ROTINRA.COM

BOIWRA.COM

```
$SAY
      ************************************************************

        THIS PROCEDURE SELECTS NECESSARY DATA FROM HISTORY AND
        ALLOCATIONS FILES, THEN RUNS DAILY BOISE RIVER WATER
        RIGHT ACCOUNTING FOR CORRESPONDING PERIOD.

      ************************************************************

$INQUIRE SEL "TO SELECT DATA FOR ACCOUNTING PERIOD ENTER YES OR NO (Y/N)"
$IF .NOT. SEL THEN GOTO 300
$ ASSIGN BOIWRA.HST HISTORY
$ ASSIGN BOIWRA.HSD HISTDIV
$ ASSIGN BOIWRA.HSF HISTFLO
$ ASSIGN BOIWRA.HSR HISTRVR
$ ASSIGN BOIWRA.ALD ALOCDIV
$ ASSIGN BOIWRA.ALR ALOCRVR
$ ASSIGN BOIWRA.ALM ALOCMIS
$ ASSIGN BOIWRA.ALC ALOCTNS
$ ASSIGN/USER TT: SYS$INPUT
$ RUN WRASEL
$SAY
      ************************************************************

        FILES ARE NOW BEING SORTED FOR PROPER SEQUENCE AS INPUT TO
        ACCOUNTING PROGRAM.

      ************************************************************
$SORT BOIWRA.HSD BOIWRA.HSD-
/KEY=(POSITION:10,SIZE:7)/KEY=(POSITION:1,SIZE:8)
$SORT BOIWRA.HSF BOIWRA.HSF-
/KEY=(POSITION:10,SIZE:7)/KEY=(POSITION:1,SIZE:8)
$SORT BOIWRA.HSR BOIWRA.HSR-
/KEY=(POSITION:10,SIZE:7)/KEY=(POSITION:1,SIZE:8)
$SORT BOIWRA.ALD BOIWRA.ALD-
/KEY=(POSITION:10,SIZE:7)/KEY=(POSITION:1,SIZE:8)
$SORT BOIWRA.ALR BOIWRA.ALR-
/KEY=(POSITION:10,SIZE:7)/KEY=(POSITION:1,SIZE:8)
$SORT BOIWRA.ALM BOIWRA.ALM-
/KEY=(POSITION:10,SIZE:7)/KEY=(POSITION:1,SIZE:8)
$SAY
      ************************************************************

        FILES ARE SORTED AND OLD FILES WILL NOW BE DELETED.

      ************************************************************
$ PU BOIWRA.HSD
$ PU BOIWRA.HSF
$ PU BOIWRA.HSR
$ PU BOIWRA.ALD
$ PU BOIWRA.ALR
$ PU BOIWRA.ALM
$300:
$INQUIRE CHG "TO ENTER ALL NEW CANAL/SYSTEM STORAGE DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 320
$INQUIRE CHGS "TO ENTER ALL NEW SYSTEM STORAGE DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHGS THEN GOTO 310
$@BOICHGS
$310:
$INQUIRE CHGC "TO ENTER ALL NEW CANAL STORAGE DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHGC THEN GOTO 345
$@BOICHGC
$320:
$INQUIRE CHG "TO CHANGE ONE OR TWO CANAL/SYSTEM STORAGE DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 345
$INQUIRE CHGS "TO CHANGE ONE OR TWO SYSTEM STORAGE DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHGS THEN GOTO 340
$@BOIACHGS
$340:
$INQUIRE CHGC "TO CHANGE ONE OR TWO CANAL STORAGE DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHGC THEN GOTO 345
$@BOIACHGC
$345:
$INQUIRE WRA "TO RUN BOISE WATER RIGHT ACCOUNTING ENTER YES OR NO (Y/N)"
$IF .NOT. WRA THEN GOTO 100
$SAY

      ************************************************************

        ACCOUNTING PROGRAM WILL NOW BEGIN.

      ************************************************************
```

340 ← Change this sometime RJS 17FEB 88

CHANGED 4.22.97 DLL

```
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOIWRA
$GOTO 200
$100:
$SAY

      ************************************************************

        ACCOUNTING PROGRAM NOT RUN.

      ************************************************************

$GOTO 350
$200:
$SAY
      ************************************************************

        ACCOUNTING PROGRAM COMPLETED.  OUTPUT IN BOIWRA.RPT.

      ************************************************************
$350:
$SAY

      ************************************************************

        IN ORDER FOR TOTALS FROM THIS RUN TO BE CARRIED TO THE
        NEXT RUN, THEY MUST BE PLACED IN THE ALLOCATIONS FILE.

      ************************************************************

$INQUIRE ALC "TO PUT PREVIOUS RUN'S TOTALS IN ALLOCATIONS ENTER YES OR NO (Y/N)"
$IF .NOT. ALC THEN GOTO 400
$@BOIALCUP
$400:
$SAY

      ************************************************************

               IF YOU WANT A PRINTOUT OF THE ACCOUNTING,
               THERE ARE TWO OPTIONS FOR PRINT LOCATION:

                          1.  YOUR PRINTER
                          2.  IDWR VAX

      ************************************************************

$INQUIRE PRNT "TO PRINT REPORT ON YOUR PRINTER ENTER YES OR NO (Y/N)"
$IF .NOT. PRNT THEN GOTO 250
$SAY
      ************************************************************

              SET YOUR TERMINAL FOR PRINTER OUTPUT.

      ************************************************************

$INQUIRE RDY "WHEN PRINTER IS READY ENTER YES OR NO (Y/N)"
$IF .NOT. RDY THEN GOTO 250
$S132
$TY BOIWRA.RPT
$TY DUMMY.DAT
$S80
$250:
$INQUIRE VAX "TO PRINT REPORT ON IDWR VAX ENTER YES OR NO (Y/N)"
$IF .NOT. VAX THEN GOTO 450
$PA BOIWRA.RPT
$450:
```

BOIDENT.COM

BOIDENT.COM

BOIDENT.COM

```
$50:
$INQUIRE BDF "TO MAKE A FILE OF NEW DATA ENTER YES OR NO (Y/N). "
$IF .NOT. BDF THEN GOTO 75
$ASSIGN/USER TT: SYS$INPUT
$RUN BOIDLYFLW
$75:
$INQUIRE SRT1 "TO SORT DATA ENTRY FILE ENTER YES OR NO (Y/N). "
$IF .NOT. SRT1 THEN GOTO 200
$SAY

      ************************************************************
                DATA ENTRY FILE IS BEING SORTED
      ************************************************************

$SORT BOIDLYFLW.OUT BOIDLYFLW.OUT-
/KEY=(POSITION:1,SIZE:8)/KEY=(POSITION:10,SIZE:7)/STABLE
$200:
$INQUIRE SRT2 "TO SORT OLD HISTORY FILE ENTER YES OR NO (Y/N). "
$IF .NOT. SRT2 THEN GOTO 300
$SAY

      ************************************************************
                OLD HISTORY FILE IS BEING SORTED
      ************************************************************

$SORT BOIWRA.HST BOIWRA.HST-
/KEY=(POSITION:1,SIZE:8)/KEY=(POSITION:10,SIZE:7)
$300:
$SAY

       BEFORE UPDATING HISTORY FILE YOU MAY WANT TO CREATE A
       BACKUP FILE.

$INQUIRE BKUP "TO CREATE A NEW BACKUP FILE ENTER YES OR NO (Y/N). "
$IF .NOT. BKUP THEN GOTO 310
$COPY BOIWRA.HST BOIHST.BKU
$SAY

       BACKUP FILE CREATED

       THE FOLLOWING BACKUP FILES ARE NOW IN YOUR DIRECTORY.  DELETE
       AS MANY AS POSSIBLE BY ENTERING THE NUMBER TO BE KEPT.

$DIR/DAT/SIZ BOIHST.BKU
$INQUIRE KF  "       HOW MANY DO YOU WANT TO KEEP?  "
$PU/KEEP='KF' BOIHST.BKU
$WRITE SYS$OUTPUT " "
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT "              ",KF," BACKUP HISTORY FILES RETAINED"
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT " "
$310:
$INQUIRE UPD "TO PUT NEW DATA IN HISTORY FILE ENTER YES OR NO (Y/N). "
$IF .NOT. UPD THEN GOTO 400
$ASSIGN BOIWRA.HST OLDHIST
$ASSIGN BOIDLYFLW.OUT ENTRY
$ASSIGN BOIWRANEW.HST NEWHIST
$ASSIGN/USER TT: SYS$INPUT
$RUN BOIHISTUP
$REN BOIWRANEW.HST BOIWRA.HST
$PU/KEEP=3 BOIDLYFLW.OUT
$SAY

       THE FOLLOWING HISTORY FILES ARE NOW IN YOUR DIRECTORY.  DELETE
       AS MANY AS POSSIBLE BY ENTERING THE NUMBER TO BE KEPT.

$DIR/DAT/SIZ BOIWRA.HST
$325:
$INQUIRE KH "       HOW MANY DO YOU WANT TO KEEP?  "
$IF 'KH'.GE.2 THEN GOTO 350
$SAY

       YOU SHOULD PROBABLY KEEP AT LEAST TWO HISTORY FILES.

$GOTO 325
$350:
$PU/KEEP='KH' BOIWRA.HST
$WRITE SYS$OUTPUT " "
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT "                      ",KH," HISTORY FILES RETAINED"
```

```
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT " "
$GOTO 450
$400:
$SAY

      ************************************************************
                HISTORY FILE NOT UPDATED WITH NEW DATA
      ************************************************************

$450:
$INQUIRE PRNT "TO LIST DATA, ENTER YES OR NO (Y/N). "
$IF .NOT. PRNT THEN GOTO 500
$@BOILST
$500:
$INQUIRE CONT "TO RETURN TO ENTER ADDITIONAL DATA, ENTER YES OR NO (Y/N). "
$IF CONT THEN GOTO 50
$SAY

      ************************************************************

           DATA ENTRY AND HISTORY FILE UPDATE ROUTINES COMPLETED

      ************************************************************

$INQUIRE WRA "TO RUN WATER RIGHT ACCOUNTING, ENTER YES OR NO (Y/N). "
$IF .NOT. WRA THEN GOTO 550
$@BOIWRA
$550:
$GOTO 600
$SAY

      ************************************************************

           WATER RIGHT ACCOUNTNG NOT RUN, DATA ENTRY COMPLETED

      ************************************************************

$600:
```

BOIALCUP.COM

BOIALCUP.COM

```
$100:
$SAY

      ************************************************************
             NEW ALLOCATIONS DATA INPUT FILE IS BEING SORTED
      ************************************************************

$SORT BOIWRA.NAL BOIWRA.NAL-
/KEY=(POSITION:1,SIZE:8)/KEY=(POSITION:10,SIZE:7)
$SAY

      IF NEW ALLOCATIONS ARE TO REPLACE ALL DATA IN OLD ALLOCATIONS
      FILE, OR IF THIS IS THE FIRST RUN OF A NEW YEAR, THERE IS NO
      NEED TO UPDATE THE OLD FILE.  THE OLD FILE CAN BE DISCARDED.

$INQUIRE NFILE "TO DISCARD OLD ALLOCATIONS FILE ENTER YES OR NO (Y/N). "
$IF NFILE THEN GOTO 350
$SAY

      ************************************************************
               OLD ALLOCATIONS FILE IS BEING SORTED
      ************************************************************

$SORT BOIWRA.ALC BOIWRA.ALC-
/KEY=(POSITION:1,SIZE:8)/KEY=(POSITION:10,SIZE:7)
$SAY

       BEFORE UPDATING ALLOCATIONS FILE YOU MAY WANT TO CREATE A
       BACKUP FILE.

$INQUIRE BKUP "TO CREATE A NEW BACKUP FILE ENTER YES OR NO (Y/N). "
$IF .NOT. BKUP THEN GOTO 310
$COPY BOIWRA.ALC BOIALC.BKU
$SAY

       BACKUP FILE CREATED

       THE FOLLOWING BACKUP FILES ARE NOW IN YOUR DIRECTORY.  DELETE
       AS MANY AS POSSIBLE BY ENTERING THE NUMBER TO BE KEPT.

$DIR/DAT/SIZ BOIALC.BKU
$INQUIRE KF  "       HOW MANY DO YOU WANT TO KEEP?  "
$PU/KEEP='KF' BOIALC.BKU
$WRITE SYS$OUTPUT " "
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT "            ",KF," BACKUP ALLOCATIONS FILES RETAINED"
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT " "
$GOTO 310
$350:
$COPY BOIEMPTY.DAT SNKWRA.ALC
$310:
$INQUIRE UPD "TO PUT NEW DATA IN ALLOCATIONS FILE ENTER YES OR NO (Y/N). "
$IF .NOT. UPD THEN GOTO 400
$ASSIGN BOIWRA.ALC OLDALC
$ASSIGN BOIWRA.NAL INPUT
$ASSIGN BOIWRANEW.ALC NEWALC
$ASSIGN/USER TT: SYS$INPUT
$RUN BOIALCUP
$REN BOIWRANEW.ALC BOIWRA.ALC
$PU/KEEP=3 BOIWRA.NAL
$SAY

       THE FOLLOWING ALLOCATIONS FILES ARE NOW IN YOUR DIRECTORY.  DELETE
       AS MANY AS POSSIBLE BY ENTERING THE NUMBER TO BE KEPT.

$DIR/DAT/SIZ BOIWRA.ALC
$380:
$INQUIRE KH "       HOW MANY DO YOU WANT TO KEEP?  "
$IF 'KH'.GE.2 THEN GOTO 385
$SAY

       YOU SHOULD PROBABLY KEEP AT LEAST TWO ALLOCATIONS FILES

$GOTO 380
$385:
$PU/KEEP='KH' BOIWRA.ALC
```

```
$WRITE SYS$OUTPUT " "
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT "               ",KH," ALLOCATIONS FILES RETAINED"
$WRITE SYS$OUTPUT "      ******************************************************"
$WRITE SYS$OUTPUT " "
$GOTO 450
$400:
$SAY

      ************************************************************
                ALLOCATIONS FILE NOT UPDATED WITH NEW DATA
      ************************************************************

$450:
$SAY

      ************************************************************

                ALLOCATIONS FILE UPDATE ROUTINE COMPLETED

      ************************************************************
```

BOTLIST.COM

BOILST.COM

```
$25:
$SAY
      ************************************************************

               THIS ROUTINE LISTS DATA FROM HISTORY FILE.

        TO MAKE ADDITIONAL COPIES OF A DATA PREVIOUSLY LISTED,
          ANSWER NO TO THE FOLLOWING QUESTION, OTHERWISE NEW
         LISTINGS WILL BE MADE ACCORDING TO YOUR INSTRUCTIONS.

      ************************************************************

$INQUIRE PRE "TO MAKE NEW LISTINGS OF DATA ENTER YES OR NO (Y/N)"
$IF .NOT. PRE THEN GOTO 50
$ASSIGN/USER TT: SYS$INPUT
$IF P1.EQS. "" THEN GOTO 30
$ASSIGN P1 INPUTDATA
$GOTO 35
$30:
$ASSIGN BOIWRA.HST INPUTDATA
$35:
$RUN BOIHSTLST
$50:
$SAY

      ************************************************************

               THERE ARE THREE OPTIONS FOR PRINT LOCATIONS:

                          1.  YOUR PRINTER
                          2.  IDWR VAX
                          3.  IDWR LASER

      ************************************************************

$INQUIRE PRNT "TO PRINT DATA ON YOUR PRINTER ENTER YES OR NO (Y/N)"
$IF .NOT. PRNT THEN GOTO 100
$SAY
      ************************************************************

              SET YOUR TERMINAL FOR PRINTER OUTPUT.

      ************************************************************

$INQUIRE RDY "WHEN PRINTER IS READY ENTER YES OR NO (Y/N)"
$IF .NOT. RDY THEN GOTO 300
$S132
$TY BOIHSTLST.OUT
$TY DUMMY.DAT
$S80
$GOTO 300
$100:
$INQUIRE VAX "TO PRINT DATA ON IDWR VAX ENTER YES OR NO (Y/N)"
$IF .NOT. VAX THEN GOTO 200
$PA BOIHSTLST.OUT
$GOTO 300
$200:
$INQUIRE LAS "TO PRINT DATA ON IDWR LASER ENTER YES OR NO (Y/N)"
$IF .NOT. LAS THEN GOTO 300
$@LASERLNDR BOIHSTLST.OUT
$300:
$PU BOIHSTLST.OUT
$INQUIRE END "TO LIST ADDITIONAL STATIONS ENTER YES OR NO (Y/N)"
$IF END THEN GOTO 25
$SAY
      ************************************************************
              LISTING ROUTINE COMPLETED
      ************************************************************
```

```
$25:
$SAY
      ***************************************************************

                THIS ROUTINE LISTS DATA FROM HISTORY FILE.

         TO MAKE ADDITIONAL COPIES OF A DATA PREVIOUSLY LISTED,
           ANSWER NO TO THE FOLLOWING QUESTION, OTHERWISE NEW
          LISTINGS WILL BE MADE ACCORDING TO YOUR INSTRUCTIONS.

      ***************************************************************

$INQUIRE PRE "TO MAKE NEW LISTINGS OF DATA ENTER YES OR NO (Y/N)"
$IF.NOT.PRE THEN GOTO 50
$IF P1.EQS."" THEN GOTO 30
$ASSIGN 'P1' INPUTDATA
$GOTO 35
$30:
$ASSIGN BOIWRA.HST INPUTDATA
$35:
$SAY
      ***************************************************************

           YOU HAVE THE OPTION TO LIST DATA IN IRRIGATION YEAR
           OR WATER YEAR.  ANSWER YES TO THE FOLLOWING QUESTION
           FOR WATER YEAR LISTINGS OTHERWISE LISTINGS WILL BE
           MADE FOR THE IRRIGATION YEAR (NOVEMBER THRU OCTOBER).

      ***************************************************************

$INQUIRE WYR "      TO MAKE WATER YEAR LISTINGS ENTER YES OR NO (Y/N)"
$IF WYR THEN GOTO 40
$ASSIGN/USER TT: SYS$INPUT
$RUN BOIHSTLST
$GOTO 50
$40:
$ASSIGN/USER TT: SYS$INPUT
$RUN BOIHSTLSTWY
$50:
$SAY
      ***************************************************************

                THERE ARE THREE OPTIONS FOR PRINT LOCATIONS:

                            1.  YOUR PRINTER
                            2.  IDWR VAX
                            3.  IDWR LASER

      ***************************************************************

$INQUIRE PRNT "TO PRINT DATA ON YOUR PRINTER ENTER YES OR NO (Y/N)"
$IF .NOT. PRNT THEN GOTO 100
$SAY
      ***************************************************************

               SET YOUR TERMINAL FOR PRINTER OUTPUT.

      ***************************************************************

$INQUIRE RDY "WHEN PRINTER IS READY ENTER YES OR NO (Y/N)"
$IF .NOT. RDY THEN GOTO 300
$S132
$TY BOIHSTLST.OUT
$TY DUMMY.DAT
$S80
$100:
$INQUIRE VAX "TO PRINT DATA ON IDWR VAX ENTER YES OR NO (Y/N)"
$IF .NOT. VAX THEN GOTO 200
$PA BOIHSTLST.OUT
$200:
$INQUIRE LAS "TO PRINT DATA ON CODY'S LASER ENTER YES OR NO (Y/N)"
$IF .NOT. LAS THEN GOTO 300
$@BOILSR BOIHSTLST.OUT
$300:
$PU BOIHSTLST.OUT
```

REPLACE THIS FILE ON LEE'S DIR WHEN SURE IT WORKS
PA 9-26-90

```
$INQUIRE END "TO LIST ADDITIONAL STATIONS ENTER YES OR NO (Y/N)"
$IF END THEN GOTO 25
$SAY
      ************************************************************
                LISTING ROUTINE COMPLETED
      ************************************************************
```

BOICHGC.COM

```
$!
$!      ************************************************************
$!
$!        THIS PROCEDURE EXECUTES PROGRAM TO CHANGE CANAL STORAGE
$!        VALUES ON DATA SET BOICAN.IND FOR INPUT TO BOISE RIVER
$!        ACCOUNTING PROGRAM.  RJS JUN 1987.
$!
$!      ************************************************************
$!
$ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGC
$REN BOICAN.TMP BOICAN.IND
$PU BOICAN.IND
```

```
$!
$!      ************************************************************
$!
$!        THIS PROCEDURE EXECUTES PROGRAM TO CHANGE RESERVOIR STORAGE
$!        VALUES AND SYSTEM ACCUMULATED TOTALS ON DATA SET BOISYS.IND
$!        FOR INPUT TO BOISE RIVER ACCOUNTING PROGRAM.  RJS JUN 1987.
$!
$!      ************************************************************
$!
$ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGS
$REN BOISYS.TMP BOISYS.IND
$PU BOISYS.IND
```

BOISTO.com

```
$SAY
      ***************************************************************

        THIS PROCEDURE SUMMARIZES BOISE RIVER STORAGE ACCOUNTS

      ***************************************************************

$100:
$INQUIRE CHG "TO ENTER NEW RESERVOIR FILL DATA ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 150
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGSI
$REN BOISTO.TMP BOISTO.IND
$PU/KEEP=2 BOISTO.IND
$150:
$INQUIRE CHG "TO CHANGE RESERVOIR SPACE ALLOCATIONS ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 200
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGSS
$REN BOISTO.TMP BOISTO.SPA
$PU/KEEP=2 BOISTO.SPA
$200:
$INQUIRE CHG "TO ENTER OR CHANGE CARRYOVER STORAGE ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 225
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGSC
$REN BOISTO.TMP BOISTO.CRY
$PU/KEEP=2 BOISTO.CRY
$225:
$INQUIRE CHG "TO ENTER OR CHANGE STORAGE TRANSFERS ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 250
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGST
$REN BOISTO.TMP BOISTO.CRY
$PU/KEEP=2 BOISTO.CRY
$250:
$INQUIRE CHG "TO ENTER OR CHANGE WTR BANK TRANSACTIONS ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 255
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGWB
$REN BOISTO.TMP BOISTO.CRY
$PU/KEEP=2 BOISTO.CRY
$255:
$INQUIRE GET "TO GET STORAGE USE FROM LATEST ACCOUNTING ENTER YES OR NO (Y/N)"
$IF .NOT. GET THEN GOTO 260
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOIGETSU
$260:
$INQUIRE CHG "TO ENTER OR CHANGE STORAGE USE ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 270
$ ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGSU
$REN BOISTO.TMP BOISTO.USE
$PU/KEEP=2 BOISTO.USE
$270:
$INQUIRE CHG "TO ENTER UNACCOUNTED STORAGE USE ENTER YES OR NO (Y/N)"
$IF .NOT. CHG THEN GOTO 275
$ASSIGN/USER TT: SYS$INPUT
$RUN BOICHGUN
$REN BOISTO.TMP BOISTO.UNC
```

```
$PU/KEEP=2 BOISTO.UNC
$275:
$INQUIRE STO "TO COMPUTE BOISE STORAGE ACCOUNTS ENTER YES OR NO (Y/N)"
$IF .NOT. STO THEN GOTO 300
  \SSIGN/USER TT: SYS$INPUT
$RUN BOISTO
$GOTO 350
$300:
$SAY

      ***************************************************************

          BOISE RIVER RESERVOIR STORAGE ACCOUNT PROGRAM NOT RUN.

      ***************************************************************

$GOTO 400
$350:
$SAY
      ***************************************************************

           BOISE RIVER RESERVOIR STORAGE ACCOUNT PROGRAM COMPLETED,
           - OUTPUT IN BOISTO.RPT

      ***************************************************************
$400:
$SAY

      ***************************************************************

                IF YOU WANT A PRINTOUT OF THE STORAGE ACCOUNTS,
                THERE ARE THREE OPTIONS FOR PRINT LOCATION:

                          1.  YOUR PRINTER
                          2.  IDWR VAX
                          3.  IDWR LASER

      ***************************************************************

$INQUIRE PRNT "TO PRINT REPORT ON YOUR PRINTER ENTER YES OR NO (Y/N)"
$IF .NOT. PRNT THEN GOTO 450
$SAY
      ***************************************************************

               SET YOUR TERMINAL FOR PRINTER OUTPUT.

      ***************************************************************

$INQUIRE RDY "WHEN PRINTER IS READY ENTER YES OR NO (Y/N)"
$IF .NOT. RDY THEN GOTO 450
$S132
$TY BOISTO.RPT
$TY DUMMY.DAT
$S80
$450:
$INQUIRE VAX "TO PRINT REPORT ON IDWR VAX ENTER YES OR NO (Y/N)"
$IF .NOT. VAX THEN GOTO 500
$PA BOISTO.RPT
$500:
$INQUIRE LAS "TO PRINT REPORT ON IDWR LASER ENTER YES OR NO (Y/N)"
```

```
$IF .NOT. LAS THEN GOTO 550
$@LASERLNDF BOISTO.RPT
$550:
$SAY
      ***************************************************************

             STORAGE ACCOUNT PROCEDURE IS FINISHED

      ***************************************************************
```

GETDATA.COM

GETDATA.COM

```
$       SET NOON
$       !
$       !       GET DATA FROM HYDROMET AND BUILD FILE FOR THE
$       !       TRANSFER OF THE HYDROMET DATA TO THE STATE WRD
$       !
$ IF P1 .EQS. "" THEN INQUIRE P1 "HOW MANY DAYS DATA DO YOU WANT?"
$ IF P2 .EQS. "" THEN INQUIRE P2 "LAST DAY?   EXAMPLE: SEP1"
$ IF P3 .EQS. "" THEN INQUIRE P3 "WATER YEAR?   EXAMPLE: 1995"
$       !
$       !       SET UP THE NDAYS FILE IF P1 IS SPECIFIED
$       !
$       OPEN/WRITE TEMP NDAYS.ACP
$       WRITE TEMP "NDAYS="'P1'
$       CLOSE TEMP
$       SET PROT NDAYS.ACP
$       !
$       IF P2 .EQS. "" THEN GOTO ONE
$       !
$       !       SET DAY COMMAND FILE FOR EARLIER DAY
$       !
$       OPEN/WRITE TEMP DAY.ACP
$       IF P3.NES."" THEN  WRITE TEMP "WATER_YEAR="'P3'
$       WRITE TEMP "DAY="'P2'
$       CLOSE TEMP
$       SET PROT DAY.ACP
$       !
$       !       POLL ARCHIVE FILE ALSO GET DIVERSIONS FOR EARLIER
$       !       DAYS IF P1 AND/OR P2 ARE GIVEN
$       !
$ONE:
$       OLDARC
OPEN 76
FORMAT=2
FILE=DAY
FILE=NDAYS
FILE=LONG
EX
$       ASSIGN/USER NL: SYS$ERROR
$       ASSIGN/USER NL: SYS$OUTPUT
$       ASSIGN/USER NL: SYS$COMMAND
$       DELE SHORT.ACP;*,NDAYS.ACP;*,DAY.ACP;*
$       !
$       !       POLL DAYFILE
$       !
$       !       IF P2 IS SPECIFIED SKIP DAYFILE POLL
$       IF P2 .NES. "" THEN GOTO SKIPS
$       OLDDAY
OPEN 76
FORMAT=2
FILE=SHORT
EX
$SKIPS:
$       !
$       !       SORT INTO ORDER BY USGS # AND DATE
$       !
$ ASS/USER NL: SYS$OUTPUT
$ SORT/KEY=(POS:10,SIZE:8)/KEY=(POS:1,SIZE:8)/STABLE FOR076 FOR077
$ DELETE FOR076.DAT;*
$       !
$       !       REMOVE MISSING VALUES
$       !
$ RUN FILTER
$ PUR FOR077.DAT
$ TYPE SYS$INPUT
      FILTER HAS RUN
$ RENAME FOR076.DAT BOIHYDMET.OUT
$       !
$       !       REFORMAT DATA INTO STATE FORMAT
$       !
$ SORT BOIHYDMET.OUT BOIHYDMET.OUT-
/KEY=(POS:1,SIZE:8)/KEY=(POS:10,SIZE:7)/STABLE
$ TYPE SYS$INPUT
      HYDROMET DATA IS SORTED
$ RUN BOIFORMAT
$ TYPE SYS$INPUT
      BOIFORMAT HAS RUN
$ PUR BOIHYDMET.OUT
$ REN FOR007.DAT FOR001.DAT
$ RUN CHGDAT
$ SORT/KEY=(POS:1,SIZE:8)/KEY=(POS:10,SIZE:7) OUTFILE.DAT BOIDLYFLW.OUT
```

```
$ PUR FOR077.DAT
$ PUR FOR001.DAT
$ PUR BOIDLYFLW.OUT
$ PUR OUTFILE.DAT
$ @TOWD63 BOIDLYFLW.OUT
$ TYPE SYS$INPUT
   THE BOISE RIVER DATA HAS BEEN SENT TO WDIST63
$END:
$       EXIT
```

BOTHSTPUR.COM

```
$SAY
      ***************************************************************

        THIS PROCEDURE SELECTS DATA FROM HISTORY FILE TO CREATE
        AN ARCHIVE FILE OF LAST YEAR'S DATA AND A NEW FILE FOR
        NEXT YEAR WITH ALL DATA BEFORE OCTOBER 1 PURGED.

      ***************************************************************

$INQUIRE CRT "TO CREATE THESE HISTORY FILES ENTER YES OR NO (Y/N)"
$IF .NOT. CRT THEN GOTO 100
$INQUIRE YR "ENTER THE IRRIGATION YEAR TO BE ARCHIVED, EXAMPLE: 88 "
$ ASSIGN BOIWRA.HST HISTORY
$ ASSIGN BOIA'YR'.HST OLDHIST
$ ASSIGN BOIWRA.HSN NEWHIST
$ ASSIGN/USER TT: SYS$INPUT
$ RUN BOIHSTPUR
$SAY
      ***************************************************************

        ARCHIVE FILE OF HISTORY DATA IS IN FILE BOIA??.HST WHERE
        ?? IS THE YEAR.  NEW HISTORY FILE IS NAMED BOIWRA.HSN.
        THIS FILE SHOULD BE RENAMED TO BOIWRA.HST AS SOON AS IT
        IS DETERMINED THAT THIS PROCEDURE WORKED OK.

                          >>>>>>CAUTION<<<<<<

        BE SURE TO CHECK ARCHIVED FILE IMMEDIATELY FOR CORRECTNESS,
        OTHERWISE THIS DATA MAY BE LOST WHEN THE OLD BOIWRA.HST IS
        AUTOMATICALLY DELETED BY THE DATA ENTRY PROGRAM!!!!!!!!!!!

      ***************************************************************
$100:
```

BOTHSTPWR.COM

```
$25:
$SAY
      ************************************************************

          THIS ROUTINE LISTS AND SUMS DATA FROM HISTORY FILE.

        TO MAKE ADDITIONAL COPIES OF A DATA PREVIOUSLY SUMMED,
          ANSWER NO TO THE FOLLOWING QUESTION, OTHERWISE NEW
         LISTINGS WILL BE MADE ACCORDING TO YOUR INSTRUCTIONS.

      ************************************************************

$INQUIRE PRE "TO MAKE NEW LISTINGS OF SUMMED DATA ENTER YES OR NO (Y/N)"
$IF .NOT. PRE THEN GOTO 50
$ASSIGN/USER TT: SYS$INPUT
$IF P1 .EQS. "" THEN GOTO 30
$ASSIGN 'P1' INPUTDATA
$GOTO 35
$30:
$ASSIGN BOIWRA.HST INPUTDATA
$35:
$INQUIRE WYR "FOR WATER YR LISTING INSTEAD OF IRRIG YR ENTER YES OR NO (Y/N)"
$IF .NOT. WYR THEN GOTO 40
$RUN BOIHSTSUMW
$GOTO 50
$40:
$RUN BOIHSTSUM
$50:
$SAY
      ************************************************************

               THERE ARE THREE OPTIONS FOR PRINT LOCATIONS:

                          1.  YOUR PRINTER
                          2.  IDWR VAX
                          3.  IDWR LASER

      ************************************************************

$INQUIRE PRNT "TO PRINT DATA ON YOUR PRINTER ENTER YES OR NO (Y/N)"
$IF .NOT. PRNT THEN GOTO 100
$SAY
      ************************************************************

              SET YOUR TERMINAL FOR PRINTER OUTPUT.

      ************************************************************

$INQUIRE RDY "WHEN PRINTER IS READY ENTER YES OR NO (Y/N)"
$IF .NOT. RDY THEN GOTO 300
$S132
$TY BOIHSTSUM.OUT
$TY DUMMY.DAT
$S80
$GOTO 300
$100:
$INQUIRE VAX "TO PRINT DATA ON IDWR VAX ENTER YES OR NO (Y/N)"
$IF .NOT. VAX THEN GOTO 200
$PA BOIHSTSUM.OUT
$GOTO 300
$200:
$INQUIRE LAS "TO PRINT DATA ON IDWR LASER ENTER YES OR NO (Y/N)"
$IF .NOT. LAS THEN GOTO 300
$@LASERJETLNDF BOIHSTSUM.OUT
$300:
$PU/KEEP=6 BOIHSTSUM.OUT
$INQUIRE END "TO SUM ADDITIONAL STATIONS ENTER YES OR NO (Y/N)"
$IF END THEN GOTO 25
$SAY
      ************************************************************
             LISTING ROUTINE COMPLETED
      ************************************************************
```

BOTHISTUP 115

BOIHISTUP.LIS

```
C
C        **** PROGRAM TO ENTER RIVER DATA FROM DATA ENTRY FILES TO
C        PERMANENT HISTORY FILE.  IF DATA ALREADY EXISTS FOR STATION
C        AND DAY BEING ENTERED, OLD DATA WILL BE REPLACED WITH NEW
C        ENTRY. BOB SUTTER - OCTOBER 1986.
C
      REAL*8 VAOLD,VANEW,VANEW2
      INTEGER*4 ISTA,ISTA2,JSTA,IDATE,IDATE2,JDATE
C
      CALL ASSIGN(1,'ENTRY')
      CALL ASSIGN(2,'OLDHIST')
      OPEN(UNIT=3,NAME='NEWHIST',TYPE='NEW',
     1CARRIAGECONTROL='LIST')
C
C***** INITIALIZE INDICATORS
C
      JSKIP=0
      IREP=0
      INEW=0
      ISAM=0
      KEND=0
C
C***** READ TWO  RECORDS FROM FROM ENTRY FILE AND ONE RECORD
C***** FROM OLD HISTORY FILE.  IF THERE ARE DUPLICATE RECORDS
C***** ON ENTRY FILE, USE ONLY FINAL ENTRY.  SKIP READING DATA
C***** FROM OLD HISTORY FILE IF LAST ENTRY DATA HAS BEEN MERGED
C***** WITH DATA FROM OLD HISTORY FILE.
C
      READ(1,1,END=10)ISTA,S,IDATE,VANEW
    1 FORMAT(I8,A1,I7,F7.0)
      GO TO 19
    2 IF(KEND.EQ.1) GO TO 10
      ISTA=ISTA2
      S=S2
      IDATE=IDATE2
      VANEW=VANEW2
   19 READ(1,1,END=17) ISTA2,S2,IDATE2,VANEW2
      IF(ISTA2.EQ.ISTA.AND.IDATE2.EQ.IDATE) GO TO 2
      GO TO 3
   17 KEND=1
    3 IF(JSKIP.EQ.0) READ(2,1,END=20)JSTA,T,JDATE,VAOLD
      JSKIP=0
C
C***** CHECK RELATIVE VALUE OF STATION NUMBERS OF OLD
C***** AND NEW ENTRY DATA.
C
      IF(JSTA-ISTA) 4,5,6
C
C***** CHECK RELATIVE VALUE OF DATES OF OLD AND NEW
C***** ENTRY DATA.
C
    5 IF(JDATE-IDATE) 4,7,6
C
C***** WRITE DATA FROM ENTRY FILE TO NEW HISTORY FILE TO
C***** EITHER REPLACE OR BE MERGED WITH OLD DATA.
C
    6 JSKIP=1
```

```
          INEW=INEW+1
          GO TO 9
        7 IREP=IREP+1
        9 IF(S.NE.'R') GO TO 23
          WRITE(3,25) ISTA,S,IDATE,JIDNNT(VANEW)
          GO TO 2
       23 WRITE(3,21) ISTA,S,IDATE,VANEW
          GO TO 2
C
C***** WRITE DATA RECORDS FROM OLD HISTORY FILE TO NEW
C***** HISTORY FILE UNTIL DATA FROM ENTRY FILE IS TO
C***** REPLACE OLD DATA OR BE MERGED WITH OLD DATA.
C
        4 ISAM=ISAM+1
          IF(T.NE.'R') GO TO 8
          WRITE(3,25) JSTA,T,JDATE,JIDNNT(VAOLD)
       25 FORMAT(I8,A1,2I7)
          GO TO 3
        8 WRITE(3,21) JSTA,T,JDATE,VAOLD
       21 FORMAT(I8,A1,I7,F7.1)
          GO TO 3
C
C***** WRITE REMAINDER OF OLD HISTORY FILE TO NEW FILE
C***** AFTER DATA ENTRY FILE HAS BEEN EXHAUSTED.
C
       10 IF(JSKIP.EQ.1) GO TO 14
       15 READ(2,1,END=100)JSTA,T,JDATE,VAOLD
       14 ISAM=ISAM+1
          IF(T.NE.'R') GO TO 13
          WRITE(3,25) JSTA,T,JDATE,JIDNNT(VAOLD)
          GO TO 15
       13 WRITE(3,21) JSTA,T,JDATE,VAOLD
          GO TO 15
C
C***** WRITE REMAINDER OF DATA ENTRY FILE TO NEW HISTORY
C***** FILE AFTER OLD HISTORY FILE HAS BEEN EXHAUSTED.
C
       22 IF(KEND.EQ.1) GO TO 100
       24 ISTA=ISTA2
          S=S2
          IDATE=IDATE2
          VANEW=VANEW2
          READ(1,1,END=37) ISTA2,S2,IDATE2,VANEW2
          IF(ISTA2.EQ.ISTA.AND.IDATE2.EQ.IDATE) GO TO 24
          GO TO 20
       37 KEND=1
       20 INEW=INEW+1
          IF(S.NE.'R') GO TO 16
          WRITE(3,25) ISTA,S,IDATE,JIDNNT(VANEW)
          GO TO 22
       16 WRITE(3,21) ISTA,S,IDATE,VANEW
          GO TO 22
C
C***** END OF ALL DATA SETS
C
      100 TYPE *, ' HISTORY FILE HAS NOW BEEN UPDATED '
          ITOT=IREP+ISAM+INEW
```

```
0115          TYPE *, ' '
0116          TYPE 11, IREP,INEW,ISAM,ITOT
0117       11 FORMAT(/2X,'UPDATED FILE HAS',I10,3X,'RECORDS REPLACED'/
0118         1 18X,I10,3X,'NEW RECORDS'/18X,I10,3X,'RECORDS TRANSFERRED'/
0119         2 4X,'FOR A TOTAL OF',I10,3X,'HISTORY FILE RECORDS')
0120          TYPE *, ' '
0121      500 STOP
0122          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1273	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	232	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	152	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	1657	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIHISTUP$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
2-00000024	I*4	IDATE		8 29= 35= 38 52 62 64 98= 101 106 108
2-00000028	I*4	IDATE2		8 35 37= 38 98 100= 101
**	I*4	INEW		19= 58(2)= 104(2)= 114 116
**	I*4	IREP		18= 60(2)= 114 116
**	I*4	ISAM		20= 71(2)= 85(2)= 114 116
2-00000018	I*4	ISTA		8 29= 33= 38 47 62 64 96= 101 106 108
2-0000001C	I*4	ISTA2		8 33 37= 38 96 100= 101
**	I*4	ITOT		114= 116
2-0000002C	I*4	JDATE		8 41= 52 73 76 84= 87 89
**	I*4	JSKIP		17= 41 42= 57= 83
2-00000020	I*4	JSTA		8 41= 47 73 76 84= 87 89
**	I*4	KEND		21= 32 40= 95 103=
2-00000030	R*4	S		29= 34= 61 62 64 97= 105 106 108
2-00000034	R*4	S2		34 37= 97 100=

2-00000038	R*4	T	41=	72	73	76	84=	86	
				87	89				
2-00000008	R*8	VANEW	7	29=	36=	62	64	99=	
				106	108				
2-00000010	R*8	VANEW2	7	36	37=	99	100=		
2-00000000	R*8	VAOLD	7	41=	73	76	84=	87	
				89					

LABELS

Address	Label	References					
1-00000042	1'	29	30#	37	41	84	100
0-000002CF	2	32#	38	63	65		
0-000000CB	3	39	41#	75	78		
0-000002F0	4	47	52	71#			
**	5	47	52#				
0-00000240	6	47	52	57#			
0-0000023C	7	52	60#				
0-00000340	8	72	76#				
0-00000245	9	59	61#				
0-00000384	10	29	32	83#			
1-0000005F	11'	116	117#				
0-000003D8	13	86	89#				
0-0000038C	14	83	85#				
0-00000416	15	84#	88	90			
0-00000168	16	105	108#				
0-000000C8	17	37	40#				
0-00000070	19	31	37#				
0-00000118	20	41	102	104#			
1-00000055	21'	64	76	77#	89	108	
0-000001A7	22	95#	107	109			
0-00000290	23	61	64#				
**	24	96#	101				
1-0000004C	25'	62	73	74#	87	106	
0-00000218	37	100	103#				
0-0000045B	100	84	95	113#			
**	500	121#					

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	10	11
	FOR$OPEN	12	

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|    =    - Value Modified                      |
|    #    - Defining Reference                  |
|    A    - Actual Argument, possibly modified  |
|    D    - Data Initialization                 |
|   (n)   - Number of occurrences on line       |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOIHISTUP

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
Run Time:           5.91 seconds
Elapsed Time:       7.18 seconds
Page Faults:        733
Dynamic Memory:     418 pages
```

```
C
C     **** PROGRAM TO ENTER RIVER DATA FROM DATA ENTRY FILES TO
C     PERMANENT HISTORY FILE.  IF DATA ALREADY EXISTS FOR STATION
C     AND DAY BEING ENTERED, OLD DATA WILL BE REPLACED WITH NEW
C     ENTRY. BOB SUTTER - OCTOBER 1986.
C
      REAL*8 VAOLD,VANEW,VANEW2
      INTEGER*4 ISTA,ISTA2,JSTA,IDATE,IDATE2,JDATE
C
      CALL ASSIGN(1,'ENTRY')
      CALL ASSIGN(2,'OLDHIST')
      OPEN(UNIT=3,NAME='NEWHIST',TYPE='NEW',
     1CARRIAGECONTROL='LIST')
C
C      TYPE *, ' ************************************************'
C      TYPE *, ' '
C      TYPE *, ' TO UPDATE HISTORY FILE WITH NEW DATA, HIT "RETURN",'
C      TYPE *, ' OTHERWISE ENTER "N". '
C      ACCEPT 12, IACT
C   12 FORMAT(A1)
C      IF(IACT.EQ.'N') GO TO 500
C
      JSKIP=0
      IREP=0
      INEW=0
      ISAM=0
      KEND=0
C
C***** READ ONE RECORD FROM FROM ENTRY FILE AND ONE RECORD
C***** FROM OLD HISTORY FILE.  SKIP READING DATA FROM OLD
C***** HISTORY FILE IF LAST ENTRY DATA HAS BEEN MERGED WITH
C***** DATA FROM OLD HISTORY FILE.
C
      READ(1,1,END=10)ISTA,S,IDATE,VANEW
    1 FORMAT(I8,A1,I7,F7.0)
      GO TO 19
    2 IF(KEND.EQ.1) GO TO 10
      ISTA=ISTA2
      S=S2
      IDATE=IDATE2
      VANEW=VANEW2
   19 READ(1,1,END=17) ISTA2,S2,IDATE2,VANEW2
      IF(ISTA2.EQ.ISTA.AND.IDATE2.EQ.IDATE) GO TO 2
      GO TO 3
   17 KEND=1
    3 IF(JSKIP.EQ.0) READ(2,1,END=20)JSTA,T,JDATE,VAOLD
      JSKIP=0
C
C***** CHECK RELATIVE VALUE OF STATION NUMBERS OF OLD
C***** AND NEW ENTRY DATA.
C
      IF(JSTA-ISTA) 4,5,6
C
C***** CHECK RELATIVE VALUE OF DATES OF OLD AND NEW
C***** ENTRY DATA.
C
    5 IF(JDATE-IDATE) 4,7,6
```

OLD, DISCARD IF WORKS

```
C
C***** WRITE DATA FROM ENTRY FILE TO NEW HISTORY FILE TO
C***** EITHER REPLACE OR BE MERGED WITH OLD DATA.
C
    6 JSKIP=1
      INEW=INEW+1
      GO TO 9
    7 IREP=IREP+1
    9 IF(S.NE.'R') GO TO 23
      WRITE(3,25) ISTA,S,IDATE,JIDNNT(VANEW)
      GO TO 2
   23 WRITE(3,21) ISTA,S,IDATE,VANEW
      GO TO 2
C
C***** WRITE DATA RECORDS FROM OLD HISTORY FILE TO NEW
C***** HISTORY FILE UNTIL DATA FROM ENTRY FILE IS TO
C***** REPLACE OLD DATA OR BE MERGED WITH OLD DATA.
C
    4 ISAM=ISAM+1
      IF(T.NE.'R') GO TO 8
      WRITE(3,25) JSTA,T,JDATE,JIDNNT(VAOLD)
   25 FORMAT(I8,A1,2I7)
      GO TO 3
    8 WRITE(3,21) JSTA,T,JDATE,VAOLD
   21 FORMAT(I8,A1,I7,F7.1)
      GO TO 3
C
C***** WRITE REMAINDER OF OLD HISTORY FILE TO NEW FILE
C***** AFTER DATA ENTRY FILE HAS BEEN EXHAUSTED.
C
   10 IF(JSKIP.EQ.1) GO TO 14
   15 READ(2,1,END=100)JSTA,T,JDATE,VAOLD
   14 ISAM=ISAM+1
      IF(T.NE.'R') GO TO 13
      WRITE(3,25) JSTA,T,JDATE,JIDNNT(VAOLD)
      GO TO 15
   13 WRITE(3,21) JSTA,T,JDATE,VAOLD
      GO TO 15
C
C***** WRITE REMAINDER OF DATA ENTRY FILE TO NEW HISTORY
C***** FILE AFTER OLD HISTORY FILE HAS BEEN EXHAUSTED.
C
   20 READ(1,1,END=100)ISTA,S,IDATE,VANEW
   23 INEW=INEW+1
      IF(S.NE.'R') GO TO 16
      WRITE(3,25) ISTA,S,IDATE,JIDNNT(VANEW)
      GO TO 20
   16 WRITE(3,21) ISTA,S,IDATE,VANEW
      GO TO 20

  100 TYPE *, ' HISTORY FILE HAS NOW BEEN UPDATED '
      ITOT=IREP+ISAM+INEW
      TYPE *, ' '
      TYPE 11, IREP,INEW,ISAM,ITOT
   11 FORMAT(/2X,'UPDATED FILE HAS',I10,3X,'RECORDS REPLACED'/
     1 18X,I10,3X,'NEW RECORDS'/18X,I10,3X,'RECORDS TRANSFERRED'/
     2 4X,'FOR A TOTAL OF',I10,3X,'HISTORY FILE RECORDS')
```

OLD

```
0115          TYPE *, ' '
0116      500 STOP
0117          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1225	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	232	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	152	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	1609	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIHISTUP$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
2-00000024	I*4	IDATE		8	34=	40=	43	57	67
					69	100=	103	105	
2-00000028	I*4	IDATE2		8	40	42=	43		
**	I*4	INEW		25=	63(2)=	101(2)=	109	111	
**	I*4	IREP		24=	65(2)=	109	111		
**	I*4	ISAM		26=	76(2)=	90(2)=	109	111	
2-00000018	I*4	ISTA		8	34=	38=	43	52	67
					69	100=	103	105	
2-0000001C	I*4	ISTA2		8	38	42=	43		
**	I*4	ITOT		109=	111				
2-0000002C	I*4	JDATE		8	46=	57	78	81	89=
					92	94			
**	I*4	JSKIP		23=	46	47=	62=	88	
2-00000020	I*4	JSTA		8	46=	52	78	81	89=
					92	94			
**	I*4	KEND		27=	37	45=			
2-00000030	R*4	S		34=	39=	66	67	69	100=
					102	103	105		
2-00000034	R*4	S2		39	42=				
2-00000038	R*4	T		46=	77	78	81	89=	91
					92	94			
2-00000008	R*8	VANEW		7	34=	41=	67	69	100=
					103	105			
2-00000010	R*8	VANEW2		7	41	42=			
2-00000000	R*8	VAOLD		7	46=	78	81	89=	92

LABELS

Address	Label	References					
1-00000042	1'	34	35#	42	46	89	100
0-0000029F	2	37#	43	68	70		
0-000000CB	3	44	46#	80	83		
0-000002C0	4	52	57	76#			
**	5	52	57#				
0-00000210	6	52	57	62#			
0-0000020C	7	57	65#				
0-00000310	8	77	81#				
0-00000215	9	64	66#				
0-00000354	10	34	37	88#			
1-0000005F	11'	111	112#				
0-000003A8	13	91	94#				
0-0000035C	14	88	90#				
0-000003E6	15	89#	93	95			
0-000001AC	16	102	105#				
0-000000C8	17	42	45#				
0-00000070	19	36	42#				
0-00000118	20	46	100#	104	106		
1-00000055	21'	69	81	82#	94	105	
0-00000260	23	66	69#				
1-0000004C	25'	67	78	79#	92	103	
0-0000042B	100	89	100	108#			
**	500	116#					

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	10	11
	FOR$OPEN	12	

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

OLD

COMMAND QUALIFIERS

```
  FORTRAN/LIST/CROSS_REFERENCE BOIHISTUP

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
  Run Time:           6.12 seconds
  Elapsed Time:       36.59 seconds
  Page Faults:        402
  Dynamic Memory:     418 pages
```

OLD

BOTWRA.115

