHECK LOCATIO
     1N AND TIME COMMAND */,*STATEMENTS*/)
      GO TO 12
  200 WRITE(W,210)
  210 FORMAT(/,* ERROR IN THE CARD HEADING --- TRY AGAIN*/)
      GO TO 12
  250 WRITE(W,255)
  255 FORMAT(* THE FIRST COMMAND CARD SHOULD BE GET THE SPECIFIC DATA EL
     1EMENT*/)
      GO TO 2
  300 CONTINUE
C
      END
```

SUBROUTINE SETFLAG

```
      SUBROUTINE SETFLAG
C
C        SET FLAG FOR DATA CATEGORY AND INFORMATION LEVEL
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,SUBLEVL
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IPETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
C
      ISET=1
      DECODE(77,10,IY) NY,NX,(ITEMP(I),I=1,7),NZ
   10 FORMAT(A1,A3,6A10,A9,A4)
      IF(NX.EQ.3HALL) GO TO 30
      IF(NX.EQ.3HSTA) GO TO 40
      IF(NX.EQ.3HSUS) GO TO 50
      IF(NX.EQ.3HBED) GO TO 60
      IF(NX.EQ.3HCRO) GO TO 70
      IF(NX.EQ.3HCON) GO TO 80
      IF(NX.EQ.3HRES) GO TO 90
      IF(NX.EQ.3HDIS) GO TO 94
      IF(NX.EQ.3HRIV) GO TO 400
      IF(NX.EQ.3HPRE) GO TO 500
      GO TO 200
   30 ICATEG=1
      LEVEL=1
      RETURN
   40 ICATEG=2
      DECODE(69,42,ITEMP)  (IZ(I),I=1,2),NY,(JTEMP(J),J=1,6)
   42 FORMAT(A10,2A3,5A10,A3)
      GO TO 100
   50 ICATEG=3
      DECODE(69,52,ITEMP)(IZ(I),I=1,2),NY,(KTEMP(J),J=1,5)
   52 FORMAT(A10,A6,A3,5A10)
      GO TO 100
   60 ICATEG=4
      DECODE(69,62,ITEMP) NX,NY,(JTEMP(I),I=1,6)
   62 FORMAT(A10,A3,5A10,A6)
      GO TO 100
   70 ICATEG=5
      DECODE(69,72,ITEMP)(IZ(I),I=1,2),NY,(JTEMP(J),J=1,6)
   72 FORMAT(A10,A1,A3,5A10,A5)
      GO TO 100
   80 ICATEG=6
      DECODE(69,82,ITEMP)(IZ(I),I=1,2),NY,(JTEMP(J),J=1,6)
   82 FORMAT(A10,A5,A3,5A10,A1)
      GO TO 100
   90 ICATEG=7
      DECODE(69,92,ITEMP) NX,NY,(JTEMP(I),I=1,6)
   92 FORMAT(A7,A3,5A10,A9)
      GO TO 100
   94 ICATEG = 8
      DECODE(69,96,ITEMP)NX,NY,(JTEMP(I),I=1,6)
   96 FORMAT(A7,A3,5A10,A9)
      GO TO 100
  400 ICATEG=9
      DECODE(69,410,ITEMP) NX,NY,(JTEMP(I),I=1,6)
  410 FORMAT(A9,A3,5A10,A7)
      GO TO 100
  500 ICATEG=10
      DECODE(69,510,ITEMP)(IZ(I),I=1,2),NY,(JTEMP(J), I=1,6)
  510 FORMAT(A10,A1,A3,5A10,A5)
  100 IF(NY.EQ.3HALL) GO TO 110
      IF(NY.EQ.3HDAT) GO TO 130
      IF(NY.EQ.3HSTA) GO TO 105
      GO TO 300
  105 LEVEL=1
```

```
      RETURN
  110 LEVEL=3    $    SUBLEVL=5HALL
      RETURN
  130 LEVEL=2
      IF(ICATEG.EQ.7) GO TO 132
      IF(ICATEG.EQ.10) GO TO 136
      RETURN
  132 DECODE(59,134,JTEMP) NX,SUBLEVL,(KTEMP(I),I=1,5),NY
  134 FORMAT(A2,A5,5A10,A2)
      RETURN
  136 DECODE(55,138,JTEMP) NX,SUBLEVL,(KTEMP(I),I=1,5)
  138 FORMAT(A2,A5,4A10,A8)
      RETURN
  200 WRITE(W,210)
  210 FORMAT(1H0,5X,*COULD NOT IDENTIFY THE DATA CATEGORY TO BE RETRIEVE
     1D --- CHECK THE DATA CATEGORY FORMAT*)
      GO TO 350
  300 WRITE(W,310)
  310 FORMAT(/,5X,*ERROR IN THE ELEMENT CARD --- CHECK INFORMATION TYPE
     1OF THE DATA ELEMENT*)
  350 ISET=0
C
      RETURN
      END
```

SUBROUTINE SETLOC

```
      SUBROUTINE SETLOC
C
C        SET FLAG FOR DATA LOCATION
C
      INTEGER W,WF1,WF2,WF3,STNA
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      NAMST=10H                          $    NUMST=4H
      ILOC=1
      DECODE(77,10,IY) NX,LCATEG,(ITEMP(I),I=1,7)
   10 FORMAT(A6,A3,6A10,A8)
      IF(LCATEG.EQ.3HALL) GO TO 20
      IF(LCATEG.EQ.3HBAS) GO TO 30
      IF(LCATEG.EQ.3HRIV) GO TO 40
      IF(LCATEG.EQ.3HSEG) GO TO 50
      IF(LCATEG.EQ.3HSTA) GO TO 55
      GO TO 100
   20 LFLAG=1
      RETURN
   30 LFLAG=2
      DECODE(68,32,ITEMP) NX,INAME,(JTEMP(I),I=1,6)
   32 FORMAT(A3,A10,5A10,A5)
      RETURN
   40 LFLAG=3
      DECODE(68,32,ITEMP) NX,INAME,(JTEMP(I),I=1,6)
      RETURN
   50 LFLAG=4
      DECODE(68,52,ITEMP) NX,INAME,(JTEMP(I),I=1,6)
   52 FORMAT(A5,A10,5A10,A3)
      PRINT*,"FROM",
      READ*,DIST1
      PRINT*,"TO",
      READ*,DIST2
      RETURN
   55 DECODE(68,56,ITEMP) NX,IST,(JTEMP(J),J=1,6)
   56 FORMAT(A5,A3,6A10)
      IF(IST.EQ.3HLOC) GO TO 60
      IF(IST.EQ.3HNUM) GO TO 70
      IF(IST.EQ.3HCOO) GO TO 80
      IF(IST.EQ.3HON ) GO TO 90
      IF(IST.EQ.3HNOD) GO TO 95
      GO TO 100
   60 LFLAG=5
      DECODE(60,61,JTEMP) NX,NAMST,(KTEMP(K),K=1,5)
   61 FORMAT(A5,A10,4A10,A5)
      RETURN
```

```
   70 LFLAG=6
      DECODE(60,71,JTEMP) NX,NUMST,NY,(KTEMP(K),K=1,5)
   71 FORMAT(A4,A4,A2,5A10)
      RETURN
   80 LFLAG=7
      DECODE(60,81,JTEMP) NX,ILAT,NY,ILONG,(LTEMP(K),K=1,4)
   81 FORMAT(A9,A8,A1,A6,3A10,A4)
      RETURN
   90 LFLAG=8
      DECODE(60,91,JTEMP) INAME,(KTEMP(K),K=1,5)
   91 FORMAT(A10,5A10)
   92 PRINT*,"AT",
      READ*,DISTI
      RETURN
   95 LFLAG=9
      DECODE(60,96,JTEMP) NX,INODE,(KTEMP(K),K=1,5)
   96 FORMAT(A2,A9,4A10,A9)
      GO TO 92
  100 WRITE(W,82)
   82 FORMAT(/,5X,*COULD NOT IDENTIFY THE INPUT STATION LOCATION --- TRY
     1 AGAIN*)
C
      ILOC=0
      RETURN
      END
```

SUBROUTINE SETTIM

```
      SUBROUTINE SETTIM
C
C         SET FLAG FOR TIME-PERIOD
C
      INTEGER W,WF1,WF2,WF3,STNA
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIAG,NYRS,IYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
C
      ITIM=1
      DECODE(77,100,IY) NX,ITIME,(ITEMP(I),I=1,7)
  100 FORMAT(A2,A5,7A10)
      IF(ITIME.EQ.5HALL  ) GO TO 110
      IF(ITIME.EQ.5HYEAR ) GO TO 120
      IF(ITIME.EQ.5HWATER) GO TO 126
      IF(ITIME.EQ.5HDATE ) GO TO 130
      IF(ITIME.EQ.5HFROM ) GO TO 140
      GO TO 170
  110 JFLAG=1
      RETURN
  120 JFLAG=2
      DECODE(70,125,ITEMP) IYR1,NX,(JTEMP(I),I=1,6)
  125 FORMAT(A4,A6,6A10)
      JSTART=IYR1   $   ND1=1
      JEND=4H
      DECODE(4,122,IYR1) JYR1
  122 FORMAT(I4)
      RETURN
  126 JFLAG = 6
      DECODE(70,128,ITEMP)NX,IYR1,(JTEMP(I),I=1,6)
  128 FORMAT(A6,A4,6A10)
      JSTART=IYR1  $  JEND=4H
```

```
      DECODE(4,127,IYR1) NX,IYR
  127 FORMAT(2A2)
      ENCODE(6,129,IDATE1) IYR
  129 FORMAT(* 110*,A2)
      CALL DORDER(IDATE1,ND1,JYR1)
      RETURN
  130 JFLAG=3
      DECODE(70,135,ITEMP) IDATE1,NX,(JTEMP(I),I=1,6)
  135 FORMAT(A6,A4,6A10)
      JSTART=IDATE1
      JEND=6H
      CALL DORDER(JSTART,ND1,JYR1)
      RETURN
  140 DECODE(70,145,ITEMP) IDATE1,NX,(JTEMP(I),I=1,6)
  145 FORMAT(2A5,6A10)
      IF(IDATE1.EQ.5HYEAR ) GO TO 150
      IF(IDATE1.EQ.5HDATE ) GO TO 160
      IF(IDATE1.EQ.5HWATER) GO TO 166
      GO TO 170
  150 JFLAG=4
      DECODE(70,155,ITEMP) NX,IYR1,NY,IYR2,(JTEMP(I),I=1,6)
  155 FORMAT(A5,3A4,5A10,A3)
      JSTART=IYR1    $    ND1=1
      JEND=IYR2
      DECODE(4,122,IYR1) JYR1
      RETURN
  160 JFLAG=5
      DECODE(70,165,ITEMP) NX,IDATE1,NY,IDATE2,(KTEMP(I),I=1,5)
  165 FORMAT(A5,A6,A4,A6,4A10,A9)
      JSTART=IDATE1
      JEND=IDATE2
      CALL DORDER(JSTART,ND1,JYR1)
      CALL DORDER(JEND,ND2,JYR2)
      RETURN
  166 JFLAG=7
      DECODE(70,168,ITEMP)(IZ(I),I=1,2),IYR1,NY,IYR2,(KTEMP(J),J=1,5)
  168 FORMAT(A10,A1,3A4,4A10,A7)
      JSTART=IYR1    $    JEND=IYR2
      DECODE(4,127,IYR1) NX,IYR
      ENCODE(6,129,IDATE1) IYR
      CALL DORDER(IDATE1,ND1,JYR1)
      RETURN
  170 WRITE(W,175)
  175 FORMAT(/,5X,*COULD NOT IDENTIFY THE INPUT TIME-PERIOD --- TRY AGAI
     1N*)
      ITIM=0
C
      RETURN
      END
```

SUBROUTINE DDATE (JDATE, IDD)

```
      SUBROUTINE DDATE(JDATE,IDD)
C
C        CHECK THE DESIRED DATE WITH DATA AVAILABILITY
C
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
C
      DECODE(6,10,JDATE) IDAY,IMON,IYR
   10 FORMAT(3(I2))
      DECODE(6,10,JSTART) IDAY1,IMON1,IYR1
      JYEAR=1900+IYR1
      IF(JEND.EQ.6H      ) GO TO 20
      DECODE(6,10,JEND) IDAY2,IMON2,IYR2
      GO TO 30
   20 IDAY2=IDAY1
      IMON2=IMON1
      IYR2=IYR1
   30 IF(IYR.LT.IYR1.OR.IYR.GT.IYR2) GO TO 40
      IF((IYR.EQ.IYR1).AND.(IMON.LT.IMON1)) GO TO 40
      IF((IYR.EQ.IYR2).AND.(IMON.GT.IMON2)) GO TO 40
      IF((IYR.EQ.IYR1).AND.(IMON.EQ.IMON1).AND.(IDAY.LT.IDAY1)) GO TO 40
      IF((IYR.EQ.IYR2).AND.(IMON.EQ.IMON2).AND.(IDAY.GT.IDAY2)) GO TO 40
      IDD=1
      RETURN
   40 IDD=0
      RETURN
C
      END
```

SUBROUTINE DYEAR (IYR, JYD)

```
      SUBROUTINE DYEAR(IYR,JYD)
C
C        CHECK THE DESIRED YEAR WITH DATA AVAILABILITY
C
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
C
      DECODE(4,10,JSTART) IYR1
   10 FORMAT(I4)
      IF(JEND.EQ.4H    ) GO TO 20
      DECODE(4,10,JEND) IYR2
      GO TO 30
   20 IYR2=IYR1
   30 IF(IYR.LT.IYR1.OR.IYR.GT.IYR2) GO TO 40
      JYD=1
      RETURN
   40 JYD=0
      RETURN
C
      END
```

SUBROUTINE DORDER (IDATE, NDT, IYR)

```
      SUBROUTINE DORDER(IDATE,NDT,IYR)
C
C          GET THE ORDER OF A DATE IN A YEAR
C
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
C
      DECODE(6,10,IDATE) IDAY,IMO,IYR
   10 FORMAT(3I2)
      IYR=1900+IYR
      IF(IMO.GT.1) GO TO 15
      NDT=IDAY   $   RETURN
   15 CALL LEAPYR(IYR,LEAP)
      IF(LEAP.EQ.1) NDAY(2)=29
      IF(LEAP.EQ.0) NDAY(2)=28
      IM1=IMO-1
      NSUM=0
      DO 20 I=1,IM1
   20 NSUM=NSUM+NDAY(I)
      NDT=NSUM+IDAY
      RETURN
      END
```

SUBROUTINE CATEG

```
      SUBROUTINE CATEG
C
C           DATA CATEGORY IDENTIFICATION
C
      INTEGER W,WF1,WF2,WF3,DF,SUBLEVL
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
C
      GO TO(250,10,20,30,40,50,60,70,80,90),ICATEG
   10 NAMCAT=6HRIVHYD    $    GO TO 200
   20 NAMCAT=6HRIVSSD    $    GO TO 200
   30 NAMCAT=6HRIVBED    $    GO TO 200
   40 NAMCAT=6HRIVGEO    $    GO TO 200
   50 NAMCAT=6HRIVSTR    $    GO TO 200
   60 NAMCAT=6HRIVRES    $    GO TO 200
   70 NAMCAT=6HRDISCH    $    GO TO 200
   80 NAMCAT=6HRSTAGE    $    GO TO 200
   90 NAMCAT=6HPRECIP    $    GO TO 200
  200 READ(DF,210) NX,(LTIT(I),I=1,8)
  210 FORMAT(A6,7A10,A4)
      IF(EOF(DF).NE.0) GO TO 300
      IF(NX.NE.NAMCAT) GO TO 200
      IF(LEVEL.EQ.2) GO TO 250
      WRITE(W,12)(LTIT(I),I=1,8)
   12 FORMAT(1H1,//,46X,*YAZOO RIVER SYSTEM DATA BANK*//,46X,7A10,A4/)
  250 RETURN
  300 WRITE(W,310) NAMCAT
  310 FORMAT(1H0,5X,*COULD NOT FIND THE *,A6,* DATA CATEGORY*)
C
      STOP
      END
```

SUBROUTINE NAMNOD (LEVEL, LFLAG, KEY)

```
      SUBROUTINE NAMNOD(LEVEL,LFLAG,KEY)
      INTEGER W,STNA
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
C
      DECODE(78,8,IY) NCROS,NX,DIST,NY,LATI,NZ,LONG,(STNA(I),I=1,3),IX,
     1KEY
    8 FORMAT(I3,A9,F7.2,2(A2,A8),3A10,A3,A6)
      IF(LFLAG.GT.1) GO TO 100
      IF(LEVEL.EQ.2) GO TO 100
      WRITE(W,5)
    5 FORMAT(/,1X,132(1H-))
      WRITE(W,24)
   24 FORMAT(/,16X,*RIVER NAME*,15X,*RIVER-MILE*,10X,*LATITUDE*,10X,
     1*LONGITUDE*,10X,*NUMBER OF CROSS-SECTION*)
      WRITE(W,26) DIST,LATI,LONG,NCROS
   26 FORMAT(/,14X,*MAIN STEM YAZOO*,13X,F7.2,12X,A8,11X,A8,17X,I3)
      WRITE(W,5)
  100 CONTINUE
      RETURN
      END
```

SUBROUTINE STNID (IDENT)

```
      SUBROUTINE STNID(IDENT)
C
C           GAGING STATION IDENTIFICATION
C
      INTEGER STNA,STNO,STFLAG
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISFT,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
C
      IDENT=0    $    STFLAG=LFLAG-4
      GO TO(10,20,30,40,50),STFLAG
   10 IF(STNA(1).EQ.NAMST) IDENT=1
      GO TO 200
   20 IF(STNO.EQ.NUMST) IDENT=1
      GO TO 200
   30 IF((LATI.EQ.ILAT).AND.(LONG.EQ.ILONG)) IDENT=1
      GO TO 200
   40 IF((JNAME.EQ.INAME).AND.(DIST.EQ.DIST1)) IDENT=1
      GO TO 200
   50 IF((NODE.EQ.INODE).AND.(DIST.EQ.DIST1)) IDENT=1
  200 RETURN
C
      END
```

SUBROUTINE DESCRIP

```
      SUBROUTINE DESCRIP
C
C          RETRIEVE DATA STATUS AND GET KEY(S) FOR DATA RETRIEVAL
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,DATE,TYPE,YEAR
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      KOUNT1=KOUNT2=KOUNT3=KOUNT4=KOUNT5=KOUNT6=KOUNT7=KOUNT8=0
      KOUNT9=0
      NKEYS=0
      IRETRIV=0
      DF=11
      REWIND DF
      KOUNT=0
      IF(LFLAG.GT.4.AND.LFLAG.NE.8) GO TO 104
   20 READ(DF,10) ICOM,JNAME,(JTEMP(I),I=1,6),JNODE,NX
   10 FORMAT(A3,A10,5A10,A2,A9,A6)
      IF(EOF(DF).NE.0) GO TO 600
      KOUNT=KOUNT+1
      IF(ICOM.EQ.3HDAT) GO TO 550
      DECODE(9,12,JNODE) JND1
   12 FORMAT(I9)
      IF(LFLAG.EQ.8) GO TO 300
      GO TO(100,200,300,300),LFLAG
C   ---   GET THE STATUS OF ALL DATA CATEGORIES IN THE DATA BASE  ---
  100 IF(ICOM.NE.3HALL) GO TO 20
      IF(LEVEL.GT.1) GO TO 104
      WRITE(W,101)(NNAME(KOUNT,J),J=1,3)
  101 FORMAT(/,45X,*DATA STATUS FOR ALL *,3A10)
      DO 610 J=1,3
```