```
C
C----SECTION 1-1
C****THIS PROGRAM WRITTEN BY IDAHO DEPARTMENT OF WATER RESOURCES IN
C****SEPTEMBER 1977 FOR THE PURPOSE OF DISTRIBUTING WATER RIGHTS BY
C****PRIORITY DATE ON RIVER SYSTEMS OF ANY REASONABLE CONFIGURATION
C
C****INPUT DATA AVAILABLE MUST INCLUDE RIVER DISCHARGES AT ALL SECTIONS
C****WHERE GAINS OR LOSSES TO THE RIVER ARE SIGNIFICANT, MEASUREMENTS
C****OF EACH DIVERSION FROM THE RIVER, AND A LIST OF ALL RIGHTS WITH
C****BOTH DATE AND AMOUNT.  TIME ELEMENT IS ONE DAY.
C
C****IF RIVER SYSTEM CONTAINS RESERVOIRS, AMOUNT OF DIVERSION IN
C****EXCESS OF NATURAL FLOW RIGHTS IS CONSIDERED USE OF STORED WATER.
C****USE OF STORED WATER IS DEDUCTED FROM BEGINNING OF SEASON STORED
C****WATER ALLOCATION TO KEEP A DAILY ACCOUNT OF REMAINING STORAGE.
C
C****FOR FURTHER INFORMATION CONTACT HYDROLOGY SECTION IDWR 334-4484.
C
C****THIS VERSION (WRD063) OF THE WATER RIGHT ACCOUNTING PROGRAM HAS
C****BEEN ADAPTED TO THE BOISE RIVER. FALL 1982.
C
      CHARACTER RUNDATE*9
C
C----SECTION  1-2, PART 1
C****DECLARE SPACE FOR ALL DIVERSION VARIABLES.
C
      DIMENSION DIVNM(75,5),IRCH(75),RD(75),DM(75),DIV(75),AD(75)
      DIMENSION IDN(75),STOR(75),DVV(75,366),SRM(75),JC(75)
      DIMENSION II(5,75),TDIV(75),DNAM(75,5)
C
C----SECTION  1-2, PART 2
C****DECLARE SPACE FOR ALL REACH VARIABLES.
C
      DIMENSION RF(25),F(25),IDS(25),TD(25),G(25,15),CTR(25),TRM(25)
      DIMENSION PCT(25),IORD(25),RCHNM(25,6),LAG(25),LGD(25)
      DIMENSION SF(25),AF(25),TR(25),TRD(25),RIN(25),IRS(25),ICF(25)
      DIMENSION IRT(25),IDA(25),GAIN(25),KNRC(25),JRN(24),MGA(25)
      DIMENSION AG(25),PF(25),TRE(25),DS(25),GAV(25,4),A(25,366)
C
C----SECTION  1-2, PART 3
C****DECLARE SPACE FOR ALL RESERVOIR VARIABLES.
C
      DIMENSION RES(4,2),RESNM(4,4),SAC(4),DRA(4),LGG(4),ARINT(4)
      DIMENSION EVAP(4),SA(11,4),EV(4),IRESN(4),PP(4),CEV(4)
      DIMENSION EVP(4,366),RSS(4,366),PPT(4,366)
C
C----SECTION  1-2, PART 4
C****DECLARE SPACE FOR ALL WATER RIGHT VARIABLES.
C
      DIMENSION P(850),ICNL(850),IDATE(850),T(850),JR(850),IV(850)
      DIMENSION SVL(850),IVO(850),ICN(850),KEN(850),KBG(850)
      DIMENSION JDATE(850),JCNL(850),JJ(850)
C
C----SECTION  1-2, PART 5
C****DECLARE SPACE FOR ALL RESERVOIR RIGHT VARIABLES.
C
      DIMENSION RST(4),BRST(4),MR(4)
```

```
C
C----SECTION  1-2, PART 6
C****DECLARE SPACE FOR ALL EXCHANGE WELL VARIABLES.
C
      DIMENSION IEN(1),IER(1),EXCNM(1,4),TSE(1),EXC(1,366),ECH(1)
C
C----SECTION  1-2, PART 7
C****DECLARE SPACE FOR ALL TIME AND OTHER MISC VARIABLES.
C
      DIMENSION OM(12),JD(12),JDT(366),JYE(366),JDY(366),ICFMD(366)
      DIMENSION FI(20),KOUT(3),GT(10),ISPEC(24),VRD(40),TT(20)
      DIMENSION ISTP(3),IBRP(4),ISFS(11),ARICD(12),AMF(12)
C
C----SECTION  2-1, PART 1
C****ASSIGN CHARACTERS FOR IDENTIFICATION OF ESTIMATED FLOWS.
C
      DATA BLANK/' '/,AST/'*'/
C
C----SECTION  2-1, PART 2
C****ISPEC = IDENTIFICATION NUMBERS OF CANALS FOR WHICH SPECIAL
C****COMPUTATIONS MUST BE MADE.
C
      DATA ISPEC/13205640,13205641,13206090,13206092,13209480,
     1 13209482,13210984,13210989,13202995,13203000,13201990,
     2 13201991,13210049,13206205,13206220,13206260,13206270,
     3 13206290,13206292,13206295,13208450,13208740,13210005,
     4 13205643/
C
C----SECTION  2-1, PART 3
C****EXTEND PRECISION OF VARIABLES WITH LARGE VALUES
C
      REAL*8 RST,BRST,RES,TT,SAC,RSS,P,T,CTR,TR,TRM,GT
C
C----SECTION  2-1, PART 4
C****SET MONTH NAMES AND DAYS PER MONTH
C
      DATA OM/'JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT
     1','NOV','DEC'/
      DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
C
C----SECTION  2-1, PART 5
C****SET MAXIMUM STORAGE POSSIBLE FOR EACH RESERVOIR STORAGE RIGHT,
C****UNITS ARE ACRE-FEET.
C
      DATA RST/260624.,11600.,453030.,264370./
C
C----SECTION  2-2, PART 1
C**** MR = RESERVOIR ORDER NUMBER CORRESPONDING TO STORAGE RIGHT IN
C**** PREVIOUS STATEMENT.
C
      DATA MR/2,2,1,3/
C
C----SECTION  2-2, PART 2
```

```
C****INITIALIZE DIVERSION MEASUREMENTS (DVV) TO ZERO.
C
      DATA DVV/27450*0.0/
      DATA EXC/366*0.0/
C
C----SECTION  2-2, PART 3
C**** SET IDENTIFICATION NUMBERS FOR ALL RESERVOIRS IN DOWNSTREAM ORDER.
C
      DATA IRESN/13190000,13194000,13201500,13203550/
C
C----SECTION  2-2, PART 4
C**** ARINT = INTERVAL BETWEEN VALUES IN CONTENT-SURFACE AREA CURVES,
C**** UNITS ARE 1000 ACRE-FEET.
C
      DATA ARINT/50.0,30.0,30.0,0.1/
C
C----SECTION  2-2, PART 5
C**** SA = SURFACE AREA IN 1000 ACRES OF EACH RESERVOIR AT ARINT
C**** INTERVALS.  THERE ARE ELEVEN VALUES FOR EACH RESERVOIR
C**** BEGINNING WITH FIRST RESERVOIR.
C
      DATA SA/1.100,1.700,2.200,2.700,3.100,3.500,3.900,4.200,4.600,
     1 4.900,4.900,0.000,0.800,1.300,1.600,1.900,2.100,2.300,2.500,
     1 2.700,3.000,3.100,0.800,1.200,1.500,1.700,1.900,2.100,2.300,
     1 2.500,2.600,2.800,2.900,0.027,0.041,0.058,0.071,0.078,0.083,
     1 0.087,0.091,0.095,0.098,0.103/
C
C----SECTION  2-2, PART 6
C****SET IDENTIFICATION NUMBERS FOR ALL REACHES
C
      DATA JRN/13185000,13186000,13190500,13194100,13200000,13202000,
     113203600,13205500,13206000,13210050,13211000,13212500,13213000,
     213210810,13210824,13210831,13210835,13210849,13210980,13210986,
     313210988,13211445,13212550,13212890/
C
C----SECTION  2-3, PART 1
C****IF MGA = 1 PROJECTED GAINS ARE CALCULATED BY SIMPLE AVERAGING.
C
      DATA MGA/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/
C
C----SECTION  2-3, PART 2
C****MD1 AND MD2 ARE THE BEGINNING AND ENDING DAY OF THE YEAR DURING
C****WHICH DATA MUST EXIST FOR ALL MEASURED DIVERSIONS.
C
      DATA MD1/91/,MD2/288/
C
C----SECTION  2-3, PART 3
C****INITIALIZE INDICATORS OF PER CENT FILL OF RIGHTS FOR STEWART
C**** (ISTP) AND BRYAN (IBRP) DECREES ACCORDING TO SLIDING SCALE,
C****AND FOR IDENTIFICATION NUMBERS (ISFS) OF FLOWS (DRAINS) HAVING NO
C****DATA FROM OCT 16 THROUGH MAR 31, AND ESTIMATED DIVERSION (ARICD)
C****OF RIVERSIDE CANAL FROM INDIAN CREEK, AND AVERAGE GLENWOOD TO
C****MIDDLETON GAIN (AMF).
C
      DATA ISTP/60,75,100/
      DATA IBRP/0,60,75,100/
      DATA ISFS/13210810,13210824,13210831,13210835,13210849,13210980,
```

```
      113210986,13210988,13211445,13212550,13212890/
       DATA ARICD/0.0,0.0,0.0,0.0,60.,80.,140.,100.,60.,40.,0.0,0.0/
       DATA AMF/85.,90.,75.,75.,175.,230.,235.,235.,215.,140.,80.,80./
C
C----SECTION  2-3, PART 4
C****INITIALIZE INPUT-OUTPUT UNITS.
C
       INTEGER OUT
       IN=1
       IZ=5
       IX=8
C
C----SECTION  2-3, PART 5
C****ASSIGN INPUT-OUTPUT FILES TO UNIT NUMBERS
C
       CALL ASSIGN (1,'BOIWRA.IND')
       CALL ASSIGN (2,'BOIWRA.RPT')
       CALL ASSIGN (5,'BOIWRA.DPL')
       CALL ASSIGN (6,'BOISYS.IND')
       CALL ASSIGN (7,'BOICAN.IND')
       CALL ASSIGN (8,'BOIWRA.RTS')
       CALL ASSIGN (10,'BOIWRA.HSF')
       CALL ASSIGN (11,'BOIWRA.HSD')
       CALL ASSIGN (12,'BOIWRA.HSR')
       CALL ASSIGN (13,'BOIWRA.ALD')
       CALL ASSIGN (14,'BOIWRA.ALR')
       CALL ASSIGN (15,'BOIWRA.ALM')
       OPEN (UNIT=16,NAME='BOIWRA.NAL',TYPE='NEW',
      1CARRIAGECONTROL='LIST')
C
C----SECTION  3-1
C**** ND  = NUMBER OF INDIVIDUAL DIVERSION MEASUREMENTS
C**** NR  = NUMBER OF REACHES IN RIVER SYSTEM
C**** NLL = NUMBER OF WATER RIGHTS
C**** IFR = FINAL REACH NUMBER IN SYSTEM
C**** NCR = GREATEST NUMBER OF CONTINUOUS REACHES (BACK-TO-BACK)
C**** IDB = DEBUG PRINTOUT IF GREATER THAN ZERO
C**** NP  = NUMBER OF RESERVOIRS
C**** ILS = WATER RIGHT PRINTOUT BY PRIORITY AND REGULAR ACCOUNTING IF
C****       EQUAL 1.  IF ILS > 1, REGULAR ACCOUNTING NOT DONE.
C**** NEP = NUMBER OF EXCHANGE PUMPS IN SYSTEM
C**** JDP = INTERVAL IN DAYS BETWEEN PRINTOUT OF DIVERSION DATA.
C**** NSR = TOAL NUMBER OF RESERVOIR STORAGE RIGHTS IN ENTIRE SYSTEM.
C**** LBC = WATER RIGHT PRINTOUT BY CANAL IF > 0.
C**** KOD = WILL PRINT OUT DIVERSION DATA IN FLOATING POINT IN GREATER
C****       DETAIL IF > 0, MAINLY USED FOR DEBUGGING.
C**** NSP = NUMBER OF CANALS FOR WHICH SPECIAL COMPUTATIONS ARE MADE.
C**** NRC = NUMBER OF REACHES WITH DATA ON HISTORY.
C**** IBG = NOT USED
C**** LRA = NUMBER OF LINES TO BE PRINTED OF DATA GENERATED FOR
C****       ADDITION TO ALLOCATIONS FILE.
C**** NDF = NUMBER OF DAYS FOR WHICH ALLOCATIONS ARE TO BE PROJECTED
C****       BEYOND FINAL DAY AT PARMA.
C**** KED = THE FINAL KED DAYS WILL BE PRINTED AT IDAHO FALLS & BURLEY
C
       READ(IN,7,END=45) ND,NR,NLL,IFR,NCR,IDB,NP,ILS,NEP,JDP,NSR,LBC,KOD
      1,NSP,NRC,IBG,LRA,NDF,KED
```

```
C
C****SET VALUES OF IBGS AND IBGD BY INQUIRY.  IF THESE VALUES = 1 THEN
C****SYSTEM RESERVOIR AND DIVERSIONS ACCUMULATIONS (RESPECTIVELY) WILL
C****BE READ IN FROM DATA FILES (BOISYS.IND AND/OR  BOICAN.IND),
C****RATHER THAN FROM ALLOCATIONS FILES (BOIWRA.ALM AND BOIWRA.ALD).
C
      IBGS=0
      IBGD=0
      TYPE *, '  '
      TYPE *, '    IF THIS IS THE FIRST RUN OF THE YEAR, OR IF YOU'
      TYPE *, '    ARE RESETTING ALL SYSTEM AND RESERVOIR AND/OR  '
      TYPE *, '    DIVERSION STORAGE ACCUMULATED TOTALS, THEN YOU '
      TYPE *, '    MUST READ THESE VALUES FROM THE CHANGE FILE.   '
      PRINT 498
  498 FORMAT('    TO READ NEW VALUES ENTER Y/N: '$)
      ACCEPT 500,INI
  500 FORMAT(A1)
      IF(INI.NE.'Y') GO TO 499
      PRINT 497
  497 FORMAT(/'    DO YOU WISH TO READ NEW DIVERSION STORAGE TOTALS? Y/N
     1: '$)
      ACCEPT 500,INI
      IF(INI.EQ.'Y') IBGD=1
      PRINT 496
  496 FORMAT(/'    DO YOU WISH TO READ NEW RESERVOIR SYSTEM TOTALS? Y/N:
     1 '$)
      ACCEPT 500,INI
      IF(INI.EQ.'Y') IBGS=1
  499 CONTINUE
C
C----SECTION  3-2, PART 1
C****READ REPORT OUTPUT UNIT NUMBERS, INITIALIZE PRINTOUT UNIT
C
      READ(IN,7)(KOUT(L),L=1,3)
      OUT=KOUT(1)
C
C----SECTION  3-2, PART 2
C**** IDS  = REACH NUMBER DIRECTLY DOWNSTREAM FROM NRTH REACH
C**** IORD = REACH NUMBERS IN ORDER OF OPERATION
C**** IRS  = RESERVOIR NUMBER IN LTH REACH
C**** IDA  = NUMBER OF DAYS FOR WHICH GAINS ARE AVERAGED FOR LTH REACH
C**** ICF = 1 IF REACH OUTFLOW IS NOT A MEASURED VALUE, BUT INSTEAD
C**** COMPUTED BY INFLOW MINUS DIVERSIONS.  IF ICF = 2 THE REACH IS AN
C**** OFFSTREAM CHANNEL, SUCH AS NEW YORK CANAL.  VALUES ARE IN ORDER
C**** OF REACHES.
C
      READ(IN,7)(IDS(L),L=1,NR)
      READ(IN,7)(IORD(L),L=1,NR)
      READ(IN,7)(IRS(L),L=1,NR)
```

```
      READ(IN,7)(IDA(L),L=1,NR)
      READ(IN,7)(ICF(L),L=1,NR)
    7 FORMAT(16I5)
C
C----SECTION  3-2, PART 3
C**** KNRC = THE ORDER IN WHICH REACH VALUES ARE READ FROM THE HISTORY
C****        FILE
C
      READ(IN,7) (KNRC(L),L=1,NRC)
C
C----SECTION  3-2, PART 4
C****READ TIME LAGS (IN DAYS FROM REFERENCE REACH) FOR ALL REACHES,
C****DIVERSIONS AND RESERVOIRS.
C
      READ(IN,7) (LAG(L),L=1,NR)
      READ(IN,7) (LGD(L),L=1,NR)
      READ(IN,7) (LGG(L),L=1,NP)
C
C----SECTION  3-2, PART 5
C****READ REACH NAMES (RCHNM) AND RESERVOIR NAMES (RESNM).
C
      READ(IN,9) ((RCHNM(L,J),J=1,6),L=1,NR)
    9 FORMAT(18A4,8X)
      READ(IN,17)((RESNM(L,J),J=1,4),L=1,NP)
   17 FORMAT(20A4)
C
C----SECTION  3-3, PART 1
C****INITIALIZE VARIOUS DIVERSION VALUES TO ZERO.
C
      DO 59 L=1,ND
      STOR(L)=0.0
      TDIV(L)=0.0
   59 SRM(L)=0.0
C
C----SECTION  3-3, PART 2
C****IF INITIAL RUN IS BEING MADE, OR IF VALUES OF ACCUMULATED EVAP-
C****ORATION , GRAND TOTALS, RESERVOIR STORAGE RIGHT ACCOUNTS AND/OR
C****DIVERSION STORAGE ACCOUNTS ARE TO BE RESET, THESE ARE READ FROM
C****FILES BOISYS.IND (IBGS=1) AND BOICAN.IND (IBGD=1), OTHERWISE THEY
C****ARE READ FROM THE ALLOCATIONS FILE.  ALSO, ZERO VALUES OF TOTAL
C****DIVERSION BY RIGHTS WITH SEASONAL VOLUME LIMITS (VRD).
C
      IF(IBGS.EQ.0) GO TO 61
      READ(6,44) (CEV(L),L=1,NP)
      READ(6,44) (GT(L),L=1,5)
      READ(6,44) (BRST(L),L=1,NSR)
   61 IF(IBGD.EQ.1) READ(7,44) (SRM(L),L=1,ND)
   44 FORMAT(8F10.1)
      DO 156 L=1,40
  156 VRD(L)=0.0
      DO 175 L=1,1
  175 TSE(L)=0.0
C
C----SECTION  3-3, PART 3
C****READ IDENTIFICATION NUMBERS, REACH NUMBERS, DIVERSION
C****TYPE INDICATOR (JC), & NAMES OF DIVERSIONS.
C****THE VARIABLE JC IS TNE INDICATOR TO DENOTE INSTREAM USE SUCH AS
```

```
C****POWER OR MINIMUM FLOW (NEGATIVE NUMBER), CANAL OR PUMP DIVERSION
C****(ZERO), OR RESERVOIR (POSITIVE NUMBER).  JC IS NORMALLY ENTERED
C****VIA JR (SEE SECTION 6-1, PART 4) ON THE LIST OF RIGHTS, BUT IF A
C****DIVERSION HAS NO RIGHT, IT CAN BE ENTERED DIRECTLY HERE.
C
      READ(IZ,89) (IDN(L),IRCH(L),JC(L),(DIVNM(L,J),J=1,4),L=1,ND)
   89 FORMAT(2I10,I3,2X,4A4)
C
C----SECTION  3-3, PART 4
C****READ IDENTIFICATION NUMBERS, REACH NUMBERS, & NAMES OF EXCHANGE
C****PUMPS
C
      IF(NEP.LE.0) GO TO 480
      READ(IZ,89) (IEN(L),IER(L),(EXCNM(L,J),J=1,4),L=1,NEP)
  480 CONTINUE
C
C----SECTION  4-1, PART 1
C****INITIALIZE K, A PAGE CONTROL COUNTER FOR PRINTOUT OF WATER RIGHTS
C****AND NSVL, THE NUMBER OF RIGHTS WITH A SEASONAL VOLUME LIMIT.
C
      K=74
      NSVL=0
C
C----SECTION  4-1, PART 2
C****READ ALL RIGHTS ASSOCIATED WITH DIVERSIONS. P IS THE CFS AMOUNT AND
C****ICN IS THE CANAL/DIVERSION NUMBER OF THE RIGHT. IV INDICATES A
C****VARIABLE RIGHT, AND JR INDICATES THE RESERVOIR NUMBER OF A RESERVOIR
C****RIGHT (ALWAYS GE 1), A CANAL OR PUMP DIVERSION (ALWAYS = 0), OR AN
C****INSTREAM FLOW RIGHT SUCH AS POWER OR FISH FLOW (ALWAYS NEGATIVE,
C**** -1 WITHOUT A MEASURED USE AND -2 WITH A MEASURED USE.
C
      DO 52 L=1,NLL
      READ(IX,8) IY,IM,ID,P(L),ICN(L),IV(L),JR(L),SVL(L),KBG(L),KEN(L)
    8 FORMAT(I6,I2,I2,5X,F10.3,I10,2I5,F6.0,2I5)
C
C----SECTION  4-1, PART 3
C****DATE (YEAR,MONTH,DAY) IS CONVERTED TO SINGLE INTEGER FOR LATER USE.
C
      IDATE(L)=ID+(100*IM)+(IY*10000)
C
C----SECTION  4-1, PART 4
C****IF ILS IS POSITIVE, A LISTING OF RIGHTS BY PRIORITY WILL BE MADE.
C
      IF(ILS.LE.0) GO TO 54
      K=K+1
      IF(K.LE.74) GO TO 54
      WRITE(OUT,53)
   53 FORMAT(1H1///  ,10X,6H ORDER,1X            ,14HPARTY OR CANAL,12X,
     14HDATE,10X,3HCFS,10X,5HREACH/)
      K=0
   54 CONTINUE
C
C----SECTION  5-1
C****DETERMINE ORDER NUMBER OF DIVERSION TO WHICH RIGHT BELONGS (ICNL)
C****USING BINARY SEARCH METHOD.  IF ICN=0, PRINT RIGHT AS "UNDEFINED".
C
      IF(ICN(L).LE.0) GO TO 129
```

```
      IB=1
      IT=ND+1
  125 IA=(IB+IT)/2
      IF(ICN(L)-IDN(IA)) 126,127,128
  126 IF(IA.EQ.IT) GO TO 129
      IT=IA
      GO TO 125
  128 IF(IA.EQ.IB) GO TO 129
      IB=IA
      GO TO 125
  129 IF(ILS.GT.0) WRITE(OUT,130) L,OM(IM),ID,IY,P(L)
  130 FORMAT(10X,I4,23X,3X,A3,I3,1H,,I4,1X,F10.3,4X,9HUNDEFINED)
      ICNL(L)=0
      GO TO 131
  127 ICNL(L)=IA
  131 CONTINUE
      I=ICNL(L)
      IF(I.LE.0) GO TO 52
C
C----SECTION  6-1, PART 2
C****IF THE RIGHT BEING READ HAS A SEASONAL VOLUME LIMIT (SVL(L) > 0),
C****ASSIGN THAT RIGHT AN ORDER NUMBER (NSVL) FOR LATER USE.
C
      IVO(L)=0
      IF(SVL(L).LE.0.0) GO TO 155
      NSVL=NSVL+1
      IVO(L)=NSVL
  155 CONTINUE
C
C----SECTION  6-1, PART 3
C****IF ILS IS POSITIVE, A LISTING OF RIGHTS BY PRIORITY WILL BE MADE.
C
      IF(ILS.LE.0) GO TO 132
      M=IRCH(I)
      WRITE(OUT,55)L,(DIVNM(I,J),J=1,4),OM(IM),ID,IY,P(L),
     1(RCHNM(M,J),J=1,6)
   55 FORMAT(10X,I4,3X,4A4,7X,A3,I3,1H,,I4,1X,F10.3,4X,6A4)
  132 CONTINUE
C
C----SECTION  6-1, PART 4
C****MOVE RESERVOIR RIGHT INDICATOR TO BECOME A RESERVOIR & POWER RIGHT
C****INDICATOR ALSO.
C
      JC(I)=JR(L)
   52 CONTINUE
C
C----SECTION  7-1
C****IF LBC IS POSITIVE, A PRINTOUT OF RIGHTS WILL BE MADE BY CANAL.
C****FORMAT IS FOR COPY AND REDUCTION OF COMPUTER PRINTOUT SIZE SHEET.
C
      IF(LBC.LE.0) GO TO 46
      WRITE(OUT,104)
  104 FORMAT(1H1///12X,6HNUMBER,4X,14HPARTY OR CANAL,10X,8HYR-MO-DY,9X,
     1 3HCFS,3X,5HREACH/)
      LI=9
      DO 103 L=1,ND
      TCL=0.0
```

```
      I=0
      DO 106 J=1,NLL
      IF(ICNL(J).NE.L) GO TO 106
      I=I+1
      LI=LI+1
      IF(LI.LT.82) GO TO 117
      WRITE(OUT,104)
      LI=9
  117 M=IRCH(L)
      IF(I.GT.1) GO TO 154
      LI=LI+1
      WRITE(OUT,123)
  123 FORMAT(1H )
  154 WRITE(OUT,107) IDN(L),(DIVNM(L,K),K=1,4),IDATE(J),P(J),(RCHNM(M,K)
     1,K=1,6)
  107 FORMAT(10X,I10,2X,4A4,6X,I10,2X,F10.3,3X,6A4)
      TCL=TCL+P(J)
  106 CONTINUE
      IF(TCL.LE.0.0.OR.I.LE.1) GO TO 105
      LI=LI+1
      WRITE(OUT,108) TCL
  108 FORMAT(22X,5HTOTAL,29X,F10.3)
  105 IF(LI.LT.74) GO TO 103
      WRITE(OUT,104)
      LI=9
  103 CONTINUE
   46 CONTINUE
C
C----SECTION  8-1, PART 1
C****IF ILS IS GREATER THAN ONE, PROGRAM WILL TERMINATE AFTER RIGHTS
C****LISTING.
C
      IF(ILS.GT.1) GO TO 45
C
C----SECTION  8-1, PART 3
C****FIND DIVERSION ORDER NUMBER (ISPEC) FOR SPECIAL CANALS (CANALS FOR
C****WHICH SPECIAL CALCULATIONS MUST BE MADE).
C
      DO 133 L=1,NSP
      DO 135 J=1,ND
      IF(ISPEC(L).EQ.IDN(J)) GO TO 134
      IF(J.EQ.ND) WRITE(OUT,136) ISPEC(L)
  136 FORMAT(30H NO DIVERSION NUMBER FOUND FOR,I10)
  135 CONTINUE
  134 ISPEC(L)=J
  133 CONTINUE
C
C----SECTION  9-1
C****TRANSFER SPECIAL CANAL DIVERSION ORDER NUMBERS TO UNDIMENSIONED
C****VARIABLES FOR LATER USE AS SUBSCRIPTS.  WHEN ADDING MORE SPECIAL
C****DIVERSIONS, INCREASE DIMENSION TO EXACT NUMBER.
C
      I1=ISPEC(1)
      I2=ISPEC(2)
      I3=ISPEC(3)
      I4=ISPEC(4)
      I5=ISPEC(5)
```

```
      I6=ISPEC(6)
      I7=ISPEC(7)
      I8=ISPEC(8)
      I9=ISPEC(9)
      I10=ISPEC(10)
      I11=ISPEC(11)
      I12=ISPEC(12)
      I13=ISPEC(13)
      I14=ISPEC(14)
      I23=ISPEC(23)
      I24=ISPEC(24)
      I25=ISPEC(25)
C
C----SECTION 10-1
C****READ A MAXIMUM OF 366 DAYS OF FLOW MEASUREMENT DATA (A) FROM
C****SELECTED HISTORY FILE.  NRC IS THE NUMBER OF MEASUREMENT STATIONS
C****AND NED BECOMES THE ENDING DAY.  NMR IS THE STATION NUMBER, TYP IS
C**** "F" FOR FLOW STATION, NYE IS THE YEAR, AND NDY IS THE DAY.  KNRC
C****IS THE REACH NUMBER.
C
      JFI=0
      NED=-1
      DO 220 L=1,366
      NED=NED+1
      ICFMD(L)=0
      NMD=0
      DO 221 I=1,NRC
      J=KNRC(I)
      IF(JFI.EQ.1) GO TO 278
  227 READ(10,226,END=222) NMR,TYP,NYE,NDY,FTEMP
  226 FORMAT(I8,A1,I4,I3,9F7.0)
      GO TO 279
  278 JFI=0
  279 IF(NMR.GT.13213000) GO TO 227
      NDT=NDY+(NYE*1000)
      IF(I.GT.1) GO TO 225
      JDT(L)=NDT
      JYE(L)=NYE
      JDY(L)=NDY
  225 IF(NMR.LE.JRN(J)) GO TO 219
      IF(J.NE.10) GO TO 218
      ICFMD(L)=1
      JFI=1
      GO TO 221
  218 MDD1=MD1
      MDD2=MD2
      FEB=JYE(L)/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      IF(FEB.GT.0.001.OR.JDY(L).LE.59) GO TO 317
      MDD1=MDD1+1
      MDD2=MDD2+1
  317 IF(JDY(L).GE.MDD1.AND.JDY(L).LE.MDD2) GO TO 318
      DO 319 K=1,11
      IF(JRN(J).EQ.ISFS(K)) GO TO 320
  319 CONTINUE
      GO TO 318
```

```
  320 A(J,L)=0.0
      JFI=1
      GO TO 221
  318 WRITE(OUT,224) JRN(J),NDT,NMR
  224 FORMAT(14H REACH NUMBER ,I10,15H MISSING ON DAY,I10,5X,12HREACH NU
     1MBER,I10,43H  READ INSTEAD. PREVIOUS DAY VALUE ASSUMED.)
      IF(L.NE.1) GO TO 286
      WRITE(OUT,194)
  194 FORMAT(50H ERROR OCCURRED ON FIRST DAY, EXECUTION TERMINATED)
      STOP
  286 IF(NMD.LE.10) GO TO 249
      WRITE(OUT,280)
  280 FORMAT(25H MISSING FLOWS EXCEED 10 )
      STOP
  249 A(J,L)=A(J,L-1)
      NMD=NMD+1
      JFI=1
      GO TO 221
  219 IF(NDT.EQ.JDT(L)) GO TO 288
      WRITE(OUT,223) NMR,JDT(L)
  223 FORMAT(21H1 FLOW STATION NUMBER,I10,18HWRONG DATE FOR DAY,I10)
      STOP
  288 IF(NMR.LT.JRN(J)) GO TO 227
      IF(FTEMP.LT.9000000.OR.L.EQ.1) GO TO 297
      A(J,L)=A(J,L-1)
      WRITE(OUT,299) JRN(J),NDT
  299 FORMAT(17H STATION NUMBER  ,I10,41H     HAS FLOW GREATER THAN 9000
     1000 ON DAY,I10,29H ,PREVIOUS DAY VALUE ASSUMED.)
      GO TO 221
  297 A(J,L)=FTEMP
  221 CONTINUE
  220 CONTINUE
  222 CONTINUE
C
C----SECTION 11-1
C****ADVANCE DATE NDF DAYS TO ALLOW FOR PROJECTED REGULATION.
C
      NED1=NED
      IF(NDF.GT.0) NED1=NED+1
      NEDF=NED+NDF
      IF(NDF.LE.0) GO TO 68
      ITD=365
      FEB=NYE/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      IF(FEB.LT.0.001) ITD=366
      DO 257 L=NED1,NEDF
      NDY=NDY+1
      IF(NDY.LE.ITD) GO TO 247
```

```
      NDY=1
      NYE=NYE+1
  247 NDT=NDY+(NYE*1000)
      JDT(L)=NDT
      JYE(L)=NYE
      JDY(L)=NDY
  257 CONTINUE
   68 CONTINUE
C
C----SECTION 12-1, PART 1
C****READ DIVERSION MEASURMENTS (DVV) FOR ALL DAYS FROM SELECTED
C****HISTORY FILE.  MD1 AND MD2, THE BEGINNING AND ENDING DAY FOR WHICH
C****ALL DIVERSION DATA MUST BE PRESENT, ARE MODIFIED FOR LEAP YEARS.
C
      JTI=0
      JDI=0
      DO 201 L=1,366
      MDD1=MD1
      MDD2=MD2
      FEB=JYE(L)/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      IF(FEB.GT.0.001.OR.JDY(L).LE.59) GO TO 199
      MDD1=MDD1+1
      MDD2=MDD2+1
  199 IF(JDY(L).GE.MDD1.AND.JDY(L).LE.MDD2) GO TO 198
C
C----SECTION 12-1, PART 2
C****THIS LOOP USED TO READ DIVERSION DATA FROM OCT 16 TO APR 1 WHEN
C****RECORDS NEED NOT BE PRESENT FOR EACH AND EVERY CANAL OR PUMP.
C
      DO 196 I=1,ND
      IF(JTI.EQ.1) GO TO 191
      IF(JDI.EQ.1) GO TO 192
  185 READ (11,80,END=204) NMR,TYP,NDT,DTEMP
      GO TO 190
  191 JTI=0
  190 IF(NDT.EQ.JDT(L)) GO TO 186
      JTI=1
      GO TO 197
  192 JDI=0
  186 IF(NMR.LT.IDN(I)) GO TO 185
      IF(NMR.EQ.IDN(I)) GO TO 195
      JDI=1
      GO TO 196
  195 DVV(I,L)=DTEMP
  196 CONTINUE
      GO TO 197
C
C----SECTION 12-2
C****THIS LOOP USED TO READ DIVERSION DATA FROM APR 1 TO OCT 15 DURING
C****WHICH DATA RECORDS MUST BE PRESENT FOR ALL CANALS OR PUMPS.
C****CHECK FOR PROPER ORDER (IDN) AND DATE (JDT).  SKIP PIONEER-
C****DIXIE (I8) SINCE THAT DIVERSION IS COMPUTED, AND SKIP RESERVOIRS
C****(JC>0) AND UNMEASURED INSTREAM FLOWS (JC=-1).
C
  198 NMD=0
```

```
      DO 205 I=1,ND
      IF(JC(I).GT.0.OR.JC(I).EQ.-1.OR.I.EQ.I8) GO TO 205
      IF(JTI.EQ.1) GO TO 189
   87 READ(11,80,END=204) NMR,TYP,NDT,DTEMP
   80 FORMAT(I8,A1,I7,9F7.0)
      GO TO 188
  189 JTI=0
  188 IF(NMR.GT.13213000) GO TO 87
      IF(NMR.LT.IDN(I)) GO TO 87
      IF(NMR.EQ.IDN(I)) GO TO 202
      WRITE(OUT,203) IDN(I),NDT,NMR
  203 FORMAT(18H DIVERSION NUMBER ,I10,15H MISSING ON DAY,I10,5X,16HDIVE
     1RSION NUMBER,I10,43H  READ INSTEAD. PREVIOUS DAY VALUE ASSUMED.)
      IF(L.NE.1) GO TO 287
      WRITE(OUT,194)
      STOP
  287 IF(NMD.LE.50) GO TO 207
      WRITE(OUT,281)
  281 FORMAT(29H MISSING DIVERSIONS EXCEED 50)
      STOP
  207 DVV(I,L)=DVV(I,L-1)
      NMD=NMD+1
      JTI=1
      GO TO 205
  202 IF(NDT.EQ.JDT(L)) GO TO 187
      WRITE(OUT,206) NMR,JDT(L)
  206 FORMAT(18H1DIVERSION NUMBER ,I10, 19H WRONG DATE FOR DAY,I10)
      STOP
  187 IF(DTEMP.LT.9000000.OR.L.EQ.1) GO TO 301
      DVV(I,L)=DVV(I,L-1)
      WRITE(OUT,302) IDN(I),NDT
  302 FORMAT(17H DIVERSION NUMBER,I10,41H     HAS FLOW GREATER THAN 9000
     1000 ON DAY,I10,29H ,PREVIOUS DAY VALUE ASSUMED.)
      GO TO 205
  301 DVV(I,L)=DTEMP
  205 CONTINUE
  197 CONTINUE
  201 CONTINUE
C
C----SECTION 12-3, PART 2
C****NEDD IS LAST DAY OF DIVERSION DATA.  THIS MAY BE GREATER THAN NED
C****(LAST DAY OF REACH DATA) IF REGULATION IS TO BE PROJECTED.
C
  204 NEDD=L-1
C
C----SECTION 13-1, PART 1
C****READ RESERVOIR MEASUREMENTS FOR ALL DAYS FROM SELECTED HISTORY
C****FILE.  RSS IS RESERVOIR CONTENT AND EVP IS EVAPORATION IN INCHES.
C****CHECK FOR PROPER DATE (JDT).  NMR IS STATION NUMBER, TYP IS "R"
```

```
C****FOR RESERVOIR, AND NDT IS THE DATE.
C
      JRI=0
      DO 210 L=1,366
      DO 212 I=1,NP
      IF(JRI.EQ.1) GO TO 289
  215 READ(12,211,END=214) NMR,TYP,NDT,RTEMP,ETEMP,PTEMP
  211 FORMAT(I8,A1,I7,3F7.0)
      GO TO 282
  289 JRI=0
  282 IF(NMR.GT.13213000) GO TO 215
      IF(NMR.LT.IRESN(I)) GO TO 215
      IF(NMR.EQ.IRESN(I)) GO TO 283
      WRITE(OUT,284) IRESN(I),NDT,NMR
  284 FORMAT(17H RESERVOIR NUMBER,I10,16H  MISSING ON DAY,I10,5X,16HRESE
     1RVOIR NUMBER,I10,43H  READ INSTEAD. PREVIOUS DAY VALUE ASSUMED.)
      IF(L.NE.1) GO TO 285
      WRITE(OUT,194)
      STOP
  285 RSS(I,L)=RSS(I,L-1)
      EVP(I,L)=EVP(I,L-1)
      JRI=1
      GO TO 212
  283 IF(NDT.EQ.JDT(L)) GO TO 208
      WRITE(OUT,213) NMR,JDT(L)
  213 FORMAT(17H1RESERVOIR NUMBER,I10,19H WRONG DATE FOR DAY,I10)
      STOP
  208 IF(RTEMP.LT.9000000.OR.L.EQ.1) GO TO 209
      RSS(I,L)=RSS(I,L-1)
      WRITE(OUT,290) IRESN(I),NDT
  290 FORMAT(17H RESERVOIR NUMBER,I10,41H HAS CONTENTS GREATER THAN 9000
     1000 ON DAY,I10,29H ,PREVIOUS DAY VALUE ASSUMED.)
      GO TO 291
  209 RSS(I,L)=RTEMP
  291 IF(ETEMP.LT.90000.) GO TO 193
      EVP(I,L)=EVP(I,L-1)
      WRITE(OUT,292) IRESN(I),NDT
  292 FORMAT(17H RESERVOIR NUMBER,I10,42H HAS EVAPORATION GREATER THAN 9
     10000 ON DAY,I10,29H ,PREVIOUS DAY VALUE ASSUMED.)
      GO TO 212
  193 EVP(I,L)=ETEMP
  181 IF(PTEMP.LT.90000.OR.L.EQ.1) GO TO 183
      PPT(I,L)=0.0
      WRITE(OUT,184) IRESN(I),NDT
  184 FORMAT(17H RESERVOIR NUMBER,I10,44H HAS PRECIPITATION GREATER THAN
```

```
      190000 ON DAY,I10,21H ,ZERO VALUE ASSUMED.)
       GO TO 212
  183 PPT(I,L)=PTEMP
  212 CONTINUE
  210 CONTINUE
  214 CONTINUE
C
C----SECTION 13-1, PART 2
C****EXTEND RESERVOIR AND EXCHANGE PUMP DATA NDF DAYS INTO FUTURE
C****USING SAME VALUES AS LAST DAY FOR WHICH DATA WAS ENTERED.  EXTEND
C****DIVERSION DATA ONLY PAST THE NEDDTH DAY USING SAME VALUES AS THE
C****NEDDTH DAY.
C
      IF(NDF.LE.0) GO TO 12
      DO 256 L=NED1,NEDF
      DO 255 I=1,NP
      EVP(I,L)=EVP(I,NED)
      PPT(I,L)=0.0
  255 RSS(I,L)=RSS(I,NED)
      IF(L.LE.NEDD) GO TO 256
      DO 254 J=1,ND
  254 DVV(J,L)=DVV(J,NEDD)
  256 CONTINUE
   12 CONTINUE
C
C----SECTION 14-1
C****READ ALLOCATION DIVERSION (STOR,SRM), RESERVOIR(CEV), AND MISC
C****(BRST,GT) DATA FOR DAY PRECEDING FIRST DAY OF CALCULATION FROM
C****SELECTED ALLOCATION FILES.  THESE DATA ARE CALCULATED BY EARLIER
C****RUNS OF THIS PROGRAM AND ARE NEEDED TO MAINTAIN CUMULATIVE TOTALS.
C****NMR =  STATION NUMBER; TYP = "D" FOR DIVERSION DATA, "R" FOR
C****RESERVOIR DATA, "M" FOR MISCELLANEOUS TOTALS; NDT = DATE.  IF
C****ANY OF THIS DATA IS BEING RESET, IBGD AND/OR IBGS = 1 AND THIS
C****ROUTINE WILL BE BYPASSED (SEE SECTION 3-3, PART 2). THE DATE IS
C****CHECKED TO MAKE SURE IT IS YHE 4TH DAY FROM THE BEGINNING (JDT(4)),
C****SINCE REGULATION BEGINS WITH THE 5TH DAY.
C
      IF(IBGD.EQ.1) GO TO 240
      JDI=0
      DO 230 I=1,ND
      IF(JDI.EQ.1) GO TO 332
      READ(13,233,END=231) NMR,TYP,NDT,STEMP1,STEMP2
  233 FORMAT(I8,A1,I7,3F10.0)
      IF(NDT.EQ.JDT(4)) GO TO 330
      WRITE(OUT,232) NMR
  232 FORMAT(39H1RESERVOIR STORAGE VALUES FOR DIVERSION,I10,10H WRONG DA
     1Y)
      STOP
  332 JDI=0
```

```
  330 IF(NMR.EQ.IDN(I)) GO TO 331
      STOR(I)=0.0
      SRM(I)=0.0
      JDI=1
      GO TO 230
  331 STOR(I)=STEMP1
      SRM(I)=STEMP2
  230 CONTINUE
  231 CONTINUE
      DO 241 I=1,NP
      READ(14,243,END=240) NMR,TYP,NDT,CEV(I)
  243 FORMAT(I8,A1,I7,F10.0)
      IF(NDT.EQ.JDT(4)) GO TO 241
      WRITE(OUT,242) NMR
  242 FORMAT(55H1RESERVOIR STORAGE AND EVAPORATION VALUES FOR RESERVOIR,
     1I10,10H WRONG DAY)
      STOP
  241 CONTINUE
  240 CONTINUE
      IF(IBGS.EQ.1) GO TO 250
      READ(15,252,END=250) NMR,TYP,NDT,(BRST(L),L=1,NSR)
      READ(15,252) NMR,TYP,NDT,(GT(L),L=1,5)
  252 FORMAT(I8,A1,I7,10F10.0)
      IF(NDT.EQ.JDT(4)) GO TO 250
      WRITE(OUT,251) NMR
  251 FORMAT(18H1MISC DATA NUMBER,I10,10H WRONG DAY)
      STOP
  250 CONTINUE
C
C----SECTION 15-1, PART 3
C****INITIALIZE 4-DAY AVERAGE GAIN TO ZERO
C
      DO 85 L=1,NR
      DO 85 N=1,4
   85 GAV(L,N)=0.0
      KK=0
C
C----SECTION 16-1, PART 1
C****BEGIN LOOP FOR DAILY CALCULATIONS.  NEDF IS THE TOTAL NUMBER OF
C****DAYS FOR WHICH GAINS ARE TO BE COMPUTED.  THIS INCLUDES DAYS FOR
C****WHICH GAINS ARE TO BE CALCULATED SIMPLY TO GET RUNNING AVERAGES,
C****REGULAR ACCOUNTING DAYS, AND PROJECTED DAYS.
C
      DO 72 KD= 1,NEDF
C
C----SECTION 16-1, PART 2
C****CHECK FOR LEAP YEAR (FEB =0.0).
C
      FEB=JYE(KD)/4.0
      IFEB=FEB
      FEB=FEB-IFEB
C
C----SECTION 16-1, PART 3
C****ALIGN DATA FOR KDTH DAY BY TRANSFERRING DIVERSIONS TO ALTERNATE
C****ALTERNATE ARRAYS WITH PROPER LAGS.  KDPIL IS THE DAY (KD) PLUS THE
C****LAG (IL).  LGD IS THE TRAVEL TIME FROM THE POINT OF DIVERSION TO
C****PARMA. (CURRENTLY NO LAG IS USED ON THE BOISE RIVER.)
```

```
C
      DO 71 L=1,ND
      IR=IRCH(L)
      IF(IR.LE.0) GO TO 71
      IL=LGD(IR)
      KDPIL=KD+IL
      IF(KDPIL.GT.0) GO TO 122
      DIV(L)=0.0
      GO TO 71
  122 DIV(L)=DVV(L,KDPIL)
   71 CONTINUE
C
C----SECTION 16-2
C****COMPUTE DIVERSIONS WHICH ARE NOT DIRECTLY MEASURED.
C
      IF(DIV(I2).GT.DIV(I1)) DIV(I2)=DIV(I1)
      DIV(I1)=DIV(I1)-DIV(I2)
      IF(DIV(I24).GT.DIV(I1)) DIV(I24)=DIV(I1)
      DIV(I1)=DIV(I1)-DIV(I24)
      IF(DIV(I4).GT.DIV(I3)) DIV(I4)=DIV(I3)
      DIV(I3)=DIV(I3)-DIV(I4)
      IF(DIV(I6).GT.DIV(I5)) DIV(I6)=DIV(I5)
      DIV(I5)=DIV(I5)-DIV(I6)
      DIV(I8)=DIV(I7)/3.0
      DIV(I7)=DIV(I7)-DIV(I8)
C
C----SECTION 17-1
C****ALIGN DATA FOR KDTH DAY BY TRANSFERRING RESERVOIR DATA TO
C****ALTERNATE ARRAYS WITH PROPER LAGS.  KDPIL IS THE DAY (KD) PLUS THE
C****LAG (IL).  LGG IS THE TRAVEL TIME FROM THE RESERVOIR TO THE LAST
C****REACH.
C
      LD=JDY(KD)
      IF(FEB.LT.0.001.AND.LD.GT.59) LD=LD-1
      DO 74 L=1,NP
      IL=LGG(L)
      KDPIL=KD+IL
      EV(L)=0.0
      PP(L)=0.0
      RES(L,1)=0.0
      RES(L,2)=0.0
      IF(KDPIL.LE.0) GO TO 74
      IF(L.EQ.4.OR.L.EQ.3) GO TO 216
C     IF(LD.LT.91.OR.LD.GT.304) GO TO 216
C     IF(EVP(2,KDPIL).GT.-0.05.OR.EVP(2,KDPIL).LT.-0.50)
C    1EVP(2,KDPIL)=-0.175
      IF(EVP(2,KDPIL).LT.0.0) THEN
            IF(L.EQ.2.OR.L.EQ.3) EV(L)=(-1.0)*EVP(2,KDPIL)
            IF(L.EQ.1.) EV(L)=(-0.95)*EVP(2,KDPIL)
            ENDIF
      IF(PPT(L,KDPIL).GE.0.1) PP(L)=PPT(L,KDPIL)
  216 RES(L,2)=RSS(L,KDPIL)
      KDPIL=KDPIL-1
      IF(KDPIL.GT.0) RES(L,1)=RSS(L,KDPIL)
   74 CONTINUE
C
C----SECTION 18-1, PART 1
```

```
C****VARIABLES USED IN THE REACH GAIN CALCULATION ARE INITIALIZED.  RIN
C****IS TOTAL REACH INFLOW, TRD IS TOTAL REACH DIVERSION, TRE IS TOTAL
C****EXCHANGE PUMPING, AND F IS TOTAL REACH NATURAL FLOW.
C
      DO 16 L=1,NR
      RIN(L)=0.0
      TRD(L)=0.0
      TRE(L)=0.0
   16 F(L)=0.0
C
C----SECTION 18-1, PART 2
C****DIVERSIONS ARE TOTALED BY REACH (TRD).
C
      DO 40 L=1,ND
      IR=IRCH(L)
      IF(IR.LE.0) GO TO 40
      IF(JC(L).NE.0) GO TO 40
      TRD(IR)=TRD(IR)+DIV(L)
   40 CONTINUE
C
C----SECTION 18-2
C****COMPUTE EVAPORATION AT RESERVOIRS FOR REACH GAIN CALCULATION. SA
C****IS THE SURFACE AREA AT ARINT INCREMENTS OF RESERVOIR STORAGE.  EV
C****IS PAN EVAPORATION IN INCHES AND PP IS PRECIPITATION.  LAKE
C****EVAPORATION IS COMPUTED AS 0.7 * PAN EVAP.  PRECIPITATION IS THEN
C****REMOVED FROM LAKE EVAP TO GET NET EVAPORATION (EVNET).  NET EVAP
C****IS NOT ALLOWED TO BECOME NEGATIVE SO THAT PRECIP IS ONLY ALLOWED
C****TO OFFSET EVAP AND NOT BECOME A GAIN.
C
      DO 116 L=1,NP
      EVAP(L)=0.0
      EVNET=0.7*EV(L)-PP(L)
      IF(EVNET.LT.0.0) EVNET=0.0
      IF(ARINT(L).LE.0.0.OR.EV(L).LE.0.0) GO TO 116
      RI=RES(L,1)/(ARINT(L)*1000.)
      IF(RI.LT.0.0) RI=0.0
      IW=RI
      R=RI-IW
      IF(IW.GT.9) IW=9
      AREA=SA(IW+1,L)+(R*(SA(IW+2,L)-SA(IW+1,L)))
      EVAP(L)=(AREA*EVNET/0.012)
      EVAP(L)=EVAP(L)/1.9835
      EVAP(2)=EVAP(2)*3.3979
  116 CONTINUE
C
C----SECTION 19-1, PART 1A
C****ARRANGE PREVIOUS DAYS' GAIN VALUES FOR AVERAGING.  IDA IS THE
C****NUMBER OF DAYS OVER WHICH GAINS ARE TO BE AVERAGED.  AT LEAST FOUR
C****PREVIOUS GAINS ARE SAVED FOR LATER CALCULATIONS.
C
      DO 78 L=1,NR
      KA=IDA(L)
      IF(KA.LT.5) KA=5
      KT=KA-1
      DO 77 N=1,KT
      J=KA-N
   77 G(L,J+1)=G(L,J)
```

```
   78 CONTINUE
C
C----SECTION 19-1, PART 1B
C----IF MIDDLETON FLOW IS MISSING (ICFMD=1), SET INDICATOR = 1 SO THAT
C----FLOW WILL BE ESTIMATED.
C
      ICF(10)=ICFMD(KD)
C
C----SECTION 19-1, PART 2
C****ALIGN DATA FOR KDTH DAY BY TRANSFERRING FLOW DATA (AF) TO
C****ALTERNATE ARRAYS WITH PROPER LAGS.  KDPIL IS THE DAY (KD) PLUS THE
C****LAG (IL).  LAG IS THE TRAVEL TIME FROM THE FLOW POINT TO MIDDLETON.
C
C
      DO 73 J=1,NR
      L=IORD(J)
      IF(L.LE.0) GO TO 73
      IL=LAG(L)
      KDPIL=KD+IL
      IF(KDPIL.GT.0) GO TO 60
      AF(L)=0.0
      GO TO 73
C
C----SECTION 20-1, PART 1
C****IF THE DAY IS GREATER THAN NED, THE LAST DAY OF MEASURED DATA,
C****THEN THE FLOW DATA MUST BE PROJECTED FROM PREVIOUS DATA.  SET
C****RESERVOIR EVAPORATION EQUAL TO PREVIOUS DAY'S EVAPORATION.
C
   60 IF(KDPIL.LE.NED) GO TO 259
      EVPR=0.0
      IS=IRS(L)
      IF(IS.EQ.0) GO TO 258
      EVPR=EVAP(IS)
  258 SUM=0.0
C
C----SECTION 20-1, PART 2
C****FOR FIRST PROJECTED DAY (NED1), FLOW IS ESTIMATED BY USING
C****PREVIOUS CALCULATED AVERAGE GAINS.
C
      IF(KDPIL.GT.NED1) GO TO 237
C
C----SECTION 20-1, PART 3
C****IF MGA IS NOT EQUAL TO ONE, GAIN IS PROJECTED BY ADDING THE
C****AVERAGE OF THE CHANGE IN AVERAGE GAIN OVER THE PREVIOUS 3 DAYS TO
C****THE LAST DAY'S AVERAGE GAIN.
C
      IF(MGA(L).EQ.1) GO TO 239
      DO 248 IJ=1,3
  248 SUM=SUM+GAV(L,IJ)-GAV(L,IJ+1)
      GAIN(L)=GAV(L,1)+(SUM/3.0)
      GO TO 237
C
C----SECTION 20-1, PART 4
C****IF MGA EQUALS ONE THE GAIN IS SIMPLY THE AVERAGE OF THE PREVIOUS
C****FOUR DAY'S AVERAGE GAIN.
C
  239 CONTINUE
```

```
      DO 238 IJ=1,4
  238 SUM=SUM+GAV(L,IJ)
      GAIN(L)=SUM/4.0
C
C----SECTION 21-1, PART 1
C****COMPUTE THE PROJECTED FLOW WITH THE REACH GAIN EQUATION (INFLOW
C****MINUS DIVERSION MINUS EVAPORATION PLUS GAIN).  RESERVOIR CHANGE
C****IN STORAGE IS ASSUMED ZERO.
C
  237 AF(L)=RIN(L)-TRD(L)+TRE(L)-EVPR+GAIN(L)
      GO TO 245
C
C----SECTION 21-1, PART 5
C****FLOW PROJECTION IS COMPLETE AT THIS POINT.
C
  259 AF(L)=A(L,KDPIL)
C
C*****USE THE FOLLOWING STATEMENT TO READ MIDDLETON IF EVER RECEIVE
C*****DATA FROM HYDROMET FOR THAT STATION.
C
C     IF(L.EQ.10.AND.AF(L).GT.900000.) ICF(L)=1
C
C----SECTION 22-1, PART 1A
C****IF A REACH HAS ICF=1, THEN THE FLOW WAS NOT MEASURED AND IS THERE-
C****FORE ESTIMATED BY REACH INFLOW LESS TOTAL REACH DIVERSIONS.  IF A
C****NEGATIVE FLOW IS CALCULATED FOR A NONPROJECTED REACH, SET = ZERO.
C
  253 IF(ICF(L).NE.1) GO TO 245
      AF(L)=RIN(L)-TRD(L)+TRE(L)
      IF(L.NE.10) GO TO 363
      MON=(JDY(KD)/30.33)+1
      IF(MON.GT.12) MON=12
      AF(L)=AF(L)+AMF(MON)
  363 IF(AF(L).LT.0.0.AND.KDPIL.LE.NED) AF(L)=0.0
C
C----SECTION 22-1, PART 1B
C****AFTER DETERMINING FLOW OF REACH 7 (BELOW DIVERSION DAM), COMPUTE
C****OUTFLOW OF LUCKY PEAK (REACH 6) BY ADDING PENITENTIARY CANAL (I9)
C****AND NEW YORK CANAL (I10) TO DIVERSION DAM FLOW, THEN CORRECT THE
C****INFLOW TO REACH 7.
C
C 246 IF(L.NE.7) GO TO 245
C     AF(6)=AF(7)+DIV(I9)+DIV(I10)
C     RIN(7)=AF(6)
C
C----SECTION 22-1, PART 2
C****KEEP A CURRENT ACCOUNT OF REACH INFLOWS FOR ABOVE CALCULATION OF
C****UNMEASURED REACH OUTFLOWS, AND FOR GAIN CALCULATION BELOW.
C
  245 IO=IDS(L)
      IF(IO.LE.0) GO TO 73
      RIN(IO)=RIN(IO)+AF(L)
   73 CONTINUE
C
C----SECTION 23-1, PART 1
C****THE ACTUAL MEASURED RIVER FLOW (AF) AND THE TOTAL REACH DIVERSIONS
C****(TRD) ARE USED TO CALCULATE NATURAL GAINS TO EACH REACH.  IF
```

```
C****THE REACH CONTAINS A RESERVOIR THE GAIN IS CORRECTED FOR CHANGE IN
C****STORAGE (DS) AND EVAPORATION (EVAP).  RIN IS THE ACTUAL INFLOW TO
C****THE REACH. THEREFORE THE GAIN IS CALCULATED AS OUTFLOW + RESERVOIR
C****CHANGE IN STORAGE + EVAPORATION - INFLOW - EXCHANGE PUMPING
C**** + DIVERSIONS.  RETURN FLOW IS CONSIDERED A PART OF NATURAL FLOW
C****AND THUS NOT ESTIMATED.
C
      DO 41 L=1,NR
      IR=IORD(L)
      IF(IR.LE.0) GO TO 41
      IF(IRS(IR).EQ.0) GO TO 42
      IS=IRS(IR)
      DSTOR=(RES(IS,2)-RES(IS,1))/1.9835
      DS(IS)=DSTOR
      EVPR=EVAP(IS)
      GO TO 43
   42 DSTOR=0.0
      EVPR=0.0
   43 G(IR,1)=AF(IR)+DSTOR+EVPR-RIN(IR)+TRD(IR)-TRE(IR)
   41 CONTINUE
C
C----SECTION 24-1, PART 1
C****COMPUTE ESTIMATED DIVERSION OF RIVERSIDE CANAL FROM INDIAN CREEK
C
      MON=(JDY(KD)/30.33)+1
      IF(MON.GT.12) MON=12
      RICD=ARICD(MON)
      TRIV=DIV(I7)+DIV(I8)
      IF(RICD.GT.TRIV) RIDC=TRIV
C
C----SECTION 24-1, PART 2
C****DISTRIBUTE UNIDENTIFIED GAIN FROM MIDDLETON TO PARMA BY COMPUTING
C****TOTAL GAIN (MPG) AND USING COEFFICIENTS.  CORRECT FOR RIVERSIDE
C****DIVERSION FROM INDIAN CREEK (RICD) AND THEN ADD GAIN FROM ALL
C****DRAINS SINCE THEY ARE NOT CONSIDERED IN THE ACCOUNTING. ALSO,
C****CORRECT PREVIOUSLY CALCULATED FLOWS AT CALDWELL AND NOTUS.
C
      MPG=G(11,1)+G(12,1)+G(13,1)-RICD
      G(11,1)=(0.23*MPG)+RICD
      DO 321 N=14,21
  321 G(11,1)=G(11,1)+AF(N)
      G(12,1)=(0.38*MPG)+AF(22)
      G(13,1)=(0.39*MPG)+AF(23)+AF(24)
      AF(11)=AF(10)+G(11,1)-TRD(11)
      AF(12)=AF(11)+G(12,1)-TRD(12)
C
C----SECTION 25-1, PART 1
C****ARRANGE PREVIOUS AVERAGE GAINS (GAV) FOR PROJECTING GAINS BEYOND
C****AVAILABLE DATA. COMPUTE CURRENT AVERAGE GAIN FOR LAST KT (NOT MORE
C****THAN 15) DAYS FOR EACH REACH FOR USE IN NATURAL FLOW COMPUTATION.
C****IF THE GAIN IS A PROJECTED VALUE (KDPIL > NED), THEN NO AVERAGING
C****IS DONE SINCE THAT VALUE IS ALREADY AN ESTIMATED VALUE.
C
      DO 235 L=1,NR
      DO 99 N=1,3
      J=4-N
   99 GAV(L,J+1)=GAV(L,J)
```

```
      GAV(L,1)=0.0
      IL=LAG(L)
      KDPIL=KD+IL
      IF(KDPIL.LE.NED) GO TO 88
      GAV(L,1)=G(L,1)
      GO TO 235
   88 KT=IDA(L)
      IF(KT.GT.15) KT=15
      IF(KT.LT.1) KT=1
      DO 236 N=1,KT
  236 GAV(L,1)=GAV(L,1)+(G(L,N)/KT)
  235 CONTINUE
C
C----SECTION 25-1, PART 2
C****INITIAL REACH GAIN CALCULATION ENDS HERE.  IF DAY OF FIRST WATER
C****RIGHT ALLOCATION HAS NOT BEEN REACHED (KD=5), THEN CONTINUE TO
C****NEXT DAY'S GAIN CALCULATION WITHOUT DOING GAIN AVERAGING OR FLOW
C****ALLOCATION.  UNTIL DAY 5, THE GAINS ARE NEEDED ONLY FOR MAKING
C****RUNNING AVERAGES.
C
      IF(KD.LT.5) GO TO 72
C
C----SECTION 26-1, PART 1
C****BEGIN ALLOCATION - TRANSFER AVERAGE GAINS TO SINGLE DIMENSIONED
C****ARRAY (AG).
C
      DO 92 L=1,NR
      AG(L)=GAV(L,1)
   92 CONTINUE
C
C----SECTION 27-1, PART 1
C****COMPUTE NATURAL FLOW (F) FOR EACH REACH BY ACCUMULATING GAINS IN
C****DOWNSTREAM ORDER.  AGAIN DO NOT INCLUDE FEEDER CANALS (ICF=2).
C
      DO 15 L=1,NR
      IC=IORD(L)
      IF(IC.LE.0.OR.ICF(IC).EQ.2) GO TO 15
      F(IC)=F(IC)+AG(IC)
      IO=IDS(IC)
      IF(IO.LE.0) GO TO 15
      F(IO)=F(IO)+F(IC)
   15 CONTINUE
C
C----SECTION 27-1, PART 2
C****RD IS INITIALLY EQUAL TO EACH DIVERSION.  AS EACH RIGHT IS MET BY
C****NATURAL FLOW, THAT AMOUNT IS DEDUCTED SO THAT RD = REMAINING
C****DIVERSION
C
      DO 1 L=1,ND
    1 RD(L)=DIV(L)
C
C----SECTION 28-1, PART 1
C****TD IS INITIALIZED AS ZERO, AND BECOMES THE TOTAL DIVERSION IN EACH
C****REACH MET BY NATURAL FLOW.  IRT IS INITIALIZED AS THE TOTAL
C****NUMBER OF RIGHTS, AND BECOMES THE NUMBER OF THE LAST RIGHT MET IN
C****EACH REACH BY NATURAL FLOW.  RF IS INITIALIZED AS THE NATURAL
C****FLOW, AND BECOMES THE REMAINING NATURAL FLOW (AFTER UPSTREAM
```

```
C****DIVERSIONS HAVE BEEN DEDUCTED) OF EACH REACH.  PF IS INITIALIZED
C****AS ZERO AND BECOMES THE FLOW RESERVED FROM NATURAL BY POWER RIGHTS
C****IN EACH REACH.  INITIALIZE RESERVOIR DAILY ACCRUED STORAGE TO ZERO.
C
      DO 2 L=1,NR
      TD(L)=0.0
      PF(L)=0.0
      IRT(L)=NLL
    2 RF(L)=F(L)
      DO 10 L=1,NP
   10 DRA(L)=0.0
C
C----SECTION 28-2
C****REDUCE RESERVOIR STORAGE ACCUMULATED BY LUCKY PEAK RIGHT BY
C****AMOUNT OF STORAGE USED FROM FISH AND GAME AND/OR USBR FLOW
C****MAINTENANCE SPACE SO THAT THIS SPACE CAN FILL.  IF THE RIGHT
C****HAS ALREADY FILLED (BRST = RST) THEN DO NOT SUBTRACT THE STORAGE
C****USE SINCE THAT WOULD ALLOW A DOUBLE FILL OF THE SPACE.  UP UNTIL
C****THE LUCKY PEAK RIGHT FILLS, THE FLOW STORAGE USE IS CONSIDERED
C****TO HAVE COME FROM CARRYOVER, NOT THE NEW FILL.  ALSO DO NOT
C****SUBTRACT THE STORAGE IN THE LATE IRRIGATION SEASON PRIOR TO
C****NOVEMBER 1 SO THAT STORAGE USE PROGRAM CAN DEDUCT SAME.
C
      IL=LGD(7)
      KDPIL=KD+IL
      LD=JDY(KDPIL)
      IF(FEB.LT.0.001.AND.LD.GT.59) LD=LD-1
      IF(LD.GT.182.AND.LD.LT.305) GO TO 114
      DST=ABS(RST(4)-BRST(4))
      IF(DST.GT.0.02) BRST(4)=BRST(4)-(1.9835*(DIV(I11)+DIV(I12)))
  114 CONTINUE
C
C----SECTION 29-1
C****ALL RIGHTS (P) ARE PUT IN A TEMPORARY ARRAY (T) WHICH CAN BE
C****MODIFIED DURING RIGHT ALLOCATION.  FIRST THE CANAL (OR PUMP) IS
C****IDENTIFIED (J), THEN THE REACH (IR) AND THE TRAVEL TIME TO MILNER
C****(IR).  IVV IS THE NUMBER OF THE SPECIAL RIGHTS WHICH VARY WITH
C****YEAR.  IF THE RIGHT IS ONE OF THESE, CHECK DAY (KDPIL) AND MODIFY
C****RIGHT (T) IF NECESSARY.
C
C
      DO 47 L=1,NLL
      J=ICNL(L)
      IR=IRCH(J)
      IL=LGD(IR)
      KDPIL=KD+IL
      LD=JDY(KDPIL)
      IVV=IV(L)
      IF(FEB.LT.0.001.AND.LD.GT.59) LD=LD-1
      T(L)=P(L)
      IF(KBG(L).LE.0.AND.KEN(L).LE.0) GO TO 47
      IF(KBG(L).GT.KEN(L)) GO TO 69
      IF(LD.LT.KBG(L).OR.LD.GT.KEN(L)) T(L)=0.0
      GO TO 47
   69 IF(LD.LT.KBG(L).AND.LD.GT.KEN(L)) T(L)=0.0
   47 CONTINUE
C
```

```
C----SECTION 30-1, PART 1
C****TRANSFER DATE, CANAL NUMBER, AND RESERVOIR ACCOUNT NUMBER OF EACH
C****RIGHT TO ALTERNATE ARRAYS THAT CAN BE MODIFIED LATER.
C
      DO 145 L=1,NLL
      JDATE(L)=IDATE(L)
      JCNL(L)=ICNL(L)
      JJ(L)=JR(L)
  145 CONTINUE
C
C----SECTION 30-1, PART 2
C****RIGHT ALLOCATION ROUTINE BEGINS HERE.  THE INTEGER DATE IS USED
C****TO IDENTIFY RIGHTS HAVING EQUAL PRIORITY.  THE FOLLOWING LOOP
C****CONTINUES UNTIL A RIGHT OF LESSER PRIORITY IS REACHED, AT WHICH
C****TIME TRANSFER IS MADE OUT OF THE LOOP, RIGHTS NUMBERING L1 THROUGH
C****L2 HAVING BEEN IDENTIFIED AS EQUAL FOR A TOTAL OF LN RIGHTS.  ALSO
C****A CHECK IS MADE FOR PROPER ORDER; INTEGER DATE MUST INCREASE
C****THROUGH LIST OF RIGHTS, OTHERWISE PROGRAM TERMINATES.
C
      L1=1
      NLK=NLL
   24 DO 23 L=L1,NLK
      IF(L.EQ.NLK) GO TO 27
      IF(JDATE(L+1)-JDATE(L)) 25,23,27
   25 IF(JDATE(L+1).NE.18640601) GO TO 38
      GO TO 27
   38 IF(JDATE(L+1).NE.18940702) GO TO 70
      GO TO 27
   70 IF(JDATE(L+1).NE.19110113) GO TO 56
      GO TO 27
   56 WRITE(OUT,28) JDATE(L+1),JDATE(L)
   28 FORMAT(1H1,'   RIGHT',I10,' FOLLOWING',I10,16H IS OUT OF ORDER)
      STOP
   23 CONTINUE
   27 L2=L
      LN=L2-L1+1
C
C----SECTION 30-1, PART 3
C****IF THERE IS ONLY ONE RIGHT OF THE SAME PRIORITY GO DIRECTLY TO THE
C****ROUTINE WHICH TRIES TO MEET THE RIGHT FROM NATURAL FLOW.  IF THERE
C****IS MORE THAN ONE RIGHT OF THE SAME PRIORITY, THE PROPORTIONING
C****ROUTINE ALLOCATES WATER TO RIGHTS PROPORTIONALLY IF THE SUPPLY IS
C****NOT GREAT ENOUGH TO MEET THOSE RIGHTS.
C
      IF(LN.EQ.1) GO TO 29
C
C----SECTION 31-1, PART 1
C****INITIALIZE PERCENT THAT RIGHTS IN EACH REACH WILL BE MET TO
C****100 (PCT=1.00).   THIS VARIABLE IS USED TO FACTOR RIGHTS OF EQUAL
C****PRIORITY WHEN WATER SUPPLY IS INSUFFICIENT.
C
      DO 51 L=1,NR
   51 PCT(L)=1.0
C
C----SECTION 31-1, PART 2
C****JCOUNT BECOMES THE NUMBER OF PASSES NEEDED TO SATISFY THE LN
C****RIGHTS HAVING THE SAME PRIORITY.
```

```
C
      JCOUNT=0
C
C----SECTION 31-1, PART 3
C****INITIALIZE VARIABLES USED TO ALLOCATE LN RIGHTS WITH SAME PRIORITY
C****DATE.  ONE OR MORE PASSES ARE MADE THROUGH THE ENTIRE RIVER SYSTEM
C****UNTIL ALL LN RIGHTS ARE ALLOCATED OR THE FLOW IS USED UP.  EACH
C****TIME NEW VALUES OF PCT ARE CALCULATED.  IF ENOUGH FLOW EXISTS TO
C****SATISFY ALL LN RIGHTS, ONLY ONE PASS IS NEEDED AND ALL PCT'S = 1.0
C****TRM BECOMES THE TOTAL OF LN RIGHTS MET IN REACHES UPSTREAM OF THE
C****LTH REACH.  CTR BECOMES THE TOTAL OF LN UPSTREAM RIGHTS REGARDLESS
C****OF WHETHER THEY HAVE OR HAVE NOT BEEN MET.  TR BECOMES THE TOTAL
C****OF THE LN RIGHTS IN EACH REACH.

C
   37 DO 39 L=1,NR
      TRM(L)=0.0
      CTR(L)=0.0
   39 TR(L)=0.0
C
C----SECTION 31-1, PART 4
C****SUM EQUAL PRIORITY RIGHTS BY REACH.  IF FLOW AVAILABLE FOR RIGHTS
C****IN REACH HAS BEEN ENTIRELY USED (PCT=0.0) OR IF DIVERSION HAS BEEN
C****MET (RD=0.0) THEN SET THE RIGHT EQUAL TO ZERO FOR ALL CANAL AND
C****PUMP RIGHTS (J=0), AND THE SAME IF THE SEASONAL VOLUME
C****LIMIT (SVL) HAS BEEN REACHED.  IF THERE IS NO DIVERSION FOR THE
C****RIGHT (IC=0), SKIP IT.
C
      DO 30 L=L1,L2
      IC=JCNL(L)
      IF(IC.LE.0) GO TO 30
      IR=IRCH(IC)
      J=JJ(L)
      IF(PCT(IR).LE.0.0.OR.RD(IC).LE.0.0.AND.J.EQ.0) T(L)=0.0
      IF(IVO(L).LE.0) GO TO 300
      LL=IVO(L)
      IF(SVL(L).LE.VRD(LL)) T(L)=0.0
  300 TR(IR)=TR(IR)+T(L)
   30 CONTINUE
C
C----SECTION 32-1
C****BEGIN LOOP TO DETERMINE PCT, THE PORTION THAT EACH EQUAL PRIORITY
C****RIGHT CAN BE MET GIVEN THE WATER SUPPLY AT ITS LOCATION IN RELATION
C****TO OTHER RIGHTS.  PCT IS EQUAL TO THE REMAINING NATURAL FLOW IN
C****EACH REACH (RF) DIVIDED BY THE CUMULATIVE TOTAL (CTR) OF ALL
C****RIGHTS IN AND UPSTREAM OF THE REACH BEING CONSIDERED.  TRF IS THE
C****FLOW REMAINING IN THE REACH AFTER DEDUCTING ALL UPSTREAM RIGHTS
C****WITH A PHYSICALLY POSSIBLE WATER SUPPLY (TRM).  IF EITHER TRF IS
C****GREATER THAN ZERO (ENOUGH WATER EXISTS TO MEET ALL RIGHTS) OR CTR
C****IS ZERO (NO RIGHTS HAVE YET BEEN FOUND) THE PCT IS SET EQUAL TO 1.
C****THE USE OF TRM IS NOT ESSENTIAL, BUT IT SPEEDS CALCULATIONS.
C
      DO 31 L=1,NR
      IR=IORD(L)
      IF(IR.LE.0.OR.ICF(IR).EQ.2) GO TO 31
      CTR(IR)=CTR(IR)+TR(IR)
      TRM(IR)=TRM(IR)+TR(IR)
```

```
      TRF=RF(IR)-TRM(IR)
      IF(TRF.GE.0.0.OR.CTR(IR).LE.0.0) GO TO 32
      PCT(IR)=RF(IR)/CTR(IR)
      IF(RF(IR).LE.0.1) PCT(IR)=0.0
      TRM(IR)=TRM(IR)+TRF
      GO TO 33
   32 PCT(IR)=1.0
   33 IO=IDS(IR)
      IF(IO.EQ.0) GO TO 31
      CTR(IO)=CTR(IO)+CTR(IR)
      TRM(IO)=TRM(IO)+TRM(IR)
   31 CONTINUE
C
C----SECTION 33-1
C****VALUES OF PCT ARE NOW MODIFIED SUCH THAT PCT INCREASES IN DOWN-
C****STREAM ORDER.  IF A PCT VALUE IS SMALLER THAN A PCT VALUE IN ANY
C****REACH ABOVE IT, THE UPSTREAM PCT IS REPLACED BY THE DOWNSTREAM PCT.
C
      DO 84 K=1,NCR
      DO 34 L=1,NR
      IR=IORD(L)
      IF(IR.LE.0.OR.ICF(IR).EQ.2) GO TO 34
      IO=IDS(IR)
      IF(IO.EQ.0) GO TO 34
      IF(PCT(IO).LT.PCT(IR)) PCT(IR)=PCT(IO)
   34 CONTINUE
   84 CONTINUE
C
C----SECTION 33-2
C****BEGIN LOOP TO MEET RIGHT (OR LN RIGHTS IF MORE THAN ONE RIGHT HAS
C****THE SAME PRIORITY) FROM NATURAL FLOW.  RIGHTS ARE FIRST MULTIPLIED
C****BY PCT VALUES DETERMINED IN ABOVE LOOP.  PCT WILL BE LESS THAN ONE
C****IF THERE IS MORE THAN ONE RIGHT WITH THE SAME PRIORITY DATE AND
C****THESE RIGHTS MUST SHARE A LIMITED WATER SUPPLY.  THE MODIFIED
C****RIGHT (R) IS THEN LIMITED BY THE REMAINING DIVERSION (RD) AND THE
C****REMAINING NATURAL FLOW (RF) IN EACH DOWNSTREAM REACH.
C
C****A RESERVOIR RIGHT (JJ>0) IS LIMITED BY THE RIGHT'S REMAINING SPACE,
C****(RST-BRST).
C
C****A FLOW RIGHT (JJ<0) IS LIMITED BY THE REMAINING NATURAL FLOW (RF)
C****AND ALSO THE ACTUAL FLOW (AF).  THE OPERATIONAL FLOW AT MIDDLETON
C****(IC=I13) IS ALSO REDUCED BY THE FISH FLOW RELEASE (FISHSTOR) SINCE
C****THAT FLOW IS STORED WATER.
C
C****THE RIGHT IS THEN SUBTRACTED FROM THE REMAINING FLOW (RF) IN EACH
C****DOWNSTREAM REACH, THE TOTAL RIGHT (T), AND THE REMAINING DIVERSION
C****(RD). WHEN THE THE REMAINING FLOW IN A REACH IS USED UP, THE
C****NUMBER OF THE RIGHT (LL) IS STORED IN THE IRT ARRAY.  IF IDB IS
C****POSITIVE, VALUES ARE PRINTED PRIMARILY FOR DEBUG PURPOSES.
C
   29 DO 4 LL=L1,L2
      IC=JCNL(LL)
      IF(IC.LE.0) GO TO 4
      IR=IRCH(IC)
      IF(LN.EQ.1) PCT(IR)=1.0
      R=T(LL)*PCT(IR)
```

```
      IF(R.EQ.0.0) GO TO 48
      IF(RD(IC).LT.R.AND.JJ(LL).EQ.0) R=RD(IC)
      IG=IR
      IF(JJ(LL).LT.0) GO TO 160
    3 IF(R.GT.RF(IG)) R=RF(IG)
      IF(R.LE.0.0) GO TO 48
      IG=IDS(IG)
      IF(IG.NE.0) GO TO 3
      GO TO 161
  160 IF(R.GT.RF(IG)) R=RF(IG)
      IF(R.LE.0.0) GO TO 48
      GO TO 98
  161 IF(JJ(LL).EQ.0) GO TO 180
      LRRT=JJ(LL)
      RS=(RST(LRRT)-BRST(LRRT))/1.9835
      IF(RS.LE.0.01) T(LL)=0.0
      IF(R.GT.RS) R=RS
      IF(R.LE.0.0) GO TO 48
      BRST(LRRT)=BRST(LRRT)+(R*1.9835)
      LR=MR(LRRT)
      DRA(LR)=DRA(LR)+R
      GO TO 14
   98 CHK=AF(IR)-(RD(IC)*(-1.))
      IF(IC.NE.I13) GO TO 144
      IF(JYE(KD).EQ.1999) THEN
          IL=LGD(IR)
          KDPIL=KD+IL
          LD=JDY(KDPIL)
          IF(LD.GE.171.AND.LD.LE.244) THEN
              R=AF(IR)-275.0
              IF(R.LE.0.0) R=0.0
          ELSE IF(LD.EQ.170) THEN
              R=AF(IR)-200.0
              IF(R.LE.0.0) R=0.0
              ENDIF
          ENDIF
      FISHSTOR=DIV(I11)+DIV(I12)
      CHK=CHK-FISHSTOR
C      IF(AF(IR).GT.600.) R=0.0
  144 IF(CHK.LT.0.0) CHK=0.0
      IF(R.GT.CHK)  R=CHK
      PF(IR)=PF(IR)+R
      GO TO 14
  180 IF(IVO(LL).LE.0) GO TO 14
      L=IVO(LL)
      RR=(SVL(LL)-VRD(L))/1.9835
      IF(R.GT.RR) R=RR
      IF(R.LE.0.0) GO TO 48
      VRD(L)=VRD(L)+(R*1.9835)
   14 IG=IR
      IF(JJ(LL).LT.0) GO TO 162
   11 RF(IG)=RF(IG)-R
      IF(RF(IG).LE.0.1.AND.IRT(IG).EQ.NLK) IRT(IG)=LL
      IG=IDS(IG)
      IF(IG.NE.0) GO TO 11
      GO TO 163
  162 RF(IG)=RF(IG)-R
```

```
      IF(RF(IG).LE.0.1.AND.IRT(IG).EQ.NLK) IRT(IG)=LL
  163 T(LL)=T(LL)-R
      RD(IC)=RD(IC)-R
      IF(JJ(LL).GE.0) TD(IR)=TD(IR)+R
   48 IF(IDB.GT.0) WRITE(OUT,35)LL,RF(IR),RD(IC),PCT(IR)
   35 FORMAT(1H ,I5,2F10.3,F10.7)
    4 CONTINUE
C
C----SECTION 34-1, PART 1
C****IF THE REMAINING NATURAL FLOW IN THE LAST REACH HAS BEEN ENTIRELY
C****USED, DISCONTINUE RIGHT ALLOCATION.
C
      IF(RF(IFR).LE.0.1) GO TO 6
C
C----SECTION 34-1, PART 2
C****FOR MULTIPLE RIGHTS WITH EQUAL PRIORITY, JCOUNT IS THE NUMBER OF
C****PASSES USED TO ALLOCATE THE FLOW.  IF MORE THAN 20 PASSES ARE
C****MADE, THERE IS AN ERROR IN DATA OR THIS PROGRAM. PRINT DEBUG INFO
C****FOR FIVE MORE PASSES.  IF LN IS JUST ONE, SKIP NEXT TWO ROUTINES.
C
      IF(LN.EQ.1) GO TO 26
      JCOUNT=JCOUNT+1
      IF(JCOUNT.LE.20) GO TO 49
      IDB=1
      IF(JCOUNT.LE.25) GO TO 49
      WRITE(OUT,50) JDATE(L2),L2
   50 FORMAT(32H NUMBER OF ITERATIONS TO SATISFY,I10,23H RIGHTS HAS EXCE
     1EDED 25,15H - RIGHT NUMBER,I6)
      IDB=0
      GO TO 26
C
C----SECTION 34-1, PART 3
C****IF CALCULATED PCT VALUE OF ANY REACH IS BETWEEN ZERO AND ONE,
C****FURTHER ALLOCATION OF THE CURRENT LN RIGHTS IS NEEDED.  K THEN
C****BECOMES A '1' AND THE PROGRAM RETURNS FOR ANOTHER PASS THROUGH
C****THE SYSTEM TO CALCULATE NEW PCT VALUES.  OTHERWISE, CONTINUE.
C
   49 K=0
      DO 36 L=1,NR
   36 IF(PCT(L).GT.0.00001.AND.PCT(L).LT.0.999) K=1
      IF(K.EQ.1) GO TO 37
C
C----SECTION 35-1, PART 1
C****THE NEXT RIGHT TO ALLOCATE IS THE LAST RIGHT ALLOCATED PLUS 1.
C
   26 L1=L2+1
C
C----SECTION 35-1, PART 2
C****IF THERE ARE NO MORE RIGHTS, ALLOCATION IS COMPLETED, OTHERWISE
C****RETURN TO ALLOCATE NEXT RIGHT OR SERIES OF RIGHTS.
C
      IF(L1.GT.NLK) GO TO 6
      GO TO 24
```

```
      6 CONTINUE
C
C----SECTION 36-1, PART 3
C****ALLOCATION IS COMPLETED FOR ONE DAY AT THIS POINT.
C
  138 CONTINUE
C
C----SECTION 38-1, PART 2
C****THE DIVERSION MADE FROM NATURAL FLOW (DM) IS EQUAL TO THE TOTAL
C****DIVERSION LESS THE REMAINING DIVERSION (RD) IF JC IS NOT NEGATIVE.
C****IF JC IS NEGATIVE, THEN RIGHT IS FOR POWER AND DM IS NOT USED
C****SINCE DIV HAS NOT BEEN DEFINED. IF RIGHT IS EITHER A RESERVOIR
C****(JC>0) OR UNMEASURED POWER RIGHT (JC=-1), SET REMAINING DIVERSION
C****TO ZERO SINCE IT DOES NOT APPLY.  EACH REMAINING DIVERSION (RD) IS
C****CONVERTED TO ACRE-FEET (AD).  THE REMAINING DIVERSION IS SUBTRACTED
C****FROM EACH DIVERSION'S RESERVOIR STORAGE ACCOUNT (SRM), AND ADDED
C****TO ITS SEASONAL USE (STOR).
C
      DO 5 L=1,ND
      DM(L)=0.0
      IF(JC(L).GE.0)  DM(L)=DIV(L)-RD(L)
      IF(JC(L).GT.0.OR.JC(L).EQ.-1)  RD(L)=0.0
      AD(L)=RD(L)*1.9835
      SRM(L)=SRM(L)-AD(L)
      STOR(L)=STOR(L)+AD(L)
      IF(JC(L).NE.0) GO TO 5
      TDIV(L)=TDIV(L)+(DIV(L)*1.9835)
    5 CONTINUE
C
C----SECTION 38-1, PART 3
C****COMBINE STORAGE USED BY CANALS HAVING COMMON STORAGE ACCOUNTS:
C****NEW YORK(I10) AND PENITENTIARY(I9), EAGLE ISLAND CANALS(I14 THRU
C****I22), AND PHYLISS(I5) AND CALDWELL HIGHLINE.
C
      SRM(I10)=SRM(I10)+SRM(I9)
      SRM(I9)=0.0
C      DO 142 L=15,22
C      IJJ=ISPEC(L)
C      SRM(14)=SRM(14)+SRM(IJJ)
C  142 SRM(IJJ)=0.0
      SRM(I5)=SRM(I5)+SRM(I23)
      SRM(I23)=0.0
C
C----SECTION 39-1, PART 1
C****THE PORTION OF THE FLOW LEAVING EACH REACH THAT IS STORED WATER
C****(SF) IS EQUAL TO THE TOTAL FLOW LESS THE REMAINING NATURAL FLOW
C****(RF, THE NATURAL FLOW NOT DIVERTED BY UPSTREAM RIGHTS), AND THE
C****FLOW RESERVED FOR POWER RIGHTS (PF).
C
      DO 13 L=1,NR
   13 SF(L)=AF(L)-RF(L)-PF(L)
C
C----SECTION 39-1, PART 2
C****THE IRT ARRAY IS CORRECTED SUCH THAT NO UPSTREAM VALUE IS GREATER
C****THAN A DOWNSTREAM VALUE.  SINCE IRT IS THE NUMBER OF THE LAST
C****RIGHT FILLED IN A REACH FOR WHICH THE REMAINING NATURAL FLOW WAS
C****COMPLETELY USED, EACH REACH UPSTREAM IS SUBJECT TO EITHER THAT
```

```
C****RIGHT OR A RIGHT OF EARLIER PRIORITY.  IT CANNOT BE SUBJECT TO A
C****LATER RIGHT REGARDLESS OF WATER SUPPLY.
C
      DO 83 L=1,NCR
      DO 57 J=1,NR
      IR=IORD(J)
      IF(IR.LE.0.OR.ICF(IR).EQ.2) GO TO 57
      IO=IDS(IR)
      IF(IO.EQ.0) GO TO 57
      IF(IRT(IO).LT.IRT(IR)) IRT(IR)=IRT(IO)
   57 CONTINUE
   83 CONTINUE
C
C****SECTION 39-1, PART 3
C****DETERMINE WHICH PERCENT OF FILL IS IN EFFECT FOR UPPER SYSTEM (USE
C****REACH 2 AS INDICATOR) FOR STEWART AND BRYAN DECREES.
C
      IERT=IRT(2)
      ISTEW=1
      IBRYAN=1
      DO 75 L=1,IERT
      IF(L.EQ.IERT) GO TO 75
      IF(JDATE(L+1)-JDATE(L)) 86,75,75
   86 IF(JDATE(L+1).NE.18640601) GO TO 76
      ISTEW=ISTEW+1
      GO TO 75
   76 IF(JDATE(L+1).NE.18940702) GO TO 137
      IBRYAN=IBRYAN+1
      GO TO 75
  137 IF(JDATE(L+1).NE.19110113) GO TO 56
   75 CONTINUE
C
C----SECTION 40-1
C****SUM REACH AND DIVERSION DATA FOR SYSTEM TOTALS
C
      DO 63 L=1,20
   63 TT(L)=0.0
      DO 64 L=1,NR
      IF(ICF(L).EQ.2) GO TO 64
      TT(1)=TT(1)+TD(L)
      TT(2)=TT(2)+TRD(L)
      TT(3)=TT(3)+AG(L)
      TT(18)=TT(18)+TRE(L)
   64 CONTINUE
      DO 90 L=1,NP
   90 SAC(L)=0.0
      DO 93 L=1,NSR
      J=MR(L)
   93 SAC(J)=SAC(J)+BRST(L)
      DO 91 L=1,NP
      TT(15)=TT(15)+DRA(L)
      TT(10)=TT(10)+SAC(L)
  182 TT(11)=TT(11)+RES(L,1)
      TT(12)=TT(12)+RES(L,2)
```

```
      CEV(L)=CEV(L)+(EVAP(L)*1.9835)
      TT(17)=TT(17)+CEV(L)
   91 TT(4)=TT(4)+DS(L)
      DO 65 L=1,ND
      TT(5)=TT(5)+DIV(L)
      TT(6)=TT(6)+DM(L)
      TT(7)=TT(7)+RD(L)
      TT(8)=TT(8)+AD(L)
      TT(16)=TT(16)+SRM(L)
      TT(19)=TT(19)+TDIV(L)
      IF(L.LT.I13) TT(9)=TT(9)+STOR(L)
   65 CONTINUE
      DO 96 L=1,NSR
      TT(13)=TT(13)+RST(L)
   96 TT(14)=TT(14)+BRST(L)
      GT(1)=GT(1)+(TT(4)*1.9835)
      IF(SF(10).GT.0.0) GT(2)=GT(2)+(SF(10)*1.9835)
      GT(3)=GT(3)+(TT(15)*1.9835)
      IF(SF(10).LT.0.0) GT(4)=GT(4)+(SF(10)*(-1.9835))
C
C----SECTION 40-2
C****ON AUGUST 15 RESET RESERVOIR FILL TO ZERO TO ALLOW NEW FILL
C
      IF(JDY(KD).NE.227) GO TO 94
      DO 95 L=1,NSR
      BRST(L)=0.0
      J=MR(L)
   95 SAC(J)=0.0
      GT(5)=TT(14)
      TT(14)=0.0
      TT(10)=0.0
   94 CONTINUE
C
C----SECTION 41-1, PART 1
C****FIND MONTH AND DAY OF MONTH FROM ABSOLUTE DAY OF YEAR.
C
      JD(2)=28
      IF(FEB.LT.0.001) JD(2)=29
      LDY=JDY(KD)
      MON=0
   81 MON=MON+1
      IF(MON.GT.12) GO TO 82
      LDY=LDY-JD(MON)
      IF(LDY.GT.0) GO TO 81
      LDY=LDY+JD(MON)
   82 CONTINUE
C
C----SECTION 41-1, PART 2
C****BEGIN REPORT PRINTOUT - OUTPUT CAN BE MADE AT 3 DIFFERENT PLACES
C****AS KO VARIES THE OUTPUT UNIT (KOUT).  FOR OUTPUT UNITS 3 OR 4,
C****ONLY THE FINAL KED DAYS WILL BE PRINTED OUT.
C
      DO 271 KO=1,3
      OUT=KOUT(KO)
      IF(OUT.LE.0) GO TO 271
      JED=NED-KED
      IF(OUT.EQ.3.AND.KD.LE.JED) GO TO 271
```

```
      IF(OUT.EQ.4.AND.KD.LE.JED) GO TO 271
C
C----SECTION 41-2
C****PRINT OUT REACH DATA.  FIND ACTUAL DATE (UNLAGGED) FOR WHICH DATA
C****WAS MEASURED AT EACH REACH (OM AND LDAY).
C
      CALL DATE(RUNDATE)
      IF(KD.GT.NED) WRITE(OUT,165) OM(MON),LDY,JYE(KD),RUNDATE
  165 FORMAT(1H1/31X,52HWATER DISTRICT 63 - BOISE RIVER FLOW ACCOUNTING
     1 -  ,A3,I3,1H,,I5,14H   (PROJECTED),9X,A9)
      IF(KD.LE.NED) WRITE(OUT,18) OM(MON),LDY,JYE(KD),RUNDATE
   18 FORMAT(1H1/31X,52HWATER DISTRICT 63 - BOISE RIVER FLOW ACCOUNTING
     1 -  ,A3,I3,1H,,I5,18X,A9)
      WRITE(OUT,164)
  164 FORMAT(/ 6X,18HREACH FLOWS IN CFS,8X,
     1 88H ACTUAL  NATURAL   ACTUAL RMAINING  OPERATN   STORED RESRVOIR
     2 NATURAL   TOTAL     REACH/  34X,97H DATE     FLOW     FLOW NAT FL
     3OW     FLOW     FLOW     EVAP FLOW DIV  RCH DIV     GAIN LAST RIGH
     4T/)
      DO 58 I=1,NR
      L=IORD(I)
      IF(L.LE.0.OR.ICF(L).EQ.2) GO TO 58
      IL=LAG(L)
      LYEAR=JYE(KD+IL)
      LDAY=JDY(KD+IL)
      FEB=LYEAR/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      JD(2)=28
      IF(FEB.LT.0.001) JD(2)=29
      MO=0
  270 MO=MO+1
      IF(MO.GT.12) GO TO 269
      LDAY=LDAY-JD(MO)
      IF(LDAY.GT.0) GO TO 270
      LDAY=LDAY+JD(MO)
  269 CONTINUE
      ER=0.0
      J=IRS(L)
      IF(J.GT.0) ER=EVAP(J)
      M=IRT(L)
```

```
      C=BLANK
      IF(ICF(L).EQ.1) C=AST
      WRITE(OUT,19)L,(RCHNM(L,J),J=1,6),C,OM(MO),LDAY,F(L),AF(L),RF(L),P
     1F(L),SF(L),ER,TD(L),TRD(L),AG(L),IDATE(M)
   19 FORMAT(2X,I2,2X,6A4,1X,A1,1X,A3,I3,9F9.0,I11)
   58 CONTINUE
      WRITE(OUT,22) (TT(L),L=1,3)
   22 FORMAT(/26X,42H* - INDICATES FLOW ESTIMATED, NOT MEASURED,16X,
     1  9HTOTALS    ,3F9.0)
      ICF(10)=0
C
C----SECTION 42-1
C****PRINT OUT RESERVOIR DATA
C
      WRITE(OUT,97)
   97 FORMAT( /5X,74HRESERVOIR        PREV CONT CURR CONT CHNG CONT ACCR
     1 STOR TOTL STOR TOTL EV,  3X, 50HPRIORITY  RESERVOIR            RI
     2GHT      STORED  /   22X, 57H   (AF)      (AF)     (CFS)     (CFS)
     3      (AF)     (AF) ,34X,4H(AF),8X,4H(AF)/)
      DO 100 L=1,NP
      I=MR(L)
  100 WRITE(OUT,62) L,(RESNM(L,J),J=1,4),(RES(L,J),J=1,2),DS(L),DRA(L),
     2SAC(L),CEV(L),L,(RESNM(I,J),J=1,4),RST(L),BRST(L)
   62 FORMAT(2X,I2,1X,4A4,5F10.1,F8.1,3X,I5,3X,4A4,2F12.1)
      L=NP+1
      I=MR(L)
C     WRITE(OUT,66) L,(RESNM(I,J),J=1,4),RST(L),BRST(L)
C  66 FORMAT(82X,I5,3X,4A4,2F12.1)
      WRITE(OUT,101) TT(11),TT(12),TT(4),TT(15),TT(10),TT(17),TT(13),
     1TT(14)
  101 FORMAT(/5X,5HTOTAL,11X,5F10.1,F8.1,11X,5HTOTAL,11X,2F12.1)
C     WRITE(OUT,67) TT(13),TT(14)
C  67 FORMAT(90X,5HTOTAL,11X,2F12.1)
      WRITE(OUT,119)
  119 FORMAT(22X,49HCHANGE IN   STORAGE MIDDLETON     TOTAL    UNACCT)
      WRITE(OUT,120) GT(5)
  120 FORMAT(24X,47HCONTENT      USED    STORED    STORED    STORED,
     1 15X,25HTOTAL EARLY SEASON FILL -,7X,F12.1)
      WRITE(OUT,121) GT(1),TT(9),(GT(L),L=2,4),ISTP(ISTEW),IBRP(IBRYAN)
  121 FORMAT(/21H   YEAR-TO-DATE AF   ,5F10.1,15X,16HSTEWART DECREE -,
     1  I4,1H%,5X,14HBRYAN DECREE -,I4,1H%)
C
C----SECTION 43-1, PART 1A
C****PRINT DIVERSION DATA - IF THIS DATA IS NOT DESIRED EACH DAY, IT
C****CAN BE PRINTED EVERY JDP DAYS.  CHANGE DIVERSION DATA TO INTEGERS
```

```
C****(II) FOR PRINTOUT.  ELIMINATE RESERVOIR AND UNMEASURED FLOW
C****DIVERSIONS FROM PRINTOUT.  DIVIDE DIVERSIONS INTO THREE PARTS FOR
C****THREE COLUMNS IN PRINTOUT.
C
      KK=KK+1
      IF(KK.NE.JDP.AND.KD.NE.5.AND.KD.NE.NED) GO TO 271
      KK=0
      IF(TT(5).LE.0.1) GO TO 271
      K=0
      DO 118 L=1,ND
      IF(JC(L).NE.0.AND.JC(L).NE.-2) GO TO 118
      K=K+1
      II(1,K)=DIV(L)+0.5
      II(2,K)=RD(L)+0.5
      II(3,K)=STOR(L)+0.5
      II(4,K)=SRM(L)+0.5
      IF(SRM(L).LT.0.0) II(4,K)=SRM(L)-0.5
      DO 124 J=1,4
  124 DNAM(K,J)=DIVNM(L,J)
  118 CONTINUE
      NK=K
C
C----SECTION 43-1, PART 1B
C****DIVIDE DIVERSIONS INTO THREE EQUAL PARTS FOR COLUMNS ON REPORT
C
      AKE=NK/3.
      KE=AKE
      R=AKE-KE
      IF(R.GT.0.1) KE=KE+1
      KEE=KE*2
      IF(KOD.GT.0) GO TO 109
C     WRITE(OUT,20) OM(MON),LDY,JYE(KD)
   20 FORMAT( ///46X,23HDIVERSION    DATA -    ,A3,I3,1H,,I5)
      WRITE(OUT,110)
  110 FORMAT(/3(23X,21HCFS  CFS   AF     AF   )/3(44H     DIVERSION
     1   DIVN STOR  USED   RMNG)/)
      DO 112 L=1,KE
      LKE=L+KE
      LKEE=L+KEE
      IF(LKEE.GT.NK) GO TO 295
      WRITE(OUT,111)      L ,(DNAM(L,J),J=1,4),(II(K,L),K=1,4),
     1       LKE ,(DNAM(L+KE,J),J=1,4),(II(K,L+KE),K=1,4),
     2       LKEE ,(DNAM(L+KEE,J),J=1,4),(II(K,L+KEE),K=1,4)
  111 FORMAT(3(1X,I3,1X,4A4,2I5,I6,I7))
      GO TO 112
  295 WRITE(OUT,296)      L ,(DNAM(L,J),J=1,4),(II(K,L),K=1,4),
     1       LKE ,(DNAM(L+KE,J),J=1,4),(II(K,L+KE),K=1,4)
  296 FORMAT(2(1X,I3,1X,4A4,2I5,I6,I7))
  112 CONTINUE
      GO TO 102
C
C----SECTION 43-1, PART 2
C****IF KOD IS GREATER THAN ZERO, MORE DETAILED OUTPUT WILL BE MADE OF
```