```
  610 NAME(J)=NNAME(KOUNT,J)
      GO TO 104
C   ---    GET DATA STATUS AND RETRIEVAL KEYS FOR A BASIN   ---
  200 IF(JNAME.NE.INAME) GO TO 20
      DO 620 J=1,3
  620 NAME(J)=NNAME(KOUNT,J)
      IF(LEVEL.GT.1) GO TO 16
      WRITE(W,14)(NNAME(KOUNT,J),J=1,3)
   14 FORMAT(/,45X,*DATA STATUS FOR BASIN OF THE *,3A10)
   16 IF(ICOM.EQ.3HRIV) GO TO 210
      IF(ICOM.EQ.3HTR1) GO TO 220
      IF(ICOM.EQ.3HTR2) GO TO 230
      IF(ICOM.EQ.3HTR3) GO TO 240
      GO TO 20
  210 JND2=JND1+999999
      GO TO 104
  220 JND2=JND1+9999
      GO TO 104
  230 JND2=JND1+99
      GO TO 104
  240 JND2=JND1
  104 READ(DF,106) ICOM,(ITEMP(I),I=1,7),NODE,KEY
  106 FORMAT(A3,6A10,A2,A9,A6)
      IF(EOF(DF).NE.0) GO TO 600
      IF(ICOM.EQ.3HALL) GO TO 104
      IF(ICOM.EQ.3HRIV.OR.ICOM.EQ.3HTR1.OR.ICOM.EQ.3HTR2.OR.ICOM.EQ.
     13HTR3) GO TO 104
      IF(ICOM.EQ.3HDAT) GO TO 104
      IF(LFLAG.EQ.1) GO TO 108
      IF(LFLAG.EQ.2) GO TO 109
      IF(LFLAG.GT.4) GO TO 107
      IF(NODE.NE.JNODE) GO TO 104
      GO TO 108
C   ---    GET DATA STATUS AND RETRIEVAL KEY(S) FOR A STATION   ---
  107 DECODE(62,111,ITEMP)(STNA(J),J=1,3),STNO,DIST,LATI,LONG,TYPE,NYRS
  111 FORMAT(2A10,A5,A4,F7.2,2A8,A7,I3)
      CALL STNID(IDENT)
      IF(IDENT.EQ.0) GO TO 104
      IF(KOUNT.GT.0) GO TO 108
      DO 103 I=1,NSUB
      IF(NODE.NE.NNODE(I)) GO TO 103
      KI=I
      GO TO 105
  103 CONTINUE
  105 IF(LEVEL.GT.1) GO TO 102
      IF(ICATEG.NE.10) GO TO 624
      WRITE(W,622)(STNA(K),K=1,3)
  622 FORMAT(/,45X,*DATA STATUS *,2A10,A5)
      GO TO 626
  624 WRITE(W,112)(NNAME(KI,J),J=1,3),(STNA(K),K=1,3)
  112 FORMAT(/,45X,*DATA STATUS FOR *,3A10,2A10,A5)
  626 KOUNT=KOUNT+1
  102 DO 630 J=1,3
  630 NAME(J)=NNAME(KI,J)
      GO TO 108
  109 DECODE(9,12,NODE) INOD
      IF(INOD.LT.JND1.OR.INOD.GT.JND2) GO TO 104
  108 CONTINUE
C   ---    RETRIEVAL ACCORDING TO SELECTED DATA CATEGORY   ---
      GO TO(110,120,130,140,150,160,170,180,185,190),ICATEG
```

```
C   ---   GET DATA STATUS OF ALL CATEGORIES IN THE BASIN   ---
  110 IF(ICOM.EQ.3HDSD) GO TO 30
      IF(ICOM.EQ.3HDDI) GO TO 35
      IF(ICOM.EQ.3HDSG) GO TO 40
      IF(ICOM.EQ.3HDSS) GO TO 45
      IF(ICOM.EQ.3HDBD) GO TO 50
      IF(ICOM.EQ.3HDCR) GO TO 55
      IF(ICOM.EQ.3HDPR) GO TO 60
      IF(ICOM.EQ.3HDRE) GO TO 65
      IF(ICOM.EQ.3HDST) GO TO 70
      GO TO 104
   30 IF(KOUNT1.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,32)
   32 FORMAT(/,50X,*STAGE-DISCHARGE DATA STATUS*)
      KOUNT1=KOUNT1+1
   34 CALL SUBSYS
      GO TO 104
   35 IF(KOUNT2.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,36)
   36 FORMAT(/,50X,*RIVER DISCHARGE DATA STATUS*)
      KOUNT2=KOUNT2+1
      GO TO 34
   40 IF(KOUNT3.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,42)
   42 FORMAT(/,50X,*RIVER STAGE DATA STATUS*)
      KOUNT3=KOUNT3+1
      GO TO 34
   45 IF(KOUNT4.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,46)
   46 FORMAT(/,50X,*SUSPENDED SEDIMENT DATA STATUS*)
      KOUNT4=KOUNT4+1
      GO TO 34
   50 IF(KOUNT5.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,52)
   52 FORMAT(/,50X,*BED MATERIAL DATA STATUS*)
      KOUNT5=KOUNT5+1
      GO TO 34
   55 IF(KOUNT6.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,56)
   56 FORMAT(/,50X,*CHANNEL CROSS-SECTION DATA STATUS*)
      KOUNT6=KOUNT6+1
      GO TO 34
   60 IF(KOUNT7.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,62)
   62 FORMAT(/,50X,*PRECIPITATION DATA STATUS*)
      KOUNT7=KOUNT7+1
```

```
      GO TO 34
   65 IF(KOUNT8.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,66)
   66 FORMAT(/,50X,*RESERVOIR DATA STATUS*)
      KOUNT8=KOUNT8+1
      GO TO 34
   70 IF(KOUNT9.GT.0) GO TO 34
      IF(LEVEL.GT.1) GO TO 34
      CALL HEADING
      WRITE(W,72)
   72 FORMAT(/,50X,*RIVER CONTROL STRUCTURE DATA STATUS*)
      KOUNT9=KOUNT9+1
      GO TO 34
  120 IF(ICOM.NE.3HDSD) GO TO 192
      GO TO 30
  130 IF(ICOM.NE.3HDSS) GO TO 192
      GO TO 45
  140 IF(ICOM.NE.3HDBD) GO TO 192
      GO TO 50
  150 IF(ICOM.NE.3HDCR) GO TO 192
      GO TO 55
  160 IF(ICOM.NE.3HDST) GO TO 192
      GO TO 70
  170 IF(ICOM.NE.3HDRE) GO TO 192
      GO TO 65
  180 IF(ICOM.NE.3HDDI) GO TO 192
      GO TO 35
  185 IF(ICOM.NE.3HDSG) GO TO 192
      GO TO 40
  190 IF(ICOM.NE.3HDPR) GO TO 192
      GO TO 60
  192 IF(IRETRIV.EQ.1) GO TO 600
      GO TO 104
C   ---    GET DATA STATUS AND RETRIEVAL KEYS FOR A RIVER   ---
  300 IF(JNAME.NE.INAME) GO TO 20
      DO 640 J=1,3
  640 NAME(J)=NNAME(KOUNT,J)
      IF(LEVEL.GT.1) GO TO 104
      IF(LFLAG.EQ.3) GO TO 104
      IF(LFLAG.EQ.4) GO TO 400
      WRITE(W,301)(NNAME(KOUNT,J),J=1,3)
  301 FORMAT(/,45X,*DATA STATUS FOR *,3A10)
      GO TO 104
C   ---    GET DATA STATUS AND RETRIEVAL KEYS FOR A SEGMENT OF A RIVER
  400 WRITE(W,401)(NNAME(KOUNT,J),J=1,3)
  401 FORMAT(/,45X,*DATA STATUS FOR SEGMENT OF *,3A10)
      GO TO 104
  550 WRITE(W,552)
  552 FORMAT(1H0,5X,*ERROR IN SUBSYSTEM LOCATION NAME*)
  600 IF(LEVEL.GT.1) GO TO 750
      WRITE(W,700)
  700 FORMAT(/,1X,130(1H-),//)
  750 RETURN
      END
```

SUBROUTINE HEADING

```
      SUBROUTINE HEADING
C
      INTEGER W,WF1,WF2,WF3,DF
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
C
      WRITE(W,10)
   10 FORMAT(/,1X,130(1H-))
      IF(ICATEG.EQ.10) GO TO 35
      IF(ICATEG.EQ.5) GO TO 25
      IF(ICATEG.EQ.6.OR.ICATEG.EQ.7) GO TO 30
      WRITE(W,15)
   15 FORMAT(/,12X,*LOCATION*,12X,*STA.NO.*,4X,*NODE*,4X,*RIVER MILE*,
     14X,*LATITUDE*,4X,*LONGITUDE*,4X,*GAGE-0(FT(MSL))*,4X,*NO.YEARS*,
     25X,*TAPE I.D.*)
      GO TO 50
   25 WRITE(W,26)
   26 FORMAT(/,12X,*LOCATION*,12X,*STA.NO.*,4X,*NODE*,4X,*RIVER MILE*,
     14X,*LATITUDE*,4X,*LONGITUDE*,4X,*GAGE-0(FT(MSL))*,4X,*NO.DATES*,
     25X,*TAPE I.D.*)
      GO TO 50
   30 WRITE(W,32)
   32 FORMAT(/,12X,*LOCATION*,12X,*STA.NO.*,4X,*NODE*,4X,*RIVER MILE*,
     14X,*LATITUDE*,4X,*LONGITUDE*,8X,*TYPE*,11X,*NO.YEARS*,5X,*TAPE I.D
     2*)
      GO TO 50
   35 WRITE(W,36)
   36 FORMAT(/,12X,*LOCATION*,12X,*STA.NO.*,12X,*LATITUDE*,4X,
     1*LONGITUDE*,12X,*MEAN ELEV(FT)*,8X,*NO.YEARS*,5X,*TAPE I.D.*)
   50 WRITE(W,10)
C
      RETURN
      END
```

SUBROUTINE SUBSYS

```
      SUBROUTINE SUBSYS
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,DATE,YEAR,TYPE
      COMMON/ELFMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXO,NYRS,TYPE,IYEAR(100),GELEV,NCP
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      IF(LFLAG.GT.4) GO TO 10
      DECODE(62,20,ITEMP)(STNA(I),I=1,3),STNO,DIST,LATI,LONG,TYPE,NYRS
   20 FORMAT(2A10,A5,A4,F7.2,2A8,A7,I3)
   10 IF(ICATEG.EQ.5) GO TO 22
      READ(DF,30) ITAPE,(IYEAR(J),J=1,NYRS)
   30 FORMAT(A8,18I4/(8X,18I4))
      IF(ICATEG.EQ.10) GO TO 38
      IF(ICATEG.EQ.7) GO TO 35
      GO TO 32
   22 NCR=NYRS
      READ(DF,31) ITAPE,(DATE(I),I=1,NCR)
   31 FORMAT(A8,12I6/(8X,12I6))
   32 IF(LFLAG.NE.4) GO TO 35
      IF(DIST.LT.DIST1.OR.DIST.GT.DIST2) RETURN
   35 DECODE(9,36,NODE) INOD
   36 FORMAT(I9)
   38 IRETRIV=1
      IF(LEVEL.GT.1) GO TO 56
      IF(ICATEG.EQ.10) GO TO 42
      WRITE(W,40)(STNA(I),I=1,3),STNO,INOD,DIST,LATI,LONG,TYPE,NYRS
     1,ITAPE
   40 FORMAT(/,5X,2A10,A5,3X,A4,3X,I9,3X,F7.2,5X,A8,5X,A8,8X,A7,10X,I3,
     110X,A8)
      GO TO 46
   42 DECODE(7,33,TYPE) GELEV
   33 FORMAT(F7.2)
      WRITE(W,44)(STNA(I),I=1,3),STNO,LATI,LONG,GELEV,NYRS,ITAPE
   44 FORMAT(/,5X,2A10,A5,3X,A4,14X,A8,5X,A8,13X,F7.2,15X,I3,10X,A8)
   46 IF(ICATEG.EQ.6.OR.ICATEG.EQ.7) GO TO 55
      IF(ICATEG.EQ.5) GO TO 52
      WRITE(W,50)(IYEAR(J),J=1,NYRS)
   50 FORMAT(16(4X,I4))
      GO TO 55
   52 WRITE(W,51)(DATE(J),J=1,NCP)
   51 FORMAT(16(2X,I6))
   55 IF(LEVEL.EQ.1) RETURN
   56 IF(LFLAG.EQ.1) RETURN
      NKEYS=NKEYS+1
      IKEY(NKEYS)=KEY
C
      RETURN
      END
```

SUBROUTINE RIVSYS

```
      SUBROUTINE RIVSYS
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,CTYP,
     1PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SNAME,SUBLEVL,XTIT,YTIT
      REAL MWSL,MXEL,MOAVG,MOMIN,MOMAX
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),I7(2),NX,NY,N7,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVSED/CONC,XSEC1,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/RIVSTR/STAGE(50),AREA(50)
      COMMON/RIVRES/GAHT(50),NRCP,SPIQ(50),NSCP,QVOL(50),NCCP,NXG(50),
     1NXS(50),NXQ(50)
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      CALL RIVHYD
      CALL RIVSED
      CALL RIVGEO
      CALL RIVSTR
      CALL RIVRES
      CALL STADISC
      CALL PRECIP
C
      RETURN
      END
```

SUBROUTINE RIVHYD

```
      SUBROUTINE RIVHYD
C
C         RIVER HYDRAULICS DATA MANIPULATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
C   ---    CHECK DATA CATEGORY    ---
      DF=12
      IRETRIV=1
      CALL CATEG
      IF(LFLAG.GT.1) GO TO 350
C   --- RETRIEVE ALL HYDRAULICS DATA ---
   50 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) RETURN
      IF(IX.EQ.3HIST) GO TO 25
      IF(IX.EQ.3HHST) GO TO 35
      RETURN
C   ---  READ AND WRITE STATION INFORMATION  ---
   25 DECODE(77,24,IY) IX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1GAGO,TYPE,NYRS
   24 FORMAT(A3,3A10,A4,F7.2,2A9,F7.3,A5,I3)
      IF(LEVEL.EQ.2) GO TO 50
      SNAME(1)=NAME(1)     $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)     $    SNAME(4)=STNA(2)
      CALL WSTD
      GO TO 50
C   ---  READ AND WRITE HYDRAULIC DATA  ---
   35 DECODE(77,30,IY) IX,YEAR,NPTS,(ITEMP(I),I=1,7)
   30 FORMAT(A3,I4,I5,6A10,A5)
      READ(DF,34)(DATE(I),STAG(I),DISC(I),I=1,NPTS)
   34 FORMAT(4(I6,F7.2,F7.0))
      IF(JFLAG.EQ.1) GO TO 31
      CALL DYEAR(YEAR,JYD)
      IF(JYD.EQ.0) GO TO 50
   31 CALL HYDDAT(W,LEVEL)
      GO TO 50
C   ---    RETRIEVE HYDRAULIC DATA AT A GAGING STATION    ---
  350 CALL HYDSTA
      IF(NRECORD.EQ.0) IRETRIV=0
      RETURN
      END
```

SUBROUTINE HYDSTA

```
      SUBROUTINE HYDSTA
C
C           GET HYDRAULICS DATA FOR ONE PARTICULAR GAGING STATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      NRECORD=0    $    KKEY=0
  500 NYEAR=0
  356 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) GO TO 16
      IF(IX.EQ.3HIST) GO TO 360
      IF(IX.EQ.3HHST) GO TO 380
   16 REWIND DF
      RETURN
C   ---  READ AND WRITE STATION INFORMATION  ---
  360 DECODE(77,362,IY) NX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1GAGO,TYPE,NYRS
  362 FORMAT(A3,3A10,A4,F7.2,2A9,F7.3,A5,I3)
      ENCODE(6,363,KEY) NX
  363 FORMAT(*IST*,A3)
      CALL CHKEY(IDENT)
      IF(IDENT.EQ.0) GO TO 356
      KKEY=KKEY+1
      SNAME(1)=NAME(1)    $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)    $    SNAME(4)=STNA(2)
      MTIT(5)=SNAME(1)    $    MTIT(6)=SNAME(2)
      MTIT(7)=SNAME(3)    $    MTIT(8)=SNAME(4)
      XTIT=10HDISCHARGE      $    YTIT=10HGAGE STAGE
      XUNIT=6HIN CFS    $    YUNIT=6HIN FT.    $    TUNIT=4HYEAR
      IF(LEVEL.EQ.2) GO TO 356
      CALL WSTD
      GO TO 356
C   ---  READ AND WRITE STAGE-DISCHARGE DATA  ---
  380 DECODE(77,30,IY) IX,YEAR,NPTS,(ITEMP(I),I=1,7)
   30 FORMAT(A3,I4,I5,6A10,A5)
      READ(DF,34) (DATE(I),STAG(I),DISC(I),I=1,NPTS)
   34 FORMAT(4(I6,F7.2,F7.0))
      IF(IDENT.EQ.1) GO TO 400
      IF(NRECORD.EQ.0) GO TO 356
      GO TO 450
  400 CONTINUE
C   ---    RETRIEVE DATA ACCORDING TO THE SELECTED TIME OPTION    ---
      GO TO(382,384,386,392,394),JFLAG
C   ---    RETRIEVE ALL DATA AT A STATION    ---
  382 IF(IPR.NE.0) GO TO 41
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 41
```

```
      CALL HYDDAT(W,LEVEL)
   41 NRECORD=NRECORD+1
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      NYEAR=NYEAR+1
      IF(JFLAG.EQ.2) GO TO 450
      GO TO 356
C   ---    RETRIEVE DATA FOR A PARTICULAR YEAR    ---
  384 DECODE(4,385,JSTART) JYEAR
  385 FORMAT(I4)
      IF(YEAR.NE.JYEAR) GO TO 356
      GO TO 382
C   ---    RETRIEVE DATA FOR A PARTICULAR DATE    ---
  386 DO 390 I=1,NPTS
      DECODE(6,391,JSTART) JDATE
  391 FORMAT(I6)
      IF(DATE(I).NE.JDATE) GO TO 390
      NRECORD=NRECORD+1
      IF(LEVEL.EQ.2) GO TO 387
      WRITE(W,388)
  388 FORMAT(31X,*DATE*,3X,*GAGE STAGE*,2X,*DISCH.*/)
  387 WRITE(W,389) DATE(I),STAG(I),DISC(I)
  389 FORMAT(30X,I6,3X,F7.2,3X,F7.0)
      GO TO 450
  390 CONTINUE
      GO TO 356
C   ---    RETRIEVE DATA FOR A NUMBER OF YEARS    ---
  392 CALL DYEAR(YEAR,JYD)
      IF(JYD.EQ.0) GO TO 356
      GO TO 382
C   ---    RETRIEVE DATA FOR A PARTICULAR TIME-PERIOD    ---
  394 KOUNT=0
      DO 398 I=1,NPTS
      JDD=DATE(I)
      ENCODE(6,395,JDATE) JDD
  395 FORMAT(I6)
      CALL DDATE(JDATE,IDD)
      IF(IDD.EQ.0) GO TO 398
      NRECORD=NRECORD+1
      IF(LEVEL.EQ.2) GO TO 396
      KOUNT=KOUNT+1
      IF(KOUNT.GT.1) GO TO 396
      WRITE(W,388)
  396 WRITE(W,389) DATE(I),STAG(I),DISC(I)
  398 CONTINUE
      GO TO 356
  450 IF(KKEY.EQ.NKEYS) GO TO 16
      GO TO 500
C
      END
```

SUBROUTINE CHKEY (IDENT)

```
      SUBROUTINE CHKEY(IDENT)
C
C          CHECK RETRIEVAL KEY
C
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      IDENT=1
      DO 10 I=1,NKEYS
      KI=IKEY(I)
      IF(KI.EQ.KEY) GO TO 20
   10 CONTINUE
      IDENT=0
   20 RETURN
C
      END
```

SUBROUTINE WSTD

```
      SUBROUTINE WSTD
C
C          WRITE STAGE / DISCHARGE DATA
C
      INTEGER W,SNAME,STNO,TYPE
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
C
C
      WRITE(W,5)
    5 FORMAT(/,1X,132(1H-))
      WRITE(W,22)
   22 FORMAT(/,16X,*STATION NAME*,14X,*STAIN NO*,2X,*DIST FR NODE*,
     12X,*LATITUDE*,2X,*LONGITUDE*,2X,*GAGE ZERO(MSL)*,3X,*DATA TYPE*,
     24X,*NO YEARS*)
      WRITE(W,26)(SNAME(I),I=1,4),STNO,DIST,LATI,LONG,GAGO,TYPE,NYRS
   26 FORMAT(/,2X,4A10,2X,A4,4X,F7.2,*(MI)*,1X,2(2X,A9),3X,F7.3,
     1*(FT)*,6X,A5,8X,I3)
      WRITE(W,5)
C
      RETURN
      END
```

SUBROUTINE HYDDAT (W, LEVEL)

```
      SUBROUTINE HYDDAT(W,LEVEL)
C
C          WRITE STAGE-DISCHARGE DATA
C
      INTEGER W
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
C
      IF(LEVEL.EQ.2) GO TO 29
      WRITE(W,28)
   28 FORMAT(/,10X,*YEAR*,10X,*NO. DATA POINTS*)
   29 WRITE(W,32) YEAR,NPTS
   32 FORMAT(10X,I4,15X,I5/)
      IF(LEVEL.EQ.2) GO TO 37
      WRITE(W,36)
   36 FORMAT(2X,*DATE*,3X,*GAGE STAGE*,2X,*DISCH.*,9X,*DATE*,3X,*GAGE ST
     1AGE*,2X,*DISCH.*,9X,*DATE*,3X,*GAGE STAGE*,2X,*DISCH.*,9X,*DATE*,
     23X,*GAGE STAGE*,2X,*DISCH.*,/,12X,*(FT)*,5X,*(CFS)*,20X,*(FT)*,5X,
     3*(CFS)*,20X,*(FT)*,5X,*(CFS)*,20X,*(FT)*,5X,*(CFS)*/)
   37 WRITE(W,38)(DATE(I),STAG(I),DISC(I),I=1,NPTS)
   38 FORMAT(4(1X,I6,3X,F7.2,3X,F7.0,7X))
      WRITE(W,39)
   39 FORMAT(//)
C
      RETURN
      END
```

SUBROUTINE RIVSED

```
      SUBROUTINE RIVSED
C
C           RIVER SEDIMENT DATA MANIPULATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,SNAME,
     1SUBLEVL,PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,XTIT,YTIT
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXO,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      DF=15
      IRETRIV=1
      CALL CATEG
      IF(LFLAG.GT.1) GO TO 350
      IDENT=1    $    NRECORD=0
   50 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) GO TO 400
      IF(ICATEG.EQ.4) GO TO 15
      IF(IX.EQ.3HJST) GO TO 25
      IF(IX.EQ.3HSST) GO TO 35
      GO TO 400
   15 IF(IX.EQ.3HLST) GO TO 25
      IF(IX.EQ.3HRST) GO TO 35
      GO TO 400
C  ---    READ AND WRITE STATION INFORMATION   ---
   25 DECODE(77,24,IY) IX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1GAGO,TYPE,NYRS
   24 FORMAT(A3,3A10,A4,F7.2,2A9,F7.3,A5,I3)
      IF(LEVEL.EQ.2) GO TO 50
      SNAME(1)=NAME(1)    $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)    $    SNAME(4)=STNA(2)
      CALL WSTD
      GO TO 50
C  ---    READ AND WRITE SUSPENDED SEDIMENT DATA   ---
   35 DECODE(77,30,IY) IX,YEAR,NPTS,(ITEMP(I),I=1,7)
   30 FORMAT(A3,I4,I5,6A10,A5)
      IF(IOUT.EQ.1) GO TO 26
      IF(LEVEL.EQ.2) GO TO 33
   26 WRITE(W,28)
   28 FORMAT(/,10X,*YEAR*,10X,*NO. DATA POINTS*)
   33 WRITE(W,32) YEAR,NPTS
   32 FORMAT(10X,I4,15X,I5/)
      CALL SEDDAT(W,DF,IDENT,IOUT,NPTS,ICATEG,LEVEL,WF1,IDATE,NRECORD)
      GO TO 50
C  ---    RETRIEVE SEDIMENT DATA AT A GAGING STATION     ---
  350 CALL SEDSTA
  400 IF(NRECORD.EQ.0) IRETRIV=0
      REWIND DF
C
      RETURN
      END
```

SUBROUTINE SEDSTA

```
      SUBROUTINE SEDSTA
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,SNAME,
     1SUBLEVL,PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,XTIT,YTIT
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(6),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,IUNIT,LINES,IOPT
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
       NRECORD=0    $    KKEY=0
  500 NYEAR=0
  356 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) GO TO 16
      IF(ICATEG.EQ.4) GO TO 300
      IF(IX.EQ.3HJST) GO TO 360
      IF(IX.EQ.3HSST) GO TO 380
      GO TO 16
  300 IF(IX.EQ.3HLST) GO TO 360
      IF(IX.EQ.3HBST) GO TO 380
   16 RETURN
C   ---    READ AND WRITE STATION INFORMATION    ---
  360 DECODE(77,362,IY) NX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1GAGO,TYPE,NYRS
  362 FORMAT(A3,3A10,A4,F7.2,2A9,F7.3,A5,I3)
      IF(ICATEG.EQ.4) GO TO 40
      ENCODE(6,363,KEY) NX
  363 FORMAT(*JST*,A3)
      GO TO 365
   40 ENCODE(6,42,KEY) NX
   42 FORMAT(*LST*,A3)
  365 CALL CHKEY(IDENT)
      IF(IDENT.EQ.0) GO TO 356
      KKEY=KKEY+1
      SNAME(1)=NAME(1)     $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)     $    SNAME(4)=STNA(2)
      MTIT(5)=SNAME(1)     $    MTIT(6)=SNAME(2)
      MTIT(7)=SNAME(3)     $    MTIT(8)=SNAME(4)
      IF(ICATEG.EQ.4) GO TO 43
      XTIT=10HSIZE                   $     YTIT=10HCONCENTRAT
      GO TO 44
   43 XTIT=10HSIZE                  $      YTIT=10HPERCENTAGE
   44 IF(LEVEL.EQ.2) GO TO 356
      CALL WSTD
      GO TO 356
C   ---    READ AND WRITE SEDIMENT DATA    ---
  380 DECODE(77,30,IY) IX,YEAR,NPTS,(ITEMP(I),I=1,7)
   30 FORMAT(A3,I4,I5,6A10,A5)
      IF(IDENT.EQ.0) GO TO 510
      GO TO(382,384,400,400,400),JFLAG
```

```
382 IF(IOUT.EQ.1) GO TO 35
    IF(LEVEL.EQ.2) GO TO 37
 35 WRITE(W,28)
 28 FORMAT(/,10X,*YEAR*,10X,*NO. DATA POINTS*)
 37 WRITE(W,32) YEAR,NPTS
 32 FORMAT(10X,I4,15X,I5/)
510 CALL SEDDAT(W,DF,IDENT,IOUT,NPTS,ICATEG,LEVEL,WF1,IDATE,NRECORD)
    NYEAR=NYEAR+1
    IF(JFLAG.EQ.2) GO TO 450
    GO TO 356
384 DECODE(4,385,JSTART) JYEAR
385 FORMAT(I4)
    IF(YEAR.NE.JYEAR) GO TO 356
    GO TO 382
450 IF(KKEY.EQ.NKEYS) GO TO 16
    GO TO 500
400 CONTINUE
    GO TO 16
    END
```

SUBROUTINE SEDDAT (W, DF, IDENT, IOUT, NPTS, ICATEG, LEVEL, WF1, IDATE, NRECORD)

```
      SUBROUTINE SEDDAT(W,DF,IDENT,IOUT,NPTS,ICATEG,LEVEL,WF1,IDATE,
     1NRECORD)
C
C           READ AND WRITE SEDIMENT DATA
C
      INTEGER W,WF1,DF,CONC,PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6
      COMMON/RIVSED/CONC,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
C
C
      IF(ICATEG.EQ.4) GO TO 50
      IF(IDENT.EQ.0) GO TO 131
      IF(IOUT.EQ.1) GO TO 150
      IF(LEVEL.EQ.2) GO TO 131
  150 WRITE(W,130)
  130 FORMAT(/,20X,*DATE*,12X,*CONCENTRATION*,10X,*SIZE*,5X,*CUM CONCN*
     1,10X,*SIZE*,5X,*CUM CONCN*/,39X,*(PPM)*,15X,*(MM)*,7X,*(PPM)*,
     212X,*(MM)*,7X,*(PPM)*/)
  131 DO 136 I=1,NPTS
      READ(DF,132) IDATE,CONC,SIZE1,PCEN1,SIZE2,PCEN2
  132 FORMAT(2I6,3X,2(F5.0,I5),45X)
      IF(IDENT.EQ.0) GO TO 136
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 135
      WRITE(W,134) IDATE,CONC,SIZE1,PCEN1,SIZE2,PCEN2
  134 FORMAT(19X,I6,12X,I6,15X,F5.3,6X,I5,12X,F5.2,6X,I5)
  135 NRECORD=NRECORD+1
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
  136 CONTINUE
      GO TO 60
   50 IF(IDENT.EQ.0) GO TO 54
      IF(IOUT.EQ.1) GO TO 65
      IF(LEVEL.EQ.2) GO TO 54
   65 WRITE(W,52)
   52 FORMAT(/,7X,*DATE*,5X,*X-SECT LOC*,5X,*SIZE*,3X,*PCENT*,5X,
     1*SIZE*,3X,*PCENT*,5X,*SIZE*,3X,*PCENT*,5X,*SIZE*,3X,*PCENT*,
     25X,*SIZE*,3X,*PCENT*,5X,*SIZE*,3X,*PCENT*/,19X,*(FT)*,8X,*(MM)*,
     313X,*(MM)*,13X,*(MM)*,13X,*(MM)*,13X,*(MM)*,13X,*(MM)*/)
   54 DO 56 I=1,NPTS
      READ(DF,55)IDATE,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
   55 FORMAT(I6,F4.2,6(F5.3,A5))
      IF(IDENT.EQ.0) GO TO 56
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 58
      WRITE(W,57)IDATE,XSECT,SIZE1,PCEN1,SIZE2,PCEN2,SIZE3,PCEN3,
     1SIZE4,PCEN4,SIZE5,PCEN5,SIZE6,PCEN6
   57 FORMAT(6X,I6,6X,F4.2,4X,6(4X,F5.3,3X,A5))
   58 NRECORD=NRECORD+1
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
   56 CONTINUE
C
   60 RETURN
      END
```

SUBROUTINE RIVGEO

```
      SUBROUTINE RIVGEO
C
C     RIVER CROSS-SECTION DATA MANIPULATION
C
      INTEGER W.WF1.WF2,WF3.DF.STNA.STNO.TYPE,YEAR.DATE.SNAME,SURLEVL.
     1XTIT.YTIT
      COMMON/INFORM/ICOM.JTEMP(6).KTEMP(5).LTEMP(4).NTEMP(2).ICOMD.IFILE
      COMMON/ELEMT/ICATEG,LEVEL.SURLEVL.JCOMWD(4).NWORD.DF
      COMMON/LOCATE/LFLAG,INODE.INAME.NAMST,NUMST.ILAT.ILONG,DIST1.DIST2
      COMMON/TIME/JFLAG.JSTART.JEND.JNAME.ND1.JYR1.ND2.JYR2
      COMMON/WORK/IN.W.WF1.WF2.WF3.ISET.ILOC.ITIM.IRETRIV.IPRO.IOUTP.
     1NODE.STNO,DIST.LATI.LONG.GAGO.NAME(4).STNA(3).SNAME(4).LTIT(8).
     2IY(8).ITEMP(7).IZ(2).NX.NY.NZ.FIXQ.NYRS.TYPE.JYEAR(100).GELEV.NCR
     3.IEND.LFAP.TEMP(366)
      COMMON/OUTPUT/IOUT.X(366).Y(366).IPOSX(366.3).IPOSY(366.3).MTIT(8)
     1.XTIT.YTIT.YEAR.NPTS.IA(10).NYEAR.NDATA.IDATE.DATF(150).
     2NRECORD.CA(150).CB(150).IPR.IPROC.FDATE(150).ITEK
      COMMON/PLOT/MO(12).XUNIT.YUNIT.TUNIT.LINES.IOPT
      COMMON/RIVCRO/XDIS(100).YELEV(100).NCROS
      COMMON/SUB/NNAME(50.3).NNODE(50).NSUB.IKEY(300).NKEYS.KEY
C
C   ---   CHECK DATA CATEGORY   ---
      DF=16
      IRETRIV=1
      CALL CATEG
      IF(LFLAG.GT.1) GO TO 350
      NRECORD=0
C   --- RETRIEVE ALL RIVER GEOMETRY DATA ---
   50 READ(DF.14) IX.(IY(I).I=1.8)
   14 FORMAT(A2.7A10.A8)
      IF(EOF(DF).NE.0) GO TO 400
      IF(IX.EQ.2HRI) GO TO 50
      IF(IX.EQ.2HPR) GO TO 15
      IF(IX.EQ.2HCS) GO TO 35
      GO TO 400
   15 CALL NAMNOD(LEVEL,LFLAG.KEY)
      GO TO 50
   35 DECODE(78.30.IY) NPTS.NX1.IMO.NX2.IDAY.NX3.IYR.NX4.(KTEMP(I).I=1.5
     1).NX5
   30 FORMAT(I3.A10.I2.A1.I2.A1.I2.A2.4A10.A7.A8)
      READ(DF.34) (XDIS(I).YELEV(I).I=1.NPTS)
   34 FORMAT((8X.6(F6.0.F6.1)))
      YEAR=1900+IYR
      IF(JFLAG.EQ.1) GO TO 31
      CALL DYEAR(YEAR.JYD)
      IF(JYD.EQ.0) GO TO 50
   31 ENCODE(6.29.NZ) IDAY.IMO.IYR
   29 FORMAT(3I2)
      DECODE(6.27.NZ) IDATE
   27 FORMAT(I6)
      CALL GEODAT(W,LEVEL.IOUT.IPR.NRECORD.IDATE.NPTS)
      DO 40 I=1.NPTS
      X(I)=XDIS(I)
      Y(I)=YELEV(I)
   40 CONTINUE
      CALL DWRITE(WF1.ICATEG,SURLEVL)
      GO TO 50
C   --- RETRIEVE RIVER GEOMETRY DATA AT ONE PARTICULAR STATION ---
  350 CALL GEOSTA
  400 IF(NRECORD.EQ.0) IRETRIV=0
      REWIND DF
C
      RETURN
      END
```

SUBROUTINE GEOSTA

```
      SUBROUTINE GEOSTA
C
C         GET RIVER GEOMETRY DATA FOR ONE PARTICULAR STATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      REAL LATI,LONG
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITER
      COMMON/PLOT/MO(12),XUNIT,YUNIT,IUNIT,LINES,IOPT
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      NRECORD=0    $    KKEY=0
      NCR=0
      REWIND WF2
      IF(IPR.NE.3) GO TO 500
      IF(LFLAG.EQ.3) GO TO 400
      ENCODE(40,52,SNAME)(NAME(I),I=1,2),DIST1,DIST2
   52 FORMAT(2A10,*FR RM*,F6.2,* TO*,F6.2)
      GO TO 500
  400 SNAME(1)=SNAME(4)=10H
      SNAME(2)=NAME(1)     $    SNAME(3)=NAME(2)
  500 NYEAR=0
  356 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A2,7A10,A8)
      IF(EOF(DF).NE.0) GO TO 16
      IF(IX.EQ.2HRI) GO TO 356
      IF(IX.EQ.2HPR) GO TO 300
      IF(IX.EQ.2HCS) GO TO 380
   16 REWIND WF2
      RETURN
  300 CALL NAMNOD(LEVEL,LFLAG,KEY)
      CALL CHKEY(IDENT)
      IF(IDENT.EQ.0) GO TO 450
      KKEY=KKEY+1
      GO TO 356
```

```
  380 DECODE(78,30,IY) NPTS,NX1,IMO,NX2,IDAY,NX3,IYR,NX4,(KTEMP(I),I=1,5
     1),NX5
   30 FORMAT(I3,A10,A2,A1,A2,A1,A2,A2,4A10,A7,A8)
      READ(DF,34)(XDIS(I),YELEV(I),I=1,NPTS)
   34 FORMAT((8X,6(F6.0,F6.1)))
      IF(IDENT.EQ.1) GO TO 25
      IF(NRECORD.EQ.0) GO TO 356
      GO TO 450
C   ---  RETRIEVE RIVER GEOMETRY DATA OF THAT STATION  ---
   25 ENCODE(6,29,JDATE) IDAY,IMO,IYR
   29 FORMAT(3A2)
      IF(JFLAG.EQ.1) GO TO 26
      CALL DDATE(JDATE,IDD)
      IF(IDD.EQ.0) GO TO 356
   26 DECODE(6,27,JDATE) IDATE
   27 FORMAT(I6)
      IF(IPR.EQ.3) GO TO 54
      XTIT=10HX-DISTANCE   $   YTIT=10HELEVATION
      XUNIT=YUNIT=6HIN FT.   $   TUNIT=4HDATE
      ENCODE(40,33,SNAME)(NAME(I),I=1,2),DIST
   33 FORMAT(2A10,*AT RIVER-MILE*,F7.2)
   54 MTIT(5)=SNAME(1)    $    MTIT(6)=SNAME(2)
      MTIT(7)=SNAME(3)    $    MTIT(8)=SNAME(4)
      CALL GEODAT(W,LEVEL,IOUT,IPR,NRECORD,IDATE,NPTS)
      DO 60 I=1,NPTS
      X(I)=XDIS(I)
      Y(I)=YELEV(I)
   60 CONTINUE
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IPR.NE.3) GO TO 48
      NCR=NCR+1
      WRITE(WF2,45) NPTS,DIST
   45 FORMAT(I3,F7.2)
      WRITE(WF2,46)(XDIS(I),YELEV(I),I=1,NPTS)
   46 FORMAT(F6.0,F6.1)
   48 NYEAR=NYEAR+1
      IF(JFLAG.EQ.3) GO TO 450
      GO TO 356
  450 IF(KKEY.EQ.NKEYS) GO TO 16
      GO TO 500
C
      END
```

SUBROUTINE GEODAT (W, LEVEL, IOUT, IPR, NRECORD, IDATE, NPTS)