```
C****DIVERSION DATA.
C
  109 CONTINUE
      WRITE(OUT,20) OM(MON),LDY,JYE(KD)
      WRITE(OUT,113)
  113 FORMAT(//48X,3HCFS,4X,11HCFS NATURAL,2X,11HCFS STORAGE,3(1X,11H AF
     1 STORAGE),6X,2HAF / 10X,9HDIVERSION,16X,5HREACH,7X,7HDIVRTD ,
     23(3X,9HDIVERSION),2X,38HUSED TO DATE REMAINING  DIVRTD TO DATE/)
      WRITE(OUT,21) (L,(DIVNM(L,J),J=1,5),IRCH(L),DIV(L),DM(L),RD(L),
     1AD(L),STOR(L),SRM(L),TDIV(L),L=1,ND)
   21 FORMAT(1X,I7,2X,5A4,I9,7F12.1)
C
C----SECTION 44-1, PART 1
C****PRINT MISCELLANEOUS DATA AND DAILY TOTALS
C
  102 CONTINUE
C      WRITE(OUT,114)
C  114 FORMAT(/22X,58HCFS DIVN   CFS NAT  CFS STOR   AF STOR   AF USED
C     1 AF REM)
C      WRITE(OUT,115)(TT(L),L=5,9),TT(16)
C  115 FORMAT(16H   DAILY TOTALS ,4X,6F10.0)
  271 CONTINUE
C
C----SECTION 44-2
C****IF LUCKY PEAK RIGHT HAS JUST FILLED, RESET STORAGE USE AND
C****SUPPLY OF FLOW USES TO ZERO SINCE THEY CAN NO LONGER USE
C****CARRYOVER, BUT WILL BE CHARGED FROM THE NEW FILL.  RESET OUTPUT
C****UNIT TO PRIMARY VALUE.  IF DAY IS A PROJECTED DAY, SKIP THE NEXT
C****SECTION WHICH STORES DATA ON THE ALLOCATIONS FILE.
C
      DST=ABS(RST(4)-BRST(4))
      IF(DST.GT.0.05.OR.DRA(3).LT.0.05) GO TO 139
      STOR(I11)=0.0
      STOR(I12)=0.0
      SRM(I11)=0.0
      SRM(I12)=0.0
      DO 140 KO=1,3
      OUT=KOUT(KO)
      IF(OUT.LE.0) GO TO 139
  140 WRITE(OUT,141)
  141 FORMAT(/2X,'FISH & GAME AND USBR STORAGE HAS BEEN SET TO ZERO AT T
     1HE END OF THIS DAY.  THIS USE WILL NO LONGER BE SUBTRACTED FROM RE
     2SERVOIRS.')
  139 OUT=KOUT(1)
      IF(KD.GT.NED) GO TO 72
C
C----SECTION 45-1
C****WRITE DIVERSION, RESERVOIR, FLOW, AND MISC DATA ON TEMPORARY FILE
C****FOR ADDITION TO ALLOCATION FILE.
```

```
C
      DO 333 L=1,ND
  334 WRITE(16,260) (IDN(L),JDT(KD),DIV(L),RD(L),STOR(L),SRM(L),
     1 TDIV(L),L=1,ND)
  260 FORMAT(I8,1HD,I7,5F10.1)
  333 CONTINUE
      DO 261 L=1,NP
      IDNJ=IRESN(L)
C     WRITE(16,262)   IDNJ,JDT(KD),CEV(L)
      WRITE(16,262)   IDNJ,JDT(KD),CEV(L),EVAP(L),DRA(L),SAC(L),RES(L,2)
  262 FORMAT(I8,1HR,I7,5F10.1)
  261 CONTINUE
      DO 272 L=1,NR
      M=IRT(L)
      WRITE(16,263) JRN(L),JDT(KD),F(L),AF(L),SF(L),AG(L),
     1IDATE(M)
  263 FORMAT(I8,1HF,I7,4F10.1,I10)
  272 CONTINUE
      WRITE(16,264) JDT(KD),(BRST(I),I=1,NSR)
  264 FORMAT(9H99999901M,I7,4F10.1)
      WRITE(16,266) JDT(KD),(GT(I),I=1,5)
  266 FORMAT(9H99999902M,I7,5F10.1)
C
C----SECTION 46-1, PART 1
C****CONTINUE TO NEXT DAYS' CALCULATIONS
C
   72 CONTINUE
C
C----SECTION 46-1, PART 2
C****IF LRA > 0, A LISTING IS MADE OF THE FIRST LRA RECORDS FROM THE
C****TEMPORARY OUTPUT FILE FOR DEBUG PURPOSES.
C
      IF(LRA.LE.0) GO TO 45
      LR=0
      WRITE(OUT,153)
  153 FORMAT(1H1)
      ENDFILE 16
      REWIND 16
  152 READ(16,150,END=45) NMR,TYP,NDT,(FI(I),I=1,10)
  150 FORMAT(I8,A1,I7,10F10.1)
      WRITE(OUT,151) NMR,TYP,NDT,(FI(I),I=1,10)
  151 FORMAT(6X,I8,A1,I7,10F10.1)
      LR=LR+1
      IF(LR.LE.LRA) GO TO 152
   45 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	15075	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	3913	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	245528	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	264516	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIWRA$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AKE		1824=	1825	1826					
**	R*4	AREA		961=	962						
2-0003BB20	R*4	AST		74D	1755						
2-0003BB1C	R*4	BLANK		74D	1754						
**	R*4	C		1754=	1755=	1756					
**	R*4	CHK		1457=	1472(2)=	1474(2)=	1475(2)				
**	R*4	DST		1235=	1236	1882=	1883				
**	R*4	DSTOR		1105=	1106	1109=	1111				
2-0003BC5C	R*4	DTEMP		642=	653	668=	693	699			
**	R*4	ER		1750=	1752=	1756					
2-0003BC70	R*4	ETEMP		720=	748	754					
**	R*4	EVNET		953=	954(2)=	962					
**	R*4	EVPR		1008=	1011=	1045	1107=	1110=	1111		
**	R*4	FEB		548=	549	550(2)=	551	601=	602	603(2)=	604
				627=	628	629(2)=	630	856=	857	858(2)=	898
				1233	1255	1693	1738=	1739	1740(2)=	1742	
**	R*4	FISHSTOR		1471=	1472						
2-0003BC40	R*4	FTEMP		531=	582	588					
**	I*4	I		404=	405	421	422	431	445=	448(2)=	454
				463	528=	529	537	639=	649	650	653
				665=	666(3)	673	674	675	685(2)	694(2)	695
				699	718=	725	726	727	733(2)	734(2)	742(2)
				743	747	749(2)	750	754	756	757	761
				774=	775(2)	776	777(2)	798=	808	809	810
				813	814	817=	818	1732=	1733	1774=	1775
				1779=	1919(2)=	1921(2)=	1939(2)=	1941(2)=			
2-0003BBF0	I*4	I1		497=	880(2)	881(2)	882(2)	883(2)			
2-0003BC14	I*4	I10		506=	1579(2)						
2-0003BC18	I*4	I11		507=	1236	1471	1884	1886			
2-0003BC1C	I*4	I12		508=	1236	1471	1885	1887			
2-0003BC20	I*4	I13		509=	1458	1666					

**	I*4	I14	510=						
2-0003BBF4	I*4	I2	498=	880(2)	881				
2-0003BC24	I*4	I23	511=	1585	1586				
2-0003BC28	I*4	I24	512=	882(2)	883				
**	I*4	I25	513=						
2-0003BBF8	I*4	I3	499=	884(2)	885(2)				
2-0003BBFC	I*4	I4	500=	884(2)	885				
2-0003BC00	I*4	I5	501=	886(2)	887(2)	1585(2)			
2-0003BC04	I*4	I6	502=	886(2)	887				
2-0003BC08	I*4	I7	503=	888	889(2)	1120			
2-0003BC0C	I*4	I8	504=	666	888	889	1120		
2-0003BC10	I*4	I9	505=	1579	1580				
**	I*4	IA	390=	391	392	393	395	396	402
2-0003BBD8	I*4	IB	388=	390	395	396=			
2-0003BB74	I*4	IBG	223=						
2-0003BB88	I*4	IBGD	232=	247=	320	796			
2-0003BB84	I*4	IBGS	231=	252=	316	827			
**	I*4	IBRYAN	1621=	1629(2)=	1792				
**	I*4	IC	1185=	1186(2)	1187(3)	1188	1190	1350=	1351	1352
			1354	1430=	1431	1432	1436(2)	1457	1458	1494(2)
			1496							
2-0003BBD4	I*4	ID	363=	369	398	422				
2-0003BB4C	I*4	IDB	223=	1496	1515=	1520=				
**	I*4	IDNJ	1908=	1910						
2-0003BD30	I*4	IERT	1619=	1622	1623					
**	I*4	IFEB	549=	550	602=	603	628=	629	857=	858
			1739=	1740						
2-0003BB44	I*4	IFR	223=	1504						
**	I*4	IG	1437=	1439(2)	1441(2)=	1442	1444(2)	1484=	1486(2)	1487(3)
			1488(2)=	1489	1491(2)	1492(3)				
**	I*4	IJ	1026=	1027(2)	1036=	1037				
**	I*4	IL	869=	870	900=	901	996=	997	1151=	1152
			1230=	1231	1251=	1252	1460=	1461	1735=	1736
			1737							
2-0003BB54	I*4	ILS	223=	374	398	420	477			
2-0003BBD0	I*4	IM	363=	369	398	422				
2-0003BB2C	I*4	IN	176=	223	258	271	272	273	274	275
			282	288	289	290	295	297		
2-0003BB8C	I*4	INI	240=	242	246=	247	251=	252		
**	I*4	IO	1085=	1086	1087(2)	1188=	1189	1190(2)	1385=	1386
			1387(2)	1388(2)	1400=	1401	1402(2)	1609=	1610	1611(2)
**	I*4	IR	867=	868	869	936=	937	939(2)	1101=	1102
			1103	1104	1111(5)	1250=	1251	1352=	1354	1358(2)
			1374=	1375(2)	1376(3)	1377(3)	1378(2)	1379	1380(3)	1381(2)
			1382(2)	1384	1385	1387	1388	1398=	1399(2)	1400
			1402(2)	1432=	1433	1434	1437	1457	1460	1464
			1467	1476(2)	1484	1495(2)	1496(2)	1607=	1608(2)	1609
			1611(2)							
**	I*4	IS	1009=	1010	1011	1104=	1105(2)	1106	1107	
**	I*4	ISTEW	1620=	1626(2)=	1792					

2-0003BBDC	I*4	IT	389=	390	392	393=			
2-0003BC50	I*4	ITD	600=	604=	607				
**	I*4	IVV	1254=						
**	I*4	IW	958=	959	960(2)=	961(3)			
2-0003BB34	I*4	IX	178=	363					
2-0003BBCC	I*4	IY	363=	369	398	422			
2-0003BB30	I*4	IZ	177=	336	344				
**	I*4	J	295(2)=	297(2)=	336(2)=	344(2)=	422(4)=	446=	447	458(2)
			461	484=	485	486	489	529=	541	542
			556	559	562	573(2)	581	583(2)	584	588
			779=	780(2)	977=	978(2)	993=	994	1148=	1149(2)
			1249=	1250	1353=	1354	1606=	1607	1649=	1650(2)
			1682=	1683	1751=	1752(2)	1756(2)=	1775(6)=	1816=	1817(2)
			1839(6)=	1844(4)=	1860(2)=					
2-0003BD00	I*4	JCOUNT	1322=	1513(2)=	1514	1516				
**	I*4	JDI	623=	641	648=	651=	797=	799	807=	811=
2-0003BB5C	I*4	JDP	223=	1804						
**	I*4	JED	1711=	1712	1713					
**	I*4	JFI	522=	530	534=	544=	560=	575=		
**	I*4	JRI	716=	719	723=	735=				
**	I*4	JTI	622=	640	644=	646=	667	671=	687=	
**	I*4	K	351=	375(2)=	376	380=	458(4)=	555=	556	1396=
			1529=	1531=	1532	1807=	1810(2)=	1811	1812	1813
			1814	1815	1817	1819	1839(6)=	1844(4)=		
**	I*4	KA	973=	974(2)=	975	977				
**	I*4	KD	851=	856	870	897	901	985	997	1066
			1117	1152	1170	1231	1252	1459	1461	1679
			1694	1712	1713	1720(2)	1723(2)	1736	1737	1804(2)
			1855	1896	1903	1910	1915	1919	1921	
**	I*4	KDPIL	870=	871	874	901=	906	911	912	913
			915(2)	916	917(2)=	918(2)	997=	998	1007	1018
			1051	1069	1152=	1153	1231=	1232	1252=	1253
			1461=	1462						
**	I*4	KE	1825=	1826	1827(2)=	1828	1835	1836	1839(2)	1844(2)
2-0003BB80	I*4	KED	223=	1711						
**	I*4	KEE	1828=	1837	1839(2)					
**	I*4	KK	843=	1803(2)=	1804	1805=				
**	I*4	KO	1708=	1709	1888=	1889				
2-0003BB68	I*4	KOD	223=	1829						
**	I*4	KT	975=	976	1156=	1157(2)=	1158(2)=	1159	1160	
**	I*4	L	258(2)=	271(2)=	272(2)=	273(2)=	274(2)=	275(2)=	282(2)=	288(2)=
			289(2)=	290(2)=	295(2)=	297(2)=	303=	304	305	306
			317(2)=	318(2)=	319(2)=	320(2)=	322=	323	324=	325
			336(5)=	344(4)=	362=	363(7)	369	387	391	398(2)
			400	402	404	411	412	414	422(2)	431
			443=	447	453	458(2)	483=	485	486	489
			524=	526	538	539	540	543	548	551
			554(2)	559	565	573(2)	577	578	582	583(2)
			588	605=	611	612	613	624=	627	630
			633(2)	645	653	678	685(2)	689	690	693
			694(2)	699	708	717=	730	733(2)	734(2)	737
			738	741	742(2)	747	749(2)	754	755	756

			761	773=	775	776	777	778	780	828(2)=
			829(2)=	840=	842	866=	867	872	874(2)	899=
			900	902	903	904	905	907(2)	912(3)	913(2)
			915(3)	916(2)	918(2)	926=	927	928	929	930
			935=	936	938	939	951=	952	953(2)	955(2)
			956(2)	961(3)	962	963(2)	972=	973	978(2)	994=
			995	996	999	1009	1025	1027(2)	1028(2)	1037
			1038	1045(5)	1051(2)	1063	1064(4)	1065	1068(2)	1069(2)
			1085	1087	1100=	1101	1146=	1149(2)	1150	1151
			1154(2)	1156	1160(3)	1176=	1177(2)	1184=	1185	1198=
			1199(2)	1211=	1212	1213	1214	1215(2)	1216=	1217
			1248=	1249	1254	1256(2)	1257(2)	1258(2)	1259(3)	1261(3)
			1268=	1269(2)	1270(2)	1271(2)	1285=	1286	1287(2)	1288
			1290	1292	1294(2)	1298	1315=	1316	1336=	1337
			1338	1339	1349=	1350	1353	1354	1355	1356
			1357(2)	1358	1373=	1374	1397=	1398	1479=	1480
			1483(2)	1530=	1531(2)	1563=	1564	1565(4)	1566(3)	1567(2)
			1568(3)	1569(3)	1570	1571(3)	1594=	1595(4)	1605=	1622=
			1623	1624(2)	1625	1628	1631	1637=	1638	1639=
			1640	1641	1642	1643	1644	1646=	1647	1648=
			1649	1650	1651=	1652	1653	1654	1655	1656(3)
			1657	1658	1659=	1660	1661	1662	1663	1664
			1665	1666(2)	1668=	1669	1670	1680=	1681	1682
			1733=	1734(2)	1735	1751	1753	1755	1756(10)	1760(2)=
			1773=	1774	1775(10)	1778=	1779	1792(2)=	1808=	1809(2)
			1811	1812	1813	1814	1815(2)	1817	1835=	1836
			1837	1839(7)	1844(5)	1860(11)=		1902=	1903(7)=	1907=
			1908	1910(5)	1913=	1914	1915(5)			
**	I*4	L1	1283=	1285	1299	1349	1429	1537=	1543	
2-0003BCF4	I*4	L2	1298=	1299	1349	1429	1517(2)	1537		
2-0003BB64	I*4	LBC	223=	438						
**	I*4	LD	897=	898(3)=	1232=	1233(3)=	1234(2)	1253=	1255(3)=	1259(2)
			1261(2)	1462=	1463(2)	1466				
**	I*4	LDAY	1737=	1746(2)=	1747	1748(2)=	1756			
**	I*4	LDY	1694=	1698(2)=	1699	1700(2)=	1720	1723	1855	
**	I*4	LI	442=	449(2)=	450	452=	455(2)=	464(2)=	467	469=
**	I*4	LKE	1836=	1839	1844					
**	I*4	LKEE	1837=	1838	1839					
**	I*4	LL	1356=	1357	1429=	1430	1434	1436	1438	1447
			1448	1450	1478	1479	1480	1485	1487	1492
			1493(2)	1495	1496					
2-0003BCF8	I*4	LN	1299=	1308	1433	1512				
**	I*4	LR	1454=	1455(2)	1934=	1943(2)=	1944			
2-0003BB78	I*4	LRA	223=	1933	1944					
**	I*4	LRRT	1448=	1449(2)	1453(2)	1454				
**	I*4	LYEAR	1736=	1738						
**	I*4	M	421=	422	453=	458	1753=	1756	1914=	1915
2-0003BB24	I*4	MD1	155D	546	625					
2-0003BB28	I*4	MD2	155D	547	626					
**	I*4	MDD1	546=	552(2)=	554	625=	631(2)=	633		
**	I*4	MDD2	547=	553(2)=	554	626=	632(2)=	633		
**	I*4	MO	1743=	1744(2)=	1745	1746	1748	1756		

**	I*4	MON	1066=	1067(2)=	1068	1117=	1118(2)=	1119	1695=	1696(2)=
			1697	1698	1700	1720	1723	1855		
**	I*4	MPG	1130=	1131	1134	1135				
**	I*4	N	841=	842	976=	977	1132=	1133	1147=	1148
			1159=	1160						
2-0003BB48	I*4	NCR	223=	1396	1605					
2-0003BB38	I*4	ND	223=	303	320	336	389	443	484	486
			639	665	779	798	866	935	1198	1563
			1659	1808	1860	1902	1903			
2-0003BB7C	I*4	NDF	223=	597	598	599	772			
2-0003BC44	I*4	NDT	536=	538	562	577	584	610=	611	642=
			645	668=	675	689	695	720=	727	737
			743	750	757	800=	802	818=	820	828=
			829=	831	1939=	1941				
2-0003BC3C	I*4	NDY	531=	536	540	606(2)=	607	608=	610	613
**	I*4	NED	523=	525(2)=	596	597	598	775	777	1007
			1069	1153	1711	1720	1723	1804	1896	
2-0003BC48	I*4	NED1	596=	597=	605	773	1018			
2-0003BC64	I*4	NEDD	708=	778	780					
2-0003BC4C	I*4	NEDF	598=	605	773	851				
2-0003BB58	I*4	NEP	223=	343	344					
**	I*4	NK	1819=	1824	1838					
2-0003BCEC	I*4	NLK	1284=	1285	1286	1487	1492	1543		
2-0003BB40	I*4	NLL	223=	362	446	1214	1248	1268	1284	
**	I*4	NMD	527=	569	574(2)=	664=	681	686(2)=		
2-0003BC30	I*4	NMR	531=	535	541	562	578	581	642=	649
			650	668=	672	673	674	675	690	720=
			724	725	726	727	738	800=	803	808
			818=	821	828=	829=	832	1939=	1941	
2-0003BB50	I*4	NP	223=	290	297	317	718	774	817	899
			951	1216	1646	1651	1773	1778	1907	
2-0003BB3C	I*4	NR	223=	271	272	273	274	275	288	289
			295	840	926	972	993	1100	1146	1176
			1184	1211	1315	1336	1373	1397	1530	1594
			1606	1639	1732	1913				
2-0003BB70	I*4	NRC	223=	282	528					
2-0003BB6C	I*4	NSP	223=	483						
2-0003BB60	I*4	NSR	223=	319	828	1648	1668	1680	1919	
2-0003BBC4	I*4	NSVL	352=	413(2)=	414					
2-0003BC38	I*4	NYE	531=	536	539	601	609(2)=	610	612	
**	I*4	OUT	175	259=	377	398	422	439	451	456
			458	465	468	486	562	566	570	578
			584	675	679	682	690	695	727	731
			738	743	750	757	803	821	832	1294
			1496	1517	1709=	1710	1712	1713	1720	1723
			1726	1756	1760	1768	1775	1782	1787	1789
			1792	1832	1839	1844	1855	1856	1860	1889=
			1890	1891	1895=	1935	1941			
2-0003BC74	R*4	PTEMP	720=	755	761					
**	R*4	R	959=	961	1434=	1435	1436(2)=	1439(2)=	1440	1444(2)=
			1445	1451(2)=	1452	1453	1455	1464=	1465(2)=	1467=
			1468(2)=	1475(2)=	1476	1481(2)=	1482	1483	1486	1491

			1493	1494	1495	1826=	1827	
**	R*4	RI	956=	957(2)=	958	959		
2-0003BCC4	R*4	RICD	1119=	1121	1130	1131		
**	R*4	RIDC	1121=					
**	R*4	RR	1480=	1481(2)				
**	R*4	RS	1449=	1450	1451(2)			
2-0003BC6C	R*4	RTEMP	720=	741	747			
2-0003BB10	CHAR	RUNDATE	22	1719A	1720	1723		
2-0003BC88	R*4	STEMP1	800=	813				
2-0003BC8C	R*4	STEMP2	800=	814				
**	R*4	SUM	1012=	1027(2)=	1028	1037(2)=	1038	
**	R*4	TCL	444=	461(2)=	463	465		
**	R*4	TRF	1378=	1379	1382			
**	R*4	TRIV	1120=	1121(2)				
2-0003BC34	R*4	TYP	531=	642=	668=	720=	800=	818=	828=	829=
			1939=	1941				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-0002441C	R*4	A		36600	(25, 366)	38	559=	573(2)=	583(2)=	588=
						1051				
2-00007154	R*4	AD		300	(75)	27	1567=	1568	1569	1663
						1860				
2-00023CB4	R*4	AF		100	(25)	36	999=	1045=	1051=	1064=
						1068(2)=	1069(2)=	1087	1111	1133
						1134	1135(2)	1136(2)=	1137(2)=	1457
						1464	1467	1595	1756	1915
2-000240FC	R*4	AG		100	(25)	38	1177=	1187	1643	1756
						1915				
2-0003BAE0	R*4	AMF		48	(12)	69	170D	1068		
2-0003BAB0	R*4	ARICD		48	(12)	69	169D	1119		
2-0002D374	R*4	ARINT		16	(4)	43	125D	955	956	
2-000065B8	R*8	BRST		32	(4)	57	89	319=	828=	1235
						1236(2)=	1449	1453(2)=	1650	1670
						1681=	1775	1882	1919	
2-0002D474	R*4	CEV		16	(4)	44	317=	818=	1656(2)=	1657
						1775	1910			
2-00000000	R*8	CTR		200	(25)	34	89	1338=	1376(2)=	1379
						1380	1387(3)=			
2-00007028	R*4	DIV		300	(75)	27	872=	874=	880(4)=	881(3)=
						882(4)=	883(3)=	884(4)=	885(3)=	886(4)=
						887(3)=	888(2)=	889(3)=	939	1120(2)
						1199	1236(2)	1471(2)	1565	1571
						1660	1811	1860	1903	
2-000066C8	R*4	DIVNM		1500	(75, 5)	27	336=	422	458	1817
						1860				
2-00006EFC	R*4	DM		300	(75)	27	1564=	1565=	1661	1860
2-00022B20	R*4	DNAM		1500	(75, 5)	29	1817=	1839(3)	1844(2)	
2-0002D354	R*4	DRA		16	(4)	43	1217=	1455(2)=	1652	1775

```
                                                          1883      1910

2-00024228  R*4  DS                    100  (25)          38       1106=     1658      1775
2-000074D8  R*4  DVV                 109800  (75, 366)     28        113D      653=      685(2)=   694(2)=
                                                          699=      780(2)=   874
    **      R*4  ECH                     4  (1)           62
2-0002D444  R*4  EV                     16  (4)           44        902=      912=      913=       953
                                                          955
2-0002D384  R*4  EVAP                   16  (4)           44        952=      962=      963(2)=    964(2)=
                                                         1011      1107      1656      1752       1910

2-0002D484  R*4  EVP                  5856  (4, 366)      45        734(2)=   749(2)=   754=       775(2)=
                                                          911       912       913
    **      R*4  EXC                  1464  (1, 366)      62        114D
2-0003A1BC  R*4  EXCNM                  16  (1, 4)       62        344=
2-00023160  R*4  F                     100  (25)          34        930=      1187(2)=  1190(3)=   1215
                                                         1756      1915
2-0003B90C  R*4  FI                     80  (20)          68       1939=      1941

2-0002328C  R*4  G                    1500  (25, 15)      34        978(2)=   1111=     1130(3)    1131=
                                                         1133(2)=  1134=      1135=     1136       1137
                                                         1154      1160
2-00023F70  R*4  GAIN                  100  (25)          37       1028=      1038=     1045
2-0002428C  R*4  GAV                   400  (25, 4)       38        842=      1027(2)   1028       1037
                                                         1149(2)=  1150=      1154=     1160(2)=   1177
2-000065D8  R*8  GT                     80  (10)          68         89       318=      829=      1671(2)=
                                                         1672(2)=  1673(2)=   1674(2)=  1684=      1789
                                                         1792(2)   1921
2-0003BA74  I*4  IBRP                   16  (4)           69        166D      1792

2-00023E44  I*4  ICF                   100  (25)          36        275=      985=      1063       1186
                                                         1375      1399      1608      1640       1734
                                                         1755      1763=
2-0003B354  I*4  ICFMD                1464  (366)         67        526=      543=      985
2-000351F4  I*4  ICN                  3400  (850)         51        363=      387       391
2-00030244  I*4  ICNL                 3400  (850)         50        400=      402=      404        447
                                                         1249      1270
2-00023F0C  I*4  IDA                   100  (25)          37        274=      973       1156

2-00030F8C  I*4  IDATE                3400  (850)         50        369=      458       1269       1756
                                                         1915
2-00007280  I*4  IDN                   300  (75)          28        336=      391       458        485
                                                          649      650       673       674        675
                                                          695      808       1903
2-000231C4  I*4  IDS                   100  (25)          34        271=      1085      1188       1385
                                                         1400      1441      1488      1609
2-0003A1B4  I*4  IEN                     4  (1)           62        344=
2-0003A1B8  I*4  IER                     4  (1)           62        344=

2-00022418  I*4  II                   1500  (5, 75)       29       1811=      1812=     1813=      1814=
                                                         1815=     1839(3)   1844(2)
2-000238CC  I*4  IORD                  100  (25)          35        272=      994       1101       1185
                                                         1374      1398      1607      1733
2-00006CA4  I*4  IRCH                  300  (75)          27        336=      421       453        867
                                                          936      1250      1352      1432       1860
2-0002D454  I*4  IRESN                  16  (4)           44        119D      725       726        727
                                                          743      750       757       1908
```

2-00023DE0	I*4	IRS	100	(25)	36	273=	1009	1103	1104
					1751				
2-00023EA8	I*4	IRT	100	(25)	37	1214=	1487(2)=	1492(2)=	1611(4)=
					1619	1753	1914		
2-0003BA84	I*4	ISFS	44	(11)	69	167D	556		
2-0003B968	I*4	ISPEC	96	(24)	68	80D	485	486	489=
					497	498	499	500	501
					502	503	504	505	506
					507	508	509	510	511
					512	513			
2-0003BA68	I*4	ISTP	12	(3)	69	165D	1792		
2-00032A1C	I*4	IV	3400	(850)	50	363=	1254		
2-000344AC	I*4	IVO	3400	(850)	51	411=	414=	1355	1356
					1478	1479			
2-000222EC	I*4	JC	300	(75)	28	336=	431=	666(2)	938
					1565	1566(2)	1570	1809(2)	
2-00038714	I*4	JCNL	3400	(850)	52	1270=	1350	1430	
2-0003A1FC	I*4	JD	48	(12)	67	96D	1692=	1693=	1698
					1700	1741=	1742=	1746	1748
2-000379CC	I*4	JDATE	3400	(850)	52	1269=	1287(2)	1288	1290
					1292	1294(2)	1517	1624(2)	1625
					1628	1631			
2-0003A22C	I*4	JDT	1464	(366)	67	538=	577	578	611=
					645	689	690	737	738
					802	820	831	1903	1910
					1915	1919	1921		
2-0003AD9C	I*4	JDY	1464	(366)	67	540=	551	554(2)	613=
					630	633(2)	897	1066	1117
					1232	1253	1462	1679	1694
					1737				
2-0003945C	I*4	JJ	3400	(850)	52	1271=	1353	1436	1438
					1447	1448	1485	1495	
2-00031CD4	I*4	JR	3400	(850)	50	363=	431	1271	
2-00024038	I*4	JRN	96	(24)	37	141D	541	556	562
					581	584	1915		
2-0003A7E4	I*4	JYE	1464	(366)	67	539=	548	612=	627
					856	1459	1720	1723	1736
					1855				
2-00036C84	I*4	KBG	3400	(850)	51	363=	1257	1258	1259
					1261				
2-00035F3C	I*4	KEN	3400	(850)	51	363=	1257	1258	1259
					1261				
2-00023FD4	I*4	KNRC	100	(25)	37	282=	529		
2-0003B95C	I*4	KOUT	12	(3)	68	258=	259	1709	1889
					1895				
2-00023B88	I*4	LAG	100	(25)	35	288=	996	1151	1735
2-00023BEC	I*4	LGD	100	(25)	35	289=	869	1230	1251
					1460				
2-0002D364	I*4	LGG	16	(4)	43	290=	900		
2-00024098	I*4	MGA	100	(25)	37	149D	1025		
2-0003A1A4	I*4	MR	16	(4)	57	108D	1454	1649	1682
					1774	1779			

2-0003A1CC	R*4	OM	48	(12)	67	94D	398	422	1720
					1723	1756	1855		
2-00003078	R*8	P	6800	(850)	50	89	363=	398	422
					458	461	1256		
2-00023868	R*4	PCT	100	(25)	35	1316=	1354	1380=	1381=
					1384=	1402(4)=	1433=	1434	1496
					1531(2)				
2-00024160	R*4	PF	100	(25)	38	1213=	1476(2)=	1595	1756
2-0002D464	R*4	PP	16	(4)	44	903=	915=	953	
2-0002EB64	R*4	PPT	5856	(4, 366)	45	756=	761=	776=	915(2)
2-00023930	R*4	RCHNM	600	(25, 6)	35	295=	422	458	1756
2-00006DD0	R*4	RD	300	(75)	27	1199=	1354	1436(2)	1457
					1494(2)=	1496	1565	1566=	1567
					1662	1812	1860	1903	
2-00000258	R*8	RES	64	(4, 2)	43	89	904=	905=	916=
					918=	956	1105(2)	1654	1655
					1775	1910			
2-0002D314	R*4	RESNM	64	(4, 4)	43	297=	1775(2)		
2-000230FC	R*4	RF	100	(25)	34	1215=	1378	1380	1381
					1439(2)	1444(2)	1486(2)=	1487	1491(2)=
					1492	1496	1504	1595	1756
2-00023D7C	R*4	RIN	100	(25)	36	927=	1045	1064	1087(2)=
					1111				
2-000002B8	R*8	RSS	11712	(4, 366)	45	89	733(2)=	742(2)=	747=
					777(2)=	916	918		
2-00006598	R*8	RST	32	(4)	57	89	102D	1235	1449
					1669	1775	1882		
2-0002D394	R*4	SA	176	(11, 4)	44	132D	961(3)		
2-00000298	R*8	SAC	32	(4)	43	89	1647=	1650(2)=	1653
					1683=	1775	1910		
2-00023C50	R*4	SF	100	(25)	36	1595=	1672(2)	1674(2)	1756
					1915				
2-000221C0	R*4	SRM	300	(75)	28	306=	320=	810=	814=
					1568(2)=	1579(3)=	1580=	1585(3)=	1586=
					1664	1814	1815(2)	1860	1886=
					1887=	1903			
2-000073AC	R*4	STOR	300	(75)	28	304=	809=	813=	1569(2)=
					1666	1813	1860	1884=	1885=
					1903				
2-00033764	R*4	SVL	3400	(850)	51	363=	412	1357	1480
2-00004B08	R*8	T	6800	(850)	50	89	1256=	1259=	1261=
					1354=	1357=	1358	1434	1450=
					1493(2)=				
2-00023228	R*4	TD	100	(25)	34	1212=	1495(2)=	1641	1756
2-000229F4	R*4	TDIV	300	(75)	29	305=	1571(2)=	1665	1860
					1903				
2-00000190	R*8	TR	200	(25)	36	89	1339=	1358(2)=	1376
					1377				
2-00023D18	R*4	TRD	100	(25)	36	928=	939(2)=	1045	1064
					1111	1136	1137	1642	1756
2-000241C4	R*4	TRE	100	(25)	38	929=	1045	1064	1111

					1644					
2-000000C8	R*8	TRM	200	(25)	34	89	1337=	1377(2)=	1378	
					1382(2)=	1388(3)=				
**	R*4	TSE	4	(1)	62	325=				
2-00006628	R*8	TT	160	(20)	68	89	1638=	1641(2)=	1642(2)=	
					1643(2)=	1644(2)=	1652(2)=	1653(2)=	1654(2)=	
					1655(2)=	1657(2)=	1658(2)=	1660(2)=	1661(2)=	
					1662(2)=	1663(2)=	1664(2)=	1665(2)=	1666(2)=	
					1669(2)=	1670(2)=	1671	1673	1684	
					1685=	1686=	1760	1782(8)	1792	
					1806					
2-0003B9C8	R*4	VRD	160	(40)	68	323=	1357	1480	1483(2)=	

LABELS

Address	Label	References							
**	1	1198	1199#						
**	2	1211	1215#						
0-00002438	3	1439#	1442						
0-000026AE	4	1429	1431	1498#					
0-000029F6	5	1563	1570	1572#					
0-0000297C	6	1504	1543	1545#					
1-000000A8	7'	223	258	271	272	273	274	275	276#
		282	288	289	290				
1-000000CC	8'	363	364#						
1-000000AD	9'	295	296#						
**	10	1216	1217#						
0-000025DC	11	1486#	1489						
0-0000166A	12	772	782#						
**	13	1594	1595#						
0-000025D2	14	1456	1477	1478	1484#				
0-000020DB	15	1184	1186	1189	1191#				
**	16	926	930#						
1-000000B4	17'	297	298#						
1-000007EB	18'	1723	1724#						
1-00000910	19'	1756	1758#						
1-00000B64	20'	1831#	1855						
1-00000CB2	21'	1860	1862#						
1-0000092C	22'	1760	1761#						
0-00002384	23	1285	1287	1297#					
0-0000232C	24	1285#	1544						
**	25	1287	1288#						
0-00002968	26	1512	1521	1537#					
0-00002388	27	1286	1287	1289	1291	1293	1298#		
1-00000702	28'	1294	1295#						
0-000023C0	29	1308	1429#						
0-00002834	30	1349	1351	1359#					
0-00002900	31	1373	1375	1386	1389#				

0-000028D4	32	1379	1384#				
0-000028DC	33	1383	1385#				
0-0000295D	34	1397	1399	1401	1403#		
1-00000732	35'	1496	1497#				
**	36	1530	1531#				
0-00002762	37	1336#	1532				
0-00002360	38	1288	1290#				
**	39	1336	1339#				
0-00001BE7	40	935	937	938	940#		
0-00001EFD	41	1100	1102	1112#			
0-00001ED0	42	1103	1109#				
0-00001ED4	43	1108	1111#				
1-000000B9	44'	317	318	319	320	321#	
0-00003ADC	45	223	477	1933	1939	1945#	
0-00000B84	46	438	471#				
0-000022B4	47	1248	1257	1260	1262#		
0-0000265F	48	1435	1440	1445	1452	1482	1496#
0-00002728	49	1514	1516	1529#			
1-00000740	50'	1517	1518#				
**	51	1315	1316#				
0-000009F1	52	362	405	432#			
1-000000E5	53'	377	378#				
0-00000877	54	374	376	381#			
1-00000145	55'	422	424#				
0-00002B26	56	1292	1294#	1631			
0-00002AB5	57	1606	1608	1610	1612#		
0-000030C4	58	1732	1734	1759#			
**	59	303	306#				
0-00001D38	60	998	1007#				
0-00000674	61	316	320#				
1-00000A43	62'	1775	1777#				
**	63	1637	1638#				
0-00002BB5	64	1639	1640	1645#			
**	65	1659	1667#				
0-00000FD4	68	599	615#				
0-000022A4	69	1258	1261#				
0-00002370	70	1290	1292#				
0-00001978	71	866	868	873	875#		
0-000039F1	72	851	1170	1896	1927#		
0-00001E6D	73	993	995	1000	1086	1088#	
0-00001B2A	74	899	906	919#			
0-00002B5C	75	1622	1623	1624(2)	1627	1630	1632#
0-00002B04	76	1625	1628#				
**	77	976	978#				
**	78	972	979#				
1-0000034C	80'	642	668	669#			
0-00002E08	81	1696#	1699				
0-00002E30	82	1697	1701#				

```
     **       83                          1605      1613#

     **       84                          1396      1404#
     **       85                           840       841       842#
     **       86                          1624      1625#
  0-00001145  87                           668#      672       673
  0-00002024  88                          1153      1156#

  1-000000BF  89'                          336       337#      344
     **       90                          1646      1647#
     **       91                          1651      1658#
     **       92                          1176      1178#
     **       93                          1648      1650#

  0-00002DE5  94                          1679      1687#
     **       95                          1680      1683#
     **       96                          1668      1670#
  1-0000096E  97'                          1768      1769#
     **       98                          1446      1457#

     **       99                          1147      1149#
     **       100                         1773      1775#
  1-00000A65  101'                         1782      1784#
  0-00003737  102                          1848      1867#
  0-00000B7D  103                           443       467       470#

  1-00000166  104'                          439       440#      451       468
  0-00000B5A  105                           463       467#
  0-00000B29  106                           446       447       462#
  1-000001AA  107'                          458       460#
  1-000001C4  108'                          465       466#

  0-00003620  109                          1829      1854#
  1-00000B8C  110'                         1832      1833#
  1-00000BE1  111'                         1839      1842#
  0-00003614  112                          1835      1843      1847#
  1-00000C11  113'                         1856      1857#

  0-00002232  114                          1234      1237#
  0-00001C90  116                           951       955       965#
  0-00000A77  117                           450       453#
  0-00003411  118                          1808      1809      1818#
  1-00000A8A  119'                         1787      1788#

  1-00000AC0  120'                         1789      1790#
  1-00000B16  121'                         1792      1793#
  0-00001964  122                           871       874#
  1-000001A6  123'                          456       457#
     **       124                          1816      1817#

  0-0000088C  125                           390#      394       397
     **       126                           391      392#
  0-000008C4  127                           391      402#
  0-000008B4  128                           391      395#
  0-000008CC  129                           387      392       395       398#

  1-00000121  130'                          398      399#
```

0-00000924	131	401	403#		
0-000009E7	132	420	425#		
**	133	483	490#		
0-00000BE3	134	485	489#		
**	135	484	488#		
1-000001D3	136'	486	487#		
0-00002B18	137	1628	1631#		
**	138	1550#			
0-000037BB	139	1883	1890	1895#	
**	140	1888	1891#		
1-00000CC4	141'	1891	1892#		
0-000025BC	144	1458	1474#		
**	145	1268	1272#		
1-00000DA0	150'	1939	1940#		
1-00000DAC	151'	1941	1942#		
0-00003A2C	152	1939#	1944		
1-00000D9C	153'	1935	1936#		
0-00000A9D	154	454	458#		
0-00000946	155	412	415#		
**	156	322	323#		
0-00002508	160	1438	1444#		
**	161	1443	1447#		
0-00002614	162	1485	1491#		
0-0000263E	163	1490	1493#		
1-00000835	164'	1726	1727#		
1-00000791	165'	1720	1721#		
**	175	324	325#		
0-000024C8	180	1447	1478#		
**	181	755#			
**	182	1654#			
0-00001584	183	755	761#		
1-000005E0	184'	757	758#		
0-00001066	185	642#	649		
0-00001058	186	645	649#		
0-00001284	187	689	693#		
0-0000112C	188	670	672#		
0-00001128	189	667	671#		
0-00001048	190	643	645#		
0-00001044	191	640	644#		
0-00001040	192	641	648#		
0-00001530	193	748	754#		
1-00000269	194'	566	567#	679	731
0-000010C4	195	650	653#		
0-000010D9	196	639	652	654#	
0-000012FB	197	647	655	701#	
0-000010E4	198	633	664#		
0-00001015	199	630	633#		
**	201	624	702#		

0-00001244	202	674	689#				
1-00000358	203'	675	676#				
0-00001305	204	642	668	708#			
0-000012F5	205	665	666	688	698	700#	
1-000003E5	206'	690	691#				
0-00001224	207	681	685#				
0-00001484	208	737	741#				
0-000014DC	209	741	747#				
**	210	717	763#				
1-00000475	211'	720	721#				
0-0000158E	212	718	736	753	760	762#	
1-000004EE	213'	738	739#				
0-0000159F	214	720	764#				
0-00001349	215	720#	724	725			
0-00001B0C	216	907	916#				
0-00000CF4	218	542	546#				
0-00000E08	219	541	577#				
**	220	524	590#				
0-00000F15	221	528	545	561	576	587	589#
0-00000F25	222	531	591#				
1-000002BA	223'	578	579#				
1-00000204	224'	562	563#				
0-00000CD4	225	537	541#				
1-000001F6	226'	531	532#				
0-00000E56	227	531#	535	581			
0-0000174C	230	798	812	815#			
0-00001752	231	800	816#				
1-00000649	232'	803	804#				
1-0000063D	233'	800	801#				
0-00002060	235	1146	1155	1161#			
**	236	1159	1160#				
0-00001DC3	237	1018	1029	1045#			
**	238	1036	1037#				
0-00001D9C	239	1025	1035#				
0-000017EF	240	796	818	826#			
0-000017E8	241	817	820	825#			
1-0000068B	242'	821	822#				
1-00000681	243'	818	819#				
0-00001E56	245	1046	1063	1085#			
0-00000FA9	247	607	610#				
**	248	1026	1027#				
0-00000DEC	249	569	573#				
0-000018D8	250	827	828	831	835#		
1-000006DF	251'	832	833#				
1-000006D3	252'	828	829	830#			
**	253	1063#					
**	254	779	780#				

```
     **       255                    774       777#
0-00001663    256                    773       778       781#
     **       257                    605       614#
0-00001D55    258                   1010      1012#

0-00001DEC    259                   1007      1051#
1-00000D4D    260'                  1903      1905#
     **       261                   1907      1912#
1-00000D5A    262'                  1910      1911#
1-00000D67    263'                  1915      1917#

1-00000D76    264'                  1919      1920#
1-00000D89    266'                  1921      1922#
0-00002FB1    269                   1745      1749#
0-00002F9C    270                   1744#     1747
0-00003737    271                   1708      1710      1712      1713      1804      1806      1873#

     **       272                   1913      1918#
0-00000C94    278                    530       534#
0-00000C98    279                    533       535#
1-0000029E    280'                   570       571#
1-000003C5    281'                   682       683#

0-0000132C    282                    722       724#
0-00001444    283                    726       737#
1-00000481    284'                   727       728#
0-0000141C    285                    730       733#
0-00000DC8    286                    565       569#

0-00001200    287                    678       681#
0-00000E48    288                    577       581#
0-00001328    289                    719       723#
1-0000051B    290'                   743       744#
0-000014E6    291                    746       748#

1-0000057D    292'                   750       751#
0-00003570    295                   1838      1844#
1-00000BF9    296'                  1844      1846#
0-00000F04    297                    582       588#
1-000002EA    299'                   584       585#

0-00002827    300                   1355      1358#
0-000012E0    301                    693       699#
1-00000413    302'                   695       696#
0-00000D2B    317                    551       554#
0-00000D68    318                    554       558       562#

     **       319                    555       557#
0-00000D54    320                    556       559#
     **       321                   1132      1133#
0-00001716    330                    802       808#
0-00001738    331                    808       813#

0-00001714    332                    799       807#
     **       333                   1902      1906#
     **       334                   1903#
0-00001E40    363                   1065      1069#
```

```
  0-00000787  480                          343       345#

  1-00000069  496'                         248       249#
  1-00000029  497'                         243       244#
  1-00000000  498'                         238       239#
  0-000002CA  499                          242       253#
  1-00000026  500'                         240       241#      246       251


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                              References

   R*8  ABS                               1235      1882
        ASSIGN                             183       184       185       186       187       188       189       190
                                           191       192       193       194
        DATE                              1719


  +-----------------------------------------------+
  |            KEY TO REFERENCE FLAGS             |
  |    =   - Value Modified                       |
  |    #   - Defining Reference                   |
  |    A   - Actual Argument, possibly modified   |
  |    D   - Data Initialization                  |
  |   (n)  - Number of occurrences on line        |
  +-----------------------------------------------+


COMMAND QUALIFIERS

  FORTRAN/LIST/CROSS_REFERENCE BOIWRA

  /ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
  /BLAS=(INLINE,MAPPED)
  /CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
  /DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
  /DIRECTIVES=(DEPENDENCE)
  /MATH_LIBRARY=(ACCURATE,NOV5)
  /PARALLEL=(NOAUTOMATIC,NOMANUAL)
  /SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
            NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
  /CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
  /NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=WD:[WRDIST63]BOIWRA.LIS;2
  /OBJECT=WD:[WRDIST63]BOIWRA.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

Run Time:	5.03 seconds
Elapsed Time:	6.22 seconds
Page Faults:	18391
Dynamic Memory:	4960 pages

BOF HSTLST. 415

BOIHSTLST.LIS

```
C  ***************************************************
C     THIS PROGRAM LISTS DATA FROM WD-63 HISTORY FILES
C      IN USGS STYLE RECORDS BY IRRIGATION YEAR
C
C          WRITTEN USING VAX FORTRAN
C ***************************************************
C
      LOGICAL*1 FILES(40)
      BYTE CNTRL
      INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
      REAL MEAN,MAX,MIN,NOVALUE
      CHARACTER NAME*50,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
      CHARACTER TTL*50,BLANK*1
      DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
      DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12)
      DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
      CNTRL='00'X
      NOVALUE=9999999.
      BLANK=' '
      OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
      OPEN(UNIT=8,FILE='BOIHSTLST.OUT',STATUS='NEW',RECL=133)
  500 PRINT *,'*************************************'
      PRINT *,'   YOU HAVE A CHOICE:                '
      PRINT *,'                                                    '
      PRINT *,'   LIST ALL DIVERSIONS, FLOWS, DRAINS, OR RESERVOIRS '
      PRINT *,'  (STATION NUMBERS AND TITLES AUTOMATICALLY PROVIDED)'
      PRINT *,'                                                    '
      PRINT *,'              OR                    '
      PRINT *,'                                                    '
      PRINT *,'   LIST STATIONS INDIVIDUALLY                        '
      PRINT *,'   (YOU PROVIDE THE STATION NUMBER AND TITLE) '
      PRINT *,'                                  '
      PRINT *,'   TO LIST ALL DIVERSIONS, ANSWER  D         '
      PRINT *,'   TO LIST ALL RIVER FLOWS, ANSWER  F         '
      PRINT *,'   TO LIST ALL DRAINS, ANSWER  W         '
      PRINT *,'   TO LIST ALL RESERVOIRS, ANSWER  R         '
      PRINT *,'   TO LIST INSTREAM FLOW RELEASES, ANSWER  S    '
      PRINT *,'   TO LIST INDIVIDUAL STATIONS, ANSWER  I  '
      PRINT *,'   TO ENTER YOUR OWN GROUP OF STATIONS, ANSWER G '
      PRINT *,'   TO QUIT, ANSWER  Q  '
      PRINT *,'                                    '
      PRINT 904
  904 FORMAT('        ENTER CHOICE:'$)
      ACCEPT 902, IND
  902 FORMAT(A1)
      IF( IND .EQ. 'Q' ) GO TO 470
      PRINT *,' '
      PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
      PRINT *,'   IRIGATION YEAR FOR WHICH DATA IS TO BE LISTED - '
      PRINT *,' '
      PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR IRRIGATION YEAR  '
      PRINT *,'   1990, ENTER 89-90   '
      PRINT *,'                __-__   '
      PRINT 901
  901 FORMAT('+                '$)
      KY=0
      ACCEPT 900, NP,NP1
```

```
  900 FORMAT(I2,1X,I2)
C
C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
C
      NP=NP+1900
      NP1=NP1+1900
      IF(JMOD(NP,4).EQ.0) KY=1
      ID=(NP*1000)+305+KY
      KY=0
      IF(JMOD(NP1,4).EQ.0) KY=1
      ID1=(NP1*1000)+304+KY
C
      IF( IND .EQ. 'I' ) THEN
                  OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
  1               PRINT *,'   ENTER STATION NUMBER AND TITLE '
                  PRINT *,'   TO END   ENTER   99999999  '
                  PRINT *,'              ________ '
                  PRINT 903
903               FORMAT('+            '$)
                  ACCEPT 905, ISTA,TTL
905               FORMAT(I8,A50)
                  IF( ISTA .EQ. 99999999 ) GO TO 9
                  WRITE(3,906) ISTA,TTL
906               FORMAT(I8,A50)
                  GO TO 1
  9               REWIND 3
                  GO TO 10
                  END IF
      IF( IND .NE. 'G' ) GO TO 8
      PRINT 907
  907 FORMAT('     ENTER FILE NAME:'$)
      ACCEPT 908, IC,(FILES(I),I=1,IC)
  908 FORMAT(Q,40A1)
      CALL ASSIGN(3,FILES)
      GO TO 10
    8 IF(IND.EQ.'D') OPEN(UNIT=3,FILE='BOIDIV.TTL',STATUS='OLD')
      IF(IND.EQ.'F') OPEN(UNIT=3,FILE='BOIFLO.TTL',STATUS='OLD')
      IF(IND.EQ.'W') OPEN(UNIT=3,FILE='BOIDRN.TTL',STATUS='OLD')
      IF(IND.EQ.'R') OPEN(UNIT=3,FILE='BOIRES.TTL',STATUS='OLD')
      IF(IND.EQ.'S') OPEN(UNIT=3,FILE='BOIIFR.TTL',STATUS='OLD')
   10 READ(3,301,END=450) STA_ID,NAME
  301 FORMAT(I8,A50)
C
C***** FILL ARRAY WITH MISSING VALUE INDICATORS
C
         DO 20 I=1,12
         DO 20 J=1,31
20       DV(I,J)=NOVALUE