```
      SUBROUTINE GEODAT(W,LEVEL,IOUT,IPR,NRECORD,IDATE,NPTS)
C
C           WRITE RIVER CROSS-SECTION DATA
C
      INTEGER W
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
C
C
      IF(LEVEL.EQ.2) GO TO 37
      WRITE(W,28)
   28 FORMAT(/,10X,*DATE*,10X,*NO. DATA POINTS*)
      WRITE(W,32) IDATE,NPTS
   32 FORMAT(9X,I6,15X,I3/)
      WRITE(W,36)
   36 FORMAT(2X,*XDIST*,2X,*YELEV*,5X,*XDIST*,2X,*YELEV*,5X,*XDIST*,2X,
     1*YELEV*,5X,*XDIST*,2X,*YELEV*,5X,*XDIST*,2X,*YELEV*,5X,*XDIST*,2X,
     2*YELEV*,5X,*XDIST*,2X,*YELEV*,5X,*XDIST*,2X,*YELEV*,/,2X,*(FT)*,
     33X,*(FT)*,6X,*(FT)*,3X,*(FT)*,6X,*(FT)*,3X,*(FT)*,6X,*(FT)*,3X,
     4*(FT)*,6X,*(FT)*,3X,*(FT)*,6X,*(FT)*,3X,*(FT)*,6X,*(FT)*,3X,*(FT)*
     5,6X,*(FT)*,3X,*(FT)*/)
   37 IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 41
      IF(IPR.NE.0) GO TO 41
      WRITE(W,38)(XDIS(I),YELEV(I),I=1,NPTS)
   38 FORMAT(8(1X,F6.0,F6.1,4X))
      WRITE(W,42)
   42 FORMAT(//)
   41 NRECORD=NRECORD+1
C
      RETURN
      END
```

SUBROUTINE RIVSTR

```
      SUBROUTINE RIVSTR
C
C         RIVER CONTROL STRUCTURE DATA MANIPULATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CTYP,SNAME,
     1SUBLEVL,XTIT,YTIT
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/RIVSTR/STAGE(50),AREA(50)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      DF=18
      IRETRIV=1
      NRECORD=KKEY=0
      CALL CATEG
   50 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) RETURN
      IF(IX.EQ.3HSTR) GO TO 50
      IF(IX.EQ.3HMST) GO TO 425
      RETURN
  425 DECODE(77,424,IY) IX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1CTYP,YEAR,NPTS
  424 FORMAT(A3,3A10,A4,F7.2,2A9,A7,2I4)
      READ(DF,428)(STAGE(I),AREA(I),I=1,NPTS)
  428 FORMAT(6(F6.1,F7.0),2X)
      ENCODE(6,20,KEY) IX
   20 FORMAT(*MST*,A3)
      CALL CHKEY(IDENT)
      IF(IDENT.EQ.0) GO TO 350
      KKEY=KKEY+1
      IF(LEVEL.EQ.3) GO TO 15
      IF(LEVEL.EQ.2.AND.IOUT.EQ.1) GO TO 10
      GO TO 431
   15 WRITE(W,5)
    5 FORMAT(/,1X,132(1H-))
      WRITE(W,422)
  422 FORMAT(/,16X,*STATION NAME*,14X,*STATION NO.*,5X,*RIVER MILE*,
     1 3X,*LATITUDE*,2X,*LONGITUDE*,2X,*STRUCTURE TYPE*,2X,*YEAR*,
```

```
     22X,*NO DATA POINTS*)
      SNAME(1)=STNA(1)    $    SNAME(2)=STNA(2)
      SNAME(3)=STNA(3)    $    SNAME(4)=10H
      WRITE(W,426)(SNAME(I),I=1,4),STNO,DIST,LATI,LONG,CTYP,YEAR,NPTS
  426 FORMAT(/,2X,4A10,3X,A4,9X,F7.2,2X,2(3X,A8),6X,A7,6X,I4,
     17X,I3/)
      WRITE(W,5)
   10 WRITE(W,430)
  430 FORMAT(2X,*STAGE*,4X,*AREA*,7X,*STAGE*,4X,*AREA*,7X,*STAGE*,4X,
     1*AREA*,7X,*STAGE*,4X,*AREA*,7X,*STAGE*,4X,*AREA*,7X,*STAGE*,4X,
     2*AREA*/,3X,*(FT)*,3X,*(SQFT)*,7X,*(FT)*,3X,*(SQFT)*,7X,*(FT)*,3X,
     3*(SQFT)*,7X,*(FT)*,3X,*(SQFT)*,7X,*(FT)*,3X,*(SQFT)*,7X,*(FT)*,3X,
     4*(SQFT)*/)
  431 WRITE(W,432)(STAGE(I),AREA(I),I=1,NPTS)
  432 FORMAT(6(1X,F6.1,2X,F7.1,4X))
      NRECORD=NRECORD+1
      IF(LEVEL.EQ.3) WRITE(W,5)
  350 IF(KKEY.EQ.NKEYS) GO TO 400
      GO TO 50
  400 IF(NRECORD.EQ.0) IRETRIV=0
C
      RETURN
      END
```

SUBROUTINE RIVRES

```
      SUBROUTINE RIVRES
C
C           RESERVOIR DATA MANIPULATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      REAL MWSL,MXEL
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITER
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVRES/GAHT(50),NRCP,SPIQ(50),NSCP,QVOL(50),NCCP,NXG(50),
     1NXS(50),NXQ(50)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      DF=18
      IRETRIV=1
      NYEAR=1
      NRECORD=KKEY=0
      CALL CATEG
   50 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) RETURN
      IF(IX.EQ.3HRES) GO TO 515
      IF(IX.EQ.3HRST) GO TO 525
      RETURN
  515 DECODE(77,518,IY) IX,(NAME(I),I=1,4),NODE,RMIL,LATI,LONG
  518 FORMAT(A3,4A10,A9,F7.2,2A9)
      ENCODE(6,60,KEY) IX
```

```
 60 FORMAT(*RES*,A3)
    CALL CHKEY(IDENT)
    IF(IDENT.EQ.0) GO TO 350
    KKEY=KKEY+1
    MTIT(5)=NAME(1)    $    MTIT(6)=NAME(2)
    MTIT(7)=NAME(3)    $    MTIT(8)=NAME(4)
    NRECORD=NRECORD+1
    IF(LEVEL.EQ.2) GO TO 50
    WRITE(W,5)
  5 FORMAT(/,1X,132(1H-))
    WRITE(W,516)
516 FORMAT(//,16X,*RESERVOIR NAME*,22X,*NODE*,7X,*RIVER-MILE*,8X,*LATI
   1TUDE*,8X,*LONGITUDE*)
    WRITE(6,520) (NAME(I),I=1,4),NODE,RMIL,LATI,LONG
520 FORMAT(/,5X,4A10,3X,I8,8X,F7.2,1X,2(8X,A9))
    WRITE(W,5)
    GO TO 50
525 DECODE(77,524,IY) IX,NRCP,NSCP,NCCP,MWSL,SPEL,MXEL,(JTEMP(I),I=1,5
   1)
524 FORMAT(A3,3I5,3F6.1,4A10,A1)
    IF(IDENT.EQ.0) GO TO 527
    IF(LEVEL.EQ.3) GO TO 62
    GO TO 527
 62 WRITE(W,522)
522 FORMAT(/,10X,*NO.OF POINTS*,10X,*NO.OF POINTS*,10X,*NO.OF POINTS*
   1,12X,*MINIMUM*,12X,*SPILLWAY*,10X,*MAXIMUM*/,10X,*ON RULE CURVE*,
   28X,*ON SPILL.CURVE*,8X,*ON CAPAC.CURVE*,5X,*WATER SURFACE LEVEL*,
   36X,*ELEVATION*,8X,*ELEVATION*)
    WRITE(W,526) NRCP,NSCP,NCCP,MWSL,SPEL,MXEL
526 FORMAT(/,12X,I5,17X,I5,17X,I5,17X,F6.1,*(FT)*,10X,F6.1,*(FT)*,8X,
   1F6.1,*(FT)*)
    WRITE(W,5)
527 READ(DF,528) (DATE(I),NXG(I),I=1,NRCP)
528 FORMAT(16I5)
    IF(IDENT.EQ.0) GO TO 533
    DO 529 I=1,NRCP
    FDATE(I)=FLOAT(DATE(I))/10.
529 GAHT(I)=FLOAT(NXG(I))/10.
    IF(SUBLEVL.NE.5HALL  .AND.SUBLEVL.NE.5HRULE ) GO TO 533
    IF(LEVEL.EQ.3) GO TO 20
    IF(LEVEL.EQ.2.AND.IOUT.EQ.1) GO TO 20
    GO TO 531
 20 WRITE(W,530)
530 FORMAT(/,50X,*--- RULE CURVE   ---*//,2X,*DATE*,1X,*GATE HGHT*,2X,
   1*DATE*,1X,*GATE HGHT*,2X,*DATE*,1X,*GATE HGHT*,2X,*DATE*,1X,*GATE
   2HGHT*,2X,*DATE*,1X,*GATE HGHT*,2X,*DATE*,1X,*GATE HGHT*,2X,*DATE*,
   31X,*GATE HGHT*,2X,*DATE*,1X,*GATE HGHT*/)
531 WRITE(W,532)(FDATE(I),GAHT(I),I=1,NRCP)
532 FORMAT(8(1X,F5.1,2X,F6.1,2X))
    YEAR=9999   $    NPTS=NRCP
    DO 52 I=1,NRCP
    Y(I)=GAHT(I)
 52 X(I)=FDATE(I)
    CALL DWRITE(WF1,ICATEG,SUBLEVL)
    XTIT=10HTIME           $   YTIT=10HGATE HGHT
    XUNIT=7HIN DAYS    $    YUNIT=6HIN FT.    $    IUNIT=4HYEAR
533 READ(DF,528)(NXG(I),NXS(I),I=1,NSCP)
    IF(IDENT.EQ.0) GO TO 535
    DO 534 I=1,NSCP
    GAHT(I)=FLOAT(NXG(I))/10.
```

```
534 SPIQ(I)=FLOAT(NXS(I))/10.
    IF(SUBLEVL.NE.5HALL   .AND.SUBLEVL.NE.5HSPILL) GO TO 535
    IF(LEVEL.EQ.3) GO TO 30
    IF(LEVEL.EQ.2.AND.IOUT.EQ.1) GO TO 30
    GO TO 537
 30 WRITE(W,536)
536 FORMAT(/,50X,*--- SPILLWAY CURVE ---*//,2X,*G HGHT*,1X,*SPILL.Q*,
   12X,*G HGHT*,1X,*SPILL.Q*,2X,*G HGHT*,1X,*SPILL.Q*,2X,*G HGHT*,1X,
   2*SPILL.Q*,2X,*G HGHT*,1X,*SPILL.Q*,2X,*G HGHT*,1X,*SPILL.Q*,2X,
   3*G HGHT*,1X,*SPILL.Q*,2X,*G HGHT*,1X,*SPILL.Q*/)
537 WRITE(W,538)(GAHT(I),SPIQ(I),I=1,NSCP)
538 FORMAT(8(1X,F6.1,2X,F6.1,1X))
    YEAR=9999   $    NPTS=NSCP
    DO 54 I=1,NSCP
    Y(I)=GAHT(I)
 54 X(I)=SPIQ(I)
    CALL DWRITE(WF1,ICATEG,SUBLEVL)
    XTIT=10HSPILLWAY Q      $     YTIT=10HGATE HGHT
    XUNIT=6HIN CFS    $    YUNIT=5HIN FT.    $    TUNIT=4HYEAR
535 READ(DF,528)(NXG(I),NXQ(I),I=1,NCCP)
    IF(IDENT.EQ.0) GO TO 50
    DO 539 I=1,NCCP
    GAHT(I)=FLOAT(NXG(I))/10.
539 QVOL(I)=FLOAT(NXQ(I))/10.
    IF(SUBLEVL.NE.5HALL   .AND.SUBLEVL.NE.5HCAPAC) GO TO 50
    IF(LEVEL.EQ.3) GO TO 40
    IF(LEVEL.EQ.2.AND.IOUT.EQ.1) GO TO 40
    GO TO 541
 40 WRITE(W,540)
540 FORMAT(/,50X,*--- CAPACITY CURVE ---*//,1X,*GATE HGHT*,1X,*QVOL*,
   12X,*GATE HGHT*,1X,*QVOL*,2X,*GATE HGHT*,1X,*QVOL*,2X,*GATE HGHT*,
   21X,*QVOL*,2X,*GATE HGHT*,1X,*QVOL*,2X,*GATE HGHT*,1X,*QVOL*,2X,
   3*GATE HGHT*,1X,*QVOL*,2X,*GATE HGHT*,1X,*QVOL*/)
541 WRITE(W,538)(GAHT(I),QVOL(I),I=1,NCCP)
    YEAR=9999   $    NPTS=NCCP
    DO 56 I=1,NCCP
    Y(I)=GAHT(I)
 56 X(I)=QVOL(I)
    CALL DWRITE(WF1,ICATEG,SUBLEVL)
    XTIT=10HQ VOLUME       $    YTIT=10HGATE HGHT
    XUNIT=7HIN A-FT    $    YUNIT=6HIN FT.    $    TUNIT=4HYEAR
    WRITE(W,542)
542 FORMAT(//)
350 IF(KKEY.EQ.NKEYS)  GO TO 400
    GO TO 50
400 IF(NRECORD.EQ.0) IRETRIV=0
C
    RETURN
    END
```

SUBROUTINE STADISC

```
      SUBROUTINE STADISC
C
C         RIVER STAGE/DISCHARGE DATA MANIPULATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      REAL MOAVG,MOMIN,MOMAX
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCONWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
C   ---    CHECK DATA CATEGORY    ---
      IF(ICATEG.EQ.8) DF=14
      IF(ICATEG.EQ.9) DF=13
      IRETRIV=1
      CALL CATEG
      IF(LFLAG.GT.1) GO TO 350
C   ---  RETRIEVE ALL RIVER STAGE OR DISCHARGE DATA  ---
   50 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) RETURN
      IF(ICATEG.EQ.9) GO TO 10
      IF(IX.EQ.3HDST) GO TO 25
      IF(IX.EQ.3HDVL) GO TO 27
      RETURN
   10 IF(IX.EQ.3HSTA) GO TO 25
      IF(IX.EQ.3HSTV) GO TO 27
      RETURN
C   ---  READ AND WRITE STATION INFORMATION  ---
   25 IF(LEVEL.EQ.2) GO TO 50
      DECODE(77,24,IY) IX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1GAGO,TYPE,NYRS
   24 FORMAT(A3,3A10,A4,F7.2,2A9,F7.3,A5,I3)
      SNAME(1)=NAME(1)    $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)    $    SNAME(4)=STNA(2)
      CALL WSTD
      GO TO 50
C   ---  READ AND WRITE STAGE-DISCHARGE DATA  ---
   27 CALL RSTD(DF)
      NMON=12    $    MSTART=1
      CALL STDVAL(W,ICATEG,LEVEL,DISTOT,FLOWH,MAXMO,MAXDAY,FLOWL,MINMO,
     1MINDAY,NMON,MSTART)
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IPR.NE.0) GO TO 50
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 50
      CALL WSTDS(W,ICATEG,IOUT,NPTS,DISTOT,FLOWH,MAXMO,MAXDAY,FLOWL,
     1MINMO,MINDAY)
      GO TO 50
  350 CALL STDSTA
      IF(NRECORD.EQ.0) IRETRIV=0
C
      RETURN
      END
```

SUBROUTINE STDSTA

```
      SUBROUTINE STDSTA
C
C         GET RIVER STAGE/DISCHARGE DATA AT A GAGING STATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      REAL MOAVG,MOMIN,MOMAX
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      NRECORD=0   $   KKEY=0
  700 NYEAR=0
  356 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A3,7A10,A7)
      IF(EOF(DF).NE.0) GO TO 16
      IF(ICATEG.EQ.9) GO TO 10
      IF(IX.EQ.3HDST) GO TO 360
      IF(IX.EQ.3HDVL) GO TO 53
      GO TO 16
   10 IF(IX.EQ.3HSTA) GO TO 360
      IF(IX.EQ.3HSTV) GO TO 53
   16 REWIND DF
      RETURN
C  ---  READ AND WRITE STATION INFORMATION  ---
  360 DECODE(77,362,IY) NX,(STNA(I),I=1,3),STNO,DIST,LATI,LONG,
     1GAGO,TYPE,NYRS
  362 FORMAT(A3,3A10,A4,F7.2,2A9,F7.3,A5,I3)
      IF(ICATEG.EQ.9) GO TO 361
      ENCODE(6,358,KEY) NX
  358 FORMAT(*DST*,A3)
      GO TO 364
  361 ENCODE(6,363,KEY) NX
  363 FORMAT(*STA*,A3)
  364 CALL CHKEY(IDENT)
      IF(IDENT.EQ.0) GO TO 356
      KKEY=KKEY+1
      SNAME(1)=NAME(1)     $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)     $    SNAME(4)=STNA(2)
      MTIT(5)=SNAME(1)     $    MTIT(6)=SNAME(2)
      MTIT(7)=SNAME(3)     $    MTIT(8)=SNAME(4)
      IF(ICATEG.EQ.9) GO TO 51
      XTIT=10HTIME              $    YTIT=10HDISCHARGE
      XUNIT=7HIN DAYS    $   YUNIT=6HIN CFS    $   TUNIT=4HYEAR
      GO TO 52
   51 XTIT=10HTIME              $    YTIT=10HRIV. STAGE
      XUNIT=7HIN DAYS    $   YUNIT=6HIN FT.    $   TUNIT=4HYEAR
   52 IF(LEVEL.EQ.2) GO TO 356
      CALL WSTD
      GO TO 356
C  ---  START MANIPULATING STAGE-DISCHARGE DATA  ---
   53 CALL RSTD(DF)
      IF(IDENT.EQ.1) GO TO 600
      IF(NRECORD.EQ.0) GO TO 356
      GO TO 550
  600 CONTINUE
      NMON=12   $   MSTART=1
```

```
C   ---  START RETRIEVAL ACCORDING TO TIME COMMAND OPTION  ---
      GO TO(382,384,386,392,394,400,500),JFLAG
  382 CALL STDVAL(W,ICATEG,LEVEL,DISTOT,FLOWH,MAXMO,MAXDAY,FLOWL,
     1MINMO,MINDAY,NMON,MSTART)
   41 NRECORD=NRECORD+1
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IPR.NE.0) GO TO 67
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 67
      CALL WSTDS(W,ICATEG,IOUT,NPTS,DISTOT,FLOWH,MAXMO,MAXDAY,FLOWL,
     1MINMO,MINDAY)
   67 NYEAR=NYEAR+1
      IF(JFLAG.EQ.2.OR.JFLAG.EQ.6) GO TO 550
      IF(JFLAG.EQ.7) GO TO 458
      GO TO 356
C   ---  RETRIEVE DATA IN CALENDAR YEAR FORMAT  ---
  384 DECODE(4,385,JSTART) JYEAR
  385 FORMAT(I4)
      IF(YEAR.NE.JYEAR) GO TO 356
      GO TO 382
C   ---   RETRIEVE DATA FOR A PARTICULAR DATE    ---
  386 IF(YEAR.NE.JYR1) GO TO 356
      NYEAR=NYEAR+1
      NRECORD=NRECORD+1
       IF(ICATEG.EQ.9)GO TO 640
       WRITE(W,630) X(ND1)
  630 FORMAT(/,* THE RIVER DISCHARGE FOR THAT DATE = *,F7.0,*CFS*)
       GO TO 550
  640 WRITE(W,650) X(ND1)
  650 FORMAT(/,* THE GAGE-STAGE FOR THAT DATE = *,F7.2,*FT.*)
       GO TO 550
C   ---     RETRIEVE DATA FOR A NUMBER OF CALENDAR YEARS    ---
  392 CALL DYEAR (YEAR,JYD)
      IF(JYD.EQ.0) GO TO 356
      GO TO 382
C   ---   RETRIEVE DATA FOR A PARTICULAR TIME-PERIOD    ---
  394 IF(YEAR.LT.JYR1) GO TO 356
      IF(YEAR.GT.JYR2) GO TO 550
      IF(JYR2.EQ.JYR1) GO TO 398
      IF(YEAR.EQ.JYR1) GO TO 396
      IF(YEAR.EQ.JYR2) GO TO 397
      N1=1
  395 IF(LEAP.EQ.1) N2=366
      IF(LEAP.EQ.0) N2=365
      GO TO 399
  396 N1=ND1    $   GO TO 395
  397 N1=1   $   N2=ND2   $   GO TO 399
  398 N1=ND1   $   N2=ND2
  399 NPTS=0
      DO 660 I=N1,N2
       NPTS=NPTS+1
       XX=X(I)
  660 X(NPTS)=XX
      NYEAR=NYEAR+1
      NRECORD=NRECORD+1
       CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 356
      IF(NYEAR.GT.1) GO TO 685
      IF(ICATEG.EQ.9) GO TO 680
      WRITE(W,670) JSTART,JEND
  670 FORMAT(/,40X,*DISCHARGE VALUES (CFS) FOR THAT PERIOD ARE*/,
```

```
   146X,*FROM *,A6,* TO *,A6)
      GO TO 685
  680 WRITE(W,690) JSTART,JEND
  690 FORMAT(/,40X,*GAGE-STAGE VALUES (FT) FOR THAT PERIOD ARE*/,
     146X,*FROM *,A6,* TO *,A6)
  685 WRITE(W,692) YEAR,NPTS
  692 FORMAT(/,2X,2I10,/)
      IF(ICATEG.EQ.9) GO TO 687
      WRITE(W,686)(X(I),I=1,NPTS)
  686 FORMAT(10(2X,F10.0))
      GO TO 356
  687 WRITE(W,688)(X(I),I=1,NPTS)
  688 FORMAT(10(2X,F10.2))
      GO TO 356
C   --- RETRIEVE DATA IN WATER-YEAR FORMAT  ---
  400 DECODE (4,385,JSTART) JYEAR
      IF(YEAR.NE.JYEAR) GO TO 356
  410 IF(LEVEL.EQ.2) GO TO 433
      IF(ICATEG.EQ.9) GO TO 76
      WRITE (W,428) YEAR
  428 FORMAT(//,45X,*DAILY DISCHARGE FOR WATER YEAR *,I4//,38X,
     1*COMPUTED DAILY DISCHARGE IN CUBIC FEET PER SECOND*)
      GO TO 78
   76 WRITE(W,77) YEAR
   77 FORMAT(//,45X,*DAILY RIVER STAGE FOR WATER YEAR *,I4//,36X,*COMPUT
     1ED DAILY RIVER STAGE IN FEET ABOVE GAGE ZERO ELEVATION*)
   78 WRITE(W,32)(MO(I),I=10,12),(MO(J),J=1,9)
   32 FORMAT(//,6X,*DAY*,12(5X,A3,1X),/)
  433 DISTOT=0.    $    NPTS=0
      FLOWL=100000.    $   FLOWH=0.
      MI=0
      DO 441 I=10,12
      MI=MI+1
      DO 436 J=1,31
  436 DVALUE(MI,J)=0.
      ND=NDAY(I)
      DO 440 J=1,ND
      NPTS=NPTS+1
      FVALUE(J)=DVALUE(I,J)
      DVALUE(MI,J)=FVALUE(J)
      X(NPTS)=FVALUE(J)
      IF(FVALUE(J).GE.FLOWL) GO TO 446
      FLOWL=FVALUE(J)
      MINDAY=J    $   MINMO=I
  446 IF(FVALUE(J).LE.FLOWH) GO TO 448
      FLOWH=FVALUE(J)
      MAXDAY=J    $   MAXMO=I
  448 DISTOT=DISTOT+FVALUE(J)
  440 MOAVG(I)=MOAVG(I)+FVALUE(J)
      MOAVG(I)=MOAVG(I)/FLOAT(ND)
      CALL RANGE(FVALUE,1,ND,MOMAX(IM),MHIGH,MOMIN(IM),MLOW)
  441 CONTINUE
      READ(DF,14) 1X,(IY(I),I=1,8)
      IF(EOF(DF).NE.0) GO TO 550
      NDAY(2)=28
      DECODE(77,54,IY) NX,YEAR,(ITEMP(I),I=1,7)
   54 FORMAT(A3,I4,7A10)
      CALL LEAPYR(YEAR,LEAP)
      IF(LEAP.EQ.1) NTOT=366
      IF(LEAP.EQ.0) NTOT=365
```

```
      READ(DF,31)(Y(I),I=1,NTOT)
   31 FORMAT(8F10.2)
      IF(LEAP.EQ.1) NDAY(2)=29
      IF(LEAP.EQ.0) NDAY(2)=28
       KOUNT=0
      DO 445 I=1,9
      MI=I+3
      DO 442 J=1,31
  442 DVALUE(MI,J)=0.
      ND=NDAY(I)
      DO 81 K=1,ND
      KOUNT=KOUNT+1
   81 DVALUE(MI,K)=Y(KOUNT)
  445 CONTINUE
      DO 450 I=1,9
      ND=NDAY(I)
      MI=I+3
      DO 456 J=1,ND
      NPTS=NPTS+1
      FVALUE(J)=DVALUE(MI,J)
      X(NPTS)=FVALUE(J)
      IF(FVALUE(J).GE.FLOWL) GO TO 452
      FLOWL=FVALUE(J)
      MINDAY=J    $    MINMO=I
  452 IF(FVALUE(J).LE.FLOWH) GO TO 454
      FLOWH=FVALUE(J)
      MAXDAY=J    $    MAXMO=I
  454 DISTOT=DISTOT+FVALUE(J)
  456 MOAVG(I)=MOAVG(I)+FVALUE(J)
      MOAVG(I)=MOAVG(I)/FLOAT(ND)
      CALL RANGE(FVALUE,1,ND,MOMAX(I),MHIGH,MOMIN(I),MLOW)
  450 CONTINUE
      YEAR=YEAR-1
      GO TO 41
  500 CALL DYEAR(YEAR,JYD)
      IF(JYD.EQ.0) GO TO 356
      GO TO 410
  458 YEAR=YEAR+1
      CALL DYEAR(YEAR,JYD)
      IF(JYD.EQ.0) GO TO 550
      DO 460 I=10,12
      ND=NDAY(I)
      DO 82 K=1,ND
      KOUNT=KOUNT+1
   82 DVALUE(I,K)=Y(KOUNT)
  460 CONTINUE
      GO TO 410
  550 IF(KKEY.EQ.NKEYS) GO TO 16
      GO TO 700
C
      END
```

SUBROUTINE RSTD (DF)

```
      SUBROUTINE RSTD(DF)
C
C          READ STAGE / DISCHARGE DATA
C
      INTEGER DF,YEAR
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXO,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(9)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
C
      DECODE(77,28,IY) NX,YEAR,(ITEMP(I),I=1,7)
   28 FORMAT(A3,I4,7A10)
      CALL LEAPYR(YEAR,LEAP)
      IF(LEAP.EQ.1) NPTS=366
      IF(LEAP.EQ.0) NPTS=365
      READ(DF,31)(X(I),I=1,NPTS)
   31 FORMAT(8F10.2)
      IF(LEAP.EQ.1) NDAY(2)=29
      IF(LEAP.EQ.0) NDAY(2)=28
      DO 32 I=1,12
      DO 32 J=1,31
   32 DVALUE(I,J)=0.
       KOUNT=0
      DO 34 I=1,12
      ND=NDAY(I)
      DO 34 J=1,ND
      KOUNT=KOUNT+1
   34 DVALUE(I,J)=X(KOUNT)
C
      RETURN
      END
```

SUBROUTINE STDVAL (W, ICATEG, LEVEL, DISTOT, FLOWH, MAXMO, MAXDAY, FLOWL, MINMO, MINDAY, NMON, MSTART)

```
      SUBROUTINE STDVAL(W,ICATEG,LEVEL,DISTOT,FLOWH,MAXMO,MAXDAY,FLOWL,
     1MINMO,MINDAY,NMON,MSTART)
C
C          RETRIEVE AND PROCESS STAGE / DISCHARGE DATA
C
      INTEGER W
      REAL MOAVG,MOMIN,MOMAX
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),NRECORD,
     2CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
C
      IF(LEVEL.EQ.2) GO TO 33
      IF(ICATEG.EQ.9) GO TO 57
      WRITE (W,28) YEAR
   28 FORMAT(//,45X,*DAILY DISCHARGE FOR *,I4//,32X,*COMPUTED DAILY DISC
     1HARGE IN CUBIC FEET PER SECOND*)
      GO TO 59
   57 WRITE(W,58) YEAR
   58 FORMAT(//,45X,*DAILY RIVER STAGE FOR *,I4//,30X,*COMPUTED DAILY RI
     1VER STAGE IN FEET ABOVE GAGE ZERO ELEVATION*)
   59 WRITE(W,32)(MO(I),I=1,12)
   32 FORMAT(//,6X,*DAY*,12(5X,A3,1X),/)
   33 DISTOT=0.   $   NPTS=0
      FLOWL=100000.   $   FLOWH=0.
      DO 40 I=1,NMON
      IM=MSTART+I-1
      ND=NDAY(IM)
      SUM=0.
      DO 36 J=1,ND
      NPTS=NPTS+1
      FVALUE(J)=DVALUE(IM,J)
      X(NPTS)=FVALUE(J)
      IF(FVALUE(J).GE.FLOWL) GO TO 46
      FLOWL=FVALUE(J)
      MINDAY=J   $   MINMO=IM
   46 IF(FVALUE(J).LE.FLOWH) GO TO 48
      FLOWH=FVALUE(J)
      MAXDAY=J   $   MAXMO=IM
   48 DISTOT=DISTOT+FVALUE(J)
   36 SUM=SUM+FVALUE(J)
      MOAVG(IM)=SUM/FLOAT(ND)
      CALL RANGE(FVALUE,1,ND,MOMAX(IM),MHIGH,MOMIN(IM),MLOW)
   40 CONTINUE
C
      RETURN
      END
```

SUBROUTINE WSTDS (W, ICATEG, IOUT, NPTS, DISTOT, FLOWH, MAXMO, MAXDAY, FLOWL, MINMO, MINDAY)

```
      SUBROUTINE WSTDS(W,ICATEG,IOUT,NPTS,DISTOT,FLOWH,MAXMO,MAXDAY,
     1FLOWL,MINMO,MINDAY)
C
C           WRITE STAGE / DISCHARGE DATA AND SUMMARY STATISTICS
C
      INTEGER W
      REAL MOAVG,MOMIN,MOMAX
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
C
C
      KOUNT=0
      DO 60 J=1,31
      KOUNT=KOUNT+1
      IF(ICATEG.EQ.9) GO TO 70
      WRITE (W,42) J, (DVALUE(I,J),I=1,12)
   42 FORMAT(7X,I2,12(2X,F7.0))
      GO TO 72
   70 WRITE(W,71) J,(DVALUE(I,J),I=1,12)
   71 FORMAT(7X,I2,12(2X,F7.2))
   72 IF(KOUNT.NE.5) GO TO 60
      IF(J.EQ.30) GO TO 60
      WRITE (W,44)
   44 FORMAT(/)
      KOUNT=0
   60 CONTINUE
      IF(IOUT.NE.3) RETURN
      WRITE(W,61)
   61 FORMAT(/,50X,*MONTHLY STATISTICS*)
      IF(ICATEG.EQ.9) GO TO 90
      WRITE(W,62)(MOAVG(I),I=1,12)
   62 FORMAT(//,5X,*MEAN*,12(2X,F7.0))
      WRITE (W,63)(MOMAX(I),I=1,12)
   63 FORMAT(/,5X,*MAX.*,12(2X,F7.0))
      WRITE(W,64)(MOMIN(I),I=1,12)
   64 FORMAT(/,5X,*MIN.*,12(2X,F7.0))
      GO TO 99
   90 WRITE(W,92)(MOAVG(I),I=1,12)
   92 FORMAT(//,5X,*MEAN*,12(2X,F7.2))
      WRITE(W,93)(MOMAX(I),I=1,12)
   93 FORMAT(/,5X,*MAX.*,12(2X,F7.2))
      WRITE(W,94)(MOMIN(I),I=1,12)
   94 FORMAT(/,5X,*MIN.*,12(2X,F7.2))
   99 YAVG=DISTOT/FLOAT(NPTS)
      IF(ICATEG.EQ.9) GO TO 73
      WRITE(W,66) DISTOT,YAVG
   66 FORMAT(//,15X,*TOTAL DISCHARGE FOR YEAR WAS*,F10.0,15X,*MEAN DISCH
     1ARGE FOR YEAR WAS *,F7.1)
      WRITE(W,68) FLOWH,MO(MAXMO),MAXDAY,FLOWL,MO(MINMO),MINDAY
   68 FORMAT(//,15X,*HIGHEST DISCHARGE VALUE WAS*,F7.0,*CFS, OCCURRED ON
     1 *,A3,I3/,15X,*LOWEST DISCHARGE VALUE WAS*,F7.0,*CFS, OCCURRED ON
     2*,A3,I3//)
      RETURN
   73 WRITE(W,74) YAVG
   74 FORMAT(//,45X,*MEAN RIVER STAGE FOR YEAR WAS *,F7.2)
      WRITE(W,75) FLOWH,MO(MAXMO),MAXDAY,FLOWL,MO(MINMO),MINDAY
   75 FORMAT(//,15X,*HIGHEST RIVER STAGE VALUE WAS*,F7.2,*FT. ABOVE GAGE
     1 ZERO ELEVATION, OCCURRED ON *,A3,I3/,15X,*LOWEST RIVER STAGE VALU
     2E WAS*,F7.2,*FT. ABOVE GAGE ZERO ELEVATION, OCCURRED ON *,A3,I3//)
C
      RETURN
      END
```

SUBROUTINE PRECIP

```
      SUBROUTINE PRECIP
C
C         PRECIPITATION DATA MANIPULATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT
      REAL MOAVG,MOMIN,MOMAX
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NFTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITER
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      DF=17
      IRETRIV=1
      CALL CATEG
      ITER=0
      IF(LFLAG.GT.1) GO TO 350
C   ---  RETRIEVE ALL PRECIPITATION DATA  ---
   90 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A2,7A10,A8)
      IF(EOF(DF).NE.0) RETURN
      IF(IX.EQ.2HPS) GO TO 25
      IF(IX.EQ.2HPV) GO TO 35
      RETURN
   25 DECODE(78,24,IY) IX,(STNA(I),I=1,3),STNO,LATI,LONG,
     1GELEV,TYPE,NYRS
   24 FORMAT(A4,3A10,2X,A4,2A9,A10,A5,I5)
      SNAME(1)=NAME(1)    $    SNAME(2)=NAME(2)
      SNAME(3)=STNA(1)    $    SNAME(4)=STNA(2)
```

```
      IF(LEVEL.EQ.2) GO TO 90
      CALL PREDAT(W)
      WRITE(W,50)
   50 FORMAT(/,8X,*DATE*,36X,*PRECIPITATION DATA (0.1 INCH)*,36X,*DAILY
     1TOT*/,7X,6(1H-),1X,96(1H-),3X,9(1H-)//)
      GO TO 90
   35 DECODE(78,30,IY) IDAY,IMON,IYR,(JVAL(I),I=1,24)
   30 FORMAT(3I2,24I3)
      IF(ITER.EQ.0) GO TO 36
      IF(IMON.EQ.IMONO) GO TO 40
      WRITE(W,34) MO(IMONO),MTOT
   34 FORMAT(/,45X,*MONTHLY TOTAL FOR *,A3,* IS  *,I4/)
      MTOT=0   $   IMONO=IMON
      IF(IYR.EQ.IYRO) GO TO 40
      WRITE(W,32) IYR
   32 FORMAT(/,10X,*YEAR  19*,I2//)
      IYRO=IYR
      GO TO 40
   36 WRITE(W,32) IYR
      MTOT=0   $   IMONO=IMON   $   IYRO=IYR
   40 JTOT=0
      DO 42 I=1,24
   42 JTOT=JTOT+JVAL(I)
      MTOT=MTOT+JTOT
      WRITE(W,46) IDAY,IMON,IYR,(JVAL(I),I=1,24),JTOT
   46 FORMAT(7X,3I2,1X,24(1X,I3),5X,I4)
      ITER=ITER+1
      GO TO 90
  350 CALL PRESTA
      IF(NRECORD.EQ.0) IRETRIV=0
      RETURN
      END
```

SUBROUTINE PRESTA