C
         IF(STA_ID.EQ.STATION_NUM) GO TO 60

25       READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
701      FORMAT(I8,A1,I4,I3,F7.0)
         IF(STA_ID.LT. STATION_NUM ) THEN
                                IF( TD .NE.STA_ID) THEN
                                                PRINT 601,STA_ID
```

```
601     FORMAT(1X,'STATION ',I8,' NOT FOUND')
                                                GO TO 10
                                                END IF
        GO TO 30
        END IF
      IF(STA_ID.EQ. STATION_NUM ) GO TO 60
      IF(STA_ID.GT. STATION_NUM ) GO TO 25
   30 IF(STA_ID.NE. STATION_NUM .AND. TD .EQ.STA_ID) GO TO 100
C
C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
C
   60 IDATE=(YEAR*1000)+ND
      IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 25
      JD(2)=28
      IF( JMOD(YEAR,4) .EQ. 0 ) JD(2)=29
      LD=ND
      MON=0
   62 MON=MON+1
      IF(MON .GT. 12 ) GO TO 68
      LD=LD-JD(MON)
      IF( LD .GT. 0 ) GO TO 62
      LD=LD+JD(MON)
      DV(MON,LD)=VALUE
C
C***** SET VARIABLES TO PRESENT VALUES IN PREPARATION FOR
C***** READING NEXT RECORD
C
   68 CONTINUE
      TD=STATION_NUM
      P_YEAR=YEAR
      P_TYPE=TYPE
      GO TO 25
C
C***** ENTIRE YEAR FOR ONE STATION READ IN -
C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
C
  100 LEAP=0
      IF(JMOD(NP1,4).EQ.0) LEAP=1
      IF( P_TYPE .NE. 'R' ) GO TO 120
         DELTA_S(1)=DV(1,31)-DV(12,31)
         DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
         DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
         DELTA_S(4)=DV(4,30)-DV(3,31)
         DELTA_S(5)=DV(5,31)-DV(4,30)
         DELTA_S(6)=DV(6,30)-DV(5,31)
         DELTA_S(7)=DV(7,31)-DV(6,30)
         DELTA_S(8)=DV(8,31)-DV(7,31)
         DELTA_S(9)=DV(9,30)-DV(8,31)
         DELTA_S(10)=DV(10,31)-DV(9,30)
         DELTA_S(11)=NOVALUE
         DELTA_S(12)=DV(12,31)-DV(11,30)
        DO 119 I=2,12
119     IF( ABS(DELTA_S(I)) .GE. 8000000. ) DELTA_S(I)=NOVALUE
120      DO 135 I=1,12
         D=31
         IF( I .EQ. 2 ) D=28+LEAP
         IF( I .EQ. 4 .OR. I .EQ. 6 .OR. I .EQ. 9 ) D=30
```

```
            CFSD(I)=DV(I,1)
            D_ADJ=0
            IF( DV(I,1) .EQ. NOVALUE )THEN
                                    CFSD(I)=0.0
                                    D_ADJ=1
                                    END IF
            MAX(I)=DV(I,1)
           IF( DV(I,1) .EQ. NOVALUE ) MAX(I)=-1.0
            MIN(I)=DV(I,1)
            DO 130 J=2,D
           DZ=DV(I,J)
            IF( DV(I,J) .EQ. NOVALUE ) THEN
                                     DZ=0.0
                                     D_ADJ=D_ADJ+1
                                     END IF
            CFSD(I)=CFSD(I)+DZ
           IF( DV(I,J) .EQ. NOVALUE ) THEN
                                     GO TO 129
                                     END IF
            IF( DV(I,J) .GT. MAX(I) ) MAX(I)=DV(I,J)
    129     IF( DV(I,J) .LT. MIN(I) ) MIN(I)=DV(I,J)
    130     CONTINUE
           IF( D_ADJ .GE. D ) THEN
                              CFSD(I)=NOVALUE
                              DELTA_S(I)=NOVALUE
                              MEAN(I)=NOVALUE
                              GO TO 133
                              END IF
            MEAN(I)=CFSD(I)/(D-D_ADJ)
    133     ACFT(I)=CFSD(I)*1.9835
    135     CONTINUE
            CALL DATE(RUNDATE)
            WRITE(8,801) RUNDATE
      801 FORMAT('1',/,120X,A9)
          IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
      800 FORMAT(1H+,'PRELIMINARY DATA')
            WRITE(8,802) TD,NAME
    802     FORMAT(//,40X,I8,6X,A50)
            IF( P_TYPE .EQ. 'R' ) THEN
                                WRITE(8,803) NP,NP1
    803     FORMAT(33X,'CONTENTS IN ACRE FEET, IRRIGATION YEAR NOVEMBER',
         *        I5,' TO OCTOBER',I5)
                                GO TO 160
            ELSE
           WRITE(8,804) NP,NP1
    804     FORMAT(33X,'DISCHARGE, CUBIC FEET PER SECOND, IRRIGATION YEAR',
         *        ' NOVEMBER',I5,' TO OCTOBER',I5)
            WRITE(8,805)
    805     FORMAT(59X,'MEAN VALUES')
           END IF
    160     WRITE(8,806)
    806     FORMAT(//,1X,'DAY',11X,'NOV',7X,'DEC',7X,'JAN',7X,'FEB',7X,
         *        'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',7X,'AUG',
         *        7X,'SEP',7X,'OCT')
            WRITE(8,807)
    807     FORMAT(' ')
            DO 180 J=1,31
```

```
            IF( J .EQ.6 .OR. J .EQ. 11 .OR. J .EQ. 16 .OR. J .EQ. 21
     *         .OR. J .EQ. 26 ) THEN
                                WRITE(8,807)
                                 END IF
            DO 170 I=1,12
            II=I - 2
           IF( I .EQ. 1 ) II=11
            IF( I .EQ. 2 ) II=12
           IF( II .EQ. 11 ) WRITE(8,807)
           IF( II .EQ. 11 ) WRITE(8,811) CNTRL,J
811                          FORMAT(A1,I3,'    ')
           IF( DV(II,J) .EQ. NOVALUE ) THEN
                                        WRITE(8,809) CNTRL
                                        GO TO 170
                                        END IF
            IF( DV(II,J) .GE. 10.0 ) THEN
                                       IDV=JNINT(DV(II,J))
                                       WRITE(8,810) CNTRL,IDV
810         FORMAT(A1,I10)
                                       GO TO 170
                                       END IF
           IF( DV(II,J) .LT. 10.0 ) THEN
                                        WRITE(8,808) CNTRL,DV(II,J)
808         FORMAT(A1,F10.1)
                                        GO TO 170
                                        END IF
809                                        FORMAT(A1,'       ---')
170         CONTINUE
180         CONTINUE

812         FORMAT('0')
            IF( P_TYPE .EQ. 'R' ) THEN
                        WRITE(8,812)
                        WRITE(8,813) CNTRL
813                     FORMAT(A1,2X,'MAX  ')
                        DO 190 I=1,12
                        II=I-2
                        IF( I .EQ. 1 ) II=11
                        IF( I .EQ. 2 ) II=12
                        IF( MAX(II) .EQ. -1.0 ) THEN
                                  WRITE(8,860) CNTRL
860                               FORMAT(A1,'          ')
                                  GO TO 190
                                  END IF
                        IDV=JNINT(MAX(II))
                        WRITE(8,810) CNTRL,IDV
190                     CONTINUE
                        WRITE(8,807)
                        WRITE(8,814) CNTRL
814                     FORMAT(A1,2X,'MIN  ')
                        DO 200 I=1,12
                        II=I-2
                        IF( I .EQ. 1 ) II=11
                        IF( I .EQ. 2 ) II=12
                        IF( MIN(II) .EQ. NOVALUE ) THEN
                                                   WRITE(8,860) CNTRL
                                                   GO TO 200
```

```
                                                END IF
                        IDV=JNINT(MIN(II))
                        WRITE(8,810) CNTRL,IDV
    200                 CONTINUE
                        WRITE(8,807)
                        WRITE(8,815) CNTRL
    815                 FORMAT(A1,1X,'CHNG            ')
                        DO 210 I=2,12
                        II=I-2
                        IF( I .EQ. 2 ) II=12
                        IDV=JNINT(DELTA_S(II))
                        IF( DELTA_S(II) .EQ. NOVALUE )THEN
                                                WRITE(8,860) CNTRL
                                                GO TO 210
                                                END IF
                        WRITE(8,810) CNTRL,IDV
    210                 CONTINUE
                        GO TO 10
                        END IF


             WRITE(8,812)
             WRITE(8,816) CNTRL
    816      FORMAT(A1,'TOTAL  ')
             DO 220 I=1,12
            II=I-2
            IF( I .EQ. 1 ) II=11
            IF( I .EQ. 2 ) II=12
            IF( CFSD(II) .GE. NOVALUE ) THEN
                                    WRITE(8,860) CNTRL
                                    GO TO 220
                                    END IF
            IDV=JNINT(CFSD(II))
            WRITE(8,810) CNTRL,IDV
    220     CONTINUE
            WRITE(8,807)
            WRITE(8,818) CNTRL
    818      FORMAT(A1,' MEAN  ')
            DO 230 I=1,12
            II=I-2
            IF( I .EQ. 1 ) II=11
            IF( I .EQ. 2 ) II=12
            IF( MEAN(II) .GE. NOVALUE ) THEN
                                    WRITE(8,860) CNTRL
                                    GO TO 230
                                    END IF
            IF( MEAN(II) .LT. 10.0 ) THEN
                                WRITE(8,808) CNTRL,MEAN(II)
                                GO TO 230
                                END IF
            IDV=JNINT(MEAN(II))
            WRITE(8,810) CNTRL,IDV
    230     CONTINUE
             WRITE(8,807)
            WRITE(8,819) CNTRL
    819      FORMAT(A1,'  MAX  ')
            DO 240 I=1,12
```

```
            II=I-2
            IF( I .EQ. 1 ) II=11
            IF( I .EQ. 2 ) II=12
            IF( MAX(II) .EQ. -1.0 ) THEN
                                       WRITE(8,860) CNTRL
                                       GO TO 240
                                       END IF
            IF( MAX(II) .LT. 10.0 ) THEN
                                    WRITE(8,808) CNTRL,MAX(II)
                                    GO TO 240
                                    END IF
            IDV=JNINT(MAX(II))
            WRITE(8,810) CNTRL,IDV
    240      CONTINUE
            WRITE(8,807)
            WRITE(8,820) CNTRL
    820      FORMAT(A1,'  MIN  ')
            DO 250 I=1,12
            II=I-2
            IF( I .EQ. 1 ) II=11
            IF( I .EQ. 2 ) II=12
            IF( MIN(II) .GE. NOVALUE ) THEN
                                        WRITE(8,860) CNTRL
                                        GO TO 250
                                        END IF
            IF( MIN(II) .LT. 10.0 ) THEN
                                    WRITE(8,808) CNTRL,MIN(II)
                                    GO TO 250
                                    END IF
            IDV=JNINT(MIN(II))
            WRITE(8,810) CNTRL,IDV
    250     CONTINUE

    C        DO 260 I=1,12
    C        IFT(I)=JNINT(ACFT(I)/100.)
    C        IFT(I)=IFT(I)*100
    C260     CONTINUE

             WRITE(8,807)
             WRITE(8,821) CNTRL
    821      FORMAT(A1,'AC-FT  ')
             DO 270 I=1,12
            II=I-2
            IF( I .EQ. 1 ) II=11
            IF( I .EQ. 2 ) II=12
            IF( ACFT(II) .GE. 19834900. ) THEN
                                          WRITE(8,860) CNTRL
                                          GO TO 270
                                          END IF
    C        IF( ACFT(II) .LE. 999. ) THEN
    C                                  IFT(II)=JNINT(ACFT(II))
    C                                 WRITE(8,810) CNTRL,IFT(II)
    C                                 GO TO 270
    C                                 END IF
            IFT(II)=JNINT(ACFT(II))
            WRITE(8,810) CNTRL,IFT(II)
    270      CONTINUE
```

```
0400
0401            YT=0.0
0402            DO 280 I=1,12
0403            IF( CFSD(I) .EQ. NOVALUE ) CFSD(I)=0.0
0404    280      YT=YT + CFSD(I)
0405            IYT=JNINT(YT)
0406            IYM=JNINT(YT/(365+LEAP))
0407    C        NT=JNINT(YT*1.9835/100. )
0408    C        IYA=NT*100
0409    C        IF( IYA .LT. 9999 ) IYA=1.9835*YT
0410            IYA=1.9835*YT
0411             WRITE(8,807)
0412            WRITE(8,831) NP1,IYT,IYM,IYA
0413    831      FORMAT(11X,'IRRIGATION YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
0414         *         I6,4X,'AC-FT',I8)
0415
0416            GO TO 10       ! RETURN TO START ANOTHER STATION
0417    450     REWIND 7
0418            CLOSE (UNIT=3)
0419            PRINT *,'  '
0420            PRINT *,'  '
0421            PRINT *,'  REQUESTED OF STATIONS COMPLETE - RETURN TO START'
0422            PRINT *,'  '
0423            PRINT *,'  '
0424            GO TO 500
0425    470     PRINT *,'                                    '
0426            PRINT *,'  OUTPUT IS  THE LATEST VERSION OF BOIHSTLST.OUT'
0427            STOP
0428            END
```

PROGRAM SECTIONS

```
    Name                                    Bytes   Attributes

  0 $CODE                                   4843   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                  1969   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                  2636   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG

    Total Space Allocated                   9448
```

ENTRY POINTS

```
    Address  Type  Name                                  References

  0-00000000       BOIHSTLST$MAIN
```

VARIABLES

```
    Address  Type  Name                          Attributes  References

      **      CHAR BLANK                                      13        19=
      **      L*1  CNTRL                                       9        17=       238       241       246       251
                                                                     262       269       274       277       284
                                                                     291       298       301       308       315
                                                                     322       329       333       337       340
                                                                     351       355       358       365       369
                                                                     382       389       398
      **      R*4  D                                         169=      170=      171=      181       194       200
      **      R*4  DZ                                        182=      184=      187
      **      I*4  D_ADJ                                      10       173=      176=      185(2)=   194       200

      **      I*4  I                                          89(2)=   103=      105       166=      167(2)    168=
                                                                     170       171(3)    172(2)    174       175
                                                                     179(2)    180(2)    182       183       187(2)
                                                                     191(4)    192(4)    195       196       197
                                                                     201(2)    233=      234       235       236
                                                                     265       266       267       279=      280
                                                                     282       293=      294       295       310=
                                                                     312       313       324=      325       326
                                                                     342=      343       344       345       360=
                                                                     362       363       384=      385       386
                                                                     402=      403(2)    404
  2-00000810  I*4  IC                                          89(2)=
  2-00000804  I*4  ID                                          65=      127
  2-00000808  I*4  ID1                                         68=      127
      **      I*4  IDATE                                     126=      127(2)

      **      I*4  IDV                                       245=      246       273=      274       287=      288
                                                                     296=      301       318=      319       336=
                                                                     354=      355       372=      373
      **      I*4  II                                        234=      235=      236=      237       238       240
                                                                     244       245       250       251       265=
                                                                     267=      268       273       280=      281=
                                                                     283       287       294=      295=      296
                                                                     311=      312=      313=      314       318
```

```
                                                                                 326=      327=      328       332       333
                                                                                 343=      344=      345=      346       350
                                                                                 354       361=      362=      363=      364
                                                                                 369       372       385=      386=      387=
                                                                                 397(2)    398
  2-000007F8  I*4  IND                                   44=      46        70        86        93        94
                                                                                 95        96        97        206(2)
  2-0000080C  I*4  ISTA                                  77=      79        80
      **      I*4  IYA                                  410=     412

      **      I*4  IYM                                  406=     412
      **      I*4  IYT                                  405=     412
      **      I*4  J                                    104=     105       181=      182       183       188
                                                                                 191(2)    192(2)    228=      229(5)    238
                                                                                 244       245       250       251
      **      I*4  KY                                    56=      64=       65        66=       67=       68
      **      I*4  LD                                   130=     134(2)=   135       136(2)=   137

      **      I*4  LEAP                                 151=     152=      155       156       170       406
      **      I*4  MON                                  131=     132(2)=   133       134       136       137
  2-00000778  CHAR NAME                                  12       98=      208
  2-00000814  I*4  ND                                   110=     126       130
      **      R*4  NOVALUE                               11       18=      105       164       167       174
                                                                                 179       183       188       195       196
                                                                                 240       283       297       314       328
                                                                                 403

  2-000007FC  I*4  NP                                    57=      62(2)=    64        65       211       216
  2-00000800  I*4  NP1                                   57=      63(2)=    67        68       152       211
                                                                                 216       412
  2-000007AB  CHAR P_TYPE                                12      145=      153       210       260
      **      I*4  P_YEAR                                10      144=
  2-000007AC  CHAR RUNDATE                               12      203A      204

  2-000007F0  I*4  STATION_NUM                           10      108       110=      112       120       121
                                                                                 122       143
  2-000007EC  I*4  STA_ID                                10       98=      108       112       113       114
                                                                                 120       121       122(2)
  2-000007E8  I*4  TD                                    10      113       122       143=      208
      **      CHAR TITLE                                 12
  2-000007B5  CHAR TTL                                   13       77=       80

  2-000007AA  CHAR TYPE                                  12      110=      145
  2-00000818  R*4  VALUE                                110=     137
  2-000007F4  I*4  YEAR                                  10      110=      126       129       144
      **      R*4  YT                                   401=     404(2)=   405       406       410
      **      CHAR ZLCH                                  12
```

ARRAYS

```
    Address  Type  Name          Attributes        Bytes  Dimensions     References

  2-000006F0  R*4  ACFT                              48  (12)              15       201=      388       397
  2-00000690  R*4  CFSD                              48  (12)              14       172=      175=      187(2)=   195=
                                                                                                 200       201       314       318
```

							404				
2-000006C0	R*4	DELTA_S	48	(12)	15	154=	155=	156=	157=		
							158=	159=	160=	161=	
							163=	164=	165=	167(	
							296	297			
2-000000C0	R*4	DV	1488	(12, 31)	14	105=	137=	154(2)	155(2)		
							156(2)	157(2)	158(2)	159(	
							161(2)	162(2)	163(2)	165(	
							174	178	179	180	
							183	188	191(2)	192(	
							244	245	250	251	
2-00000750	L*1	FILES	40	(40)	8	89=	91A				
2-00000720	I*4	IFT	48	(12)	15	397=	398				
2-00000090	I*4	JD	48	(12)	14	16D	128=	129=	134		
							136				
2-00000030	R*4	MAX	48	(12)	11	14	178=	179=	191(2)=		
							268	273	346	350	
							354				
2-00000000	R*4	MEAN	48	(12)	11	15	197=	200=	328		
							332	333	336		
2-00000060	R*4	MIN	48	(12)	11	14	180=	192(2)=	283		
							287	364	368	369	

LABELS

Address	Label	References					
0-00000418	1	72#	82				
0-00000560	8	86	93#				
0-000004E8	9	79	83#				
0-000005BF	10	84	92	98#	116	303	416
**	20	103	104	105#			
0-0000067F	25	110#	121	127	146		
0-00000713	30	118	122#				
0-0000061C	60	108	120	126#			
0-00000654	62	132#	135				
0-00000676	68	133	142#				
0-00000726	100	110	122	151#			
**	119	166	167#				
0-000007DF	120	153	168#				
0-00000880	129	189	192#				
**	130	181	193#				
0-000008C1	133	198	201#				
**	135	168	202#				
0-000009BF	160	214	222#				
0-00000B2C	170	233	242	248	253	256#	
**	180	228	257#				
0-00000BEF	190	264	271	275#			
0-00000C9F	200	279	285	289#			
0-00000D45	210	293	299	302#			

```
  0-00000DFB  220                    310      316      320#
  0-00000EE3  230                    324      330      334      338#

  0-00000FCF  240                    342      348      352      356#
  0-000010BB  250                    360      366      370      374#
  0-00001175  270                    384      390      399#
      **      280                    402      404#
  1-00000579  301'                    98       99#

  0-00001208  450                     98      417#
  0-000012AC  470                     46      425#
  0-00000020  500                     22#     424
  1-0000058A  601'                   114      115#
  1-0000057E  701'                   110      111#

  1-000005AE  800'                   206      207#
  1-000005A5  801'                   204      205#
  1-000005C4  802'                   208      209#
  1-000005CF  803'                   211      212#
  1-00000614  804'                   216      217#

  1-00000666  805'                   219      220#
  1-00000676  806'                   222      223#
  1-000006D4  807'                   226      227#     231      237      276      290
                                                 321      339      357      381      411
  1-000006E8  808'                   251      252#     333      351      369
  1-000006EE  809'                   241      255#

  1-000006E3  810'                   246      247#     274      288      301      319
                                                 337      355      373      398
  1-000006D8  811'                   238      239#
  1-000006FD  812'                   259#     261      307
  1-00000701  813'                   262      263#
  1-0000071C  814'                   277      278#

  1-00000728  815'                   291      292#
  1-0000073F  816'                   308      309#
  1-0000074B  818'                   322      323#
  1-00000757  819'                   340      341#
  1-00000763  820'                   358      359#

  1-0000076F  821'                   382      383#
  1-0000077B  831'                   412      413#
  1-0000070D  860'                   269      270#     284      298      315      329
                                                 347      365      389
  1-00000539  900'                    57       58#
  1-00000525  901'                    54       55#

  1-00000522  902'                    44       45#
  1-00000540  903'                    75       76#
  1-00000509  904'                    42       43#
  1-00000550  905'                    77       78#
  1-00000555  906'                    80       81#

  1-0000055A  907'                    87       88#
  1-00000573  908'                    89       90#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	91
	FOR$CLOSE	418
	FOR$DATE_T_DS	203
	FOR$OPEN	20 21 71 93 94 95 96 97

0001

```
+-----------------------------------------+
|          KEY TO REFERENCE FLAGS         |
|   =   - Value Modified                  |
|   #   - Defining Reference              |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization             |
|  (n)  - Number of occurrences on line   |
+-----------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOIHSTLST

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
Run Time:           25.72 seconds
Elapsed Time:       110.69 seconds
Page Faults:        2557
Dynamic Memory:     803 pages
```

BODLYFLW.LIS

BOIDLYFLW.LIS

```
C
C       **** PROGRAM TO ENTER RIVER DATA FROM PARTIAL RECORDS AND
C       PERFORM STRAIGHT LINE INTERPOLATION BETWEEN KNOWN POINTS.
C       OUTPUTS DATA IN FORMAT ACCEPTABLE TO WATER RIGHT ACCOUNTING
C       PROGRAMS.   RHL - JUNE 1981.
C
C       **** MODIFIED TO ENTER BOISE RIVER WATER RIGHT ACCOUNTING
C       DATA.  RJS - DECEMBER 1986.
C
C
      REAL*8 XFLOWA,XFLOWB,DIFF,XD,DAYFL
      INTEGER*4 NDAYA,NDAYB,NNDAY,JDAY,KDAY
      DIMENSION JDAY(12),TITLE(11),KDAY(12)
      DATA JDAY/0,31,59,90,120,151,181,212,243,273,304,334/
      DATA KDAY/31,28,31,30,31,30,31,31,30,31,30,31/
      DATA BLANK/' '/
C
C***** ASSIGN INPUT-OUTPUT UNITS.
C
      OPEN(UNIT=1,NAME='BOIDLYFLW.OUT',TYPE='NEW',
     1CARRIAGECONTROL='LIST')
      CALL ASSIGN(2,'BOIWRA.DTL')
      CALL ASSIGN(3,'BOIWRA.DRT')
      CALL ASSIGN(4,'BOIFLO.TTL')
      CALL ASSIGN(5,'BOIRES.TTL')
      CALL ASSIGN(6,'BOISTR.TTL')
C
C***** READ DATA USING TITLE PROMPTS OR HAND ENTER STATION NUMBER.
C
      IN=1
      NEWYR=0
      MTHB=0
      TYPE *, ' ************************************************'
      TYPE *, ' '
      TYPE *, ' BOISE RIVER DATA ENTRY AND INTERPOLATION PROGRAM. '
      TYPE *, ' STRAIGHT LINE INTERPOLATION BETWEEN KNOWN POINTS.'
      TYPE *, ' '
      TYPE *, ' ************************************************'
      TYPE *, ' '
   16 CONTINUE
      TYPE *, ' '
      TYPE *, ' TYPE "D" TO ENTER A GROUP OF DIVERSIONS, '
      TYPE *, '      "F" TO ENTER A GROUP OF RIVER FLOWS, '
      TYPE *, '      "W" TO ENTER A GROUP OF DRAINS, '
      TYPE *, '      "R" TO ENTER A GROUP OF RESERVOIRS,'
      TYPE *, '      "I" TO ENTER INSTREAM FLOW RELEASES,'
      TYPE *, '      "S" TO ENTER ONE STATION, OR   '
      TYPE *, '      "Q" TO QUIT.  '
      TYPE *, ' '
      ACCEPT 102, IMODE
      IF(IMODE.EQ.'Q') GO TO 500
      TYPE *, ' '
      TYPE *, ' ENTER TWO DIGIT CALENDAR YEAR IN WHICH DATA BEGINS. '
      TYPE *, ' EXAMPLE : 90, 91, ETC.'
      ACCEPT 7, IYR
    7 FORMAT(I2)
      JYR=IYR
```

```
      IF(IMODE.EQ.'D') GO TO 12
      IF(IMODE.EQ.'F') GO TO 13
      IF(IMODE.EQ.'W') GO TO 2
      IF(IMODE.EQ.'R') GO TO 14
      IF(IMODE.EQ.'S') GO TO 5
      IF(IMODE.EQ.'I') GO TO 4
   12 S='D'
      IN=2
      GO TO 15
    2 S='F'
      IN=3
      GO TO 15
   13 S='F'
      IN=4
      GO TO 15
   14 S='R'
      IN=5
      GO TO 15
    4 S='D'
      IN=6
   15 REWIND IN
      S3=S
   29 S=S3
      READ(IN,1,END=16) ID,S2,(TITLE(I),I=1,11)
    1 FORMAT(I8,A1,11A4)
      IF(S2.NE.BLANK) S=S2
      TYPE 3, (TITLE(I),I=1,11)
    3 FORMAT(' HIT "RETURN" TO ENTER DATA FOR '//5X,11A4//
     1 ' OTHERWISE ENTER "N" TO GO TO NEXT STATION, OR   '/
     2 '           ENTER "E" TO EXIT THIS GROUP ',$)
      ACCEPT 102, ILOGIC
      IF(ILOGIC.EQ.'N') GO TO 29
      IF(ILOGIC.EQ.'E') GO TO 16
      GO TO 17
C
C***** IF MODE IS 'S' , ENTER STATION ID NUMBERS INDIVIDUALLY
C
   22 IYR=JYR
      IF(IMODE.NE.'S') GO TO 29
    5 TYPE *, ' ENTER 9 DIGIT STATION NO. - EX. 13059800F'
      TYPE *, ' SUFFIX: D=DIV, F=FLOW OR DRAIN, R=RESERVR.'
      TYPE *, 'NNNNNNNNA    '
      ACCEPT 6, ID,S
    6 FORMAT(I8,A1)
   17 CONTINUE
   10 TYPE *, ' '
      TYPE *, ' ENTER FIRST MONTH, DAY, AND FLOW.'
      TYPE *, ' EXAMPLE: MAY 9, 222.2CFS = 0409222.2'
      TYPE *, ' TO BEGIN NEXT STATION, ENTER 80 FOR MONTH.'
      TYPE *, ' TO EXIT OR BEGIN NEW GROUP ENTER 90.'
      TYPE *, ' '
      TYPE *, 'MMDDFLOW.00    '
      ACCEPT 11, MTHA,IDAYA,XFLOWA
   11 FORMAT(I2,I2,F7.0)
      IF(MTHA.EQ.80) GO TO 22
      IF(MTHA.EQ.90) GO TO 16
      TYPE 101, MTHA,IDAYA,XFLOWA
```

```
  101 FORMAT(3X,'DATA ACCEPTED: MONTH=',I2,/,20X,'DAY=',I2,/,19X,'DATA
     1=',F8.1,/,'   **** IF DATA IS INCORRECT ENTER "N".',$)
      ACCEPT 102, ILOGIC
  102 FORMAT(1A1)
      IF(ILOGIC.EQ.'N') GO TO 10
      MTHC=MTHA
      IDAYC=IDAYA
C
C***** CHECK FOR LEAP YEAR CORRECTION, THEN COMPUTE DAY OF *****
C***** THE YEAR FOR FIRST DATA POINT                       *****
C
    9 ILEAP=0
      ADIFF=MOD(IYR,4)
      IF(ADIFF.EQ.0) ILEAP=1
      IF(NEWYR.EQ.1) GO TO 110
      NDAYA=JDAY(MTHA)+IDAYA
      IF(MTHA.GT.2) NDAYA=NDAYA+ILEAP
      IF(S.NE.'R') GO TO 23
      WRITE(1,25) ID,S,IYR,NDAYA,JIDNNT(XFLOWA)
   25 FORMAT(I8,A1,'19',I2,I3,I7)
      GO TO 24
   23 WRITE(1,21) ID,S,IYR,NDAYA,XFLOWA
   21 FORMAT(I8,A1,'19',I2,I3,F7.1)
C
C***** IF DATA BEING ENTERED IS 13203600 - BELOW DIVERSION DAM,
C***** COMPUTE RELEASES FOR FLOW MAINTENANCE FROM FISH AND GAME
C***** STORAGE AND USBR STORAGE IN LUCKY PEAK.
C
   24 IF(ID.NE.13203600) GO TO 31
      FGF=80.
      IF(FGF.GT.XFLOWA) FGF=XFLOWA
      USBRF=XFLOWA-FGF
      IF(USBRF.GT.170.) USBRF=170.
      IF(XFLOWA.GT.325.) FGF=0.0
      IF(XFLOWA.GT.325.) USBRF=0.0
      IF(FGF.GT.0.0.OR.USBRF.GT.0.0) GO TO 34
      NA=NDAYA-ILEAP
      IF(NA.LT.91.OR.NA.GT.304) GO TO 31
   34 WRITE(1,32) IYR,NDAYA,FGF
   32 FORMAT('13201990D19',I2,I3,F7.1)
      WRITE(1,33) IYR,NDAYA,USBRF
   33 FORMAT('13201991D19',I2,I3,F7.1)
   31 NCTN=1
      TYPE 150, NCTN
  110 NEWYR=0
C
C***** ACCEPT SECOND DAY AND FLOW. *****
C
      GO TO 27
  112 TYPE *, ' **** SECOND DATE IS SAME AS PREVIOUS - NN=0'
   27 TYPE *, ' '
      TYPE *, ' ENTER SECOND OR NEXT DAY AND FLOW.'
      TYPE *, ' TO SELECT NEW FIRST DATE, ENTER 70 FOR MONTH.'
```

```
      TYPE *, ' TO BEGIN NEXT STATION, ENTER 80 FOR MONTH.'
      TYPE *, ' TO EXIT OR BEGIN NEW GROUP ENTER 90.'
      TYPE *, ' '
      TYPE *, 'MMDDFLOW.00 '
      ACCEPT 11, MTHB,IDAYB,XFLOWB
      IF(MTHB.LE.0) GO TO 30
      IF(MTHB.LE.12) GO TO 19
      IF(MTHB.EQ.80) GO TO 22
      IF(MTHB.EQ.90) GO TO 16
      IF(MTHB.EQ.70) GO TO 17
C
C***** GENERATE DATE AS ONE DAY LATER IF NO DATE ENTERED. *****
C
   30 MTHB=MTHC
      IDAYB=IDAYC+1
      KD=KDAY(MTHB)
      IF(ILEAP.GT.0.AND.MTHB.EQ.2) KD=29
      IF(IDAYB.LE.KD) GO TO 19
      MTHB=MTHB+1
      IDAYB=1
      IF(MTHB.GT.12) MTHB=1
C
C***** WRITE DATA TO SCREEN FOR VERIFICATION. *****
C
   19 TYPE 101, MTHB,IDAYB,XFLOWB
      ACCEPT 102, ILOGIC
      IF(ILOGIC.EQ.'N') GO TO 27
C
C***** DETERMINE DAY OF YEAR AND CORRECT FOR LEAP YEAR. *****
C
      NDAYB=JDAY(MTHB)+IDAYB
      IF(MTHB.GT.2) NDAYB=NDAYB+ILEAP
      MTHC=MTHB
      IDAYC=IDAYB
C
C***** PERFORM INTERPOLATION. *****
C
      NN=NDAYB-NDAYA
      IF(NDAYA.GT.NDAYB) NN=365-NDAYA+ILEAP+NDAYB
      IF(NN.LT.1) GO TO 112
      DIFF=XFLOWB-XFLOWA
      XD=DIFF/NN
   18 DO 28 J=1,NN
      DAYFL=XFLOWA+FLOAT(J)*XD
      NNDAY=NDAYA+J
C
C***** ERROR CHECK FOR RUN AWAY DATA - MORE THAN 35 DAYS. ******
C
      IF(J.EQ.36) GO TO 35
      GO TO 39
   35 TYPE *, ' ==> CAUTION I HAVE CREATED >35 DAYS OF DATA. <==='
      TYPE *, ' IS THIS OK????  ANSWER Y/N.'
      ACCEPT 36, JLOGIC
   36 FORMAT(1A1)
      IF(JLOGIC.EQ.'N') GO TO 500
C
C***** CORRECT FOR END OF YEAR. [ DEC==>JAN ] *****
```

```
C
   39 IF(NNDAY.GT.(365+ILEAP)) GO TO 40
      GO TO 20
   40 NNDAY=NNDAY-365-ILEAP
      IF(NEWYR.EQ.1) GO TO 20
      IYR=IYR+1
      NEWYR=1
      TYPE *, ' ++++ NOTICE:  ENTERING NEW YEAR. ++++'
   20 IF(S.NE.'R') GO TO 26
      WRITE(1,25) ID,S,IYR,NNDAY,JIDNNT(DAYFL)
      GO TO 37
   26 WRITE(1,21) ID,S,IYR,NNDAY,DAYFL
C
C***** IF DATA BEING ENTERED IS 13203600 - BELOW DIVERSION DAM,
C***** COMPUTE RELEASES FOR FLOW MAINTENANCE FROM FISH AND GAME
C***** STORAGE AND USBR STORAGE IN LUCKY PEAK.
C
   37 IF(ID.NE.13203600) GO TO 28
      FGF=70.
      IF(FGF.GT.DAYFL) FGF=DAYFL
      USBRF=DAYFL-FGF
      IF(USBRF.GT.150.) USBRF=150.
      IF(DAYFL.GT.325.) FGF=0.0
      IF(DAYFL.GT.325.) USBRF=0.0
      IF(FGF.GT.0.0.OR.USBRF.GT.0.0) GO TO 38
      NA=NNDAY-ILEAP
      IF(NA.LT.91.OR.NA.GT.304) GO TO 28
   38 WRITE(1,32) IYR,NNDAY,FGF
      WRITE(1,33) IYR,NNDAY,USBRF
   28 CONTINUE
C
C***** WRITE OUTPUT INFORMATION TO SCREEN. *****
C
      NCTN=NN
      TYPE 150, NCTN
  150 FORMAT(1X,I4,' LINES ENTERED IN OUTPUT FILE.')
C
C***** RENAME SECOND DAY AND FLOW TO FIRST. *****
C
      NDAYA=NDAYB
      XFLOWA=XFLOWB
C
C***** RETURN TO ACCEPT NEXT DAY AND FLOW. *****
C
      IF(NEWYR.EQ.1) GO TO 9
      GO TO 27
  500 STOP 'OUTPUT IN "BOIDLYFLW.OUT"'
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	3331	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	1497	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	688	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	5516	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIDLYFLW$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
**	R*4	ADIFF		127=	128				
2-000000A4	R*4	BLANK		16D	83				
**	R*8	DAYFL		11	211=	234	236	244(2)	245
				247	248				
**	R*8	DIFF		11	208=	209			
**	R*4	FGF		144=	145(2)=	146	148=	150	153
				243=	244(2)=	245	247=	249	252
2-000000C4	I*4	I		81(2)=	84(2)=				
2-000000BC	I*4	ID		81=	100=	133	136	143	234
				236	242				
2-000000D0	I*4	IDAYA		110=	114	121	130		
2-000000D4	I*4	IDAYB		172=	182=	185	187=	192	198
				201					
**	I*4	IDAYC		121=	182	201=			
**	I*4	ILEAP		126=	128=	131	151	184	199
				206	226	228	250		
2-000000C8	I*4	ILOGIC		88=	89	90	117=	119	193=
				194					
2-000000AC	I*4	IMODE		50=	51	58	59	60	61
				62	63	96			
**	I*4	IN		30=	65=	68=	71=	74=	77=
				78	81				
2-000000B0	I*4	IYR		55=	57	95=	127	133	136
				153	155	230(2)=	234	236	252
				253					
**	I*4	J		210=	211	212	216		
2-000000DC	I*4	JLOGIC		220=	222				
**	I*4	JYR		57=	95				
**	I*4	KD		183=	184=	185			
2-000000CC	I*4	MTHA		110=	112	113	114	120	130
				131					

2-000000A8	I*4	MTHB	32=	172=	173	174	175	176
			177	181=	183	184	186(2)=	188(2)=
			192	198	199	200		
**	I*4	MTHC	120=	181	200=			
**	I*4	NA	151=	152(2)	250=	251(2)		
**	I*4	NCTN	157=	158	258=	259		
2-000000A0	I*4	NDAYA	12	130=	131(2)=	133	136	151
			153	155	205	206(2)	212	264=
**	I*4	NDAYB	12	198=	199(2)=	205	206(2)	264
**	I*4	NEWYR	31=	129	159=	229	231=	269
**	I*4	NN	205=	206=	207	209	210	258
**	I*4	NNDAY	12	212=	226	228(2)=	234	236
			250	252	253			
2-000000B4	R*4	S	64=	67=	70=	73=	76=	79
			80=	83=	100=	132	133	136
			233	234	236			
2-000000C0	R*4	S2	81=	83(2)				
2-000000B8	R*4	S3	79=	80				
**	R*4	USBRF	146=	147(2)=	149=	150	155	245=
			246(2)=	248=	249	253		
**	R*8	XD	11	209=	211			
2-00000090	R*8	XFLOWA	11	110=	114	133	136	145(2)
			146	148	149	208	211	265=
2-00000098	R*8	XFLOWB	11	172=	192	208	265	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000000	I*4	JDAY		48	(12)	12	13	14D	130	198
2-00000030	I*4	KDAY		48	(12)	12	13	15D	183	
2-00000060	R*4	TITLE		44	(11)	13	81=	84		

LABELS

Address	Label	References		
1-00000003	1'	81	82#	
0-00000334	2	60	67#	
1-0000000C	3'	84	85#	
0-00000314	4	63	76#	
0-000008BA	5	62	97#	
1-00000098	6'	100	101#	
1-00000000	7'	55	56#	
0-000005C0	9	126#	269	
0-00000434	10	103#	119	
1-0000009D	11'	110	111#	172
0-00000354	12	58	64#	
0-00000344	13	59	70#	
0-00000324	14	61	73#	

0-0000035F	15	66	69	72	75	78#
0-00000124	16	40#	81	90	113	176
0-00000434	17	91	102#	177		
**	18	210#				
0-00000999	19	174	185	192#		
0-00000B46	20	227	229	233#		
1-00000116	21'	136	137#	236		
0-000008A9	22	95#	112	175		
0-00000648	23	132	136#			
0-0000068F	24	135	143#			
1-00000107	25'	133	134#	234		
0-00000B98	26	233	236#			
0-0000078C	27	163	165#	194	270	
0-00000CBD	28	210	242	251	254#	
0-00000370	29	80#	89	96		
0-0000095C	30	173	181#			
0-00000767	31	143	152	157#		
1-00000126	32'	153	154#	252		
1-0000013B	33'	155	156#	253		
0-00000705	34	150	153#			
0-00000A9C	35	216	218#			
1-00000150	36'	220	221#			
0-00000BDE	37	235	242#			
0-00000C5B	38	249	252#			
0-00000B02	39	217	226#			
0-00000B14	40	226	228#			
1-000000A5	101'	114	115#	192		
1-00000104	102'	50	88	117	118#	193
0-00000787	110	129	159#			
0-00000A36	112	164#	207			
1-00000153	150'	158	259	260#		
0-00000CF8	500	51	222	271#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References				
	ASSIGN	22	23	24	25	26
R*4	FLOAT	211				
I*4	JIDNNT	133	234			
I*4	MOD	127				

```
+-------------------------------------------+
|            KEY TO REFERENCE FLAGS         |
|    =   - Value Modified                   |
|    #   - Defining Reference               |
|    A   - Actual Argument, possibly modified |
|    D   - Data Initialization              |
|   (n)  - Number of occurrences on line    |
+-------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOIDLYFLW

/ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
/BLAS=(INLINE,MAPPED)
/CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
/DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
/DIRECTIVES=(DEPENDENCE)
/MATH_LIBRARY=(ACCURATE,NOV5)
/PARALLEL=(NOAUTOMATIC,NOMANUAL)
/SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
/STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
/WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
           NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
/CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
/F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
/NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
/NOANALYSIS_DATA
/NODIAGNOSTICS
/LIST=HYDRO:[BSUTTER.BOISE]BOIDLYFLW.LIS;1
/OBJECT=HYDRO:[BSUTTER.BOISE]BOIDLYFLW.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

```
Run Time:           0.71 seconds
Elapsed Time:       1.45 seconds
Page Faults:        1530
Dynamic Memory:     1184 pages
```

BOICOR.LIS

BOICOR.LIS

```
C
C        PROGRAM TO ELIMINATE DATA IN DATA ENTRY FILES THAT HAS BEEN
C       CORRECTED BY A SUBSEQUENT ENTRY.  SECOND ENTRY IS RETAINED
C       WHILE FIRST ENTRYIS DISCARDED.  RJS - NOVEMBER 1986.
C
      DIMENSION CODE(8000),REM(8000)
      CHARACTER*16 CODE
      CHARACTER*7 REM
      CALL ASSIGN (1,'BOIDLYFLW.OUT')
      OPEN (UNIT=2,NAME='BOIDLY.OUT',TYPE='NEW',CARRIAGECONTROL='LIST')
      IW=0
      DO 1 J=1,5000
    1 READ(1,2,END=10) CODE(J),REM(J)
    2 FORMAT(A16,A7)
   10 JEND=J-2
      K=1
      IDROP=0
      DO 4 K=1,JEND
      DO 3 I=K,JEND
      IF(CODE(I+1).NE.CODE(K)) GO TO 3
      IDROP=IDROP+1
      GO TO 4
    3 CONTINUE
      WRITE(2,2) CODE(K),REM(K)
      IW=IW+1
      IF(JMOD(IW,300).EQ.0) TYPE 7,IW
    7 FORMAT(5X,I4,' RECORDS WRITTEN')
    4 CONTINUE
      JEND=JEND+1
      WRITE(2,2) CODE(JEND),REM(JEND)
      IW=IW+1
      TYPE 7,IW
      TYPE 8,IDROP
    8 FORMAT(5X,I4,' RECORDS DROPPED')
      IF(JEND.LT.8000) GO TO 6
      TYPE *, '   WARNING! FILE > 8000 RECORDS - TOO BIG TO CHECK'
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	505	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	130	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	184088	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	184723	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICOR$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
**	I*4	I		19=	20				
**	I*4	IDROP		17=	21(2)=	33			
**	I*4	IW		11=	25(2)=	26(2)	31(2)=	32	
**	I*4	J		12=	13(2)	15			
**	I*4	JEND		15=	18	19	29(2)=	30(2)	35
**	I*4	K		16=	18=	19	20	24(2)	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000000	CHAR	CODE		128000	(8000)	6	7	13=	20(2)	24
								30		
2-0001F400	CHAR	REM		56000	(8000)	6	8	13=	24	30

LABELS

Address	Label	References			
**	1	12	13#		
1-0000004F	2'	13	14#	24	30
0-000000B8	3	19	20	23#	
0-00000138	4	18	22	28#	
0-000001F2	6	35	37#		
1-00000054	7'	26	27#	32	
1-0000006B	8'	33	34#		
0-00000079	10	13	15#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	9
	FOR$OPEN	10

```
+-----------------------------------------+
|          KEY TO REFERENCE FLAGS         |
|   =   - Value Modified                  |
|   #   - Defining Reference              |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization             |
|  (n)  - Number of occurrences on line   |
+-----------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICOR

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
Run Time:           3.47 seconds
Elapsed Time:       10.53 seconds
Page Faults:        370
Dynamic Memory:     354 pages
```

ROTATE UP 1.5

BOIALCUP.LIS

```
C
C       **** PROGRAM TO ENTER RESULTS FROM WATER RIGHT ACCOUNTING TO
C       PERMANENT ALLOCATIONS FILE.  IF DATA ALREADY EXISTS FOR STATION
C       AND DAY BEING ENTERED, OLD DATA WILL BE REPLACED WITH NEW
C       DATA. BOB SUTTER - OCTOBER 1986.
C
C       MODIFIED FOR BOISE RIVER - DECEMBER 24, 1992 - RJS
C
      REAL*8 VAOLD(5),VANEW(5)
      INTEGER*4 ISTA,JSTA,IDATE,JDATE
C
      CALL ASSIGN(1,'INPUT')
      CALL ASSIGN(2,'OLDALC')
      OPEN(UNIT=3,NAME='NEWALC',TYPE='NEW',
     1CARRIAGECONTROL='LIST')
C
      JSKIP=0
      IREP=0
      INEW=0
      ISAM=0
      IN=0
      IR=0
      KW=1
C
C***** READ ONE RECORD FROM FROM INPUT FILE AND ONE RECORD
C***** FROM OLD ALLOCATIONS FILE.  SKIP READING DATA FROM OLD
C***** ALLOCATIONS FILE IF LAST INPUT DATA HAS BEEN MERGED WITH
C***** DATA FROM OLD ALLOCATIONS FILE.
C
    2 READ(1,1,END=10)ISTA,S,IDATE,(VANEW(I),I=1,5)
    1 FORMAT(I8,A1,I7,5F10.0)
    3 IF(JSKIP.EQ.0) READ(2,1,END=20)JSTA,T,JDATE,(VAOLD(I),I=1,5)
      JSKIP=0
C
C***** CHECK RELATIVE VALUE OF STATION NUMBERS OF OLD
C***** AND NEW DATA.
C
      IF(JSTA-ISTA) 4,5,6
C
C***** CHECK RELATIVE VALUE OF DATES OF OLD AND NEW
C***** INPUT DATA.
C
    5 IF(JDATE-IDATE) 4,7,6
C
C***** WRITE DATA FROM INPUT FILE TO NEW ALLOCATIONS FILE TO
C***** EITHER REPLACE OR BE MERGED WITH OLD DATA.
C
    6 JSKIP=1
      INEW=INEW+1
      IN=1
      GO TO 9
    7 IREP=IREP+1
      IR=1
    9 IF(S.EQ.'D') THEN
         KW=0
         DO 12 I=1,4
   12    IF(VANEW(I).GT.0.0) KW=1
```

```
            IF(KW.EQ.1) GO TO 23
            IF(IN.EQ.1) INEW=INEW-1
            IF(IR.EQ.1) IREP=IREP-1
            GO TO 23
            END IF
         IF(S.EQ.'F') THEN
            IVANEW5=VANEW(5)
            WRITE(3,18) ISTA,S,IDATE,(VANEW(I),I=1,4),IVANEW5
   18       FORMAT(I8,A1,I7,4F10.1,I10)
            GO TO 19
            END IF
   23 IF(KW.GT.0) WRITE(3,21) ISTA,S,IDATE,(VANEW(I),I=1,5)
   19 IN=0
      IR=0
      KW=1
      GO TO 2
C
C***** WRITE DATA RECORDS FROM OLD ALLOCATIONS FILE TO NEW
C***** ALLOCATIONS FILE UNTIL DATA FROM INPUT FILE IS TO
C***** REPLACE OLD DATA OR BE MERGED WITH OLD DATA.
C
    4 ISAM=ISAM+1
      IF(S.EQ.'F') THEN
         IVAOLD5=VAOLD(5)
         WRITE(3,18) JSTA,T,JDATE,(VAOLD(I),I=1,4),IVAOLD5
         GO TO 3
         END IF
    8 WRITE(3,21) JSTA,T,JDATE,(VAOLD(I),I=1,5)
   21 FORMAT(I8,A1,I7,5F10.1)
      GO TO 3
C
C***** WRITE REMAINDER OF OLD ALLOCATIONS FILE TO NEW FILE
C***** AFTER DATA INPUT FILE HAS BEEN EXHAUSTED.
C
   10 IF(JSKIP.EQ.1) GO TO 14
   15 READ(2,1,END=100)JSTA,T,JDATE,(VAOLD(I),I=1,5)
   14 ISAM=ISAM+1
      IF(S.EQ.'F') THEN
         IVAOLD5=VAOLD(5)
         WRITE(3,18) JSTA,T,JDATE,(VAOLD(I),I=1,4),IVAOLD5
         GO TO 15
         END IF
   13 WRITE(3,21) JSTA,T,JDATE,(VAOLD(I),I=1,5)
      GO TO 15
C
C***** WRITE REMAINDER OF DATA INPUT FILE TO NEW ALLOCATIONS
C***** FILE AFTER OLD ALLOCATIONS FILE HAS BEEN EXHAUSTED.
C
   22 READ(1,1,END=100)ISTA,S,IDATE,(VANEW(I),I=1,5)
      KW=1
   20 INEW=INEW+1
      IF(S.EQ.'D') THEN
         KW=0
         DO 17 I=1,4
   17    IF(VANEW(I).GT.0.0) KW=1
         IF(KW.EQ.0) INEW=INEW-1
         GO TO 16
```

– THIS ROUTINE HAS NOT BEEN TESTED IF NEW ALLOCATIONS GO BEYOND DATA IN OLD ALLOCATIONS, THIS SHOULD PREVENT RECORDS W/ ZEROES FROM BEING WRITTEN TO NEW ALLOCATIONS FILE