```
      SUBROUTINE PRESTA
C
C         GET PRECIPITATION DATA FOR ONE PARTICULAR GAGING STATION
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,SNAME,SUBLFVL,
     1XTIT,YTIT,GFLEV
      REAL MOAVG,MOMIN,MOMAX
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/LOCATE/LFLAG,INODE,INAME,NAMST,NUMST,ILAT,ILONG,DIST1,DIST2
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),I7(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/SUB/NNAME(50,3),NNODE(50),NSUB,IKEY(300),NKEYS,KEY
C
      NRECORD=0    $    KKEY=0
  650 NYEAR=0
      DO 6 I=1,366
      DO 6 J=1,25
    6 HVALUE(I,J)=0.
      DO 10 I=1,12
      MOAVG(I)=0.
      DO 10 J=1,31
   10 DVALUE(I,J)=0.
  356 READ(DF,14) IX,(IY(I),I=1,8)
   14 FORMAT(A2,7A10,A8)
      IF(EOF(DF).NE.0) GO TO 16
      IF(IX.EQ.2HPS) GO TO 360
      IF(IX.EQ.2HPV) GO TO 380
   16 REWIND DF
      RETURN
C   ---  READ AND WRITE STATION INFORMATION  ---
  360 DECODE(78,362,IY) NX,(STNA(I),I=1,3),STNO,LATI,LONG,GELEV,TYPE,
     1NYRS
  362 FORMAT(A4,3A10,2X,A4,2A9,A10,A5,I5)
      ENCODE(6,363,KEY) NX
  363 FORMAT(*PS*,A4)
      CALL CHKEY(IDENT)
      IF(IDENT.EQ.0) GO TO 356
      KKEY=KKEY+1
      SNAME(1)=10H                 $    SNAME(2)=STNA(1)
      SNAME(3)=STNA(2)             $    SNAME(4)=STNA(3)
      MTIT(5)=SNAME(1)    $    MTIT(6)=SNAME(2)
```

```
      MTIT(7)=SNAME(3)    $    MTIT(8)=SNAME(4)
      XTIT=10HTIME            $    YTIT=10HRAINFALL
      YUNIT=7HIN INCH    $    TUNIT=4HYEAR
      IF(SUBLEVL.EQ.5HHOURL) XUNIT=6HIN HRS
      IF(SUBLEVL.EQ.5HDAILY) XUNIT=7HIN DAYS
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 356
      CALL PREDAT
      GO TO 356
C    ---  GET PRECIPITATION DATA OF ONE PARTICULAR STATION  ---
  380 IF(IDENT.EQ.0) GO TO 356
      DECODE(78,30,IY) JDATE,(JVAL(I),I=1,24)
   30 FORMAT(A6,24I3)
      DECODE(6,8,JDATE) IDAY,IMON,IYR
    8 FORMAT(3I2)
      YEAR=1900+IYR
      CALL LEAPYR(YEAR,LEAP)
      IF(LEAP.EQ.1) NDAY(2)=29
      IF(LEAP.EQ.0) NDAY(2)=28
      IF(LEAP.EQ.1) NPTS=366
      IF(LEAP.EQ.0) NPTS=365
C    ---  RETRIEVE DATA ACCORDING TO TIME COMMAND OPTION  ---
      GO TO (382,384,386,392,394,400,500),JFLAG
  382 IF(NRECORD.EQ.0) GO TO 38
      IF(IMON.EQ.IMONO) GO TO 40
      MOAVG(IMONO)=MTOT    $    MTOT=0    $    IMONO=IMON
      IF(YEAR.EQ.IYRO) GO TO 40
      IF(SUBLEVL.EQ.5HHOURL) GO TO 25
      NPTS=0
      DO 48 I=1,12
      ND=NDAY(I)
      DO 48 J=1,ND
      NPTS=NPTS+1
      DVALUE(I,J)=DVALUE(I,J)/100.
   48 X(NPTS)=DVALUE(I,J)
      GO TO 33
   25 DO 18 I=1,NPTS
      DO 18 J=1,25
   18 HVALUE(I,J)=HVALUE(I,J)/100.
   33 DO 20 I=1,12
   20 MOAVG(I)=MOAVG(I)/100.
      NEWYR=YEAR
      YEAR=IYRO
      IF(JFLAG.EQ.3) GO TO 387
      IF(JFLAG.EQ.5) GO TO 393
  100 CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IOUT.EQ.2) GO TO 58
      IF(SUBLEVL.EQ.5HHOURL) GO TO 22
      IF(JFLAG.GT.5) GO TO 620
      WRITE(W,51)(MO(I),I=1,12)
   51 FORMAT(//,7X,*DAY*,12(5X,A3,1X)/)
      GO TO 630
  620 WRITE(W,51)(MO(I),I=10,12),(MO(J),J=1,9)
  630 KOUNT=0
      DO 54 J=1,31
      KOUNT=KOUNT+1
      WRITE(W,52) J,(DVALUE(I,J),I=1,12)
   52 FORMAT(7X,I2,12(2X,F7.2))
      IF(KOUNT.NE.5) GO TO 54
      IF(J.EQ.30) GO TO 54
      WRITE(W,53)
```

```
 53 FORMAT(/)
    KOUNT=0
 54 CONTINUE
    WRITE(W,56)(MOAVG(I),I=1,12)
 56 FORMAT(//,50X,*MONTHLY SUMMARY*//,2X,*MON.TOT*,12(2X,F7.2)//,1X,
   1130(1H-)//)
    GO TO 58
 22 IF(JFLAG.GT.5) GO TO 28
    NPTS=0
    DO 26 I=1,12
    CALL WHRAIN(I,W,IYR,NPTS)
 26 CONTINUE
    GO TO 58
 28 NPTS=0
    DO 27 I=10,12
    CALL WHRAIN(I,W,IYR,NPTS)
 27 CONTINUE
    DO 29 I=1,9
    CALL WHRAIN(I,W,IYR,NPTS)
 29 CONTINUE
 58 NYEAR=NYEAR+1
 39 YEAR=NEWYR
    IF(JFLAG.EQ.2.OR.JFLAG.EQ.6) GO TO 550
    IF(JFLAG.EQ.4.OR.JFLAG.EQ.7) GO TO 458
    IF(IOUT.EQ.2) GO TO 62
    WRITE(W,34) YEAR
 34 FORMAT(/,45X,*RAINFALL DATA FOR YEAR *,I4/)
    IF(SUBLEVL.EQ.5HDAILY) GO TO 60
    WRITE(W,35)
 35 FORMAT(2X,*DATE*,34X,*MEASURED HOURLY RAINFALL IN .01 INCH*,47X,
   1*DAILY TOT*/)
    GO TO 62
 60 WRITE(W,41)
 41 FORMAT(44X,*DAILY TOTAL RAINFALL IN INCH*)
 62 IYRO=IYR
    IF(SUBLEVL.EQ.5HHOURL) GO TO 75
    DO 36 I=1,12
    MOAVG(I)=0.
    DO 36 J=1,31
 36 DVALUE(I,J)=0.
    GO TO 40
 75 DO 76 I=1,366
    DO 76 J=1,25
 76 HVALUE(I,J)=0.
    GO TO 40
458 DECODE(4,385,JEND) KYEAR
    IF(YEAR.GT.KYEAR) GO TO 550
    GO TO 139
 38 IF(IOUT.EQ.2) GO TO 65
    IF(JFLAG.EQ.3.OR.JFLAG.EQ.5) GO TO 65
    WRITE(W,34) YEAR
    IF(SUBLEVL.EQ.5HDAILY) GO TO 63
    WRITE(W,35)
    GO TO 65
 63 WRITE(W,41)
 65 MTOT=0    $    IMONO=IMON    $    IYRO=YEAR
 40 JTOT=0
    DO 42 I=1,24
 42 JTOT=JTOT+JVAL(I)
    MTOT=MTOT+JTOT
```

```
      IF(SUBLEVL.EQ.5HDAILY) GO TO 46
      CALL DORDER(JDATE,NDT,YEAR)
      DO 44 I=1,24
   44 HVALUE(NDT,I)=JVAL(I)
      HVALUE(NDT,25)=JTOT
      GO TO 47
   46 DVALUE(IMON,IDAY)=JTOT
   47 NRECORD=NRECORD+1
      GO TO 356
C   ---  RETRIEVE DATA IN CALENDAR YEAR FORMAT  ---
  384 DECODE(4,385,JSTART) JYEAR
  385 FORMAT(I4)
      IF(YEAR.NE.JYEAR) GO TO 78
      GO TO 382
   78 IF(NRECORD.EQ.0) GO TO 356
      GO TO 382
C   ---   RETRIEVE DATA FOR A PARTICULAR DATE   ---
  386 IF(YEAR.NE.JYR1) GO TO 78
      GO TO 382
  387 IF(SUBLEVL.EQ.5HHOURL) GO TO 70
      WRITE(W,710) X(ND1)
  710 FORMAT(/,* TOTAL DAILY RAINFALL FOR THAT DATE = *,F5.2,* INCHES*)
      GO TO 550
   70 WRITE(W,711)(HVALUE(ND1,I),I=1,25)
  711 FORMAT(/,* HOURLY RAINFALL FOR THAT DATE ARE*/,5X,24F5.2/,
     1* TOTAL DAILY RAINFALL FOR THAT DATE = *,F5.2,* INCHES*)
      GO TO 550
C   ---   RETRIEVE DATA FOR A NUMBER OF CALENDAR YEARS   ---
  392 CALL DYEAR(YEAR,JYD)
      IF(JYD.EQ.0) GO TO 78
      GO TO 382
C   ---   RETRIEVE DATA FOR A PATICULAR TIME-PERIOD   ---
  394 IF(YEAR.LT.JYR1.OR.YEAR.GT.JYR2) GO TO 78
      GO TO 382
  393 IF(JYR2.EQ.JYR1) GO TO 398
      IF(YEAR.EQ.JYR1) GO TO 396
      IF(YEAR.EQ.JYR2) GO TO 397
      N1=1
  395 IF(LEAP.EQ.1) N2=366
      IF(LEAP.EQ.0) N2=365
      GO TO 399
  396 N1=ND1   $   GO TO 395
  397 N1=1   $   N2=ND2   $   GO TO 399
  398 N1=ND1   $   N2=ND2
  399 NPTS=0
      NYEAR=NYEAR+1
      IF(SUBLEVL.EQ.5HHOURL) GO TO 730
      DO 720 I=N1,N2
      NPTS=NPTS+1
  720 X(NPTS)=X(I)
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 740
      IF(NYEAR.GT.1) GO TO 725
      WRITE(W,722) JSTART,JEND
  722 FORMAT(/,40X,*DAILY RAINFALL (IN INCHES) FROM *,A6,* TO *,A6/)
  725 WRITE(W,726) YEAR,NPTS
  726 FORMAT(/,2X,2I10,* DAYS*/)
      WRITE(W,728)(X(I),I=1,NPTS)
  728 FORMAT(10(2X,F10.2))
      GO TO 740
```

```
  730 DO 732 I=N1,N2
      NPTS=NPTS+1
      DO 732 J=1,25
  732 HVALUE(NPTS,J)=HVALUE(I,J)
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      IF(IOUT.EQ.0.OR.IOUT.EQ.2.OR.IOUT.EQ.4) GO TO 356
      IF(NYEAR.GT.1) GO TO 735
      WRITE(W,734) JSTART,JEND
  734 FORMAT(/,40X,*HOURLY RAINFALL (IN INCHES) FROM *,A6,* TO *,A6/)
  735 WRITE(W,736) YEAR,NPTS
  736 FORMAT(/,2X,2I10,* DAYS*/)
      DO 739 I=1,NPTS
      WRITE(W,738)(HVALUE(I,J),J=1,24)
  738 FORMAT(5X,24F5.2)
  739 CONTINUE
  740 IF(YEAR.EQ.JYR2) GO TO 550
      YEAR=NEWYR
      GO TO 65
C   ---  RETRIEVE DATA IN WATER YEAR FORMAT  ---
  400 IF(NRECORD.EQ.0) GO TO 138
      IF(IMON.EQ.IMONO) GO TO 40
      IF(SUBLEVL.EQ.5HDAILY) GO TO 133
      WRITE(W,32) MO(IMON),MTOT
   32 FORMAT(/,42X,*MONTHLY TOTAL FOR *,A3,* IS  *,I4/)
  133 MOAVG(IMONO)=MTOT   $   MTOT=0   $   IMONO=IMON
      IF(IMON.EQ.10) GO TO 190
      GO TO 40
  190 IF(SUBLEVL.EQ.5HHOURL) GO TO 39
      DO 198 I=1,3
      DO 192 J=1,31
  192 FVALUE(J)=DVALUE(12,J)
      TEMP=MOAVG(12)
      DO 194 K=1,11
      K12=12-K   $   K21=K12+1
      MOAVG(K21)=MOAVG(K12)
      DO 194 J=1,31
  194 DVALUE(K21,J)=DVALUE(K12,J)
      DO 196 J=1,31
  196 DVALUE(1,J)=FVALUE(J)
      MOAVG(1)=TEMP
  198 CONTINUE
      NPTS=0   $   MI=9
      DO 200 I=1,3
      MI=MI+1   $   ND=NDAY(MI)
      DO 200 J=1,ND
      NPTS=NPTS+1
      DVALUE(I,J)=DVALUE(I,J)/100.
  200 X(NPTS)=DVALUE(I,J)
      DO 210 I=4,12
      MI=I-3   $   ND=NDAY(MI)
      DO 210 J=1,ND
      NPTS=NPTS+1
      DVALUE(I,J)=DVALUE(I,J)/100.
  210 X(NPTS)=DVALUE(I,J)
      DO 220 I=1,12
  220 MOAVG(I)=MOAVG(I)/100.
      GO TO 100
  138 DECODE(4,385,JSTART) JYEAR
      IF(YEAR.LT.JYEAR) GO TO 356
      IF(NRECORD.NE.0) GO TO 139
```

```
      IF(IMON.LT.10) GO TO 356
  139 IF(IOUT.NE.2) GO TO 142
      IF(LEVEL.EQ.2) GO TO 65
  142 WRITE(W,140) YEAR
  140 FORMAT(/,45X,*RAINFALL DATA FOR WATER YEAR *,I4/)
      IF(SUBLEVL.EQ.5HDAILY) GO TO 600
      WRITE(W,35)
      GO TO 65
  600 WRITE(W,41)
      GO TO 65
C   ---   RETRIEVE DATA FOR A NUMBER OF WATER YEARS   ---
  500 CALL DYEAR(YEAR,JYD)
      IF(JYD.EQ.0) GO TO 356
      GO TO 400
  550 IF(KKEY.EQ.NKEYS) GO TO 16
      GO TO 650
C
      END
```

SUBROUTINE PREDAT

```
      SUBROUTINE PREDAT
C
C           WRITE PRECIPITATION DATA
C
      INTEGER W,TYPE,GELEV
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(6),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
C
C
      WRITE(W,5)
    5 FORMAT(/,1X,132(1H-))
      WRITE(W,22)
   22 FORMAT(/,16X,*STATION NAME*,14X,*STATN NO*,5X,*LATITUDE*,5X,
     1*LONGITUDE*,5X,*ELEVATION(MSL)*,5X,*DATA TYPE*,7X,*NO YEARS*)
      WRITE(W,26)(SNAME(I),I=1,4),STNO,LATI,LONG,GELEV,TYPE,NYRS
   26 FORMAT(/,2X,4A10,3X,A4,2(5X,A9),5X,A10,*(FT)*,7X,A5,9X,I5)
      WRITE(W,5)
C
      RETURN
      END
```

SUBROUTINE WHRAIN (I, W, IYR, NPTS)

```
      SUBROUTINE WHRAIN(I,W,IYR,NPTS)
C
      INTEGER W
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
C
      ND=NDAY(I)
      DO 24 J=1,ND
      NPTS=NPTS+1
      WRITE(W,23) J,I,IYR,(HVALUE(NPTS,K),K=1,25)
   23 FORMAT(1X,3I2,24F5.2,F6.2)
   24 CONTINUE
      WRITE(W,32) MO(I),MOAVG(I)
   32 FORMAT(/,42X,*MONTHLY TOTAL FOR *,A3,* IS  *,I4/)
      RETURN
      END
```

PROGRAM PROCESS

```
      OVERLAY(TEST,2,0)
      PROGRAM PROCESS
C
C         PROCESS THE RETRIEVED DATA
C
      INTEGER W,WF1,WF2,WF3,OF,STNA,STNO,YEAR,DATE,SNAME,SUBLEVL,
     1XTIT,YTIT,XUNIT,YUNIT,TUNIT
      REAL MOAVG,MOMIN,MOMAX
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/TIME/JFLAG,JSTART,JEND,JNAME,ND1,JYR1,ND2,JYR2
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITER
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
C
      IF(IPR.GT.9) GO TO 800
      GO TO(50,50,150,250,350,550,600,650,700),IPR
C  ---    FREQUENCY ANALYSIS    ---
   50 XLOW=1000000.   $   XHIGH=0.   $    NDATA=0
      DO 10 II=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      NDATA=NDATA+NPTS
      CALL RANGE(X,1,NPTS,XHI,IPH,XLI,IPL)
      IF(XLI.LE.XLOW) XLOW=XLI
      IF(XHI.GE.XHIGH) XHIGH=XHI
   10 CONTINUE
      REWIND WF1
      DO 20 II=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      CALL CDFPDF(II,XHIGH,XLOW)
   20 CONTINUE
      IF(ICATEG.EQ.8)XTIT=10HDISCHARGE
      IF(ICATEG.EQ.9)XTIT=10HGAGE STAGE
      IF(ICATEG.EQ.10)XTIT=10HRAINFALL
```

```
      IF(ICATEG.EQ.8)XUNIT=6HIN CFS
      IF(ICATEG.EQ.9)XUNIT=6HIN FT.
      IF(ICATEG.EQ.10)XUNIT=7HIN INCH
      YUNIT=6H            $   TUNIT=4H
      IOUT=2
      GO TO 1000
C   ---  THALWEG LEVEL PLOT  ---
  150 DO 155 I=1,NCR
      READ(WF2,152) NPTS,DIST
  152 FORMAT(I3,F7.2)
      READ(WF2,154)(XDIS(J),YELEV(J),J=1,NPTS)
  154 FORMAT(F6.0,F6.1)
      X(I)=DIST
      CALL XMIN(YELEV,1,NPTS,YLOW,IPMIN)
      Y(I)=YLOW
  155 CONTINUE
      IF(IOUT.NE.1.AND.IOUT.NE.3) GO TO 157
      WRITE(W,151)(SNAME(I),I=1,4)
  151 FORMAT(//,50X,*THALWEG LEVEL FOR*/,40X,4A10//,35X,*RIVER MILE (MI)
     1*,20X,*BED ELEV.(FT)*/)
      WRITE(W,153)(X(I),Y(I),I=1,NCR)
  153 FORMAT(37X,F6.0,29X,F6.1)
  157 NPTS=NCR
      CALL DWRITE(WF1,ICATEG,SUBLEVL)
      XTIT=10HRIVER MILE   $   YTIT=10HBED ELEV.
      XUNIT=6HIN MI.       $   YUNIT=6HIN FT.    $   TUNIT=4H
      GO TO 1000
C   ---   CHANGING STAGE FOR A GIVEN DISCHARGE   ---
  250 CONTINUE
      DECODE(40,251,JCOMWD) NX,NY,FIXQ,NZ,(IZ(I),I=1,2)
  251 FORMAT(A10,A1,F7.0,A3,A10,A9)
      GO TO 550
  252 DO 254 II=1,NYEAR
      X(II)=0.   $   A1=CA(II)   $   B1=CB(II)
      CALL PINV(XX,FIXQ,A1,B1,ANEW,BNEW)
      Y(II)=XX
      X(II)=IYEAR(II)
  254 CONTINUE
      IF(IOUT.EQ.2) GO TO 265
      WRITE(W,256) FIXQ,(MTIT(I),I=4,8)
  256 FORMAT(//,15X,*CHANGING RIVER STAGE FOR Q = *,F7.0,* CFS*//,
     17X,5A10/)
      WRITE(W,258) GAGO
  258 FORMAT(17X,*(GAGE-O ELEV. = *,F7.3,* FT(MSL))*//)
      WRITE(W,259)
  259 FORMAT(/,20X,*YEAR*,10X,*RIVER-STAGE IN FT*/)
      DO 262 I=1,NYEAR
      IXX=X(I)
      WRITE(W,260) IXX,Y(I)
  260 FORMAT(20X,I4,13X,F7.2)
  262 CONTINUE
      IF(IOUT.EQ.1) GO TO 1000
  265 YTIT=10HRIV. STAGE   $   XTIT=10HTIME
      XUNIT=7HIN YEAR   $   YUNIT=6HIN FT.   $   TUNIT=4H
      NPTS=NYEAR
      IOUT=2
      GO TO 1000
C   ---   CUMULATIVE RAINFALL ANALYSIS   ---
  350 DO 380 N=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
```

```
      SUM=0.
      DO 352 I=1,NPTS
      XX=X(I)+SUM
      X(I)=XX
      SUM=X(I)
  352 CONTINUE
      CALL DWRITE(WF2,ICATEG,SUBLEVL)
  380 CONTINUE
      ENDFILE WF2
      REWIND WF2
      IF(IOUT.EQ.2) GO TO 370
      DO 390 N=1,NYEAR
      CALL DREAD(WF2,ICATEG,SUBLEVL)
      DO 353 I=1,12
      DO 353 J=1,31
  353 DVALUE(I,J)=0.
      IF(JFLAG.EQ.6.OR.JFLAG.EQ.7) GO TO 360
      WRITE(W,385)
  385 FORMAT(1X,130(1H-))
      WRITE(W,354) YEAR
  354 FORMAT(/,25X,*TABLE OF CUMULATIVE RAINFALL FOR CALENDAR YEAR *,
     1I4/)
      WRITE(W,385)
      IF(LEAP.EQ.1) NDAY(2)=29
      IF(LEAP.EQ.0) NDAY(2)=28
      NDATA=0
      DO 355 I=1,12
      ND=NDAY(I)
      DO 355 J=1,ND
      NDATA=NDATA+1
  355 DVALUE(I,J)=X(NDATA)
      WRITE(W,356)(MO(I),I=1,12)
  356 FORMAT(/,6X,*DAY*,12(5X,A3,1X),/)
  359 KOUNT=0
      DO 357 J=1,31
      KOUNT=KOUNT+1
      WRITE(W,358) J,(DVALUE(I,J),I=1,12)
  358 FORMAT(7X,I2,12(4X,F5.2))
      IF(KOUNT.NE.5) GO TO 357
      IF(J.EQ.30) GO TO 357
      WRITE(W,361)
  361 FORMAT(/)
      KOUNT=0
  357 CONTINUE
      WRITE(W,385)
      GO TO 390
  360 WRITE(W,385)
      WRITE(W,362) YEAR
  362 FORMAT(/,25X,*TABLE OF CUMULATIVE RAINFALL FOR WATER YEAR *,I4/)
      WRITE(W,385)
      NDATA=0
      DO 364 I=10,12
      ND=NDAY(I)
      K=I-9
      DO 364 J=1,ND
      NDATA=NDATA+1
  364 DVALUE(K,J)=X(NDATA)
      YEAR=YEAR+1
      CALL LEAPYR(YEAR,LEAP)
      IF(LEAP.EQ.1) NDAY(2)=29
```

```
      IF(LEAP.EQ.0) NDAY(2)=28
      DO 365 I=1,9
      ND=NDAY(I)
      K=I+3
      DO 365 J=1,ND
      NDATA=NDATA+1
  365 DVALUE(K,J)=X(NDATA)
      WRITE(W,356)(MO(I),I=10,12),(MO(J),J=1,9)
      GO TO 359
  390 CONTINUE
      REWIND WF2
      GO TO 1000
  370 YTIT=10HC.RAINFALL
      GO TO 1000
C   ---  REGRESSION ANALYSIS  ---
  550 DO 555 II=1,NYEAR
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      IYEAR(II)=YEAR
      DO 553 IJ=1,NPTS
      TEMP=X(IJ)
      X(IJ)=Y(IJ)
  553 Y(IJ)=TEMP
      CALL PLSQ(A1,B1,RC,SBAR)
      CA(II)=A1    $   CB(II)=B1
      IF(IPR.NE.6) GO TO 555
      WRITE(W,551) YEAR
  551 FORMAT(//,34X,*YEAR = *,I4)
      WRITE(W,552) A1,B1,RC,SBAR
  552 FORMAT(/,10X,*REGRESSION EQUATION IS*,5X,*Q =*,F10.5,*((S).EXP(*,
     1F7.4,*))*//,10X,*CORRELATION COEFFICIENT = *,F7.4,10X,*STANDARD ER
     2ROR = *,F9.6)
      WRITE(WF2,554) A1,B1
  554 FORMAT(2F8.4)
  555 CONTINUE
      REWIND WF1
      REWIND WF2
      IF(IPR.EQ.4) GO TO 252
      GO TO 1000
C   ---  CORRELATION ANALYSIS  ---
  600 CONTINUE
      GO TO 1000
C   ---   DISCHARGE HYDROGRAPH GENERATION   ---
  650 DO 670 II=1,NYEAR
      READ(WF2,554) A1,B1
      CALL DREAD(WF1,9,SUBLEVL)
      DO 652 I=1,NPTS
      XX=X(I)
      CALL PEVAL(XX,YY,A1,B1,ANEW,BNEW)
  652 X(I)=YY
      CALL DWRITE(WF3,8,SUBLEVL)
      IF(IOUT.EQ.2) GO TO 670
      WRITE(W,660) YEAR,NPTS
  660 FORMAT(//,40X,*TABLE OF GENERATED DISCHARGE VALUES*//,40X,
     1*YEAR = *,I4,4X,*NO.OF POINTS =*,I4//)
      WRITE(W,662)(X(I),I=1,NPTS)
  662 FORMAT(10(2X,F10.0))
  670 CONTINUE
      REWIND WF1
      REWIND WF2
      ENDFILE WF3
```

```
      REWIND WF3
      CALL DREAD(WF3,8,SUBLEVL)
      CALL DWRITE(WF1,8,SUBLEVL)
      ICATEG=8
      IF(IOUT.EQ.1) GO TO 1000
      YUNIT=6HIN CFS   $   XUNIT=7HIN DAYS   $   TUNIT=4HYEAR
      XTIT=10HDISCHARGE      $    YTIT=10HTIME
C     WRITE(W,666)
C 666 FORMAT(//,40X,*PLOT OF GENERATED DISCHARGE VALUES*//)
      GO TO 1000
C   ---   STAGE HYDROGRAPH GENERATION   ---
  700 DO 730 II=1,NYEAR
      READ(WF2,554) A1,B1
      CALL DREAD(WF1,8,SUBLEVL)
      DO 702 I=1,NPTS
      YY=X(I)
      CALL PINV(XX,YY,A1,B1,ANEW,BNEW)
  702 X(I)=XX
      CALL DWRITE(WF3,9,SUBLEVL)
      IF(IOUT.EQ.2) GO TO 730
      WRITE(W,710) YEAR,NPTS
  710 FORMAT(//,40X,*TABLE OF GENERATED RIVER STAGE VALUES*//,40X,
     1*YEAR = *,I4,4X,*NO.OF POINTS =*,I4//)
      WRITE(W,712)(X(I),I=1,NPTS)
  712 FORMAT(10(2X,F10.2))
  730 CONTINUE
      REWIND WF1
      REWIND WF2
      ENDFILE WF3
      REWIND WF3
      CALL DREAD(WF3,9,SUBLEVL)
      CALL DWRITE(WF1,9,SUBLEVL)
      ICATEG=9
      IF(IOUT.EQ.1) GO TO 1000
      YUNIT=6HIN FT.   $   XUNIT=7HIN DAYS   $   TUNIT=4HYEAR
      XTIT=10HRIV. STAGE     $    YTIT=10HTIME
C     WRITE(W,716) GAGO
C 716 FORMAT(//,40X,*PLOT OF GENERATED VALUES FOR RIVER STAGE*/,50X,
C    1*(GAGE-O ELEV. = *,F7.3,* FT(MSL))*//)
      GO TO 1000
  800 IF(IPR.EQ.10) GO TO 810
      IF(IPR.EQ.11) GO TO 820
      IF(IPR.EQ.12) GO TO 830
      IF(IPR.EQ.13) GO TO 810
      IF(IPR.EQ.14) GO TO 50
      GO TO 1000
C   ---   FIND MIN. VALUE   ---
  810 CALL YXMIN(YXLOW,MYEAR,IPYMIN)
      IF(MYEAR.EQ.1) IPYMIN=IPYMIN+ND1-1
      MYEAR=MYEAR+JYR1-1
      CALL DCONVRT(MYEAR,IPYMIN,IMO,IDAY)
      WRITE(W,812) YXLOW,MO(IMO),IDAY,MYEAR
  812 FORMAT(//,10X,*THE MINIMUM VALUE IS *,F10.2,
     1*  WHICH OCCURRED ON *,A3,1X,I2,*,*,I4)
      IF(IPR.EQ.13) GO TO 820
      GO TO 1000
C   ---   FIND MAX. VALUE   ---
  820 CALL YXMAX(YXHIGH,MYEAR,IPYMAX)
      IF(MYEAR.EQ.1) IPYMAX=IPYMAX+ND1-1
      MYEAR=MYEAR+JYR1-1
```

```
      CALL UCONVRT(MYEAR,IPYMAX,IMO,IDAY)
      WRITE(W,822) YXHIGH,MO(IMO),IDAY,MYEAR
  822 FORMAT(//,10X,*THE MAXIMUM VALUE IS *,F10.2,
     1*  WHICH OCCURRED ON *,A3,1X,I2,*,*,I4)
      GO TO 1000
C   ---   CALCULATE THE BASIC STATISTICS   ---
  830 CALL USTAT(XLOW,XHIGH,XMEAN,XSTDV)
      WRITE(W,832) XLOW,XHIGH,XMEAN,XSTDV
  832 FORMAT(//,10X,*MINIMUM VALUE =*,F10.2,10X,*MAXIMUM VALUE =*,
     1F10.2/,10X,*MEAN VALUE =*,F10.2,10X,*STANDARD DEVIATION =*,
     2F10.2)
 1000 CONTINUE
      REWIND WF1
      REWIND WF2
      REWIND WF3
      END
```

SUBROUTINE CDFPDF (MULT, XMAX, XMIN)

```
      SUBROUTINE CDFPDF(MULT,XMAX,XMIN)
C
C     THIS ROUTINE WILL COMPUTE A DISCRETE PROBABILITY DENSITY FUNCTION
C     AND ITS CORRESPONDING  CUMULATIVE DENSITY FCN FOR A GIVEN SAMPLE X
C     OF SIZE = NDATA.

      INTEGER W,WF1,WF2,WF3,STNA,STNO,YEAR,SNAME,XTIT,YTIT,XUNIT,YUNIT
      DIMENSION PDFX(50),CDFX(50),V(50),OFEQ(50)
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT

C     THE TOTAL RANGE = XMAX - XMIN, IS DIVIDED  INTO RN-1 =50 SUBRANGES
C     THEN THE FREQUENCY IS COMPUTED CORRESPONDING TO EACH SUBRANGE.
      IF(MULT.GT.1) GO TO 400
      NR=50
      IOPT=0
      LINES=NR
      XUNIT=YUNIT=5H
      RN=NR
      RDATA=NDATA
      CDF=0.
      SUBRAN=(XMAX-XMIN)/(RN-1.)
      RL=XMIN-SUBRAN/2.
      RLO=RL
      DO 3 K=1,NR
    3 OFEQ(K)=0.
  400 RL=RLO
      DO 420 K=1,NR
      FEQ=0.
      RL=RL+SUBRAN
      DO 410 J=1,NPTS
      IF(X(J).GE.RL) GO TO 410
      IF(X(J).LT.(RL-SUBRAN)) GO TO 410
      FEQ=FEQ+1
  410 CONTINUE
  420 OFEQ(K)=OFEQ(K)+FEQ
      IF(MULT.LT.NYEAR) RETURN
      RL=RLO
      DO 430 K=1,NR
      RL=RL+SUBRAN
      V(K)=RL-SUBRAN/2.
      PDFX(K)=OFEQ(K)/RDATA
      CDF=CDF+PDFX(K)
  430 CDFX(K)=CDF
      IF(IOUT.EQ.2) GO TO 108
      IF(IPR.EQ.14) GO TO 450
      IF(IPR.EQ.2) GO TO 440
      IF(IOUT.NE.4) GO TO 431
      REWIND WF1
```

```
      WRITE(WF1,436) NR
  436 FORMAT(I5)
      WRITE(WF1,438)(CDFX(I),I=1,NR)
  438 FORMAT(8F10.5)
      RETURN
  431 WRITE(W,432) MTIT
  432 FORMAT(/,33X,*TABLE OF THE COMPUTED CUMULATIVE FREQUENCIES (CDF)*
     1//,1X,8A10//,1X,119(1H-)/,26X,*NO.*,22X,*OBS. FREQ.*,19X,*CDF*/,
     21X,119(1H-))
      WRITE(W,434)(I,OFEQ(I),CDFX(I),I=1,NR)
  434 FORMAT(25X,I3,20X,F10.0,15X,F10.5)
      GO TO 104
  440 IF(IOUT.NE.4) GO TO 441
      REWIND WF1
      WRITE(WF1,436) NR
      WRITE(WF1,438)(PDFX(I),I=1,NR)
      RETURN
  441 WRITE(W,442) MTIT
  442 FORMAT(/,33X,*TABLE OF THE COMPUTED RELATIVE FREQUENCIES (PDF)*//,
     11X,8A10//,1X,119(1H-)/,26X,*NO.*,22X,*OBS. FREQ.*,19X,*PDF*/,1X,
     2119(1H-))
      WRITE(W,434)(I,OFEQ(I),PDFX(I),I=1,NR)
      GO TO 104
  450 IF(IOUT.NE.4) GO TO 445
      REWIND WF1
      WRITE(WF1,436) NR
      WRITE(WF1,439)(OFEQ(I),CDFX(I),PDFX(I),I=1,NR)
  439 FORMAT(F10.0,2F10.5)
      RETURN
  445 WRITE(6,102)    MTIT
  102 FORMAT(/,14X,*TABLE OF THE COMPUTED RELATIVE FREQUENCIES (PDF),
     1AND THE CUMULATIVE FREQUENCIES (CDF)*//,1X,8A10//,1X,119(1H-)/
     /  5X,*NO.* 16X, *RANGE*. 17X,  *MID-RANGE *,   13X, *OBS. FREQ.*,
     314X,*CDF*,16X,*PDF*/,1X,119(1H-))
      DO 6 I=1,NR
      VL=V(I)-SUBRAN/2.
      RV=V(I)+SUBRAN/2.
      WRITE(6,103) I,VL,RV, V(I),OFEQ(I),CDFX(I),  PDFX(I)
  103 FORMAT(5X,I3,5X,*(*,F10.2,2X,F10.2,2X,*)*,5X,F10.2,12X,F10.0,2X,
     12(9X,F10.5))
    6 CONTINUE
  104 PRINT 105
  105 FORMAT ( 1X,119(1H-))
      IF(IOUT.EQ.1) RETURN

C     READY FOR PLOTTING ACCORDING TO OPTION FLAG.

  108 DO 20 K=1,NR
   20 X(K)=V(K)
      NPTS=NR
      IF(IPR.EQ.2) GO TO 326
      IF(IPR.EQ.1) GO TO 340
      WRITE(6,22)
   22 FORMAT(/,* SURRY.THE ONLY OUTPUT OPTION IS "LIST"*)
      GO TO 900
  326 DO 24 K=1,NR
   24 Y(K)=PDFX(K)
      YTIT=10HREL. FREQ.
      GO TO 900
  340 DO 26 K=1,NR
   26 Y(K)=CDFX(K)
      YTIT=10HCUM. FREQ.
  900 WRITE(WF3,910) NR
  910 FORMAT(I5)
      WRITE(WF3,920)(X(I),Y(I),I=1,NR)
  920 FORMAT(F10.0,F10.6)
C
      RETURN
      END
```

SUBROUTINE PLSQ (A1, B1, RC, SBAR)

```
      SUBROUTINE PLSQ(A1,B1,RC,SBAR)
C
C        LEAST-SQUARES CURVE FITTING OF POWER FUNCTION  Y=A1*(X**B1)
C
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      SUMX=SUMXX=SUMY=SUMXY=0.
      SUMA=SUMB=SUMC=SUMD=0.
      EPSI=.01
      DO 10 I=1,NPTS
      IF(X(I).LT.EPSI) X(I)=EPSI
      IF(Y(I).LT.EPSI) Y(I)=EPSI
      XLOG=ALOG(X(I))
      YLOG=ALOG(Y(I))
      SUMX=SUMX+XLOG
      SUMY=SUMY+YLOG
      SUMXX=SUMXX+XLOG*XLOG
      SUMXY=SUMXY+XLOG*YLOG
   10 CONTINUE
      FN=FLOAT(NPTS)
      FX=SUMX/FN      $      FY=SUMY/FN
C
C        DERIVE THE EQUATION
C
      B=(SUMXY-FN*FX*FY)/(SUMXX-FN*FX*FX)
      B1=B
      A=FY-B*FX
      A1=EXP(A)
C
C        CALCULATE THE COEFFICIENT OF CORRELATION AND THE STANDARD ERROR
C        OF THE ESTIMATE
C
      DO 20 I=1,NPTS
      XLOG=ALOG(X(I))
      YLOG=ALOG(Y(I))
      SUMA=SUMA+(XLOG-FX)*(YLOG-FY)
      SUMB=SUMB+(XLOG-FX)**2
      SUMC=SUMC+(YLOG-FY)**2
      SUMD=SUMD+(YLOG-A-B*XLOG)**2
   20 CONTINUE
      RC=SUMA/SQRT(SUMB*SUMC)
      SBAR=SQRT(SUMD/(FN-1.))
      RETURN
      END
```

SUBROUTINE PEVAL (XX, YY, A1, B1, ANEW, BNEW)

```
      SUBROUTINE PEVAL(XX,YY,A1,B1,ANEW,BNEW)
C
C        EVALUATE YY FROM XX BY THE POWER FUNCTION    YY=A1*(XX**B1)
C
      YY=A1*(XX**B1)
      AA1=1./A1    $    BNEW=1./B1
      ANEW=AA1**BNEW
      RETURN
      END
```

SUBROUTINE PINV (XX, YY, A1, B1, ANEW, BNEW)