```
00115            END IF
00116         IF(S.EQ.'F') THEN
00117            IVANEW5=VANEW(5)
00118            WRITE(3,18) ISTA,S,IDATE,(VANEW(I),I=1,4),IVANEW5
00119            GO TO 22
00120            END IF
00121       16 IF(KW.GT.0) WRITE(3,21) ISTA,S,IDATE,(VANEW(I),I=1,5)
00122         GO TO 22
00123    C
00124    C***** ALL DATA HAS BEEN ADDED OR TRANSFERRED TO NEW ALLOCATIONS FILE.
00125    C
00126      100 TYPE *, ' BOISE ALLOCATIONS FILE HAS NOW BEEN UPDATED '
00127         ITOT=IREP+ISAM+INEW
00128         TYPE *, ' '
00129         TYPE 11, IREP,INEW,ISAM,ITOT
00130       11 FORMAT(/2X,'UPDATED FILE HAS',I10,3X,'RECORDS REPLACED'/
00131         1 18X,I10,3X,'NEW RECORDS'/18X,I10,3X,'RECORDS TRANSFERRED'/
00132         2 4X,'FOR A TOTAL OF',I10,3X,'ALLOCATIONS FILE RECORDS')
00133         TYPE *, ' '
00134      500 STOP
00135         END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1378	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	254	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	336	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	1968	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIALCUP$MAIN	

VARIABLES

Address	Type	Name	Attributes	References								
2-00000064	I*4	I		30(2)=	32(2)=	56=	57	65(2)=	69(2)=	82(2)=	85(2)=	
				93(2)=	97(2)=	100(2)=	106(2)=	111=	112	118(2)=	121(2)=	
2-00000058	I*4	IDATE		10	30=	43	65	69	106=	118	121	
**	I*4	IN		21=	50=	59	70=					
**	I*4	INEW		19=	49(2)=	59(2)=	108(2)=	113(2)=	127	129		
**	I*4	IR		22=	53=	60	71=					
**	I*4	IREP		18=	52(2)=	60(2)=	127	129				
**	I*4	ISAM		20=	79(2)=	94(2)=	127	129				
2-00000050	I*4	ISTA		10	30=	38	65	69	106=	118	121	

**	I*4	ITOT	127=	129					
**	I*4	IVANEW5	64=	65	117=	118			
**	I*4	IVAOLD5	81=	82	96=	97			
2-0000005C	I*4	JDATE	10	32=	43	82	85	93=	97	100
**	I*4	JSKIP	17=	32	33=	48=	92			
2-00000054	I*4	JSTA	10	32=	38	82	85	93=	97	100
**	I*4	KW	23=	55=	57=	58	69	72=	107=	110=
			112=	113	121					
2-00000060	R*4	S	30=	54	63	65	69	80	95	106=
			109	116	118	121				
2-00000068	R*4	T	32=	82	85	93=	97	100		

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000028	R*8	VANEW		40	(5)	9	30=	57	64	65
						69	106=	112	117	118
						121				
2-00000000	R*8	VAOLD		40	(5)	9	32=	81	82	85
						93=	96	97	100	

LABELS

Address	Label	References				
1-00000000	1'	30	31#	32	93	106
0-00000034	2	30#	73			
0-00000084	3	32#	83	87		
0-0000031C	4	38	43	79#		
**	5	38	43#			
0-00000224	6	38	43	48#		
0-0000021C	7	43	52#			
**	8	85#				
0-0000022C	9	51	54#			
0-000003CC	10	30	92#			
1-00000026	11'	129	130#			
**	12	56	57#			
**	13	100#				
0-000003D4	14	92	94#			
0-00000479	15	93#	98	101		
0-00000168	16	114	121#			
**	17	111	112#			
1-0000000C	18'	65	66#	82	97	118
0-00000310	19	67	70#			
0-000000DC	20	32	108#			
1-0000001A	21'	69	85	86#	100	121
0-000001B0	22	106#	119	122		
0-000002C4	23	58	61	69#		

```
  0-000004C8  100                          93       106       126#
      **      500                         134#


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                              References

        ASSIGN                               12        13


  +-----------------------------------------+
  |           KEY TO REFERENCE FLAGS        |
  |    =   - Value Modified                 |
  |    #   - Defining Reference             |
  |    A   - Actual Argument, possibly modified |
  |    D   - Data Initialization            |
  |   (n)  - Number of occurrences on line  |
  +-----------------------------------------+


COMMAND QUALIFIERS

  FORTRAN/LIST/CROSS_REFERENCE BOIALCUP

  /ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
  /BLAS=(INLINE,MAPPED)
  /CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
  /DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
  /DIRECTIVES=(DEPENDENCE)
  /MATH_LIBRARY=(ACCURATE,NOV5)
  /PARALLEL=(NOAUTOMATIC,NOMANUAL)
  /SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
             NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
  /CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
  /NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=HYDRO:[BSUTTER.BOISE]BOIALCUP.LIS;7
  /OBJECT=HYDRO:[BSUTTER.BOISE]BOIALCUP.OBJ;1


COMPILER:  DEC Fortran V6.3-141

COMPILATION STATISTICS

  Run Time:           0.42 seconds
  Elapsed Time:       1.10 seconds
  Page Faults:        905
  Dynamic Memory:     840 pages
```

```
C     ***************************************************************
C
C             PROGRAM BOISEL - WATER RIGHT ACCOUNTING SELECT
C
C       THIS PROGRAM SELECTS FROM THE HISTORY FILE THE HYDROLOGIC
C       DATA NEEDED FOR THE WATER RIGHT ACCOUNTING OVER A SPECIFIED
C       PERIOD.  THESE DATA ARE BROKEN INTO SEPARATE FILES ACCORDING
C       TO DATA TYPE (DIVERSION, FLOW, RESERVOIR) IN SORTED FORM
C       SUCH THAT THEY CAN BE DIRECTLY READ BY THE ACCOUNTING PROGRAM.
C
C                       BOB SUTTER  JULY 31,1984
C
C     ***************************************************************
C
C
      DIMENSION OM(12),JD(12),IYD(4),A(25),D(4)
      CHARACTER*7 D
      DATA OM/'OCT','NOV','DEC','JAN','FEB','MAR','APR','MAY','JUN',
     !  'JUL','AUG','SEP'/
      DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
      CALL ASSIGN(1,'HISTORY')
      OPEN (UNIT=2,NAME='HISTDIV',TYPE='NEW',CARRIAGECONTROL='LIST')
      OPEN (UNIT=3,NAME='HISTFLO',TYPE='NEW',CARRIAGECONTROL='LIST')
      OPEN (UNIT=4,NAME='HISTRVR',TYPE='NEW',CARRIAGECONTROL='LIST')
      OPEN (UNIT=7,NAME='ALOCDIV',TYPE='NEW',CARRIAGECONTROL='LIST')
      OPEN (UNIT=8,NAME='ALOCRVR',TYPE='NEW',CARRIAGECONTROL='LIST')
      OPEN (UNIT=9,NAME='ALOCMIS',TYPE='NEW',CARRIAGECONTROL='LIST')
      CALL ASSIGN(10,'ALOCTNS')
C
      PRINT 1
    1 FORMAT (' ')
      PRINT *,' ******************************************************'
      PRINT *,' '
      PRINT *,'  THIS PROCEDURE SELECTS THE HYDROLOGIC DATA FROM THE  '
      PRINT *,'  HISTORY REQUIRED TO PERFORM THE WATER RIGHT ACCOUNTING'
      PRINT *,'  OVER A SELECTED PERIOD SPECIFIED BY A BEGINNING AND '
      PRINT *,'  ENDING DATE.  ANSWER FOLLOWING QUESTIONS USING Y FOR '
      PRINT *,'  YES, N FOR NO, AND C IF YOU WISH TO CANCEL ALL.  '
      PRINT *,' '
      PRINT *,' ******************************************************'
      PRINT 1
C
    5 PRINT *,'  ENTER THE MONTH, DAY, AND YEAR FOR THE FIRST DAY YOU'
      PRINT *,'  WISH TO PERFORM THE ACCOUNTING. '
      PRINT *,'  FOR EXAMPLE : 06-12-1989  '
      PRINT 1
      PRINT *,'                ??????????  '
      PRINT 35
   35 FORMAT('+                '$)
      ACCEPT 2, IM1,ID1,IY1
    2 FORMAT(I2,1X,I2,1X,I4)
      PRINT 1
      PRINT 3, IM1,ID1,IY1
    3 FORMAT (1X,' THE FIRST ACCOUNTING DAY IS ',I2,'-',I2,'-',I4/
     !   2X,'IS THIS CORRECT?  ANSWER Y/N/C.  ')
      PRINT 36
   36 FORMAT(28X,' '$)
```

```
      ACCEPT 4,IANSWR
    4 FORMAT (1A1)
      PRINT 1
      IF (IANSWR.EQ.'C') GO TO 10
      IF (IANSWR.EQ.'N') GO TO 5
C
      PRINT *,'  DO YOU WISH TO DO MORE THAN ONE DAY OF WATER RIGHT '
      PRINT *,'  ACCOUNTING?  ANSWER Y/N/C. '
      PRINT 36
      ACCEPT 4,IANSWR
      PRINT 1
      IF (IANSWR.EQ.'C') GO TO 10
      IF (IANSWR.EQ.'N') GO TO 9
C
    7 PRINT *,'  ENTER THE MONTH, DAY, AND YEAR FOR THE FINAL DAY '
      PRINT *,'  YOU WISH TO PERFORM THE ACCOUNTING. '
      PRINT *,'  FOR EXAMPLE : 07-15-1989. '
      PRINT 1
      PRINT *,'                 ??????????? '
      PRINT 35
      ACCEPT 2, IM2,ID2,IY2
      PRINT 1
      PRINT 6, IM2,ID2,IY2
    6 FORMAT (1X,' THE LAST ACCOUNTING DAY WILL BE ',I2,'-',I2,'-',I4,
     !  /1X,' IS THIS CORRECT?  ANSWER Y/N/C. ')
      PRINT 36
      ACCEPT 4,IANSWR
      PRINT 1
      IF (IANSWR.EQ.'C') GO TO 10
      IF (IANSWR.EQ.'N') GO TO 7
      GO TO 8
    9 IM2=IM1
      ID2=ID1
      IY2=IY1
    8 CONTINUE
C
C***** CONVERT BEGINNING AND ENDING DATES TO SEVEN DIGIT DATE YYYYDDD,
C***** WHERE YYYY IS THE YEAR AND DDD IS THE NUMERICAL DAY OF THE YEAR.
C***** ALSO COMPUTE A SIMILAR DATE FOR THE DAY THREE DAYS PRIOR TO THE
C***** FIRST ACCOUNTING DAY SO THAT FOUR DAY AVERAGES CAN BE COMPUTED
C***** FOR CERTAIN REACH GAINS. THEN PRINT RESULTS.
C
      IM=IM1
      ID=ID1
      IY=IY1
      DO 13 K=2,3
      FEB=IY/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      NDY=0
      DO 11 I=1,IM
      IF(I.EQ.IM) GO TO 12
      IF(I.EQ.2.AND.FEB.LT.0.001) JD(2)=29
   11 NDY=NDY+JD(I)
   12 JD(2)=28
      NDY=NDY+ID
      IYD(K)=(IY*1000)+NDY
```

```
      IF(K.EQ.3) GO TO 14
      IM=IM2
      ID=ID2
      IY=IY2
   13 CONTINUE
   14 IYD(1)=IYD(2)-4
      IY=IYD(1)/1000
      IF(IY1.EQ.IY) GO TO 15
      FEB=IY/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      J=635
      IF(FEB.LT.0.001) J=634
      IYD(1)=IYD(1)-J
   15 CONTINUE
      PRINT 16, IYD(1),IYD(3),IYD(2),IYD(3)
   16 FORMAT(3X,'ALL DATA FROM DAY ',I8,' TO DAY ', I8,' IS BEING RETRIE
     !VED FROM'/3X,'THE HISTORY FILE TO RUN DAY ',I8,' THROUGH DAY ',I8)
      PRINT *,'  SO PLEASE HANG ON FOR A SECOND. '
      PRINT 1
C
C***** SELECT ALL DATA FROM HISTORY FILE FROM DATE IYD(1) THROUGH IYD(3)
C***** AND WRITE DATA ON THREE OUTPUT FILES (R'S, F'S, AND {D,X}'S).
C
      IERR=0
   19 READ(1,17,END=18,ERR=24) (A(I),I=1,3),IDATE,(D(I),I=1,4)
   17 FORMAT(2A4,A1,I7,4A7)
      IF(IDATE.LT.IYD(1)) GO TO 19
      IF(IDATE.GT.IYD(3)) GO TO 19
      IF(A(3).EQ.'D') GO TO 21
      IF(A(3).EQ.'F') GO TO 22
      IF(A(3).EQ.'R') GO TO 23
      IF(A(3).EQ.'X') GO TO 21
      GO TO 19
   24 IERR=IERR+1
      PRINT *,'  AN ERROR HAS OCCURRED DURING HISTORY FILE READ.'
      PRINT *,'  EXECUTION CONTINUING.'
      IF(IERR.LE.10) GO TO 19
      PRINT *,'  THE NUMBER OF ERRORS IN READING THE HISTORY FILE'
      PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
      GO TO 10
   23 WRITE(4,17) (A(I),I=1,3),IDATE,D(1),D(3),D(4)
      GO TO 19
   22 WRITE(3,20) (A(I),I=1,3),IDATE,D(1)
   20 FORMAT(2A4,A1,I7,A7)
      GO TO 19
   21 WRITE(2,20) (A(I),I=1,3),IDATE,D(1)
      GO TO 19
   18 PRINT 1
      PRINT *,'  OUTPUT FILES OF DIVERSION, FLOW, AND RESERVOIR DATA '
      PRINT *,'  FROM HISTORY FILE FOR SPECIFIED ACCOUNTING PERIOD'
      PRINT *,'  HAVE NOW BEEN CREATED AND ARE READY TO BE '
      PRINT *,'  SORTED FOR INPUT TO ACCOUNTING PROGRAM. '
```

```
C
C***** COMPUTE DATE FOR WHICH DATA IS TO BE RETRIEVED FROM ALLOCATIONS
C***** FILE (ONE DAY BEFORE FIRST ACCOUNTING DAY).
C
      IYD(4)=IYD(2)-1
      IY=IYD(4)/1000
      IF(IY1.EQ.IY) GO TO 25
      FEB=IY/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      J=635
      IF(FEB.LT.0.001) J=634
      IYD(4)=IYD(4)-J
   25 CONTINUE
      PRINT 1
      PRINT 26, IYD(4)
   26 FORMAT(3X,'ALL DATA FOR DAY ',I8,' IS BEING RETRIEVED FROM THE '/
     !3X,'ALLOCATIONS FILE.')
C
C***** SELECT ALL DATA FROM ALLOCATIONS FILE FOR DAY IYD(4) AND WRITE
C***** THESE DATA ON THREE OUTPUT FILES (D'S, R'S, AND M'S).
C
      IALK=0
      IERR=0
   29 READ(10,27,END=28,ERR=34) (A(I),I=1,3),IDATE,(A(I),I=4,16)
   27 FORMAT(2A4,A1,I7,12A4,A2)
      IF(IDATE.NE.IYD(4)) GO TO 29
      IALK=1
      IF(A(3).EQ.'D') GO TO 31
      IF(A(3).EQ.'R') GO TO 32
      IF(A(3).EQ.'M') GO TO 33
      GO TO 29
   34 IERR=IERR+1
      PRINT *,'  AN ERROR HAS OCCURRED DURING ALLOCATIONS FILE READ.'
      PRINT *,'  EXECUTION CONTINUING.'
      IF(IERR.LE.10) GO TO 29
      PRINT *,'  THE NUMBER OF ERRORS IN READING THE ALLOCATIONS FILE'
      PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
      GO TO 10
   33 WRITE(9,30) (A(I),I=1,3),IDATE,(A(I),I=4,16)
   30 FORMAT(2A4,A1,I7,12A4,A2)
      GO TO 29
   32 WRITE(8,30) (A(I),I=1,3),IDATE,(A(I),I=4,6)
      GO TO 29
   31 WRITE(7,30) (A(I),I=1,3),IDATE,(A(I),I=9,16)
      GO TO 29
   28 IF(IALK.EQ.0) GO TO 45
      PRINT 1
      PRINT *,'  OUTPUT FILES OF DIVERSION, RESERVOIR, AND'
      PRINT *,'  MISCELLANEOUS DATA FROM ALLOCATIONS FILE FOR '
      PRINT *,'  PREVIOUS DAY HAVE BEEN CREATED AND ARE NOW READY'
      PRINT *,'  TO BE SORTED FOR INPUT TO ACCOUNTING PROGRAM.'
   10 STOP
   45 PRINT 46, IYD(4)
   46 FORMAT(/3X,'  NO DATA FOR DAY ',I8,' WAS FOUND IN ALLOCATIONS')
      STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	3171	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	1980	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	916	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	6067	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOISEL$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	FEB		104= 105 106(2)= 110 123= 124 125(2)= 127 175= 176 177(2)= 179
**	I*4	I		108= 109 110 111 140(4)= 156(2)= 158(2)= 161(2)= 192(4)= 207(4)= 210(4)= 212(4)=
**	I*4	IALK		190= 195= 214
2-000000CC	I*4	IANSWR		58= 61 62 67= 69 70 84= 86 87
**	I*4	ID		101= 113 117=
2-000000C4	I*4	ID1		50= 53 90 101
2-000000D4	I*4	ID2		78= 80 90= 117
2-000000E0	I*4	IDATE		140= 142 143 156 158 161 192= 194 207 210 212
**	I*4	IERR		139= 149(2)= 152 191= 200(2)= 203
**	I*4	IFEB		105= 106 124= 125 176= 177
**	I*4	IM		100= 108 109 116=
2-000000C0	I*4	IM1		50= 53 89 100
2-000000D0	I*4	IM2		78= 80 89= 116
**	I*4	IY		102= 104 114 118= 121= 122 123 173= 174 175
2-000000C8	I*4	IY1		50= 53 91 102 122 174
2-000000D8	I*4	IY2		78= 80 91= 118
**	I*4	J		126= 127= 128 178= 179= 180
**	I*4	K		103= 114 115
**	I*4	NDY		107= 111(2)= 113(2)= 114

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000040	R*4	A		100	(25)	16	140=	144	145	146
						147	156	158	161	192(2)=
						196	197	198	207(2)	210(2)
						212(2)				
2-000000A4	CHAR	D		28	(4)	16	17	140=	156(3)	158
						161				
2-00000030	I*4	IYD		16	(4)	16	114=	120(2)=	121	128(2)=
						130(4)	142	143	172(2)=	173
						180(2)=	183	194	221	
2-00000000	I*4	JD		48	(12)	16	20D	110=	111	112=
**	R*4	OM		48	(12)	16	18D			

LABELS

Address	Label	References							
1-00000000	1'	30	31#	41	46	52	60	68	75
		79	85	134	163	182	215		
1-00000019	2'	50	51#	78					
1-00000024	3'	53	54#						
1-0000007F	4'	58	59#	67	84				
0-00000178	5	43#	62						
1-00000082	6'	80	81#						
0-000003A0	7	72#	87						
0-0000053A	8	88	92#						
0-0000052C	9	70	89#						
0-00000C5C	10	61	69	86	155	206	220#		
**	11	108	111#						
0-0000058E	12	109	112#						
**	13	103	119#						
0-000005B6	14	115	120#						
0-000005F9	15	122	129#						
1-000000DA	16'	130	131#						
1-0000014D	17'	140	141#	156					
0-000008B0	18	140	163#						
0-00000674	19	140#	142	143	148	152	157	160	162
1-0000015A	20'	158	159#	161					
0-000007E4	21	144	147	161#					
0-00000794	22	145	158#						
0-0000071C	23	146	156#						
0-00000830	24	140	149#						
0-0000097A	25	174	181#						
1-00000165	26'	183	184#						
1-000001B2	27'	192	193#						
0-00000BA0	28	192	214#						
0-000009B8	29	192#	194	199	203	209	211	213	

```
  1-000001C1  30'                          207       208#      210       212

  0-00000AD8  31                           196       212#
  0-00000A90  32                           197       210#
  0-00000A40  33                           198       207#
  0-00000B20  34                           192       200#
  1-00000004  35'                           48        49#       77

  1-00000078  36'                           56        57#       66        83
  0-00000C30  45                           214       221#
  1-000001D0  46'                          221       222#


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                              References

        ASSIGN                                21        28


  +-----------------------------------------+
  |            KEY TO REFERENCE FLAGS       |
  |    =    - Value Modified                |
  |    #    - Defining Reference            |
  |    A    - Actual Argument, possibly modified |
  |    D    - Data Initialization           |
  |   (n)   - Number of occurrences on line |
  +-----------------------------------------+


COMMAND QUALIFIERS

  FORTRAN/LIST/CROSS_REFERENCE BOISEL

  /ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
  /BLAS=(INLINE,MAPPED)
  /CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
  /DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
  /DIRECTIVES=(DEPENDENCE)
  /MATH_LIBRARY=(ACCURATE,NOV5)
  /PARALLEL=(NOAUTOMATIC,NOMANUAL)
  /SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
             NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
  /CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
  /NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=HYDRO:[BSUTTER.BOISE]BOISEL.LIS;3
  /OBJECT=HYDRO:[BSUTTER.BOISE]BOISEL.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

Run Time:	0.71 seconds
Elapsed Time:	1.43 seconds
Page Faults:	1222
Dynamic Memory:	1120 pages

BOTHSTSUM.LIS

```
C  ***************************************************
C     THIS PROGRAM LISTS DATA FROM WD-63 HISTORY FILES
C      IN USGS STYLE RECORDS BY IRRIGATION YEAR
C        AND SUMS STATIONS BETWEEN DESIGNATED
C                      POINTS.
C
C     THIS PROGRAM IS A MODIFICATION OF BOIHSTLST.FOR
C                  BOB SUTTER - MARCH 1987
C ***************************************************
C
      LOGICAL*1 FILES(40)
      BYTE CNTRL
      INTEGER TD,STA_ID,STATION_NUM,YEAR,O_ADJ,P_YEAR
      REAL MEAN,MAX,MIN,NOVALUE
      CHARACTER NAME*70,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
      CHARACTER TTL*70,BLANK*1,SNAME*70,NNAME*70
      DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
      DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12),SDV(12,31)
      DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
      CNTRL='00'X
      NOVALUE=9999999.
      BLANK=' '
      OPEN(UNIT=4,FILE='BOIHSTSUM.TTL',STATUS='OLD')
      OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
      OPEN(UNIT=8,FILE='BOIHSTSUM.OUT',STATUS='NEW',RECL=133)
  500 PRINT *,'*************************************'
      PRINT *,'   YOU HAVE A CHOICE:                '
      PRINT *,'                                                  '
      PRINT *,'   LIST ALL DIVERSIONS, FLOWS, DRAINS, OR RESERVOIRS '
      PRINT *,'  (STATION NUMBERS AND TITLES AUTOMATICALLY PROVIDED)'
      PRINT *,'                                                  '
      PRINT *,'              OR                  '
      PRINT *,'                                                  '
      PRINT *,'   LIST STATIONS INDIVIDUALLY                     '
      PRINT *,'   (YOU PROVIDE THE STATION NUMBER AND TITLE) '
      PRINT *,'                                  '
      PRINT *,'   TO SUM ALL DIVERSIONS, ANSWER  D         '
      PRINT *,'   TO SUM ALL RIVER FLOWS, ANSWER  F         '
      PRINT *,'   TO SUM ALL DRAINS, ANSWER  W         '
      PRINT *,'   TO SUM ALL RESERVOIRS, ANSWER  R         '
      PRINT *,'   TO SUM INDIVIDUAL STATIONS, ANSWER  I  '
      PRINT *,'   TO ENTER YOUR OWN GROUP OF STATIONS, ANSWER G '
      PRINT *,'   TO QUIT, ANSWER  Q  '
      PRINT *,'                                  '
      PRINT 904
  904 FORMAT('        ENTER CHOICE:'$)
      ACCEPT 902, IND
  902 FORMAT(A1)
      IF( IND .EQ. 'Q' ) GO TO 470
      PRINT *,' '
      PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
      PRINT *,'   IRIGATION YEAR FOR WHICH DATA IS TO BE LISTED - '
      PRINT *,' '
      PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR IRRIGATION YEAR  '
      PRINT *,'   1997, ENTER 1996-1997   '
      PRINT *,'              ----'----   '
      PRINT 901
```

```
  901 FORMAT('+                '$)
      KY=0
      ACCEPT 900, NP,NP1
  900 FORMAT(I4,1X,I4)
C
C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
C
C     NP=NP+1900
C     NP1=NP1+1900
      IF(JMOD(NP,4).EQ.0) KY=1
      ID=(NP*1000)+305+KY
      KY=0
      IF(JMOD(NP1,4).EQ.0) KY=1
      ID1=(NP1*1000)+304+KY
C
      IF( IND .EQ. 'I' ) THEN
                OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
  1             PRINT *,'   ENTER STATION NUMBER AND TITLE '
                PRINT *,'   TO END   ENTER   99999999  '
                PRINT *,'            -------- '
                PRINT 903
903             FORMAT('+            '$)
                ACCEPT 301, ISTA,TTL
                IF( ISTA .EQ. 99999999 ) GO TO 9
                WRITE(3,301) ISTA,TTL
                GO TO 1
  9             REWIND 3
                GO TO 2
                ENO IF
      IF( IND .NE. 'G' ) GO TO 8
      PRINT 907
  907 FORMAT('     ENTER FILE NAME:'$)
      ACCEPT 908, IC,(FILES(I),I=1,IC)
  908 FORMAT(Q,40A1)
      CALL ASSIGN(3,FILES)
      GO TO 2
    8 IF(IND.EQ.'D') OPEN(UNIT=3,FILE='BOIDIVSUM.TTL',STATUS='OLD')
      IF(IND.EQ.'F') OPEN(UNIT=3,FILE='BOIFLO.TTL',STATUS='OLD')
      IF(IND.EQ.'W') OPEN(UNIT=3,FILE='BOIDRN.TTL',STATUS='OLD')
      IF(IND.EQ.'R') OPEN(UNIT=3,FILE='BOIRES.TTL',STATUS='OLD')
    2 INSTA=0
      ISR=1
      STATION_NUM=0
    5 ISUM=0
      DO 40 I=1,12
      DO 40 J=1,31
   40 SDV(I,J)=0.0
   10 IF (INSTA.EQ.1)THEN
       STA_ID=NSTA_ID
       NAME=NNAME
       INSTA=0
       GO TO 15
       ENDIF
      READ(3,301,END=450) STA_ID,NAME
  301 FORMAT(I8,A70)
C
C***** FILL ARRAY WITH MISSING VALUE INDICATORS
```

```
C
   15 DO 20 I=1,12
      DO 20 J=1,31
   20 DV(I,J)=NOVALUE
C
C***** SKIP READ STATEMENT IF FIRST RECORD PREVIOUSLY READ
C
      IF(STATION_NUM.EQ.STA_ID) GO TO 60
      IF(STATION_NUM.GT.STA_ID) GO TO 651
C
C***** READ A RECORD
C
   25 READ(7,701,END=100) STATION_NUM,TYPE,YEAR,NO,VALUE
  701 FORMAT(I8,A1,I4,I3,F7.0)
      IF(STATION_NUM.LT.STA_ID) GO TO 25
      IF(STATION_NUM.EQ.STA_ID) GO TO 60
  651 IF(TD.NE.STA_ID) THEN
        PRINT 601,STA_ID
  601   FORMAT(1X,'STATION ',I8,' NOT FOUND')
        GO TO 10
        END IF
      READ(3,301,END=99) NSTA_ID,NNAME
      INSTA=1
   35 IF(STATION_NUM.LT.NSTA_ID) THEN
       READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
       GO TO 35
       END IF
      GO TO 100
C
C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
C
   60 IDATE=(YEAR*1000)+ND
      IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 25
      JD(2)=28
      IF( JMOD(YEAR,4) .EQ. 0 ) JD(2)=29
      LD=ND
      MON=0
   62 MON=MON+1
      IF(MON .GT. 12 ) GO TO 68
      LD=LD-JD(MON)
      IF( LD .GT. 0 ) GO TO 62
      LD=LD+JD(MON)
      DV(MON,LD)=VALUE
C
C***** SET VARIABLES TO PRESENT VALUES IN PREPARATION FOR
C***** READING NEXT RECORD
C
   68 CONTINUE
      TD=STATION_NUM
      P_YEAR=YEAR
      P_TYPE=TYPE
      GO TO 25
C
C***** ENTIRE YEAR FOR ONE STATION READ IN -
C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
C***** IF LAST STATION TITLE HAS BEEN READ, SET NEXT STATION
C***** NUMBER = 13999999 SO THAT SUMS WILL BE PRINTED.
```

```
C
   99 STATION_NUM=13999999
  100 LEAP=0
      IF(JMOD(NP1,4).EQ.0) LEAP=1
      IF( P_TYPE .NE. 'R' ) GO TO 120
        DELTA_S(1)=DV(1,31)-DV(12,31)
        DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
        DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
        DELTA_S(4)=DV(4,30)-DV(3,31)
        DELTA_S(5)=DV(5,31)-DV(4,30)
        DELTA_S(6)=DV(6,30)-DV(5,31)
        DELTA_S(7)=DV(7,31)-DV(6,30)
        DELTA_S(8)=DV(8,31)-DV(7,31)
        DELTA_S(9)=DV(9,30)-DV(8,31)
        DELTA_S(10)=DV(10,31)-DV(9,30)
        DELTA_S(11)=NOVALUE
        DELTA_S(12)=DV(12,31)-DV(11,30)
        DO 119 I=2,12
119     IF( ABS(DELTA_S(I)) .GE. 8000000. ) DELTA_S(I)=NOVALUE
120     DO 135 I=1,12
        D=31
        IF( I .EQ. 2 ) D=28+LEAP
        IF( I .EQ. 4 .OR. I .EQ. 6 .OR. I .EQ. 9 ) D=30
        CFSD(I)=DV(I,1)
        D_ADJ=0
        IF( DV(I,1) .EQ. NOVALUE )THEN
                                  CFSD(I)=0.0
                                  D_ADJ=1
                                  END IF
        MAX(I)=DV(I,1)
        IF( DV(I,1) .EQ. NOVALUE ) MAX(I)=-1.0
        MIN(I)=DV(I,1)
        DO 130 J=2,D
        DZ=DV(I,J)
        IF( DV(I,J) .EQ. NOVALUE ) THEN
                                   DZ=0.0
                                   D_ADJ=D_ADJ+1
                                   END IF
        CFSD(I)=CFSD(I)+DZ
        IF( DV(I,J) .EQ. NOVALUE ) THEN
                                   GO TO 129
                                   END IF
        IF( DV(I,J) .GT. MAX(I) ) MAX(I)=DV(I,J)
129     IF( DV(I,J) .LT. MIN(I) ) MIN(I)=DV(I,J)
130     CONTINUE
        IF( D_ADJ .GE. D ) THEN
                           CFSD(I)=NOVALUE
                           DELTA_S(I)=NOVALUE
                           MEAN(I)=NOVALUE
                           GO TO 133
                           END IF
      MEAN(I)=CFSD(I)/(D-D_ADJ)
  133 ACFT(I)=CFSD(I)*1.9835
  135 CONTINUE
      CALL DATE(RUNDATE)
      WRITE(8,801) RUNDATE
  801 FORMAT('1',/,120X,A9)
```

```
      IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
  800 FORMAT(1H+,'PRELIMINARY DATA')
      WRITE(8,802) STA_ID,NAME
  802 FORMAT(//37X,I8,3X,A70)
      IF( P_TYPE .EQ. 'R' ) THEN
                            WRITE(8,803) NP,NP1
  803 FORMAT(30X,'CONTENTS IN ACRE FEET, IRRIGATION YEAR NOVEMBER',
     *         I5,' TO OCTOBER',I5)
                            GO TO 160
      ELSE
      WRITE(8,804) NP,NP1
  804 FORMAT(30X,'DISCHARGE, CUBIC FEET PER SECOND, IRRIGATION YEAR',
     *         ' NOVEMBER',I5,' TO OCTOBER',I5)
      WRITE(8,805)
  805 FORMAT(59X,'MEAN VALUES')
      END IF
  160 WRITE(8,806)
  806 FORMAT(//,1X,'DAY',11X,'NOV',7X,'DEC',7X,'JAN',7X,'FEB',7X,
     *         'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',7X,'AUG',
     *         7X,'SEP',7X,'OCT')
      WRITE(8,807)
  807 FORMAT(' ')
      DO 180 J=1,31
        IF( J .EQ.6 .OR. J .EQ. 11 .OR. J .EQ. 16 .OR. J .EQ. 21
     *      .OR. J .EQ. 26 ) THEN
                              WRITE(8,807)
                              END IF
      DO 170 I=1,12
      II=I - 2
      IF( I .EQ. 1 ) II=11
      IF( I .EQ. 2 ) II=12
      IF( II .EQ. 11 ) WRITE(8,807)
      IF( II .EQ. 11 ) WRITE(8,811) CNTRL,J
  811 FORMAT(A1,I3,'    ')
        IF( DV(II,J) .EQ. NOVALUE ) THEN
                                    WRITE(8,809) CNTRL
                                    GO TO 170
                                    END IF
        IF( DV(II,J) .GE. 10.0 ) THEN
                                IDV=JNINT(DV(II,J))
                                WRITE(8,810) CNTRL,IDV
  810 FORMAT(A1,I10)
                                GO TO 170
                                END IF
        IF( DV(II,J) .LT. 10.0 ) THEN
                                 WRITE(8,808) CNTRL,DV(II,J)
808     FORMAT(A1,F10.1)
                                 GO TO 170
                                 END IF
809                                 FORMAT(A1,'       ---')
170     CONTINUE
180     CONTINUE

812     FORMAT('0')
        IF( P_TYPE .EQ. 'R' ) THEN
                      WRITE(8,812)
                      WRITE(8,813) CNTRL
```

```
813                 FORMAT(A1,2X,'MAX  ')
                    DO 190 I=1,12
                    II=I-2
                    IF( I .EQ. 1 ) II=11
                    IF( I .EQ. 2 ) II=12
                    IF( MAX(II) .EQ. -1.0 ) THEN
                              WRITE(8,860) CNTRL
860                           FORMAT(A1,'          ')
                              GO TO 190
                              END IF
                    IDV=JNINT(MAX(II))
                    WRITE(8,810) CNTRL,IDV
190                 CONTINUE
                    WRITE(8,807)
                    WRITE(8,814) CNTRL
814                 FORMAT(A1,2X,'MIN  ')
                    DO 200 I=1,12
                    II=I-2
                    IF( I .EQ. 1 ) II=11
                    IF( I .EQ. 2 ) II=12
                    IF( MIN(II) .EQ. NOVALUE ) THEN
                                               WRITE(8,860) CNTRL
                                               GO TO 200
                                               END IF
                    IDV=JNINT(MIN(II))
                    WRITE(8,810) CNTRL,IDV
200                 CONTINUE
                    WRITE(8,807)
                    WRITE(8,815) CNTRL
815                 FORMAT(A1,1X,'CHNG           ')
                    DO 210 I=2,12
                    II=I-2
                    IF( I .EQ. 2 ) II=12
                    IDV=JNINT(DELTA_S(II))
                    IF( DELTA_S(II) .EQ. NOVALUE )THEN
                                                 WRITE(8,860) CNTRL
                                                 GO TO 210
                                                 END IF
                    WRITE(8,810) CNTRL,IDV
210                 CONTINUE
                    GO TO 10
                    END IF


          WRITE(8,812)
          WRITE(8,816) CNTRL
816       FORMAT(A1,'TOTAL  ')
          DO 220 I=1,12
          II=I-2
          IF( I .EQ. 1 ) II=11
          IF( I .EQ. 2 ) II=12
          IF( CFSD(II) .GE. NOVALUE ) THEN
                                      WRITE(8,860) CNTRL
                                      GO TO 220
                                      END IF
          IDV=JNINT(CFSD(II))
          WRITE(8,810) CNTRL,IDV
```

```
220     CONTINUE
        WRITE(8,807)
        WRITE(8,818) CNTRL
818     FORMAT(A1,' MEAN  ')
        DO 230 I=1,12
        II=I-2
        IF( I .EQ. 1 ) II=11
        IF( I .EQ. 2 ) II=12
        IF( MEAN(II) .GE. NOVALUE ) THEN
                                     WRITE(8,860) CNTRL
                                     GO TO 230
                                     END IF
        IF( MEAN(II) .LT. 10.0 ) THEN
                                  WRITE(8,808) CNTRL,MEAN(II)
                                  GO TO 230
                                  END IF
        IDV=JNINT(MEAN(II))
        WRITE(8,810) CNTRL,IDV
230     CONTINUE
        WRITE(8,807)
        WRITE(8,819) CNTRL
819     FORMAT(A1,'  MAX  ')
        DO 240 I=1,12
        II=I-2
        IF( I .EQ. 1 ) II=11
        IF( I .EQ. 2 ) II=12
        IF( MAX(II) .EQ. -1.0 ) THEN
                                   WRITE(8,860) CNTRL
                                   GO TO 240
                                   END IF
        IF( MAX(II) .LT. 10.0 ) THEN
                                 WRITE(8,808) CNTRL,MAX(II)
                                 GO TO 240
                                 END IF
        IDV=JNINT(MAX(II))
        WRITE(8,810) CNTRL,IDV
240     CONTINUE
        WRITE(8,807)
        WRITE(8,820) CNTRL
820     FORMAT(A1,'  MIN  ')
        DO 250 I=1,12
        II=I-2
        IF( I .EQ. 1 ) II=11
        IF( I .EQ. 2 ) II=12
        IF( MIN(II) .GE. NOVALUE ) THEN
                                     WRITE(8,860) CNTRL
                                     GO TO 250
                                     END IF
        IF( MIN(II) .LT. 10.0 ) THEN
                                 WRITE(8,808) CNTRL,MIN(II)
                                 GO TO 250
                                 END IF
        IDV=JNINT(MIN(II))
        WRITE(8,810) CNTRL,IDV
250     CONTINUE

C        DO 260 I=1,12
```

```
C        IFT(I)=JNINT(ACFT(I)/100.)
C        IFT(I)=IFT(I)*100
C260     CONTINUE

         WRITE(8,807)
         WRITE(8,821) CNTRL
821      FORMAT(A1,'AC-FT  ')
         DO 270 I=1,12
         II=I-2
         IF( I .EQ. 1 ) II=11
         IF( I .EQ. 2 ) II=12
         IF( ACFT(II) .GE. 19834900. ) THEN
                                       WRITE(8,860) CNTRL
                                       GO TO 270
                                       END IF
C        IF( ACFT(II) .LE. 999. ) THEN
C                                  IFT(II)=JNINT(ACFT(II))
C                                WRITE(8,810) CNTRL,IFT(II)
C                                 GO TO 270
C                                 END IF
         IFT(II)=JNINT(ACFT(II))
         WRITE(8,810) CNTRL,IFT(II)
270      CONTINUE

         YT=0.0
         DO 280 I=1,12
         IF( CFSD(I) .EQ. NOVALUE ) CFSD(I)=0.0
280      YT=YT + CFSD(I)
         IYT=JNINT(YT)
         IYM=JNINT(YT/(365+LEAP))
C         NT=JNINT(YT*1.9835/100. )
C         IYA=NT*100
C         IF( IYA .LT. 9999 ) IYA=1.9835*YT
       IYA=1.9835*YT
       WRITE(8,807)
       WRITE(8,831) NP1,IYT,IYM,IYA
  831 FORMAT(11X,'IRRIGATION YEAR',I4,4X,'TOTAL',I10,4X,'MEAN',
     *         I6,4X,'AC-FT',I8)
C
C***** ENTER SUMMING ROUTINE UNLESS SUM HAS JUST BEEN PRINTED.
C
       IF(ISUM.EQ.1) GO TO 5
       DO 460 I=1,12
       DO 460 J=1,31
       IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).EQ.NOVALUE) GO TO 460
       IF(DV(I,J).GT.0.0.AND.SDV(I,J).EQ.NOVALUE) SDV(I,J)=0.0
       IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).GT.0.0) DV(I,J)=0.0
       SDV(I,J)=SDV(I,J)+DV(I,J)
  460 CONTINUE
       IF(ISR.EQ.0) GO TO 480
       READ(4,301,END=10) IDSUM,SNAME
       ISR=0
  480 IF(STATION_NUM.LT.IDSUM) GO TO 10 ! GO TO NEXT STATION
  485 STA_ID=IDSUM
       NAME=SNAME
       DO 465 I=1,12
       DO 465 J=1,31
```

```
00457     465 DV(I,J)=SDV(I,J)
00458         ISUM=1
00459         ISR=1
00460         GO TO 100  ! ORGANIZE AND PRINT SUMS
00461     450 CONTINUE
00462         IF(ISR.EQ.0) GO TO 485
00463         PRINT *,'  '
00464         PRINT *,'  '
00465         PRINT *,'  REQUESTED STATION LISTINGS AND SUMS COMPLETE '
00466         PRINT *,'  '
00467         PRINT *,'  '
00468     470 PRINT *,'                                    '
00469         PRINT *,'  OUTPUT IS  THE LATEST VERSION OF BOIHSTSUM.OUT'
00470         PRINT *,'  '
00471         STOP
00472         END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	5251	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	1917	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	4368	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	11536	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIHSTSUM$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
**	CHAR	BLANK		16	22=				
**	L*1	CNTRL		12	20=	261	264	269	274
				285	292	297	300	307	311
				314	321	324	331	338	342
				345	352	356	360	363	370
				374	378	381	388	392	396
				405	412	421			
**	R*4	D		192=	193=	194=	204	217	223
**	R*4	DZ		205=	207=	210			

```
      **        I*4  D_ADJ          13        196=      199=      208(2)=   217       223

      **        I*4  I              90(2)=    102=      104       116=      118       189=
                                   190(2)    191=      193       194(3)    195(2)    197
                                   198       201(2)    202(2)    203(2)    205       206
                                   210(2)    211       214(4)    215(4)    218       219
                                   220       223(2)    224(2)    256=      257       258
                                   259       287=      288       289       290       302=
                                   303       304       305       316=      317       318
                                   333=      334       335       336       347=      348
                                   349       350       365=      366       367       368
                                   383=      384       385       386       407=      408
                                   409       410       425=      426(2)    427       442=
                                   444(2)    445(3)    446(3)    447(3)    455=      457(2)
  2-00000E8C  I*4  IC               90(2)=
      **        I*4  ID             68=       147
      **        I*4  ID1            71=       147
      **        I*4  IDATE          146=      147(2)

  2-00000EA4  I*4  IDSUM            450=      452       453
      **        I*4  IDV            268=      269       296=      297       310=      311
                                    319=      324       341=      342       359=      360
                                    377=      378       395=      396
      **        I*4  II             257=      258=      259=      260       261       263
                                    267       268       273       274       288=      289=
                                    290=      291       296       303=      304=      305=
                                    306       310       317=      318=      319       320
                                    334=      335=      336=      337       341       348=
                                    349=      350=      351       355       356       359
                                    366=      367=      368=      369       373       374
                                    377       384=      385=      386=      387       391
                                    392       395       408=      409=      410=      411
                                    420(2)    421
  2-00000E7C  I*4  IND              47=       49        73        87        94        95
                                    96        97        229(2)
      **        I*4  INSTA          98=       105       108=      137=

      **        I*4  ISR            99=       449       451=      459=      462
  2-00000E88  I*4  ISTA             80=       81        82
      **        I*4  ISUM           101=      441       458=
      **        I*4  IYA            433=      435
      **        I*4  IYM            429=      435

      **        I*4  IYT            428=      435
      **        I*4  J              103=      104       117=      118       204=      205
                                    206       211       214(2)    215(2)    251=      252(5)
                                    261       263       267       268       273       274
                                    443=      444(2)    445(3)    446(3)    447(3)    456=
                                    457(2)
      **        I*4  KY             59=       67=       68        69=       70=       71
      **        I*4  LD             150=      154(2)=   155       156(2)=   157
      **        I*4  LEAP           174=      175=      178       179       193       429

      **        I*4  MON            151=      152(2)=   153       154       156       157
  2-00000D48  CHAR NAME             15        107=      111=      231       454=
  2-00000E98  I*4  ND               127=      139=      146       150
  2-00000E25  CHAR NNAME            16        107       136=
```

**	R*4	NOVALUE	14	21=	118	187	190	197
			202	206	211	218	219	220
			263	306	320	337	351	387
			426	444(2)	445	446		
2-00000E80	I*4	NP	60=	67	68	234	239	
2-00000E84	I*4	NP1	60=	70	71	175	234	239
			435					
2-00000E94	I*4	NSTA_ID	106	136=	138			
2-00000D8F	CHAR	P_TYPE	15	165=	176	233	283	
**	I*4	P_YEAR	13	164=				
2-00000D90	CHAR	RUNDATE	15	226A	227			
2-00000DDF	CHAR	SNAME	16	450=	454			
2-00000E74	I*4	STATION_NUM	13	100=	122	123	127=	129
			130	138	139=	163	173=	452
2-00000E70	I*4	STA_ID	13	106=	111=	122	123	129
			130	131	132	231	453=	
2-00000E6C	I*4	TD	13	131	163=			
**	CHAR	TITLE	15					
2-00000D99	CHAR	TTL	16	80=	82			
2-00000D8E	CHAR	TYPE	15	127=	139=	165		
2-00000E9C	R*4	VALUE	127=	139=	157			
2-00000E78	I*4	YEAR	13	127=	139=	146	149	164
**	R*4	YT	424=	427(2)=	428	429	433	
**	CHAR	ZLCH	15					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-000006F0	R*4	ACFT		48	(12)	18	224=	411	420	
2-00000690	R*4	CFSD		48	(12)	17	195=	198=	210(2)=	218=
						223	224	337	341	426(2)=
						427				
2-000006C0	R*4	DELTA_S		48	(12)	18	177=	178=	179=	180=
						181=	182=	183=	184=	185=
						186=	187=	188=	190(2)=	219=
						319	320			
2-000000C0	R*4	DV		1488	(12, 31)	17	118=	157=	177(2)	178(2)
						179(2)	180(2)	181(2)	182(2)	183(2)
						184(2)	185(2)	186(2)	188(2)	195
						197	201	202	203	205
						206	211	214(2)	215(2)	263
						267	268	273	274	444
						445	446(2)=	447	457=	
2-00000D20	L*1	FILES		40	(40)	11	90=	92A		
2-00000720	I*4	IFT		48	(12)	18	420=	421		
2-00000090	I*4	JD		48	(12)	17	19D	148=	149=	154
						156				
2-00000030	R*4	MAX		48	(12)	14	17	201=	202=	214(2)=
						291	296	369	373	374
						377				

```
  2-00000000  R*4  MEAN                    48  (12)            14        18        220=      223=      351
                                                              355       356       359
  2-00000060  R*4  MIN                     48  (12)            14        17        203=      215(2)=   306
                                                              310       387       391       392       395

  2-00000750  R*4  SDV                   1488  (12, 31)        18       104=       444       445(2)=   446
                                                              447(2)=   457
```

LABELS

```
    Address   Label                                   References

  0-00000404  1                                         75#       83
  0-00000598  2                                         85        93        98#
  0-000005A4  5                                        101#      441
  0-0000054C  8                                         87        94#
  0-000004D4  9                                         81        84#

  0-000005C8  10                                       105#      134       326       450       452
  0-00000614  15                                       109       116#
      **      20                                       116       117       118#
  0-000006AA  25                                       127#      129       147       166
  0-00000768  35                                       138#      140

      **      40                                       102       103       104#
  0-00000644  60                                       122       130       146#
  0-0000067C  62                                       152#      155
  0-0000069E  68                                       153       162#
  0-000007C4  99                                       136       173#

  0-0000080F  100                                      127       139       142       174#      460
      **      119                                      189       190#
  0-000008CB  120                                      176       191#
  0-0000096C  129                                      212       215#
      **      130                                      204       216#

  0-000009AD  133                                      221       224#
      **      135                                      191       225#
  0-00000AAB  160                                      237       245#
  0-00000C18  170                                      256       265       271       276       279#
      **      180                                      251       280#

  0-00000CDB  190                                      287       294       298#
  0-00000D8B  200                                      302       308       312#
  0-00000E31  210                                      316       322       325#
  0-00000EE7  220                                      333       339       343#
  0-00000FCF  230                                      347       353       357       361#

  0-000010BB  240                                      365       371       375       379#
  0-000011A7  250                                      383       389       393       397#
  0-00001261  270                                      407       413       422#
      **      280                                      425       427#
  1-00000066  301'                                      80        82       111       112#      136       450

  0-000007D0  450                                      111       461#
  0-0000134D  460                                      442       443       444       448#
```

```
     **        465                                    455       456       457#
  0-00001428   470                                     49       468#
  0-0000138E   480                                    449       452#

  0-000007D8   485                                    453#      462
     **        500                                     26#
  1-00000077   601'                                   132       133#
  0-00000706   651                                    123       131#
  1-0000006B   701'                                   127       128#      139

  1-0000009B   800'                                   229       230#
  1-00000092   801'                                   227       228#
  1-000000B1   802'                                   231       232#
  1-000000BC   803'                                   234       235#
  1-00000101   804'                                   239       240#

  1-00000153   805'                                   242       243#
  1-00000163   806'                                   245       246#
  1-000001C1   807'                                   249       250#      254       260       299       313
                                                      344       362       380       404       434
  1-000001D5   808'                                   274       275#      356       374       392
  1-000001DB   809'                                   264       278#