```
      SUBROUTINE PINV(XX,YY,A1,B1,ANEW,BNEW)
C
C        EVALUATE XX FROM YY BY THE INVERSE OF THE POWER FUNCTION
C                                     YY=A1*(XX**B1)
C
      AA1=1./A1    $    BNEW=1./B1
      ANEW=AA1**BNEW
      XX=ANEW*(YY**BNEW)
      RETURN
      END
```

PROGRAM DOUTPUT

```
      OVERLAY(TEST,3,0)
      PROGRAM DOUTPUT
C
C          DISPLAY THE RETRIEVED INFORMATION AND DATA PROCESSING RESULTS
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,CTYP,
     1PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SNAME,SUBLEVL,XTIT,YTIT,
     2XUNIT,YUNIT,TUNIT
      REAL LATI,LONG,MWSL,MXEL,MOAVG,MOMIN,MOMAX
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXG,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
C
      WRITE(W,10)
   10 FORMAT(/,* DO YOU WISH THE RESULTS TO BE DISPLAYED ON THE TEKTRONI
     1X SCREEN*/)
      READ(IN,20) ITEK
   20 FORMAT(A3)
      IF(ITEK.EQ.3HNO ) GO TO 30
      CALL DPLOTS
      GO TO 50
   30 WRITE(W,40)
   40 FORMAT(/,* DO YOU WISH THE RESULTS TO BE DISPLAYED BY A LINE PRINT
     1ER*/)
      READ(IN,20) ICOM
      IF(ICOM.EQ.3HNO ) GO TO 50
      CALL PPLOTS
   50 CONTINUE
      END
```

SUBROUTINE PPLOTS

```
      SUBROUTINE PPLOTS
C
C          DISPLAY THE INFORMATION BY A LINE PRINTER
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,CTYP,
     1PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SNAME,SUBLEVL,XTIT,YTIT,
     2XUNIT,YUNIT,TUNIT
      REAL LATI,LONG,MWSL,MXEL,MOAVG,MOMIN,MOMAX
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISFT,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(6),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
C
      MOST=NYEAR
      IF(MOST.GT.3) MOST=3
      XL=YL=100000.
      XH=YH=0.
      DO 36 NP=1,MOST
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      CALL RANGE(X,1,NPTS,XHI,IPH,XLI,IPL)
      CALL RANGE(Y,1,NPTS,YHI,IPH,YLI,IPL)
      IF(XLI.LE.XL) XL=XLI
      IF(XHI.GE.XH) XH=XHI
      IF(YLI.LE.YL) YL=YLI
      IF(YHI.GE.YH) YH=YHI
   36 CONTINUE
      REWIND WF1
      IF(ICATEG.NE.10) GO TO 10
      XLOW=XL    $    XHIGH=XH
      XL=YL    $    XH=YH
      YL=XLOW    $    YH=XHIGH
   10 CALL AXISTL(ICATEG)
      DO 40 IPLOT=1,MOST
      IF(IPLOT.EQ.1) ISYMBOL=1H*
      IF(IPLOT.EQ.2) ISYMBOL=1H+
      IF(IPLOT.EQ.3) ISYMBOL=1H0
      CALL DREAD(WF1,ICATEG,SUBLEVL)
      IF(ICATEG.NE.10) GO TO 20
      CALL SWITCH
   20 CALL PLTITL(ICATEG,IPLOT,ISYMBOL)
      IF(ICATEG.EQ.9) WRITE(W,38) GAGO
   38 FORMAT(/,55X,*(GAGE-O ELEV. = *,F7.3,*FT(MSL))*)
      CALL MULPLOT(ISYMBOL,IPLOT,MOST,XL,XH,YL,YH)
   40 CONTINUE
      REWIND WF1
      IF(ICATEG.EQ.2) GO TO 74
      WRITE(W,72)(MTIT(I),I=1,8)
   72 FORMAT(/,5X,8A10)
      GO TO 200
   74 WRITE(W,76)(MTIT(I),I=1,8),GAGO
   76 FORMAT(/,10X,8A10,8X,*GAGE-O ELEV. = *,F7.3,* FT(MSL)*)
  200 CONTINUE
C
      RETURN
      END
```

SUBROUTINE AXISTL (ICATEG)

```
      SUBROUTINE AXISTL(ICATEG)
      COMMON/PLOT/MO(12),XUNIT,YUNIT,TUNIT,LINES,IOPT
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROG,DATE(150),ITER
      GO TO(100,20,100,100,50,100,100,80,85,90),ICATEG
   20 XUNIT=5H(CFS)
      YUNIT=5H(FT.)
      IOPT=0   $   LINES=50
      RETURN
   50 XUNIT=5H(FT.)
      YUNIT=5H(FT.)
      IOPT=0   $   LINES=50
      RETURN
   80 XUNIT=5H(CFS)
      YUNIT=5H(DAY)
      IOPT=2   $   LINES=NPTS
      RETURN
   85 XUNIT=5H(FT.)
      YUNIT=5H(DAY)
      IOPT=2   $   LINES=NPTS
      RETURN
   90 XUNIT=5H(IN.)
      YUNIT=5H(DAY)
      XTIT=10HRAINFALL     $    YTIT=10HTIME
      IOPT=2   $   LINES=NPTS
  100 RETURN
      END
```

SUBROUTINE PLTITL (ICATEG, IPLOT, ISYMBOL)

```
      SUBROUTINE PLTITL(ICATEG,IPLOT,ISYMBOL)
C
      INTEGER W,WF1,WF2,WF3,YEAR,DATE
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOUT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
C
      IF(IPR.EQ.1.OR.IPR.EQ.2) GO TO 200
      GO TO(100,20,100,100,50,100,100,80,85,90),ICATEG
   20 IF(IPLOT.GT.1) GO TO 24
      WRITE(W,23)
   23 FORMAT(///,55X,*STAGE-DISCHARGE PLOT*/)
   24 WRITE(W,25) ISYMBOL,YEAR,NPTS
   25 FORMAT(45X,A1,* = YEAR *,I4,5X,*NO. DATA POINTS = *,I4/)
      RETURN
   50 IF(IPLOT.GT.1) GO TO 64
      WRITE(W,63)
   63 FORMAT(///,55X,*RIVER CROSS-SECTION*/)
   64 WRITE(W,65) ISYMBOL,IDATE,NPTS
   65 FORMAT(45X,A1,* = DATE *,I6,5X,*NO. DATA POINTS = *,I4/)
      RETURN
   80 IF(IPLOT.GT.1) GO TO 84
      WRITE(W,83)
   83 FORMAT(///,55X,*DISCHARGE-HYDROGRAPH PLOT*//)
   84 WRITE(W,25) ISYMBOL,YEAR,NPTS
      RETURN
   85 IF(IPLOT.GT.1) GO TO 88
      WRITE(W,86)
   86 FORMAT(///,55X,*STAGE-HYDROGRAPH PLOT*//)
   88 WRITE(W,25) ISYMBOL,YEAR,NPTS
      RETURN
   90 IF(IPLOT.GT.1) GO TO 94
      WRITE(W,92)
   92 FORMAT(///,55X,*DAILY RAINFALL PLOT*//)
   94 WRITE(W,25) ISYMBOL,YEAR,NPTS
  100 RETURN
  200 IF(IPR.EQ.2) GO TO 210
      WRITE(W,202)
  202 FORMAT(///,52X,*CUMULATIVE FREQUENCY CURVE*//)
      RETURN
  210 WRITE(W,212)
  212 FORMAT(///,52X,*RELATIVE FREQUENCY HISTOGRAM*//)
C
      RETURN
      END
```

SUBROUTINE DPLOTS

```
      SUBROUTINE DPLOTS
C
C          DISPLAY THE INFORMATION ON A TEKTRONIX SCREEN
C
      INTEGER W,WF1,WF2,WF3,DF,STNA,STNO,TYPE,YEAR,DATE,CONC,CTYP,
     1PCEN1,PCEN2,PCEN3,PCEN4,PCEN5,PCEN6,SNAME,SUBLEVL,XTIT,YTIT,
     2XUNIT,YUNIT,IUNIT
      REAL LATI,LONG,MWSL,MXEL,MOAVG,MOMIN,MOMAX
      COMMON/INFORM/ICOM,JTEMP(6),KTEMP(5),LTEMP(4),NTEMP(2),ICOMD,IFILE
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      COMMON/OUTPUT/IOT,X(366),Y(366),IPOSX(366,3),IPOSY(366,3),MTIT(8)
     1,XTIT,YTIT,YEAR,NNPTS,IA(10),NYEAR,NDATA,IDATE,DATE(150),
     2NRECORD,CA(150),CB(150),IPR,IPROC,FDATE(150),ITEK
      COMMON/PLOT/MO(12),XUNIT,YUNIT,IUNIT,LINES,IOPT
      COMMON/RIVCRO/XDIS(100),YELEV(100),NCROS
      COMMON/STADIS/MOAVG(12),MOMIN(12),MOMAX(12),DVALUE(12,31),NDAY(12)
     1,FVALUE(31),JVAL(24),STAG(150),DISC(150),HVALUE(366,25)
      DIMENSION HEADER(10),IOUT(100),IDASH(5)
      DATA IDASH/96,98,97,99,100/
C
      MOST=NYEAR
      IF(MOST.GT.5) MOST=5
      IF(IPR.EQ.1.OR.IPR.EQ.2.OR.IPR.EQ.4) MOST=1
      IF(IPR.EQ.3) MOST=1
      XL=YL=100000.
      XH=YH=0.
      IF(IPR.EQ.5) WF1=2
      DO 36 NP=1,MOST
      IF(IPR.EQ.1.OR.IPR.EQ.2.OR.IPR.EQ.4) GO TO 10
      IF(IPR.EQ.3) GO TO 10
      CALL DREAD(WF1,ICATEG,SUBLEVL)
   10 CALL RANGE(X,1,NNPTS,XHI,IPH,XLI,IPL)
      CALL RANGE(Y,1,NNPTS,YHI,IPH,YLI,IPL)
      IF(XLI.LE.XL) XL=XLI
      IF(XHI.GE.XH) XH=XHI
      IF(YLI.LE.YL) YL=YLI
      IF(YHI.GE.YH) YH=YHI
   36 CONTINUE
      REWIND WF1
      IF(IPR.NE.8.AND.IPR.NE.9) GO TO 40
      ITITLE=10H
      ITITLE=XTIT
      XTIT=YTIT
      YTIT=ITITLE
   40 CONTINUE
      CALL ERASE
      CALL BINITT
      CALL TERM(3,4096)
      CALL SLIMX(250,3995)
      CALL SLIMY(250,2500)
      CALL TOUTPT(27)
      CALL TOUTPT(96)
      CALL GETINFO(IPR,HEADER)
      CALL TOASCII (HEADER,30,IOUT)
      CALL CHRSIZ(2)
      CALL MOVABS (1415,3020)
      CALL HLABEL(30,IOUT)
      CALL TOASCII (YTIT,10,IOUT)
      IOUT(11)=32
      DO 45 I=1,10
      II=11-I
      IF (IOUT(II).NE.32) GO TO 47
   45 CONTINUE
      II=1
   47 CALL TOASCII (YUNIT,10,IOUT(II+2))
      CALL CHRSIZ(3)
```

```
      CALL MOVABS(5,2000)
      CALL VLABEL(II+10,IOUT)
      CALL TOASCII(XTIT,10,IOUT)
      IOUT(11)=32
      DO 55 I=1,10
      II=11-I
      IF (IOUT(II).NE.32) GO TO 57
   55 CONTINUE
      II=1
   57 CALL TOASCII(XUNIT,10,IOUT(II+2))
      CALL CHRSIZ(3)
      CALL MOVABS(1800,75)
      CALL HLABEL(II+11,IOUT)
      CALL TOASCII(MTIT,80,IOUT)
      CALL MOVABS(20,5)
      CALL HLABEL(80,IOUT)
      IF (ICATEG.NE.2 .AND. ICATEG.NE.9) GO TO 95
      ENCODE (30,85,HEADER) GAGO
   85 FORMAT (*GAGE-O ELEV. =* F7.3 * FT(MSL)* )
      CALL TOASCII(HEADER,30,IOUT)
      CALL MOVABS(3070,5)
      CALL HLABEL(30,IOUT)
   95 CONTINUE
      IF(IPR.EQ.1.OR.IPR.EQ.2.OR.IPR.EQ.4) GO TO 96
      IF(IPR.EQ.3) GO TO 96
      CALL DREAD (WF1,ICATEG,SUBLEVL)
   96 CALL NPTS(NNPTS)
      CALL DLIMX(XL,XH)
      CALL DLIMY(YL,YH)
      CALL CHECK (X,Y)
      CALL DSPLAY (X,Y)
      III=YEAR
      IF (TUNIT.EQ.4HDATE) III=IDATE
      IF(TUNIT.EQ.4H    ) GO TO 120
      ENCODE(40,105,HEADER) TUNIT,III,NNPTS
  105 FORMAT(A5,I6,5X,*NO. DATA POINTS =*,I4)
      GO TO 125
  120 ENCODE(40,115,HEADER) TUNIT,NNPTS
  115 FORMAT(A5,5X,*NO. DATA POINTS =*,I4)
  125 CALL TOASCII (HEADER,40,IOUT)
      CALL CHRSIZ(3)
      CALL MOVABS(1020,2960+18)
      CALL DRWABS(1320,2960+18)
      CALL MOVABS(1370,2960)
      CALL HLABEL(40,IOUT)
      IF (MOST.EQ.1) GO TO 300
      IYY=2960
      DO 250 I=2,MOST
      CALL TOUTPT(27)
      CALL TOUTPT(IDASH(I))
      CALL DREAD (WF1,ICATEG,SUBLEVL)
      CALL NPTS(NNPTS)
      CALL CPLOT(X,Y)
      III=YEAR
      IF (TUNIT.EQ.4HDATE) III=IDATE
      IF(TUNIT.EQ.4H    ) GO TO 130
      ENCODE(40,105,HEADER) TUNIT,III,NNPTS
      GO TO 135
  130 ENCODE(40,115,HEADER) TUNIT,NNPTS
  135 CALL TOASCII(HEADER,40,IOUT)
      CALL CHRSIZ(3)
      IYY=IYY-52
      CALL MOVABS(1020,IYY+18)
      CALL DRWABS(1320,IYY+18)
      CALL MOVABS (1370,IYY)
      CALL HLABEL(40,IOUT)
  250 CONTINUE
  300 CALL MOVABS(0,2600)
      CALL TINPUT(I)
      CALL ERASE   $   CALL ANMODE
      CALL TSEND
      IF(IPR.EQ.5) WF1=1
      RETURN
      END
```

SUBROUTINE GETINFO (IPR, HEADER)

```
      SUBROUTINE GETINFO(IPR,HEADER)
      INTEGER DF,SUBLEVL
      COMMON/ELEMT/ICATEG,LEVEL,SUBLEVL,JCOMWD(4),NWORD,DF
      COMMON/WORK/IN,W,WF1,WF2,WF3,ISET,ILOC,ITIM,IRETRIV,IPRO,IOUTP,
     1NODE,STNO,DIST,LATI,LONG,GAGO,NAME(4),STNA(3),SNAME(4),LTIT(8),
     2IY(8),ITEMP(7),IZ(2),NX,NY,NZ,FIXQ,NYRS,TYPE,IYEAR(100),GELEV,NCR
     3,IEND,LEAP,TEMP(366)
      DIMENSION HEADER(3)
C
      IF(IPR.EQ.5) GO TO 95
      IF(IPR.EQ.1) GO TO 150
      IF(IPR.EQ.2) GO TO 160
      IF(IPR.EQ.3) GO TO 170
      IF(IPR.EQ.4) GO TO 200
      GO TO (100,20,100,100,50,100,70,80,85,90), ICATEG
   20 HEADER(1)=10HSTAGE-DISC
      HEADER(2)=10HHARGE PLOT
      HEADER(3)=10H

      RETURN

   50 HEADER(1)=10HRIVER CROS
      HEADER(2)=10HS-SECTION
      HEADER(3)=10H
      RETURN
   70 IF(SUBLEVL.EQ.5HRULE ) GO TO 72
       IF(SUBLEVL.EQ.5HSPILL) GO TO 74
       IF(SUBLEVL.EQ.5HCAPAC) GO TO 76
       GO TO 100
   72 HEADER(1)=10HRESERVOIR
      HEADER(2)=10HOPERATN -
      HEADER(3)=10HRULE CURVE
      GO TO 100
   74 HEADER(1)=10HRESERV.OPE
      HEADER(2)=10HRATN - SPI
      HEADER(3)=10HLLWAY CURV
      GO TO 100
   76 HEADER(1)=10HRESERV.OPE
      HEADER(2)=10HRATN - CAP
      HEADER(3)=10HACITY CURV
      GO TO 100

   80 HEADER(1)=10HDISCHARGE-
      HEADER(2)=10HHYDROGRAPH
      HEADER(3)=10H PLOT
      RETURN

   85 HEADER(1)=10HSTAGE-HYDR
      HEADER(2)=10HOGRAPH PLO
      HEADER(3)=10HT
      RETURN
   90 HEADER(1)=10HDAILY TOTA
      HEADER(2)=10HL RAINFALL
      HEADER(3)=10H PLOT
      RETURN

   95 HEADER(1)=10HDAILY CUMU
      HEADER(2)=10HLATIVE RAI
      HEADER(3)=10HNFALL PLOT
  100 RETURN
  150 HEADER(1)=10HCUMULATIVE
      HEADER(2)=10H FREQUENCY
      HEADER(3)=10H CURVE
      RETURN
  160 HEADER(1)=10HRELATIVE F
      HEADER(2)=10HREQUENCY H
      HEADER(3)=10HISTOGRAM
```

```
      RETURN
  170 HEADER(1)=10HTHALWEG LE
      HEADER(2)=10HVEL PLOT
      HEADER(3)=10H
      RETURN
  200 ENCODE(30,205,HEADER) FIXQ
  205 FORMAT(*CHANGING STAGE FOR Q=*,F6.0,*CFS*)
      RETURN
      END
```

SUBROUTINE TDASCII (IARY, NCHAR, NEW)

```
      SUBROUTINE TOASCII(IARY,NCHAR,NEW)
C
      INTEGER X
      DIMENSION ITABL(52),IARY(10),NEW(NCHAR)
      DATA (ITABL(I),I=1,52)/65,66,67,68,69,70,71,72,73,74,75,76,77,78,
     1                       79,80,81,82,83,84,85,86,87,88,89,90,48,49,
     2                       50,51,52,53,54,55,56,57,43,45,42,47,40,41,
     3                       36,61,32,44,46,34,33,64,63,59/
C
      K=0
      DO 30 I=1,10
      X=IARY(I)
      DO 20 J=1,10
      X=SHIFT(X,6)
      IX=X .AND. 77B
      K=K+1
      NEW(K)=ITABL(IX)
      IF(K .EQ. NCHAR) RETURN
   20 CONTINUE
   30 CONTINUE
C
      RETURN
      END
```