  1-000001D0   810'                                   269       270#      297       311       324       342
                                                      360       378       396       421
  1-000001C5   811'                                   261       262#
  1-000001EA   812'                                   282#      284       330
  1-000001EE   813'                                   285       286#
  1-00000209   814'                                   300       301#

  1-00000215   815'                                   314       315#
  1-0000022C   816'                                   331       332#
  1-00000238   818'                                   345       346#
  1-00000244   819'                                   363       364#
  1-00000250   820'                                   381       382#

  1-0000025C   821'                                   405       406#
  1-00000268   831'                                   435       436#
  1-000001FA   860'                                   292       293#      307       321       338       352
                                                      370       388       412
  1-00000030   900'                                    60        61#
  1-0000001C   901'                                    57        58#

  1-00000019   902'                                    47        48#
  1-00000037   903'                                    78        79#
  1-00000000   904'                                    45        46#
  1-00000047   907'                                    88        89#
  1-00000060   908'                                    90        91#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References					
R*4	ABS	190					
	ASSIGN	92					
	DATE	226					
I*4	JMOD	67	70	149	175		
I*4	JNINT	268	296	310	319	341	359
		377	395	420	428	429	

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|    =   - Value Modified                       |
|    #   - Defining Reference                   |
|    A   - Actual Argument, possibly modified   |
|    D   - Data Initialization                  |
|   (n)  - Number of occurrences on line        |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
  FORTRAN/LIST/CROSS_REFERENCE BOIHSTSUM

  /ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
  /BLAS=(INLINE,MAPPED)
  /CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
  /DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
  /DIRECTIVES=(DEPENDENCE)
  /MATH_LIBRARY=(ACCURATE,NOV5)
  /PARALLEL=(NOAUTOMATIC,NOMANUAL)
  /SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
             NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
  /CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
  /NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=HYDRO:[BSUTTER.BOISE]BOIHSTSUM.LIS;1
  /OBJECT=HYDRO:[BSUTTER.BOISE]BOIHSTSUM.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

```
  Run Time:           1.44 seconds
  Elapsed Time:       2.55 seconds
  Page Faults:        3579
  Dynamic Memory:     1824 pages
```

BOICHGS. LIS

```
C*************************************************************************
C
C        PROGRAM TO CHANGE VALUES IN DATA SET BOISYS.IND.
C       THESE VALUES ARE THE INITIAL RESERVOIR RIGHT STORAGE ACCUMULATIONS,
C       AND OTHER SYSTEM ACCUMULATIONS.   RJS - JUNE 1987.
C
C*************************************************************************

      DIMENSION CEV(3),GT(5),BRST(4),NAME(5),RIGHT(4)
      CHARACTER*17 NAME,RIGHT
      DATA NAME/'CHANGE IN STORAGE','MIDDLETON STORED ',
     1'TOTAL NEW STORAGE','UNACCOUNTED STORE','TOTAL EARLY FILL'/
      DATA RIGHT/'ARROWROCK 1911   ','ARROWROCK 1938   ',
     1'ANDERSON RANCH   ','LUCKY PEAK       '/
      CALL ASSIGN (1,'BOISYS.IND')
      OPEN (UNIT=2,NAME='BOISYS.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      READ(1,2,END=6) (CEV(L),L=1,3)
    2 FORMAT(8F10.1)
      WRITE(2,2) (CEV(L),L=1,3)
      READ(1,2,END=6) (GT(L),L=1,5)
      READ(1,2,END=6) (BRST(L),L=1,4)
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE YEAR TO DATE TOTALS
     1  '
      PRINT *,' '
      DO 101 I=1,5
      TYPE 11, NAME(I),GT(I)
   11 FORMAT(' ',A17,' =',F10.1,' NEW VALUE? Y/N '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 101
      PRINT 103
  103 FORMAT(' ENTER NEW VALUE   ='$)
      ACCEPT 4, GT(I)
    4 FORMAT(F10.0)
  101 CONTINUE
      WRITE(2,2) (GT(L),L=1,5)
      PRINT *,'                                                        '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR STORAGE
     1RIGHTS '
      PRINT *,' '
      DO 104 I=1,4
      TYPE 11, RIGHT(I),BRST(I)
      ACCEPT 102, IANS
      IF(IANS.NE.'Y') GO TO 104
      PRINT 103
      ACCEPT 4, BRST(I)
  104 CONTINUE
      WRITE(2,2) (BRST(L),L=1,4)
      PRINT *,' '
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	827	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	273	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	356	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1456	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGS$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
**	I*4	I		26=	27(2)	34	42=	43(2)	47
2-000000D0	I*4	IANS		29=	31	44=	45		
2-000000CC	I*4	L		17(2)=	19(2)=	20(2)=	21(2)=	37(2)=	49(2)=

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000020	R*4	BRST		16	(4)	9	21=	43	47=	49
2-00000000	R*4	CEV		12	(3)	9	17=	19		
2-0000000C	R*4	GT		20	(5)	9	20=	27	34=	37
2-00000030	CHAR	NAME		85	(5)	9	10	11D	27	
2-00000085	CHAR	RIGHT		68	(4)	9	10	13D	43	

LABELS

Address	Label	References						
1-000000CC	2'	17	18#	19	20	21	37	49
1-0000010D	4'	34	35#	47				
0-00000334	6	17	20	21	51#			
1-000000D2	11'	27	28#	43				
0-000001C8	101	26	31	36#				
1-000000F2	102'	29	30#	44				
1-000000F5	103'	32	33#	46				
0-000002EB	104	42	45	48#				

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	15
	FOR$OPEN	16

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGS

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
Run Time:           3.95 seconds
Elapsed Time:       15.59 seconds
Page Faults:        367
Dynamic Memory:     364 pages
```

```
C*************************************************************************
C
C        PROGRAM TO CHANGE VALUES IN DATA SET BOICAN.IND.
C       THESE VALUES ARE THE INITIAL CANAL STORAGE ALLOCATIONS.
C       RJS - JUNE 1987.
C
C*************************************************************************

      DIMENSION SRM(100),NAME(100),ID(100),IST(100)
      CHARACTER*16 NAME
      CALL ASSIGN (1,'BOICAN.IND')
      CALL ASSIGN (8,'BOIWRA.DPL')
      OPEN (UNIT=2,NAME='BOICAN.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      READ(1,2,END=1) (SRM(L),L=1,100)
    2 FORMAT(8F10.1)
    1 DO 10 L=1,100
      READ(8,9,END=7) ID(L),IST(L),NAME(L)
    9 FORMAT(I10,14X,I1,A16)
   10 CONTINUE
    7 L=L-1
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE CANAL STORAGE ACCOU
     1NTS  '
      PRINT *,' '
      DO 101 I=1,L
      IF(IST(I).EQ.1) GO TO 101
      TYPE 11, ID(I),NAME(I),SRM(I)
   11 FORMAT(' ',I8,1X,A16,' =',F10.1,' NEW VALUE? Y/N '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 101
      PRINT 103
  103 FORMAT('         ENTER NEW VALUE   ='$)
      ACCEPT 4, SRM(I)
    4 FORMAT(F10.0)
  101 CONTINUE
      WRITE(2,2) (SRM(I),I=1,L)
      PRINT *,' '
      PRINT *,' FILE BOICAN.IND HAS BEEN UPDATED WITH NEW VALUES'
      PRINT *,' '
      PRINT *,' '
      PRINT *,' '
      STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	649	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	243	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	2936	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3828	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
**	I*4	I		25=	26	27(3)	34	37(2)=	
2-00000AF8	I*4	IANS		29=	31				
2-00000AF0	I*4	L		14(2)=	16=	17(3)	20(2)=	25	37

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000190	I*4	ID		400	(100)	9	17=	27		
2-00000320	I*4	IST		400	(100)	9	17=	26		
2-000004B0	CHAR	NAME		1600	(100)	9	10	17=	27	
2-00000000	R*4	SRM		400	(100)	9	14=	27	34=	37

LABELS

Address	Label	References			
0-00000058	1	14	16#		
1-00000099	2'	14	15#	37	
1-000000EF	4'	34	35#		
0-000000B0	7	17	20#		
1-0000009F	9'	17	18#		
**	10	16	19#		
1-000000A8	11'	27	28#		
0-000001CA	101	25	26	31	36#
1-000000CC	102'	29	30#		
1-000000CF	103'	32	33#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	11	12
	FOR$OPEN	13	

```
+-----------------------------------------+
|          KEY TO REFERENCE FLAGS         |
|   =   - Value Modified                  |
|   #   - Defining Reference              |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization             |
|  (n)  - Number of occurrences on line   |
+-----------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGC

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
Run Time:           3.31 seconds
Elapsed Time:       10.45 seconds
Page Faults:        709
Dynamic Memory:     355 pages
```

ROTCHGC.LIS

```
C*************************************************************************
C
C        PROGRAM TO CHANGE VALUES IN DATA SET BOICAN.IND.
C       THESE VALUES ARE THE INITIAL CANAL STORAGE ALLOCATIONS.
C       RJS - JUNE 1987.
C
C*************************************************************************

      DIMENSION SRM(100),NAME(100),ID(100),IST(100)
      CHARACTER*16 NAME
      CALL ASSIGN (1,'BOICAN.IND')
      CALL ASSIGN (8,'BOIWRA.DPL')
      OPEN (UNIT=2,NAME='BOICAN.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      READ(1,2,END=1) (SRM(L),L=1,100)
    2 FORMAT(8F10.1)
    1 DO 10 L=1,100
      READ(8,9,END=7) ID(L),IST(L),NAME(L)
    9 FORMAT(I10,14X,I1,A16)
   10 CONTINUE
    7 L=L-1
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE CANAL STORAGE ACCOU
     1NTS  '
      PRINT *,' '
      DO 101 I=1,L
      IF(IST(I).EQ.1) GO TO 101
      TYPE 11, ID(I),NAME(I),SRM(I)
   11 FORMAT(' ',I8,1X,A16,' =',F10.1,' NEW VALUE? Y/N '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 101
      PRINT 103
  103 FORMAT('          ENTER NEW VALUE   ='$)
      ACCEPT 4, SRM(I)
    4 FORMAT(F10.0)
  101 CONTINUE
      WRITE(2,2) (SRM(I),I=1,L)
      PRINT *,' '
      PRINT *,' FILE BOICAN.IND HAS BEEN UPDATED WITH NEW VALUES'
      PRINT *,' '
      PRINT *,' '
      PRINT *,' '
      STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	649	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	243	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	2936	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3828	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
**	I*4	I		25=	26	27(3)	34	37(2)=	
2-00000AF8	I*4	IANS		29=	31				
2-00000AF0	I*4	L		14(2)=	16=	17(3)	20(2)=	25	37

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000190	I*4	ID		400	(100)	9	17=	27		
2-00000320	I*4	IST		400	(100)	9	17=	26		
2-000004B0	CHAR	NAME		1600	(100)	9	10	17=	27	
2-00000000	R*4	SRM		400	(100)	9	14=	27	34=	37

LABELS

Address	Label	References			
0-00000058	1	14	16#		
1-00000099	2'	14	15#	37	
1-000000EF	4'	34	35#		
0-000000B0	7	17	20#		
1-0000009F	9'	17	18#		
**	10	16	19#		
1-000000A8	11'	27	28#		
0-000001CA	101	25	26	31	36#
1-000000CC	102'	29	30#		
1-000000CF	103'	32	33#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	11	12
	FOR$OPEN	13	

```
+-------------------------------------------+
|          KEY TO REFERENCE FLAGS           |
|   =   - Value Modified                    |
|   #   - Defining Reference                |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization               |
|  (n)  - Number of occurrences on line     |
+-------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGC

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
Run Time:           3.31 seconds
Elapsed Time:       10.45 seconds
Page Faults:        709
Dynamic Memory:     355 pages
```

ROTACNGC. LIS

```
C***********************************************************************
C
C       PROGRAM TO CHANGE CANAL REMAINING STORAGE FROM ALLOCATIONS
C       FILE FOR INPUT TO ACCOUNTING PROGRAM.  RJS - JUNE 1987.
C
C***********************************************************************

      DIMENSION SRM(100),NAME(100),ID(100),CODE(100),SRU(100)
      DIMENSION ID1(100),JK(100)
      CHARACTER*8 CODE,CODEK
      CHARACTER*16 NAME
      CALL ASSIGN (1,'BOIWRA.ALD')
      CALL ASSIGN (8,'BOIWRA.DPL')
      OPEN (UNIT=2,NAME='BOIALD.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      READ(1,2,END=1) (ID1(K),CODE(K),SRU(K),SRM(K),K=1,100)
    2 FORMAT(I8,A8,2F10.1)
    1 KEND=K-1
      DO 10 L=1,100
      READ(8,9,END=7) ID(L),JK(L),NAME(L)
    9 FORMAT(I10,14X,I1,A16)
   10 CONTINUE
    7 LEND=L-1
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE CANAL STORAGE ACCOU
     1NTS  '
      PRINT *,' '
      K=0
      KR=1
      DO 101 I=1,LEND
      IF(KR.GT.0) THEN
         K=K+1
         KR=0
         END IF
      IF(K.GT.KEND) GO TO 6
      CODEK=CODE(K)
      SRUI=SRU(K)
      SRMI=SRM(K)
      IF(ID1(K).EQ.ID(I)) GO TO 98
    6 IF(JK(I).GT.0) GO TO 101
      SRUI=0.0
      SRMI=0.0
      GO TO 99
   98 KR=1
   99 TYPE 11, ID(I),NAME(I),SRMI
   11 FORMAT(' ',I8,1X,A16,' =',F10.1,' NEW VALUE? Y/N/Q '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.EQ.'Y') GO TO 100
      IF(IANS.EQ.'N') GO TO 3
      IF(IANS.EQ.'Q') GO TO 104
      GO TO 99
  100 PRINT 103
  103 FORMAT('          ENTER NEW VALUE   ='$)
      ACCEPT 4, SRMI
    4 FORMAT(F10.0)
    3 IF(SRMI.EQ.0.0.AND.SRUI.EQ.0.0) GO TO 101
      WRITE(2,2) ID(I),CODEK,SRUI,SRMI
```

```
0058      101 CONTINUE
0059          GO TO 5
0060      104 IF(K.LE.KEND) WRITE(2,2) (ID1(I),CODE(I),SRU(I),SRM(I),I=K,KEND)
0061        5 PRINT *,' '
0062          PRINT *,' FILE HAS BEEN UPDATED WITH NEW VALUES'
0063          PRINT *,' '
0064          PRINT *,' '
0065          PRINT *,' '
0066          STOP
0067          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	943	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	238	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	4608	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	5789	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIACHGC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
2-00001130	CHAR	CODEK		10	35=	57					
**	I*4	I		29=	38	39	44(2)	57	60(5)=		
2-00001144	I*4	IANS		46=	48	49	50				
2-00001138	I*4	K		15(5)=	17	27=	31(2)=	34	35	36	37
				38	60(2)						
**	I*4	KEND		17=	34	60(2)					
**	I*4	KR		28=	30	32=	43=				
**	I*4	L		18=	19(3)	22					
**	I*4	LEND		22=	29						
2-00001140	R*4	SRMI		37=	41=	44	54=	56	57		
**	R*4	SRUI		36=	40=	56	57				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000E10	CHAR	CODE		800	(100)	8	10	15=	35	60
2-00000190	I*4	ID		400	(100)	8	19=	38	44	57
2-000004B0	I*4	ID1		400	(100)	9	15=	38	60	
2-00000640	I*4	JK		400	(100)	9	19=	39		
2-000007D0	CHAR	NAME		1600	(100)	8	11	19=	44	
2-00000000	R*4	SRM		400	(100)	8	15=	37	60	
2-00000320	R*4	SRU		400	(100)	8	15=	36	60	

LABELS

Address	Label	References			
0-00000087	1	15	17#		
1-0000008E	2'	15	16#	57	60
0-000002D0	3	49	56#		
1-000000EA	4'	54	55#		
0-0000031C	5	59	61#		
0-00000180	6	34	39#		
0-000000E0	7	19	22#		
1-00000098	9'	19	20#		
**	10	18	21#		
1-000000A1	11'	44	45#		
0-0000017A	98	38	43#		
0-0000018E	99	42	44#	51	
0-00000294	100	48	52#		
0-00000316	101	29	39	56	58#
1-000000C7	102'	46	47#		
1-000000CA	103'	52	53#		
0-00000220	104	50	60#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	12	13
	FOR$OPEN	14	

```
+-------------------------------------------+
|          KEY TO REFERENCE FLAGS           |
|   =   - Value Modified                    |
|   #   - Defining Reference                |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization               |
|  (n)  - Number of occurrences on line     |
+-------------------------------------------+
```

```
COMMAND QUALIFIERS

  FORTRAN/LIST/CROSS_REFERENCE BOIACHGC

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /SHOW=(NODICTIONARY,NOINCLUDE,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NODECLARATIONS,GENERAL,NOULTRIX,NOVAXELN)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOPARALLEL
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=HYDRO:[SUTTER.BOISE]BOIACHGC.LIS;2
  /OBJECT=HYDRO:[SUTTER.BOISE]BOIACHGC.OBJ;2


COMPILATION STATISTICS

  Run Time:           3.40 seconds
  Elapsed Time:       5.14 seconds
  Page Faults:        295
  Dynamic Memory:     407 pages
```

SIL SSRCVAIVA

```
C***********************************************************************
C
C       PROGRAM TO CHANGE RESERVOIR RIGHT STORAGE ACCUMULATIONS,
C       AND OTHER SYSTEM ACCUMULATIONS FROM ALLOCATIONS FILE AS
C       INPUT TO ACCOUNTING PROGRAM.   RJS - JUNE 1987.
C
C***********************************************************************

      DIMENSION CEV(3),GT(5),BRST(4),NAME(5),RIGHT(4)
      CHARACTER*16 CODE1,CODE2
      CHARACTER*17 NAME,RIGHT
      DATA NAME/'CHANGE IN STORAGE','MIDDLETON STORED ',
     1'TOTAL NEW STORAGE','UNACCOUNTED STORE','TOTAL EARLY FILL '/
      DATA RIGHT/'ARROWROCK 1911   ','ARROWROCK 1938   ',
     1'ANDERSON RANCH   ','LUCKY PEAK       '/
      CALL ASSIGN (1,'BOIWRA.ALM')
      OPEN (UNIT=2,NAME='BOIALM.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      READ(1,2,END=6) (CODE1,(BRST(L),L=1,4))
    2 FORMAT(A16,5F10.1)
      READ(1,2,END=6) (CODE2,(GT(L),L=1,5))
      PRINT *,'                                                    '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR STORAGE
     1RIGHTS '
      PRINT *,' '
      DO 104 I=1,4
      TYPE 11, RIGHT(I),BRST(I)
      ACCEPT 102, IANS
      IF(IANS.NE.'Y') GO TO 104
      PRINT 103
      ACCEPT 4, BRST(I)
  104 CONTINUE
      WRITE(2,2) (CODE1,(BRST(L),L=1,4))
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE YEAR TO DATE TOTALS
     1  '
      PRINT *,' '
      DO 101 I=1,5
      TYPE 11, NAME(I),GT(I)
   11 FORMAT(' ',A17,' =',F10.1,' NEW VALUE? Y/N '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 101
      PRINT 103
  103 FORMAT(' ENTER NEW VALUE   ='$)
      ACCEPT 4, GT(I)
    4 FORMAT(F10.0)
  101 CONTINUE
      WRITE(2,2) (CODE2,(GT(L),L=1,5))
      PRINT *,' '
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	776	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	275	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	388	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1439	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIACHGS$MAIN	

VARIABLES

Address	Type	Name	Attributes	References					
2-000000C9	CHAR	CODE1		10	18=	32			
2-000000D9	CHAR	CODE2		10	20=	48			
**	I*4	I		25=	26(2)	30	37=	38(2)	45
2-000000F0	I*4	IANS		27=	28	40=	42		
2-000000EC	I*4	L		18(2)=	20(2)=	32(2)=	48(2)=		

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000020	R*4	BRST		16	(4)	9	18=	26	30=	32
2-00000000	R*4	CEV		12	(3)	9				
2-0000000C	R*4	GT		20	(5)	9	20=	38	45=	48
2-00000030	CHAR	NAME		85	(5)	9	11	12D	38	
2-00000085	CHAR	RIGHT		68	(4)	9	11	14D	26	

LABELS

Address	Label	References				
1-000000CC	2'	18	19#	20	32	48
1-0000010F	4'	30	45	46#		
0-00000301	6	18	20	50#		
1-000000D4	11'	26	38	39#		
0-000002B0	101	37	42	47#		
1-000000F4	102'	27	40	41#		
1-000000F7	103'	29	43	44#		
0-00000183	104	25	28	31#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	16
	FOR$OPEN	17

```
+-------------------------------------------+
|          KEY TO REFERENCE FLAGS           |
|   =   - Value Modified                    |
|   #   - Defining Reference                |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization               |
|  (n)  - Number of occurrences on line     |
+-------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOIACHGS

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
Run Time:           3.81 seconds
Elapsed Time:       5.55 seconds
Page Faults:        456
Dynamic Memory:     364 pages
```

BOTSTO.115

```
C*********************************************************************
C
C       PROGRAM BOISTO.FOR TO COMPUTE BEGINNING OF IRRIGATION SEASON
C       RESERVOIR STORAGE BY CANAL OR USE FOR ANDERSON RANCH, ARROWROCK,
C       AND YUCKY PEAK RESERVOIRS.        RJS -  JUNE 1997
C*********************************************************************
      DIMENSION CTOT(10),STOR(80,3),ID(80),ID2(80),UNAME(80),STOT(9)
      DIMENSION USED(80),EXCS(80),RFWB(80),CRYO(80,3),OCTB(80)
      DIMENSION BALN(80),IK(80),KRYO(80,3),RTOT(80)
      DIMENSION FILL(4),SPACE(4),YIELD(4),EVAP(4),TOTR(80,3)
      DIMENSION CTOTC(3),CTOTR(3),SUBT(80),NSTO(80,3),CNSTO(3)
      DIMENSION T(4),RNAME(4),BANK(80)
      DIMENSION PAVAIL(3),TRANS(80),RFTR(80)
      DIMENSION IPTAB(80),FILLATE(8),TFILL(8)
      CHARACTER*9 RUNDATE
      CHARACTER*18 UNAME,U
      CHARACTER*16 RNAME
      REAL*8 CTOT,STOR,STOT,CTOTC,CRYO,CTOTR,SUBT,CSUBT,TBANK,BANK
      REAL*8 FILL,SPACE,YIELD,GTOT,T,TOTR,RTOT,KRYO
      REAL*8 CNSTO,NSTO,FILLLPL,TTRANS,TRANS,UNACCT
      CALL DATE(RUNDATE)
      CALL ASSIGN(1,'BOISTO.IND')
      CALL ASSIGN(5,'BOISTO.CRY')
      CALL ASSIGN(6,'BOISTO.RPT')
      CALL ASSIGN(7,'BOISTO.SPA')
      CALL ASSIGN(9,'BOISTO.USE')
      CALL ASSIGN(10,'BOISTO.UNC')
      L=1
      I=1
      IN2=10
   30 READ(7,10,END=25) ID(I),IK(I),UNAME(I),(STOR(I,K),K=1,3)
   10 FORMAT(I8,A1,A18,4F10.0)
      IF(ID(I).EQ.13201991) IUN=I
      IF(ID(I).EQ.13202995) IPEN=I
      IF(ID(I).EQ.13203000) INYK=I
      IF(ID(I).EQ.13209480) IPHL=I
      IF(ID(I).EQ.13210005) ICAL=I
      IF(ID(I).EQ.99999100) IEG=I
      IF(ID(I).EQ.0) GO TO 25
      I=I+1
      GO TO 30
   25 CONTINUE
      IE=I-1
      DO 40 I=1,IE
      READ(5,12,END=5) ID2(I),U,(CRYO(I,J),J=1,3),TRANS(I),BANK(I)
   12 FORMAT(I8,1X,A18,5F10.0)
      IF(ID2(I).EQ.ID(I)) GO TO 40
      TYPE 2, ID2(I),ID(I)
    2 FORMAT(' DIVERSION ',I8,' FROM BOISTO.CRY IS DIFFERENT THAN DIVERS
     1ION ',I8,' FROM BOISTO.SPA')
      STOP
   40 CONTINUE
      READ(10,305,ERR=310) UNACCT
```

```
  305 FORMAT(11X,F9.1)
      GO TO 5
  310 TYPE 306
  306 FORMAT(' ERR READING BOISTO.UNC ')
      STOP
    5 TYPE *, ' '
      TYPE *, '  ENTER FOUR DIGIT YEAR FOR WHICH YOU WANT  '
      TYPE *, '  TO COMPUTE STORAGE ALLOCATION AND USE.       '
      TYPE *, '  FOR EXAMPLE: 1998                          '
      TYPE *, ' '
      ACCEPT 49,IYR
   49 FORMAT(I4)
      READ(1,55) (RNAME(K),SPACE(K),FILL(K),EVAP(K),K=1,4)
   55 FORMAT(9X,A16,3F10.0)
C
C***** COMPUTE AND PRINT TABLE OF EACH RESERVOIR'S SPACE, FILL,
C***** EVAPORATION AND YIELD.
C
      IT=1
      DO 90 K=1,3
      YIELD(K)=FILL(K)-EVAP(K)
      PAVAIL(K)=0.0
      IF(FILL(K).GT.0.0) PAVAIL(K)=YIELD(K)/FILL(K)
   90 CONTINUE
      WRITE(6,100) RUNDATE,IT,IYR
  100 FORMAT(1H1////////102X,A9/////////////,40X,'TABLE'I3,'. ',I4,
     1  ' BOISE RIVER STORED WATER BY RESERVOIR'//55X,' (ACRE-FEET)'/)
      WRITE(6,101)
  101 FORMAT(32X,9HRESERVOIR,14X,5HSPACE, 8X,4HFILL, 1X,11HEVAPORATION,
     1  7X,5HYIELD/)
      DO 105 K=1,3
  105 T(K)=0.0
      DO 110 K=1,3
      T(1)=T(1)+SPACE(K)
      T(2)=T(2)+FILL(K)
      T(3)=T(3)+EVAP(K)
      T(4)=T(4)+YIELD(K)
      WRITE(6,120) RNAME(K),SPACE(K),FILL(K),EVAP(K),YIELD(K)
  120 FORMAT(32X,A16,4F12.1)
  110 CONTINUE
      WRITE(6,130) (T(K),K=1,4)
  130 FORMAT(/32X,5HTOTAL,11X,4F12.1)
C
C***** WRITE TABLE OF SPACE ALLOCATED TO EACH ENTITY OR USE IN
C***** ALL THREE BOISE RIVER RESERVOIRS BASED ON COMPLETELY
C***** FULL RESERVOIRS.
C
      IT=2
      DO 155 K=1,10
  155 CTOT(K)=0.0
      WRITE(6,160) RUNDATE,IT,IYR
  160 FORMAT(1H1/////120X,A9//////36X,6HTABLE ,I2,'. ',I4,
     1  ' BOISE RIVER RESERVOIR SPACE BY USER. (ACRE-FEET)'//)
      WRITE(6,97)
   97 FORMAT(1H ,31X,13HNUMBER   USER,14X,'  ARROWROCK   ANDERSON LUCKY
     1PEAK      TOTAL'/)
```

```
      DO 3 K=1,3
      CTOTR(K)=0.0
    3 CTOTC(K)=0.0
      DO 80 I=1,IE
      CRYO(I,1)=0.0
      RTOT(I)=0.0
      IPTAB(I)=0
      DO 60 K=1,3
      CTOT(K)=CTOT(K)+STOR(I,K)
      RTOT(I)=RTOT(I)+STOR(I,K)
      NSTO(I,K)=0.0
   60 CONTINUE
      CTOT(4)=CTOT(4)+RTOT(I)
      IF(RTOT(I).GT.0.0) THEN
      WRITE(6,71) ID(I),UNAME(I),(STOR(I,K),K=1,3),RTOT(I)
   71 FORMAT(32X,I8,1X,A18,4F11.1)
      IPTAB(I)=1
      ENDIF
   80 CONTINUE
      WRITE(6,21) (CTOT(K),K=1,4)
   21 FORMAT(/41X,5HTOTAL,13X,4F11.1)
C
C***** IF THE SPACE IN ANDERSON RANCH HAS ENTIRELY FILLED, ZERO
C***** OUT THE CARRYOVER FROM THE PREVIOUS YEAR. (DO NOT DO THIS
C***** FOR BOISE).  IF DID NOT FILL, CHECK FOR FLOOD CONTROL.
C
      DIFF=SPACE(2)-FILL(2)
      IF(DIFF.LE.0.01) GO TO 18
   31 TYPE *, ' '
      PRINT 32
   32 FORMAT(' WAS THE FAILURE TO FILL AT ANDERSON RANCH DUE TO'/
     !' A FLOOD CONTROL OPERATION? Y/N: '$)
      ACCEPT 26, FL
   26 FORMAT(A1)
      IF(FL.EQ.'N') GO TO 18
      IF(FL.NE.'Y') GO TO 31
      DO 19 I=1,IE
   19 CRYO(I,2)=0.0
   18 CONTINUE
C
C***** IF LUCKY PEAK SPACE DID NOT FILL, INQUIRE IF THIS WAS THE
C***** RESULT OF A FLOOD CONTROL OPERATION.  IF IT WAS, THEN
C***** COMPUTE LUCKY PEAK STORAGE ALLOCATIONS WITH THE FINAL 60000
C***** ACRE-FEET HAVING THE LAST FILL.  TO DO THIS, REMOVE THE
C***** 60000 FROM THE TOTAL ALLOCABLE STORAGE AND THE STREAMFLOW
C***** MAINTENANCE SPACE. IF ANY OF THE LAST 60000 HAS FILLED
C***** (FILLLPL) ADD THIS BACK IN AFTER THE OTHER STORAGE HAS BEEN
C***** ALLOCATED.
C
      DIFF=SPACE(3)-FILL(3)
      IF(DIFF.LE.0.01) GO TO 24
   28 TYPE *, ' '
      PRINT 27
   27 FORMAT(' WAS THE FAILURE TO FILL AT LUCKY PEAK DUE TO A'/
```

```
      !' FLOOD CONTROL OPERATION? Y/N: '$)
       ACCEPT 26, FL
       IF(FL.EQ.'N') GO TO 24
       IF(FL.NE.'Y') GO TO 28
       STOR(IUN,3)=STOR(IUN,3)-60000.
       CTOT(3)=CTOT(3)-60000
       FILLLPL=FILL(3)-CTOT(3)
       IF(FILLLPL.LT.0.0) FILLLPL=0.0
       FILL(3)=FILL(3)-FILLLPL
C
C***** IF THE SPACE IN LUCKY PEAK DID NOT FILL DUE TO
C***** FLOOD CONTROL AND IF NONE OF THE 60000 EXCLUSIVE
C***** FLOOD SPACE FILLED, ZERO OUT THE CARRYOVER FROM THE
C***** PREVIOUS YEAR. THE CARRYOVER COULD BE ZEROED OUT FOR
C***** OTHER CONDITIONS, BUT LEE SISCO WANTS TO SHOW LAST
C***** YEAR'S CARRYOVER IN THE TABLE EVEN IF THE IRRIGATION
C***** SPACE FILLS COMPLETELY.
C
       IF(FILLLPL.EQ.0.0) THEN
             DO 23 I=1,IE
   23        CRYO(I,3)=0.0
             ENDIF
   24 CONTINUE
C
C
C***** COMPUTE NEW FILL TO EACH RESERVOIR BY SUBTRACTING CARRYOVER
C***** FROM TOTAL FILL, THEN ALLOCATE NEW FILL TO INDIVIDUAL
C***** ENTITIES AND USES WITHIN EACH RESERVOIR PROPORTIONAL TO
C***** SPACE OWNED.
C
       DO 1 K=1,3
       IF(FILL(K).GT.CTOT(K)) TYPE 997, RNAME(K),FILL(K),CTOT(K)
  997 FORMAT(1X,A16,' HAS FILL =',F10.1,' > TOTAL SPACE =',F10.1)
       DO 29 I=1,IE
   29 CTOTC(K)=CTOTC(K)+CRYO(I,K)
       CNSTO(K)=FILL(K)-CTOTC(K)
       WAT=CNSTO(K)
       J=0
    6 EXCESS=0.0
       J=J+1
       IF(J.GT.2000) TYPE 998, RNAME(K)
  998 FORMAT(' RUNAWAY LOOP ALLOCATING ',A16,' STORAGE')
       TYPE 999, J,WAT
  999 FORMAT(1X,I5,F10.1)
       DO 4 I=1,IE
       NSTO(I,K)=((STOR(I,K)/CTOT(K))*WAT)+NSTO(I,K)
       TOTR(I,K)=NSTO(I,K)+CRYO(I,K)
       IF(TOTR(I,K).LE.STOR(I,K)) GO TO 4
       EXCESS=EXCESS+TOTR(I,K)-STOR(I,K)
       TOTR(I,K)=STOR(I,K)
       NSTO(I,K)=STOR(I,K)-CRYO(I,K)
    4 CONTINUE
       WAT=EXCESS
       IF(EXCESS.GT.0.01) GO TO 6
    1 CONTINUE
C
C***** RECOMPUTE LUCKY PEAK STREAMFLOW MAINTENANCE STORAGE TO INCLUDE
```

```
C***** NEW FILL IN LAST 60000 ACRE-FEET.
C
      NSTO(IUN,3)=NSTO(IUN,3)+FILLLPL
      TOTR(IUN,3)=TOTR(IUN,3)+FILLLPL
      CNSTO(3)=CNSTO(3)+FILLLPL
C
C***** PRINT TABLE OF BEGINNING OF YEAR STORAGE ACCOUNTS.
C
      IT=3
  145 WRITE(6,150) RUNDATE,IT,IYR
  150 FORMAT(1H1/////120X,A9//////36X,6HTABLE ,I2,'. ',I4,' BOISE
     1RIVER RESERVOIR STORAGE ACCOUNTS.  (ACRE-FEET)'//)
      WRITE(6,95)
   95 FORMAT(1H ,37X,'ARROWROCK *-----ANDERSON RANCH-----* *-------LUCKY
     1 PEAK-------*'/11X,13HNUMBER   USER,14X,'  STORAGE CARRYOVER NEW FI
     2LL STORAGE CARRYOVER NEW FILL STORAGE  SUBTOTAL'/)
      CSUBT=0.0
      DO 82 I=1,IE
      DO 81 M=1,3
      SUBT(I)=SUBT(I)+TOTR(I,M)
   81 CTOTR(M)=CTOTR(M)+TOTR(I,M)
      CSUBT=CSUBT+SUBT(I)
      IF(IPTAB(I).GT.0.OR.RTOT(I).GT.0.0) WRITE(6,70) ID(I),
     1UNAME(I),TOTR(I,1),CRYO(I,2),NSTO(I,2),TOTR(I,2),CRYO(I,3),
     2NSTO(I,3),TOTR(I,3),SUBT(I)
   70 FORMAT(11X,I8,1X,A18,8F9.1)
   82 CONTINUE
      WRITE(6,22) CTOTR(1),CTOTC(2),CNSTO(2),CTOTR(2),CTOTC(3),
     1 CNSTO(3),CTOTR(3),CSUBT
   22 FORMAT(/20X,5HTOTAL,13X,8F9.1)
  200 CONTINUE
C
C***** COMPUTE YIELD OF SPACE VALUES AFTER EVAPORATION.
C
      DO 165 I=1,IE
      DO 175 K=1,3
  175 TOTR(I,K)=TOTR(I,K)*PAVAIL(K)
  165 CONTINUE
C
C***** WRITE TABLE OF STORAGE ALLOCATED TO EACH ENTITY OR USER
C***** IN ALL THREE BOISE RIVER RESERVOIRS AFTER EVAPORATION
C***** WITH WATER BANK AND OTHER TRANSACTIONS.
C
      IT=4
      WRITE(6,161) RUNDATE,IT,IYR
```

```
  161 FORMAT(1H1/////120X,A9/28X,6HTABLE ,I2,'. ',I4,
     1  ' BOISE RIVER RESERVOIR NET STORAGE BY USER. (ACRE-FEET)'//)
      WRITE(6,168)
  168 FORMAT(1H ,10X,13HNUMBER   USER,16X,'ARROWROCK   ANDERSON  LU
     1CKY PEAK  SUBTOTAL  TRANSFERS   RNT POOL      TOTAL'/)
      TBANK=0.0
      TTRANS=0.0
      GTOT=0.0
      CSUBT=0.0
      DO 65 K=1,3
   65 CTOTR(K)=0.0
      DO 85 I=1,IE
      SUBT(I)=0.0
      DO 83 M=1,3
      CTOTR(M)=CTOTR(M)+TOTR(I,M)
   83 SUBT(I)=SUBT(I)+TOTR(I,M)
      RTOT(I)=SUBT(I)+TRANS(I)+BANK(I)
      CSUBT=CSUBT+SUBT(I)
      TTRANS=TTRANS+TRANS(I)
      TBANK=TBANK+BANK(I)
      GTOT=GTOT+RTOT(I)
      IF (IPTAB(I).GT.0.OR.RTOT(I).GT.0.0) WRITE (6,75) ID(I),
     1UNAME(I),TOTR(I,1),TOTR(I,2),TOTR(I,3),SUBT(I),TRANS(I),
     2BANK(I),RTOT(I)
   75 FORMAT (11X,I8,1X,A18,7F11.1)
   85 CONTINUE
      WRITE (6,35) CTOTR(1),CTOTR(2),CTOTR(3),CSUBT,TTRANS,
     1TBANK,GTOT
   35 FORMAT (/20X,5HTOTAL,13X,7F11.1)
C
C***** COMPUTE END OF SEASON STORAGE BALANCES
C
      IT=3
      JDI=0
      EGU=0.0
      DO 315 I=1,IE
      IF(JDI.EQ.1) THEN
      JDI=0
      GO TO 262
      ENDIF
      READ(9,260,END=16,ERR=400) JD,T1
  260 FORMAT(I8,8X,F10.0)
  262 IF(JD.EQ.ID(I)) GO TO 320
   16 USED(I)=0.0
      JDI=1
      GO TO 315
  320 USED(I)=T1
   17 IF(IK(I).EQ.'E') EGU=EGU+USED(I)
  315 CONTINUE
      DO 325 I=1,IE
      EXCS(I)=0.0
      RFWB(I)=0.0
      OCTB(I)=0.0
```

```
      DO 316 K=1,3
  316 KRYO(I,K)=TOTR(I,K)
      IF(I.EQ.IEG) USED(I)=EGU
      IF(I.EQ.INYK) USED(I)=USED(I)+USED(IPEN)
      IF(I.EQ.IPHL) USED(I)=USED(I)+USED(ICAL)
      OCTB(I)=RTOT(I)-USED(I)
      BALN(I)=BANK(I)-USED(I)
      RFWB(I)=BALN(I)
      IF(BALN(I).GT.0.0) GO TO 325
      RFWB(I)=0.0
      BALN(I)=BALN(I)+TRANS(I)
      RFTR(I)=BALN(I)
      IF(BALN(I).GT.0.0) GO TO 325
      RFTR(I)=0.0
      DO 295 K=1,3
      BALN(I)=BALN(I)+TOTR(I,K)
      KRYO(I,K)=BALN(I)
      IF(BALN(I).GT.0.0) GO TO 325
      KRYO(I,K)=0.0
  295 CONTINUE
      EXCS(I)=BALN(I)*(-1.0)
  325 CONTINUE
C
C***** PRINT TABLE
C
      IT=5
      WRITE(6,7) RUNDATE,IT,IYR
    7 FORMAT(1H1////120X,A9////////27X,6HTABLE ,I2,'. ',I4,' BOISE
     1RIVER RESERVOIR STORAGE ACCOUNTS - OCTOBER 31.  (ACRE-FEET)'//)
      WRITE(6,9)
    9 FORMAT(1H ,35X,'BEGINNING  STORAGE  BALANCE    UNUSED    UNUSED  E
     1XCESS ARROWROCK  ANDERSON LUCKY PEAK'/    7X,13HNUMBER   USER,16X,
     2'  STORAGE     USED   OCT 31 TRANSFERS      BANK    USED    UNUSED
     3 CARRYOVER  CARRYOVER'/)
      DO 11 M=1,9
   11 STOT(M)=0.0
      DO 8 I=1,IE
      IF(IK(I).EQ.'S'.OR.IK(I).EQ.'E') GO TO 8
      IF(I.EQ.IPEN.OR.I.EQ.ICAL) GO TO 8
      STOT(1)=STOT(1)+RTOT(I)
      STOT(2)=STOT(2)+USED(I)
      STOT(3)=STOT(3)+OCTB(I)
      STOT(4)=STOT(4)+RFTR(I)
      STOT(5)=STOT(5)+RFWB(I)
      STOT(6)=STOT(6)+EXCS(I)
```

```
      STOT(7)=STOT(7)+KRYO(I,1)
      STOT(8)=STOT(8)+KRYO(I,2)
      STOT(9)=STOT(9)+KRYO(I,3)
      IF(RTOT(I).GT.0.0.OR.IPTAB(I).GT.0.OR.USED(I).GT.0.0)
     1WRITE(6,13) ID(I),UNAME(I),RTOT(I),USED(I),OCTB(I),RFTR(I),
     2RFWB(I),EXCS(I),(KRYO(I,M),M=1,3)
   13 FORMAT(7X,I8,1X,A18,F11.1,2F9.1,2F10.1,F8.1,2F10.1,F11.1)
    8 CONTINUE
      WRITE(6,14) (STOT(M),M=1,9)
   14 FORMAT(/16X,5HTOTAL,13X,F11.1,2F9.1,2F10.1,F8.1,2F10.1,F11.1)
C
C*******SUMMARIZE CARRYOVER FOR SYSTEM
C
      GRTOT=0.0
      DO 640 I=7,9
  640 GRTOT=GRTOT+STOT(I)
C     UNACCT=0.0
C     TYPE 20
C  20 FORMAT(' DO YOU WISH TO ENTER UNACCOUNTED FOR CARRYOVER? Y/N ' $)
C      ACCEPT 26, ANSWER
C      IF(ANSWER.NE.'Y') GO TO 44
C      TYPE *,' '
C   43 TYPE 41, UNACCT
C   41 FORMAT(1X,'UNACCOUNTED CARRYOVER = ',F10.1,' ENTER NEW VALUE?
C     1Y/N '$)
C      ACCEPT 26, ANSWER
C      IF(ANSWER.NE.'Y') GO TO 44
C      TYPE 45
C   45 FORMAT(1X,'ENTER UNACCOUNTED FOR CARRYOVER = '$)
C      ACCEPT 42, UNACCT
C   42 FORMAT(F10.0)
C      GO TO 43
   44 SGRTOT=GRTOT+UNACCT
      WRITE(6,642) RUNDATE
  642 FORMAT(1H1/////////////120X,A9)
      DO 56 K=1,3
   56 FILLATE(K)=0.0
      TYPE 46
   46 FORMAT('  DO YOU WISH TO ENTER LATE SEASON RESERVOIR FILL? Y/N '$)
      ACCEPT 26,ANSWER
      TYPE *, ' '
      IF(ANSWER.NE.'Y') GO TO 54
   58 DO 69 K=1,3
      PRINT 51, RNAME(K),FILLATE(K)
   51 FORMAT(1X,'LATE SEASON FILL FOR       ',A14,' = ',F10.1,' ENTER
     1NEW FILL? Y/N/Q '$)
      ACCEPT 26,ANSWER
      IF(ANSWER.EQ.'Q') GO TO 88
      IF(ANSWER.NE.'Y') GO TO 69
      PRINT 59,RNAME(K)
   59 FORMAT(1X,'ENTER LATE SEASON FILL FOR  ',A14,' = '$)
      ACCEPT 61,FILLATE(K)
   61 FORMAT(F10.0)
```

```
 69 CONTINUE
 88 TYPE 76
 76 FORMAT(' DO YOU WISH TO REENTER LATE SEASON RESERVOIR
   1FILL? Y/N '$)
    ACCEPT 26, ANSWER
    IF(ANSWER.EQ.'Y') GO TO 58
 54 WRITE(6,142) IYR
142 FORMAT(/////////25X, 'TABLE 6.  ',I4,' WATER DISTRICT 63
   1RESERVOIR TOTAL STORAGE - OCTOBER 31 (ACRE-FEET)'
   2///43X,'RESERVOIR',12X,'CARRYOVER   LATE FILL',7X,'TOTAL'/)
    TLATE=0.0
    GTFILL=0.0
    STOT(10)=UNACCT
    DO 146 K=1,4
    IF(K.EQ.6) GO TO 146
    K1=K+6
    TFILL(K)=STOT(K1)+FILLATE(K)
    TLATE=TLATE+FILLATE(K)
    GTFILL=GTFILL+TFILL(K)
    WRITE(6,144) RNAME(K),STOT(K1),FILLATE(K),TFILL(K)
144 FORMAT(43X,A14,4X,3F12.1)
146 CONTINUE
    WRITE(6,73) SGRTOT,TLATE,GTFILL
 73 FORMAT(/43X,'TOTAL',13X,3F12.1)
 38 PRINT 91
 91 FORMAT('  DO YOU WANT TO CREATE A FILE OF THE CARRYOVER'/' BY
   1CANAL AND RESERVOIR? Y/N:  '$)
    ACCEPT 26, FILE
    IF(FILE.NE.'Y') GO TO 15
    OPEN (UNIT=10,NAME='CARRYOUT',TYPE='NEW',CARRIAGECONTROL='LIST')
    DO 37 I=1,IE
 37 WRITE(10,34) ID(I),UNAME(I),(KRYO(I,K),K=1,3)
 34 FORMAT(I8,1X,A18,6F10.1)
    GO TO 15
400 TYPE 401
401 FORMAT(' ERROR ENCOUNTERED READING STORAGE USED DATA')
 15 STOP
    END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	4974	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	2525	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	17808	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	25307	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOISTO$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
2-00004428	R*4	ANSWER		403=	405	410=	411	412	421=	422	
**	R*8	CSUBT		18	237=	242(2)=	248	274=	283(2)=	292	
**	R*4	DIFF		136=	137	159=	160				
**	R*4	EGU		300=	313(2)=	321					
**	R*4	EXCESS		202=	212(2)=	216	217				
2-0000442C	R*4	FILE		444=	445						
2-000043B8	R*8	FILLLPL		20	170=	171(2)=	172	182	223	224	225
2-000043F4	R*4	FL		142=	144	145	165=	166	167		
**	R*4	GRTOT		377=	379(2)=	396					
**	R*4	GTFILL		428=	435(2)=	439					
**	R*8	GTOT		19	273=	286(2)=	292				
**	I*4	I		29=	31(4)	33(2)	34(2)	35(2)	36(2)	37(2)	38(2)
				39	40(2)=	43	44=	45(4)	47(2)	48(2)	113=
				114	115	116	118	119(3)	120	122	123
				124(4)	126	146=	147	183=	184	197=	198
				208=	209(3)	210(3)	211(2)	212(2)	213(2)	214(3)	238=
				240(3)	241	242	243(12)	255=	257(2)	277=	278
				280	281(3)	282(4)	283	284	285	286	287(11)
				301=	308	309	312	313(2)	315=	316	317
				318	320(2)	321(2)	322(3)	323(3)	324(3)	325(3)	326(2)
				327	328	329(3)	330(2)	331	332	334(3)	335(2)
				336	337	339(2)	355=	356(2)	357(2)	358	359
				360	361	362	363	364	365	366	367(12)
				378=	379	447=	448(3)				
2-000043DC	I*4	ICAL		37=	323	357					
**	I*4	IE		43=	44	113	146	183	197	208	238
				255	277	301	315	355	447		
2-000043E0	I*4	IEG		38=	321						
**	I*4	IN2		30=							
2-000043D4	I*4	INYK		35=	322						
2-000043D0	I*4	IPEN		34=	322	357					
2-000043D8	I*4	IPHL		36=	323						

**	I*4	IT	72=	78	101=	104	229=	230	264=	265
			298=	344=	345					
2-000043CC	I*4	IUN	33=	168(2)	223(2)	224(2)				
2-000043EC	I*4	IYR	64=	78	104	230	265	345	423	
2-000043E8	I*4	J	45(2)=	201=	203(2)=	204	206			
2-00004418	I*4	JD	306=	308						
**	I*4	JDI	299=	302	303=	310=				
2-000043C8	I*4	K	31(2)=	66(5)=	73=	74(3)	75	76(4)	84=	85
			86=	87	88	89	90	91(5)	94(2)=	102=
			103	110=	111	112	117=	118(3)	119	120
			124(2)=	129(2)=	194=	195(5)	198(3)	199(3)	200	204
			209(4)	210(3)	211(2)	212(2)	213(2)	214(3)	256=	257(3)
			275=	276	319=	320(2)	333=	334	335	337
			399=	400	406=	407(2)	413	415	430=	431
			432	433(2)	434	435	436(3)	448(2)=		
**	I*4	K1	432=	433	436					
**	I*4	L	28=							
**	I*4	M	239=	240	241(3)	279=	280(3)	281	353=	354
			367(2)=	372(2)=						
2-0000439C	CHAR	RUNDATE	15	21A	78	104	230	265	345	397
**	R*4	SGRTOT	396=	439						
2-0000441C	R*4	T1	306=	312						
**	R*8	TBANK	18	271=	285(2)=	292				
**	R*4	TLATE	427=	434(2)=	439					
**	R*8	TTRANS	20	272=	284(2)=	292				
2-000043A5	CHAR	U	16	45=						
2-000043C0	R*8	UNACCT	20	53=	396	429				
**	R*4	WAT	200=	206	209	216=				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00003860	R*4	BALN		320	(80)	9	325=	326	327	329(2)=
						330	331	334(2)=	335	336
						339				
2-00002BE0	R*8	BANK		640	(80)	12	18	45=	282	285
						287	325			
2-00002BA8	R*8	CNSTO		24	(3)	11	20	199=	200	225(2)=
						248(2)				
2-00000818	R*8	CRYO		1920	(80, 3)	8	18	45=	114=	147=
						184=	198	210	214	243(2)
2-00000000	R*8	CTOT		80	(10)	7	18	103=	118(2)=	122(2)=
						129	169(2)=	170	195(2)	209
2-00002178	R*8	CTOTC		24	(3)	11	18	112=	198(2)=	199
						248(2)				
2-00002190	R*8	CTOTR		24	(3)	11	18	111=	241(2)=	248(3)
						276=	280(2)=	292(3)		
2-00003AE0	R*4	EVAP		16	(4)	10	66=	74	89	91
2-000034A0	R*4	EXCS		320	(80)	8	316=	339=	363	367
2-00001998	R*8	FILL		32	(4)	10	19	66=	74	76(2)

```
                                                                  88        91        136       159       170
                                                                172(2)=   195(2)    199

  2-00003D7C  R*4   FILLATE                  32  (8)              14       400=      407       415=      433
                                                                434       436
  2-000030E0  I*4   ID                      320  (80)              7        31=       33        34        35
                                                                 36        37        38        39        47
                                                                 48       124       243       287       308
                                                                367       448
  2-00003220  I*4   ID2                     320  (80)              7        45=       47        48
  2-000039A0  I*4   IK                      320  (80)              9        31=      313       356(2)
  2-00003C3C  I*4   IPTAB                   320  (80)             14       116=      126=      243       287
                                                                367

  2-00000F98  R*8   KRYO                   1920  (80, 3)           9        19       320=      335=      337=
                                                                364       365       366       367       448
  2-00002428  R*8   NSTO                   1920  (80, 3)          11        20       120=      209(2)=   210
                                                                214=      223(2)=   243(2)
  2-00003720  R*4   OCTB                    320  (80)              8       318=      324=      360       367
  2-00003AF0  R*4   PAVAIL                   12  (3)              13        75=       76=      257
  2-00003AFC  R*4   RFTR                    320  (80)             13       330=      332=      361       367

  2-000035E0  R*4   RFWB                    320  (80)              8       317=      326=      328=      362
                                                                367
  2-0000435C  CHAR  RNAME                    64  (4)              12        17        66=       91       195
                                                                204       407       413       436
  2-00001718  R*8   RTOT                    640  (80)              9        19       115=      119(2)=   122
                                                                123       124       243       282=      286
                                                                287(2)    324       358       367(2)
  2-000019B8  R*8   SPACE                    32  (4)              10        19        66=       87        91
                                                                136       159
  2-00000050  R*8   STOR                   1920  (80, 3)           7        18        31=      118       119
                                                                124       168(2)=   209       211       212
                                                                213       214

  2-000007D0  R*8   STOT                     72  (9)               7        18       354=      358(2)=   359(2)=
                                                                360(2)=   361(2)=   362(2)=   363(2)=   364(2)=
                                                                365(2)=   366(2)=   372       379       429=
                                                                433       436
  2-000021A8  R*8   SUBT                    640  (80)             11        18       240(2)=   242       243
                                                                278=      281(2)=   282       283       287
  2-00002BC0  R*8   T                        32  (4)              12        19        85=       87(2)=    88(2)=
                                                                 89(2)=    90(2)=    94
  2-00003D9C  R*4   TFILL                    32  (8)              14       433=      435       436
  2-000019F8  R*8   TOTR                   1920  (80, 3)          10        19       210=      211       212
                                                                213=      224(2)=   240       241       243(3)
                                                                257(2)=   280       281       287(3)    320
                                                                334

  2-00002E60  R*8   TRANS                   640  (80)             13        20        45=      282       284
                                                                287       329
  2-00003DBC  CHAR  UNAME                  1440  (80)              7        16        31=      124       243
                                                                287       367       448
  2-00003360  R*4   USED                    320  (80)              8       309=      312=      313       321=
                                                                322(3)=   323(3)=   324       325       359
                                                                367(2)
  2-000019D8  R*8   YIELD                    32  (4)              10        19        74=       76        90
```

LABELS

Address	Label	References						
**	1	194	218#					
1-00000018	2'	48	49#					
**	3	110	112#					
0-000008CB	4	208	211	215#				
0-0000023E	5	45	55	59#				
0-00000804	6	202#	217					
1-00000531	7'	345	346#					
0-00000FAF	8	355	356	357	371#			
1-0000059D	9'	348	349#					
1-00000000	10'	31	32#					
**	11	353	354#					
1-0000000C	12'	45	46#					
1-00000668	13'	367	370#					
1-00000689	14'	372	373#					
0-00001347	15	445	450	453#				
0-00000D00	16	306	309#					
**	17	313#						
0-00000696	18	137	144	148#				
**	19	146	147#					
1-00000207	21'	129	130#					
1-00000437	22'	248	250#					
**	23	183	184#					
0-00000779	24	160	166	186#				
0-00000154	25	31	39	42#				
1-00000272	26'	142	143#	165	403	410	421	444
1-00000275	27'	162	163#					
0-000006B0	28	161#	167					
**	29	197	198#					
0-00000078	30	31#	41					
0-00000608	31	138#	145					
1-00000219	32'	139	140#					
1-000008B2	34'	448	449#					
1-00000517	35'	292	294#					
**	37	447	448#					
**	38	441#						
0-00000214	40	44	47	52#				
**	44	396#						
1-000006C3	46'	401	402#					
1-0000008C	49'	64	65#					
1-000006FE	51'	407	408#					
0-000011AE	54	405	423#					
1-0000008F	55'	66	67#					

```
     **      56                    399       400#
  0-0000109C 58                    406#      422
  1-00000742 59'                   413       414#

     **      60                    117       121#
  1-0000076B 61'                   415       416#
     **      65                    275       276#
  0-00001163 69                    406       412       417#
  1-00000429 70'                   243       246#

  1-000001F9 71'                   124       125#
  1-0000084A 73'                   439       440#
  1-00000509 75'                   287       290#
  1-0000076F 76'                   418       419#
     **      80                    113       128#

     **      81                    239       241#
     **      82                    238       247#
     **      83                    279       281#
     **      85                    277       291#
  0-00001169 88                    411       418#

     **      90                     73        77#
  1-0000085C 91'                   441       442#
  1-00000383 95'                   233       234#
  1-000001B3 97'                   107       108#
  1-00000099 100'                   78        79#

  1-00000104 101'                   81        82#
     **      105                    84        85#
     **      110                    86        93#
  1-0000013C 120'                   91        92#
  1-00000146 130'                   94        95#

  1-000007AB 142'                  423       424#
  1-0000083E 144'                  436       437#
     **      145                   230#
  0-00001258 146                   430       431       438#
  1-00000324 150'                  230       231#

     **      155                   102       103#
  1-00000158 160'                  104       105#
  1-00000449 161'                  265       266#
     **      165                   255       258#
  1-000004A4 168'                  268       269#

     **      175                   256       257#
     **      200                   251#
  1-00000529 260'                  306       307#
  0-00000CD8 262                   304       308#
     **      295                   333       338#

  1-0000006B 305'                   53        54#
  1-00000071 306'                   56        57#
  0-00001350 310                    53        56#
  0-00000D08 315                   301       311       314#
     **      316                   319       320#
```

```
  0-00000CE4  320                          308      312#
  0-00000E0D  325                          315      327      331      336      340#
  0-00000D10  400                          306      451#
  1-000008BE  401'                         451      452#
      **      640                          378      379#

  1-000006AE  642'                         397      398#
  1-000002CA  997'                         195      196#
  1-000002F4  998'                         204      205#
  1-0000031C  999'                         206      207#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                              References

        ASSIGN                              22       23       24       25       26       27
        DATE                                21
```

```
  +-----------------------------------------+
  |          KEY TO REFERENCE FLAGS         |
  |    =   - Value Modified                 |
  |    #   - Defining Reference             |
  |    A   - Actual Argument, possibly modified |
  |    D   - Data Initialization            |
  |   (n)  - Number of occurrences on line  |
  +-----------------------------------------+
```

COMMAND QUALIFIERS

```
  FORTRAN/LIST/CROSS_REFERENCE BOISTO

  /ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
  /BLAS=(INLINE,MAPPED)
  /CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
  /DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
  /DIRECTIVES=(DEPENDENCE)
  /MATH_LIBRARY=(ACCURATE,NOV5)
  /PARALLEL=(NOAUTOMATIC,NOMANUAL)
  /SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
             NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
  /CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
  /NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=HYDRO:[BSUTTER.BOISE]BOISTO.LIS;210
  /OBJECT=HYDRO:[BSUTTER.BOISE]BOISTO.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

Run Time:	1.53 seconds
Elapsed Time:	2.66 seconds
Page Faults:	3117
Dynamic Memory:	1856 pages

BOTCHGSS 115

```
C*************************************************************************
C
C        PROGRAM TO CHANGE VALUES IN DATA SET BOISTO.SPA
C       THESE VALUES ARE THE STORAGE SPACE ENTITLMENTS IN ANDERSON RANCH,
C       ARROWROCK, AND LUCKY PEAK RESERVOIRS.  RJS - AUGUST 1987.
C
C*************************************************************************

      DIMENSION SPACE(3),RNAME(3)
      CHARACTER*18 DNAME
      CHARACTER*10 RNAME
      DATA RNAME/'ARROWROCK ','ANDERSON R',
     1'LUCKY PEAK'/
      DATA BLANK/' '/
      CALL ASSIGN (1,'BOISTO.SPA')
      OPEN (UNIT=2,NAME='BOISTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE SPACE ENTITLEMENTS
     1  '
      PRINT *,' '
      IANS=BLANK
    9 READ(1,2,END=6) ID,IK,DNAME,(SPACE(L),L=1,3)
    2 FORMAT(I8,A1,A18,3F10.0)
      IF(IANS.EQ.'Q') GO TO 7
      IF(IK.EQ.'S'.OR.IK.EQ.'E') GO TO 7
      DO 101 I=1,3
      TYPE 11, DNAME,RNAME(I),SPACE(I)
   11 FORMAT(' ',A18,' SPACE IN ',A10,' =',F10.1,' NEW VALUE? Y/N/Q
     1 '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 100
      PRINT 103
  103 FORMAT('                    ENTER NEW VALUE     ='$)
      ACCEPT 4, SPACE(I)
    4 FORMAT(F10.0)
  100 IF(IANS.EQ.'Q') GO TO 7
  101 CONTINUE
    7 WRITE(2,8) ID,IK,DNAME,(SPACE(L),L=1,3)
    8 FORMAT(I8,A1,A18,3F10.1)
      GO TO 9
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	511	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	210	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	168	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	889	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGSS$MAIN	

VARIABLES

Address	Type	Name	Attributes	References				
2-0000003C	R*4	BLANK		14D	21			
2-0000002A	CHAR	DNAME		10	22=	27	39	
**	I*4	I		26=	27(2)	35		
2-00000040	I*4	IANS		21=	24	30=	32	37
2-00000044	I*4	ID		22=	39			
2-00000048	I*4	IK		22=	25(2)	39		
2-0000004C	I*4	L		22(2)=	39(2)=			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-0000000C	CHAR	RNAME		30	(3)	9	11	12D	27	
2-00000000	R*4	SPACE		12	(3)	9	22=	27	35=	39

LABELS

Address	Label	References			
1-00000056	2'	22	23#		
1-000000C2	4'	35	36#		
0-000001F8	6	22	42#		
0-000001A9	7	24	25	37	39#
1-000000C6	8'	39	40#		
0-00000074	9	22#	41		
1-00000062	11'	27	28#		
0-00000199	100	32	37#		
**	101	26	38#		
1-00000092	102'	30	31#		
1-00000095	103'	33	34#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	15
	FOR$OPEN	16

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGSS

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
Run Time:           3.19 seconds
Elapsed Time:       25.03 seconds
Page Faults:        326
Dynamic Memory:     355 pages
```

ROTOHASC. LIS

```
C*********************************************************************
C
C        PROGRAM TO CHANGE VALUES IN DATA SET BOISTO.CRY
C       THESE VALUES ARE CARRYOVER STORAGE ENTITLMENTS IN ANDERSON RANCH,
C       ARROWROCK, AND LUCKY PEAK RESERVOIRS.  RJS - AUGUST 1987.
C
C*********************************************************************

      DIMENSION CRYO(3),RNAME(3)
      CHARACTER*18 DNAME
      CHARACTER*10 RNAME
      DATA RNAME/'ARROWROCK ','ANDERSON R',
     1'LUCKY PEAK'/
      DATA BLANK/' '/
      CALL ASSIGN (1,'BOISTO.CRY')
      OPEN (UNIT=2,NAME='BOISTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE CARRYOVER VALUES
     1  '
      PRINT *,' '
      IANS=BLANK
    9 READ(1,2,END=6) ID,IK,DNAME,(CRYO(L),L=1,3),TRANS,BANK
    2 FORMAT(I8,A1,A18,5F10.0)
      IF(IANS.EQ.'Q') GO TO 7
      IF(IK.EQ.'S'.OR.IK.EQ.'E') GO TO 7
      DO 101 I=1,3
      TYPE 11, DNAME,RNAME(I),CRYO(I)
   11 FORMAT(' ',A18,' CARRYOVER IN ',A10,' =',F10.1,' NEW VALUE? Y/N/Q
     1 '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 100
      PRINT 103
  103 FORMAT('                          ENTER NEW VALUE   ='$)
      ACCEPT 4, CRYO(I)
    4 FORMAT(F10.0)
  100 IF(IANS.EQ.'Q') GO TO 7
  101 CONTINUE
    7 WRITE(2,8) ID,IK,DNAME,(CRYO(L),L=1,3),TRANS,BANK
    8 FORMAT(I8,A1,A18,5F10.1)
      GO TO 9
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	551	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	215	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	176	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	942	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGSC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References				
2-00000054	R*4	BANK		22=	39			
2-0000003C	R*4	BLANK		14D	21			
2-0000002A	CHAR	DNAME		10	22=	27	39	
**	I*4	I		26=	27(2)	35		
2-00000040	I*4	IANS		21=	24	30=	32	37
2-00000044	I*4	ID		22=	39			
2-00000048	I*4	IK		22=	25(2)	39		
2-0000004C	I*4	L		22(2)=	39(2)=			
2-00000050	R*4	TRANS		22=	39			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000000	R*4	CRYO		12	(3)	9	22=	27	35=	39
2-0000000C	CHAR	RNAME		30	(3)	9	11	12D	27	

LABELS

Address	Label	References			
1-00000053	2'	22	23#		
1-000000C7	4'	35	36#		
0-00000220	6	22	42#		
0-000001BD	7	24	25	37	39#
1-000000CB	8'	39	40#		
0-00000074	9	22#	41		
1-0000005F	11'	27	28#		
0-000001AD	100	32	37#		
**	101	26	38#		
1-00000093	102'	30	31#		

```
  1-00000096  103'                              33        34#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	15
	FOR$OPEN	16

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
  FORTRAN/LIST/CROSS_REFERENCE BOICHGSC

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
  Run Time:           3.31 seconds
  Elapsed Time:       8.88 seconds
  Page Faults:        332
  Dynamic Memory:     355 pages
```

BOTCHGSIL.LIS

```
C*************************************************************************
C
C       PROGRAM TO CHANGE VALUES IN DATA SET BOISTO.USE
C       THESE VALUES ARE STORAGE USE AND ADJUSTMENTS FOR
C       ALL ENTITIES HAVING SPACE IN BOISE RIVER RESERVOIRS.
C       RJS - AUGUST 1987.
C
C       NOTE : THIS PROGRAM WAS ORIGINALLY WRITTEN TO PASS STORAGE WATER
C              REMAINING (VALU(2)) TO NEW FILE, AND TO INCLUDE AN
C              ADJUSTMENT TO USE (VALU(3)).  FOR THE TIME BEING, THEY HAVE
C              REMOVED BECAUSE THEY ARE NOT USED.    RJS FEBRUARY, 1988.
C
C*************************************************************************

      DIMENSION VALU1(3),VALU2(3),UNAME(3)
      CHARACTER*18 DNAME
      CHARACTER*10 UNAME
      DATA UNAME/'TOTAL USE ','          ','ADJUSTMENT'/
      DATA BLANK/' '/
      CALL ASSIGN (1,'BOISTO.USE')
      CALL ASSIGN (3,'BOISTO.SPA')
      OPEN (UNIT=2,NAME='BOISTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE STORAGE USE VALUES
     1  '
      PRINT *,' '
      IRD=0
      IEU=0
      IANS=BLANK
    9 READ(3,2,END=6) ID,IK,DNAME
    2 FORMAT(I8,A1,A18)
   18 IF(IRD.EQ.0) READ(1,3,END=13) ID2,CODE,IDATE,(VALU2(I),I=1,1)
    3 FORMAT(I8,A1,I7,F10.0)
      IRD=0
      IF(ID2.GE.ID) GO TO 5
      TYPE 12, ID2
   12 FORMAT('  DIVERSION ',I8,'FROM BOISTO.USE NOT FOUND IN BOISTO.SPA
     1   ')
      GO TO 18
    5 IF(ID2.EQ.ID) GO TO 17
      IRD=1
      GO TO 13
   17 DO 16 I=1,1
   16 VALU1(I)=VALU2(I)
   13 IF(IK.EQ.'S') GO TO 14
      IF(IANS.EQ.'Q') GO TO 7
      DO 101 I=1,1
      TYPE 11, DNAME,UNAME(I),VALU1(I)
   11 FORMAT(' ',A18,' STORED WATER ',A10,' =',F10.1,' NEW VALUE? Y/N/Q
     1 '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 100
      PRINT 103
  103 FORMAT('                          ENTER NEW VALUE   ='$)
      ACCEPT 4, VALU1(I)
    4 FORMAT(F10.0)
```

```
0058      100 IF(IANS.EQ.'Q') GO TO 7
0059      101 CONTINUE
0060        7 WRITE(2,8) ID,CODE,IDATE,(VALU1(L),L=1,1)
0061        8 FORMAT(I8,A1,I7,F10.1)
0062       14 DO 15 I=1,3
0063       15 VALU1(I)=0.0
0064          GO TO 9
0065        6 STOP
0066          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	659	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	296	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	212	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1167	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGSU$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
2-00000048	R*4	BLANK		19D	29						
2-0000005C	R*4	CODE		32=	60						
2-00000036	CHAR	DNAME		16	30=	48					
2-00000064	I*4	I		32(2)=	43=	44(2)	47=	48(2)	56	62=	63
2-0000004C	I*4	IANS		29=	46	51=	53	58			
2-00000050	I*4	ID		30=	35	40	60				
2-00000058	I*4	ID2		32=	35	36	40				
2-00000060	I*4	IDATE		32=	60						
**	I*4	IEU		28=							
2-00000054	I*4	IK		30=	45						
**	I*4	IRD		27=	32	34=	41=				
**	I*4	L		60(2)=							

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000018	CHAR	UNAME		30	(3)	15	17	18D	48	
2-00000000	R*4	VALU1		12	(3)	15	44=	48	56=	60
								63=		
2-0000000C	R*4	VALU2		12	(3)	15	32=	44		

LABELS

Address	Label	References		
1-00000064	2'	30	31#	
1-0000006B	3'	32	33#	
1-0000011A	4'	56	57#	
0-00000138	5	35	40#	
0-0000028C	6	30	65#	
0-00000230	7	46	58	60#
1-0000011E	8'	60	61#	
0-0000007C	9	30#	64	
1-000000B2	11'	48	49#	
1-00000075	12'	36	37#	
0-00000154	13	32	42	45#
0-00000278	14	45	62#	
**	15	62	63#	
**	16	43	44#	
0-00000144	17	40	43#	
**	18	32#	39	
0-00000220	100	53	58#	
**	101	47	59#	
1-000000E6	102'	51	52#	
1-000000E9	103'	54	55#	

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	20	21
	FOR$OPEN	22	

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

```
COMMAND QUALIFIERS

  FORTRAN/LIST/CROSS_REFERENCE BOICHGSU

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS


COMPILATION STATISTICS

  Run Time:           4.24 seconds
  Elapsed Time:       54.09 seconds
  Page Faults:        432
  Dynamic Memory:     399 pages
```

```
C*************************************************************************
C
C        PROGRAM TO CHANGE VALUES STORAGE TRANSFERS IN DATA SET BOISTO.CRY.
C       THESE VALUES ARE AMOUNTS OF STORED WATER THAT ONE USER TRANSFERS
C       TO ANOTHER USER FOR WHATEVER REASON.  ENTER POSITIVE VALUES FOR
C       THOSE RECEIVING WATER, AND NEGATIVE VALUES FOR THOSE SUPPLYING
C       WATER.   RJS - FEBRUARY, 1988.
C
C*************************************************************************

      DIMENSION RNAME(4),CRYO(3)
      CHARACTER*18 DNAME
      DATA BLANK/' '/
      CALL ASSIGN (1,'BOISTO.CRY')
      OPEN (UNIT=2,NAME='BOISTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE TRANSFER VALUES: '
      PRINT *,' ENTER POSITIVE VALUES FOR THOSE RECEIVING STORAGE,    '
      PRINT *,' AND NEGATIVE VALUES FOR THOSE SUPPLYING STORAGE.      '
      PRINT *,' '
      IANS=BLANK
    9 READ(1,2,END=6) ID,IK,DNAME,(CRYO(L),L=1,3),TRANS,BANK
    2 FORMAT(I8,A1,A18,5F10.0)
      IF(IANS.EQ.'Q') GO TO 7
      IF(IK.EQ.'S'.OR.IK.EQ.'E') GO TO 7
      TYPE 11, DNAME,TRANS
   11 FORMAT(' ',A18,' STORAGE TRANSFER  =',F10.1,' NEW VALUE? Y/N/Q
     1 '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 100
      PRINT 103
  103 FORMAT('                    ENTER NEW VALUE  ='$)
      ACCEPT 4, TRANS
    4 FORMAT(F10.0)
  100 IF(IANS.EQ.'Q') GO TO 7
    7 WRITE(2,8) ID,IK,DNAME,(CRYO(L),L=1,3),TRANS,BANK
    8 FORMAT(I8,A1,A18,5F10.1)
      GO TO 9
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	563	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	318	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	172	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1053	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGST$MAIN	

VARIABLES

Address	Type	Name	Attributes	References				
2-00000048	R*4	BANK		22=	37			
2-00000030	R*4	BLANK		13D	21			
2-0000001C	CHAR	DNAME		12	22=	26	37	
2-00000034	I*4	IANS		21=	24	29=	31	36
2-00000038	I*4	ID		22=	37			
2-0000003C	I*4	IK		22=	25(2)	37		
2-00000040	I*4	L		22(2)=	37(2)=			
2-00000044	R*4	TRANS		22=	26	34=	37	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
2-00000010	R*4	CRYO		12	(3)	11	22=	37
2-00000000	R*4	RNAME		16	(4)	11		

LABELS

Address	Label	References			
1-000000C0	2'	22	23#		
1-0000012E	4'	34	35#		
0-0000022C	6	22	40#		
0-000001CB	7	24	25	36	37#
1-00000132	8'	37	38#		
0-000000A8	9	22#	39		
1-000000CC	11'	26	27#		
0-000001C1	100	31	36#		
1-00000100	102'	29	30#		
1-00000103	103'	32	33#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	14
	FOR$OPEN	15

```
+-----------------------------------------+
|          KEY TO REFERENCE FLAGS         |
|   =   - Value Modified                  |
|   #   - Defining Reference              |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization             |
|  (n)  - Number of occurrences on line   |
+-----------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGST

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
Run Time:           3.21 seconds
Elapsed Time:       10.48 seconds
Page Faults:        322
Dynamic Memory:     355 pages
```

BOTCHGST.LIS

```
C*********************************************************************
C
C        PROGRAM TO CHANGE VALUES IN DATA SET BOISTO.IND.
C       THESE VALUES ARE THE INITIAL RESERVOIR FILL AND
C       EITHER ESTIMATED OR ACTUAL EVAPORATION. RJS - AUUGUST 1987.
C
C*********************************************************************

      DIMENSION ID(3),RNAME(3),SPACE(3),FILL(3),EVAP(3)
      CHARACTER*16 RNAME
      CALL ASSIGN (1,'BOISTO.IND')
      OPEN (UNIT=2,NAME='BOISTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      READ(1,2,END=6) (ID(K),RNAME(K),SPACE(K),FILL(K),EVAP(K),K=1,3)
    2 FORMAT(I8,1X,A16,3F10.0)
      PRINT *,' '
      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR FILL VALU
     1ES  '
      PRINT *,' '
      DO 101 I=1,3
      TYPE 11, RNAME(I),FILL(I)
   11 FORMAT(' ',A16,' =',F10.1,' NEW VALUE? Y/N '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 101
      PRINT 103
  103 FORMAT(' ENTER NEW VALUE   ='$)
      ACCEPT 4, FILL(I)
    4 FORMAT(F10.0)
  101 CONTINUE
C      PRINT *,'                                                        '
C      PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR EVAPORATI
C     1ON VALUES '
C      PRINT *,' '
C      DO 104 I=1,3
C      TYPE 11, RNAME(I),EVAP(I)
C      ACCEPT 102, IANS
C      IF(IANS.NE.'Y') GO TO 104
C      PRINT 103
C      ACCEPT 4, EVAP(I)
C  104 CONTINUE
      WRITE(2,7) (ID(K),RNAME(K),SPACE(K),FILL(K),EVAP(K),K=1,3)
    7 FORMAT(I8,1X,A16,3F10.1)
      PRINT *,' '
    6 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	512	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	175	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	184	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	871	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGSI$MAIN	

VARIABLES

Address	Type	Name	Attributes	References		
**	I*4	I		19=	20(2)	27
2-00000064	I*4	IANS		22=	24	
2-00000060	I*4	K		13(6)=	41(6)=	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000024	R*4	EVAP		12	(3)	9	13=	41		
2-00000018	R*4	FILL		12	(3)	9	13=	20	27=	41
2-00000000	I*4	ID		12	(3)	9	13=	41		
2-00000030	CHAR	RNAME		48	(3)	9	10	13=	20	41
2-0000000C	R*4	SPACE		12	(3)	9	13=	41		

LABELS

Address	Label	References		
1-00000058	2'	13	14#	
1-0000009F	4'	27	28#	
0-000001F9	6	13	44#	
1-000000A3	7'	41	42#	
1-00000064	11'	20	21#	
0-00000177	101	19	24	29#
1-00000084	102'	22	23#	
1-00000087	103'	25	26#	

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	11
	FOR$OPEN	12

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGSI

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
Run Time:           2.98 seconds
Elapsed Time:       4.84 seconds
Page Faults:        624
Dynamic Memory:     350 pages
```

ROTHSTPUB.LIS

```
CC      ****************************************************************
C
C             PROGRAM BOIWRAPUR - WATER RIGHT ACCOUNTING PURGE
C
C        THIS PROGRAM SELECTS FROM THE HISTORY FILE (BOIWRA.HST) THE
C        DATA NEEDED FOR THE WATER RIGHT ACCOUNTING FOR THE NEW YEAR.
C        THE DATA ARE BROKEN INTO TWO SEPARATE FILES, ONE FROM OCTOBER 1
C        OF THE PREVIOUS YEAR TO OCTOBER 31 OF THE CURRENT YEAR; AND ONE
C        FROM OCTOBER 1 OF THE CURRENT YEAR.  THE PREVIOUS YEAR FILE IS
C        NAMED BOIAYR.HST (WHICH SHOULD LATER BE RENAMED WITH YR EQUAL TO
C        THE PROPER YEAR) AND THE NEW FILE IS BOIWRA.PUR (WHICH WILL
C        LATER BE RENAMED TO BOIWRA.HST).            HSN
C
C                       BOB SUTTER  JULY 31,1984
C
C      ***************************************************************
C
C
         DIMENSION JD(12),IYD(3),A(5)
         DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
         CALL ASSIGN(1,'HISTORY')
         OPEN (UNIT=2,NAME='OLDHIST',TYPE='NEW',CARRIAGECONTROL='LIST')
         OPEN (UNIT=3,NAME='NEWHIST',TYPE='NEW',CARRIAGECONTROL='LIST')
C
         PRINT 1
       1 FORMAT (' ')
         PRINT *,' ******************************************************'
         PRINT *,' '
         PRINT *,'  THIS PROCEDURE CREATES TWO FILES FROM THE HISTORY '
         PRINT *,'  FILE, ONE WITH DATA FROM OCTOBER 1 OF THE PREVIOUS '
         PRINT *,'  YEAR TO OCTOBER 31 OF THE CURRENT YEAR, AND ONE FROM '
         PRINT *,'  OCTOBER 1 OF THE CURRENT YEAR.  THE LATTER FILE '
         PRINT *,'  WILL BECOME THE NEW HISTORY FILE.'
         PRINT *,' '
         PRINT *,' ******************************************************'
         PRINT 1
C
       5 PRINT *,'  ONE MORE TIME - PLEASE ENTER THE'
         PRINT *,'  IRRIGATION YEAR FOR WHICH YOU ARE CREATING'
         PRINT *,'  AN ARCHIVE FILE.  THIS IS USUALLY LAST YEAR.   '
         PRINT *,'  FOR EXAMPLE : 1989  '
         PRINT 1
         PRINT *,'                ????  '
         PRINT 35
      35 FORMAT('+                '$)
         ACCEPT 2, IY2
       2 FORMAT(I4)
         PRINT 1
         PRINT 3, IY2
       3 FORMAT (1X,' THE ARCHIVE ACCOUNTING YEAR IS ',I4/
        !   2X,'IS THIS CORRECT?  ANSWER Y/N/C.  ')
         PRINT 36
      36 FORMAT(28X,' '$)
         ACCEPT 4,IANSWR
       4 FORMAT (1A1)
         PRINT 1
         IF (IANSWR.EQ.'C') GO TO 10
```

```
      IF (IANSWR.EQ.'N') GO TO 5
C
      IY1=IY2-1
      ID1=1
      ID2=31
      IM1=10
      IM2=10
    8 CONTINUE
C
C***** CONVERT BEGINNING AND ENDING DATES TO SEVEN DIGIT DATE YYYYDDD,
C***** WHERE YYYY IS THE YEAR AND DDD IS THE NUMERICAL DAY OF THE YEAR.
C***** ALSO COMPUTE A SIMILAR DATE FOR THE DAY THREE DAYS PRIOR TO THE
C***** FIRST ACCOUNTING DAY SO THAT FOUR DAY AVERAGES CAN BE COMPUTED
C***** FOR CERTAIN REACH GAINS. THEN PRINT RESULTS.
C
      IM=IM1
      ID=ID1
      IY=IY1
      DO 13 K=1,2
      FEB=IY/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      NDY=0
      DO 11 I=1,IM
      IF(I.EQ.IM) GO TO 12
      IF(I.EQ.2.AND.FEB.LT.0.001) JD(2)=29
   11 NDY=NDY+JD(I)
   12 JD(2)=28
      NDY=NDY+ID
      IYD(K)=(IY*1000)+NDY
      IF(K.EQ.2) GO TO 13
      IM=IM2
      ID=ID2
      IY=IY2
   13 CONTINUE
      IY=IYD(1)/1000
      IF(IY1.EQ.IY) GO TO 15
      FEB=IY/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      J=635
      IF(FEB.LT.0.001) J=634
      IYD(1)=IYD(1)-J
   15 CONTINUE
      IYD(3)=IYD(2)-30
      PRINT 16, IYD(1),IYD(2),IYD(3)
   16 FORMAT(3X,'ALL DATA FROM DAY ',I8,' TO DAY ', I8,' IS BEING RETRIE
     !VED FROM'/3X,'THE HISTORY FILE TO CREATE A HISTORY ARCHIVE FILE,'/
     !3X,'AND ALL DATA FROM DAY ',I8,' IS BEING SELECTED FOR THE NEW  '/
     !3X,'(PURGED) HISTORY FILE')
      PRINT *,'  SO PLEASE HANG ON FOR A WHILE. '
      PRINT 1
C
C***** SELECT ALL DATA FROM HISTORY FILE FROM DATE IYD(1) THROUGH IYD(2)
C***** AND WRITE DATA TO THE ARCHIVE FILE.  ALSO SELECT ALL DATA FROM
C***** HISTORY FILE FROM DATE IYD(3) FOR NEW (PURGED) HISTORY FILE.
C
```

```
      IARCH=0
      IPHST=0
      IERR=0
   19 READ(1,17,END=18,ERR=24) (A(I),I=1,3),IDATE,(A(I),I=4,5)
   17 FORMAT(2A4,A1,I7,A4,A3)
      IF(IDATE.LT.IYD(1)) GO TO 19
      IF(IDATE.GT.IYD(2)) GO TO 26
      WRITE(2,17) (A(I),I=1,3),IDATE,(A(I),I=4,5)
      IARCH=IARCH+1
      GO TO 23
   24 IERR=IERR+1
      PRINT *,'  AN ERROR HAS OCCURRED DURING HISTORY FILE READ.'
      PRINT *,'  EXECUTION CONTINUING.'
      IF(IERR.LE.10) GO TO 19
      PRINT *,'  THE NUMBER OF ERRORS IN READING THE HISTORY FILE'
      PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
      GO TO 10
   23 IF(IDATE.LT.IYD(3)) GO TO 19
   26 WRITE(3,17) (A(I),I=1,3),IDATE,(A(I),I=4,5)
      IPHST=IPHST+1
      GO TO 19
   18 PRINT 1
      PRINT *,'  OUTPUT FILES OF ARCHIVE HISTORY RECORDS AND PURGED '
      PRINT *,'  HISTORY RECORDS (BEGINNING OCTOBER 1, HAVE NOW'
      PRINT *,'  BEEN CREATED.'
      PRINT *,'  '
      TYPE 25, IARCH,IPHST
   25 FORMAT(/2X,'ARCHIVE HISTORY FILE CONTAINS    ',I10,' RECORDS'/
     !        2X,'PURGED(NEW) HISTORY FILE CONTAINS',I10,' RECORDS')
      PRINT *,'  '
   10 STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1595	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1259	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	412	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3266	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIHSTPUR$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	FEB		77= 78 79(2)= 83 95= 96 97(2)= 99
**	I*4	I		81= 82 83 84 118(4)= 122(4)= 133(4)=
2-00000054	I*4	IANSWR		54= 57 58
**	I*4	IARCH		115= 123(2)= 141
**	I*4	ID		74= 86 90=
**	I*4	ID1		61= 74
**	I*4	ID2		62= 90
2-0000005C	I*4	IDATE		118= 120 121 122 132 133
**	I*4	IERR		117= 125(2)= 128
**	I*4	IFEB		78= 79 96= 97
**	I*4	IM		73= 81 82 89=
**	I*4	IM1		63= 73
**	I*4	IM2		64= 89
**	I*4	IPHST		116= 134(2)= 141
**	I*4	IY		75= 77 87 91= 93= 94 95
**	I*4	IY1		60= 75 94
2-00000050	I*4	IY2		46= 49 60 91
**	I*4	J		98= 99= 100
**	I*4	K		76= 87 88
**	I*4	NDY		80= 84(2)= 86(2)= 87

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-0000003C	R*4	A		20	(5)	19 118(2)= 122(2) 133(2)
2-00000030	I*4	IYD		12	(3)	19 87= 93 100(2)= 102(2)= 103(3) 120 121 132
2-00000000	I*4	JD		48	(12)	19 20D 83= 84 85=

LABELS

Address	Label	References
1-00000337	1'	25 26# 36 42 48 56 109 136
1-00000350	2'	46 47#
1-00000353	3'	49 50#
1-000003A7	4'	54 55#
0-00000148	5	38# 58
**	8	65#
0-00000634	10	57 131 145#
**	11	81 84#
0-00000318	12	82 85#
0-00000339	13	76 88 92#
0-00000376	15	94 101#
1-000003AA	16'	103 104#
1-00000479	17'	118 119# 122 133
0-00000568	18	118 136#
0-000003EC	19	118# 120 128 132 135
**	23	124 132#
0-000004E8	24	118 125#
1-00000486	25'	141 142#
0-0000049F	26	121 133#
1-0000033B	35'	44 45#
1-000003A0	36'	52 53#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	21
	FOR$OPEN	22 23

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|    =    - Value Modified                      |
|    #    - Defining Reference                  |
|    A    - Actual Argument, possibly modified  |
|    D    - Data Initialization                 |
|   (n)   - Number of occurrences on line       |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOIHSTPUR

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/SHOW=(NODICTIONARY,NOINCLUDE,MAP,NOPREPROCESSOR,SINGLE)
/STANDARD=(NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
/WARNINGS=(NODECLARATIONS,GENERAL,NOULTRIX,NOVAXELN)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE
/F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOPARALLEL
/NOANALYSIS_DATA
/NODIAGNOSTICS
/LIST=HYDRO:[SUTTER]BOIHSTPUR.LIS;2
/OBJECT=HYDRO:[SUTTER]BOIHSTPUR.OBJ;1
```

COMPILATION STATISTICS

```
Run Time:           5.00 seconds
Elapsed Time:       76.79 seconds
Page Faults:        777
Dynamic Memory:     443 pages
```

ROTCHGLOR 115

```
C*************************************************************************
C
C        PROGRAM TO CHANGE VALUES WATER BANK TRANSACTIONS IN DATA SET
C       BOISTO.CRY.  ENTER POSITIVE VALUES FOR THOSE RECEIVING WATER
C       THRU THE WATER BANK , AND NEGATIVE VALUES FOR THOSE SUPPLYING
C       WATER TO THE BANK.   RJS - JULY, 1988.
C
C*************************************************************************

      DIMENSION RNAME(4),CRYO(3)
      CHARACTER*18 DNAME
      DATA BLANK/' '/
      CALL ASSIGN (1,'BOISTO.CRY')
      OPEN (UNIT=2,NAME='BOISTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
      PRINT *,' '
      PRINT *,' ENTER CHANGES YOU WISH FOR THESE WATER BANK VALUES: '
      PRINT *,' ENTER POSITIVE VALUES FOR THOSE RECEIVING STORAGE,  '
      PRINT *,' ENTER NEGATIVE VALUES FOR THOSE SUPPLYING STORAGE.  '
      PRINT *,' '
      IANS=BLANK
    9 READ(1,2,END=6) ID,IK,DNAME,(CRYO(L),L=1,3),TRANS,BANK
    2 FORMAT(I8,A1,A18,5F10.0)
      IF(IANS.EQ.'Q') GO TO 7
      IF(IK.EQ.'S'.OR.IK.EQ.'E') GO TO 7
      TYPE 11, DNAME,BANK
   11 FORMAT(' ',A18,' WATER BANK TRANSFER  =',F10.1,' NEW VALUE? Y/N/Q
     1 '$)
      ACCEPT 102, IANS
  102 FORMAT(A1)
      IF(IANS.NE.'Y') GO TO 100
      PRINT 103
  103 FORMAT('                    ENTER NEW VALUE  ='$)
      ACCEPT 4, BANK
    4 FORMAT(F10.0)
  100 IF(IANS.EQ.'Q') GO TO 7
    7 WRITE(2,8) ID,IK,DNAME,(CRYO(L),L=1,3),TRANS,BANK
    8 FORMAT(I8,A1,A18,5F10.1)
      GO TO 9
    6 STOP
      END
```

DON'T NEED

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	563	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	315	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	172	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1050	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOICHGWB$MAIN	

VARIABLES

Address	Type	Name	Attributes	References				
2-00000048	R*4	BANK		21=	25	33=	36	
2-00000030	R*4	BLANK		12D	20			
2-0000001C	CHAR	DNAME		11	21=	25	36	
2-00000034	I*4	IANS		20=	23	28=	30	35
2-00000038	I*4	ID		21=	36			
2-0000003C	I*4	IK		21=	24(2)	36		
2-00000040	I*4	L		21(2)=	36(2)=			
2-00000044	R*4	TRANS		21=	36			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
2-00000010	R*4	CRYO		12	(3)	10	21=	36
2-00000000	R*4	RNAME		16	(4)	10		

LABELS

Address	Label	References			
1-000000BA	2'	21	22#		
1-0000012B	4'	33	34#		
0-0000022C	6	21	39#		
0-000001CB	7	23	24	35	36#
1-0000012F	8'	36	37#		
0-000000A8	9	21#	38		
1-000000C6	11'	25	26#		
0-000001C1	100	30	35#		
1-000000FD	102'	28	29#		
1-00000100	103'	31	32#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	13
	FOR$OPEN	14

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|    =   - Value Modified                       |
|    #   - Defining Reference                   |
|    A   - Actual Argument, possibly modified   |
|    D   - Data Initialization                  |
|   (n)  - Number of occurrences on line        |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOICHGWB

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS
```

COMPILATION STATISTICS

```
Run Time:           3.11 seconds
Elapsed Time:       8.13 seconds
Page Faults:        349
Dynamic Memory:     355 pages
```

CHGDAT.LIS

```
      REAL*8 XFLOWA,ADIFF,FLP,FNY
      DIMENSION JDAY(12)
      DATA JDAY/0,31,59,90,120,151,181,212,243,273,304,334/
      OPEN(UNIT=77,TYPE='OLD')
      OPEN(UNIT=76,TYPE='NEW',CARRIAGECONTROL='LIST')
      FNY=0.0
C
C***** READ A RECORD FROM OLD FILE
C
    1 READ(77,10,END=500)  ID,S,IC,IYR,MTHA,IDAYA,CODE,XFLOWA
   10 FORMAT(I8,A1,4I2,A3,F10.2)
C
C***** CHECK FOR LEAP YEAR CORRECTION, THEN COMPUTE DAY OF YEAR
C***** FROM MONTH AND DAY OF MONTH.
C
      ILEAP=0
      ADIFF=MOD(IYR,4)
      IF(ADIFF.EQ.0) ILEAP=1
      NDAYA=JDAY(MTHA)+IDAYA
      IF(MTHA.GT.2) NDAYA=NDAYA+ILEAP
C
C***** WRITE DATA TO NEW FILE IN PROPER FORMAT FOR ACCOUNTING
C***** PROGRAM.  THEN RETURN TO GET ANOTHER RECORD.
C
      IF(S.NE.'R') GO TO 2
      WRITE(76,11) ID,S,IC,IYR,NDAYA,JIDNNT(XFLOWA)
   11 FORMAT(I8,A1,2I2,I3,I7)
      GO TO 1
    2 WRITE(76,12) ID,S,IC,IYR,NDAYA,XFLOWA
   12 FORMAT(I8,A1,2I2,I3,F7.1)
C
C***** IF DATA BEING ENTERED IS LUCKY PEAK OUTFLOW (13202000) OR
C***** NEW YORK CANAL (13202000), SAVE DISCHARGE; THEN WHEN REACH
C***** GLENWOOD FLOW COMPUTE FLOW BELOW DIVERSION DAM, THEN COMPUTE
C***** RELEASES FOR FLOW MAINTENANCE FROM FISH AND GAME STORAGE AND
C***** USBR STORAGE IN LUCKY PEAK.
C
      IF(ID.EQ.13202000) FLP=XFLOWA
      IF(ID.EQ.13203000) FNY=XFLOWA
      IF(ID.NE.13206000) GO TO 1
      XFLOWA=FLP-FNY
      IF(XFLOWA.LE.0.0) XFLOWA=0.0
      FNY=0.0
      FGF=80.
      IF(FGF.GT.XFLOWA) FGF=XFLOWA
      USBRF=XFLOWA-FGF
      IF(USBRF.GT.170.) USBRF=170
      IF(XFLOWA.GT.500.) FGF=0.0
      IF(XFLOWA.GT.500.) USBRF=0.0
C     IF(FGF.GT.0.0.OR.USBRF.GT.0.0) GO TO 34
C     NA=NDAYA-ILEAP
C     IF(NA.LT.91.OR.NA.GT.304) GO TO 1
   34 WRITE(76,32) IYR,NDAYA,FGF
   32 FORMAT('13201990D19',I2,I3,F7.1)
      WRITE(76,33) IYR,NDAYA,USBRF
   33 FORMAT('13201991D19',I2,I3,F7.1)
      GO TO 1
```

```
00058      500 STOP
00059          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	595	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	83	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	128	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	806	

ENTRY POINTS

Address	Type	Name	References
0-00000000		CHGDAT$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
**	R*8	ADIFF		1	17=	18					
2-00000058	R*4	CODE		10=							
**	R*4	FGF		44=	45(2)=	46	48=	53			
2-00000038	R*8	FLP		1	38=	41					
**	R*8	FNY		1	6=	39=	41	43=			
2-00000048	I*4	IC		10=	26	29					
2-00000040	I*4	ID		10=	26	29	38	39	40		
2-00000054	I*4	IDAYA		10=	19						
**	I*4	ILEAP		16=	18=	20					
2-0000004C	I*4	IYR		10=	17	26	29	53	55		
2-00000050	I*4	MTHA		10=	19	20					
**	I*4	NDAYA		19=	20(2)=	26	29	53	55		
2-00000044	R*4	S		10=	25	26	29				
**	R*4	USBRF		46=	47(2)=	49=	55				
2-00000030	R*8	XFLOWA		1	10=	26	29	38	39	41=	42(2)=
				45(2)	46	48	49				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
2-00000000	I*4	JDAY		48	(12)	2	3D	19

LABELS

Address	Label	References			
0-00000020	1	10#	28	40	57
0-00000118	2	25	29#		
1-00000000	10'	10	11#		
1-0000000E	11'	26	27#		
1-0000001B	12'	29	30#		
1-00000029	32'	53	54#		
1-0000003E	33'	55	56#		
**	34	53#			
0-0000024C	500	10	58#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
I*4	JIDNNT	26
I*4	MOD	17

```
+-----------------------------------------+
|          KEY TO REFERENCE FLAGS         |
|   =   - Value Modified                  |
|   #   - Defining Reference              |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization             |
|  (n)  - Number of occurrences on line   |
+-----------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE CHGDAT

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
/SHOW=(NODICTIONARY,NOINCLUDE,MAP,NOPREPROCESSOR,SINGLE)
/STANDARD=(NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
/WARNINGS=(NODECLARATIONS,GENERAL,NOULTRIX,NOVAXELN)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE
/F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOPARALLEL
/NOANALYSIS_DATA
/NODIAGNOSTICS
/LIST=DUA4:[WRD.WD63]CHGDAT.LIS;6
/OBJECT=DUA4:[WRD.WD63]CHGDAT.OBJ;1
```

COMPILATION STATISTICS

Run Time:	2.37 seconds
Elapsed Time:	3.08 seconds
Page Faults:	645
Dynamic Memory:	520 pages

BOIHSTSUMW.LIS

```
C  ****************************************************
C     THIS PROGRAM LISTS DATA FROM WD-63 HISTORY FILES
C      IN USGS STYLE RECORDS BY WATER YEAR
C        AND SUMS STATIONS BETWEEN DESIGNATED
C                     POINTS.
C
C     THIS PROGRAM IS A MODIFICATION OF SNKHSTLSTW.FOR
C                  BOB SUTTER - FEBRUARY 1991
C ****************************************************
C
      LOGICAL*1 FILES(40)
      BYTE CNTRL
      INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
      REAL MEAN,MAX,MIN,NOVALUE
      CHARACTER NAME*70,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
      CHARACTER TTL*70,BLANK*1,SNAME*70,NNAME*70
      DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
      DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12),SDV(12,31)
      DIMENSION JDATE(12,31)
      DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
      CNTRL='00'X
      NOVALUE=9999999.
      BLANK=' '
      OPEN(UNIT=4,FILE='BOIHSTSUM.TTL',STATUS='OLD')
      OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
      OPEN(UNIT=8,FILE='BOIHSTSUM.OUT',STATUS='NEW',RECL=133)
C
C***** INQUIRE IF FILE IS TO BE USED TO SELECT SPECIFIC STATIONS TO BE
C***** SUMMED FROM DATA FILE.  IF NOT, ALL DATA IN FILE WILL BE
C***** SUMMED.
C
  500 PRINT *,'*************************************'
      PRINT *,'   YOU HAVE A CHOICE:                '
      PRINT *,'                                     '
      PRINT *,' EITHER SELECT STATIONS FROM INPUT FILE TO BE SUMMED,'
      PRINT *,' OR SUM ALL DATA OF SAME DAY IN INPUT FILE.          '
      PRINT *,'                                     '
      PRINT 904
  904 FORMAT('  TO USE A FILE OF SELECTED STATIONS ENTER Y/N: '$)
      ACCEPT 4, SEL
    4 FORMAT(A1)
      IF(SEL.NE.'Y') GO TO 11
C
C***** READ IN A LIST OF TITLES FOR EACH DATA SET TO BE SUMMED.  IF A
C***** TITLE IS NOT PRESENT, THAT DATA WILL THEN NOT BE INCLUDED IN THE
C***** SUMMED DATA SET.
C
      PRINT *,'                                     '
      PRINT *,'*************************************'
      PRINT *,'                                     '
      PRINT *,'   YOU HAVE ANOTHER CHOICE:                '
      PRINT *,'                                     '
      PRINT *,'  SUM ALL DIVERSIONS, FLOWS, PUMPS, OR RESERVOIRS '
      PRINT *,'  (STATION NUMBERS & TITLES READ FROM EXISTING FILE)'
      PRINT *,'                                     '
      PRINT *,'              OR                     '
      PRINT *,'                                     '
```

```
      PRINT *,'  SUM ONLY STATIONS WHICH YOU SELECT INDIVIDUALLY   '
      PRINT *,'  (YOU PROVIDE THE STATION NUMBER AND TITLE) '
      PRINT *,'                                            '
      PRINT *,'   TO SUM ALL DIVERSIONS, ANSWER  D          '
      PRINT *,'   TO SUM ALL RIVER FLOWS, ANSWER  F          '
      PRINT *,'   TO SUM ALL DRAINS, ANSWER  W          '
      PRINT *,'   TO SUM ALL RESERVOIRS, ANSWER  R          '
      PRINT *,'   TO SUM INDIVIDUAL STATIONS, ANSWER  I  '
      PRINT *,'   TO ENTER YOUR OWN GROUP OF STATIONS, ANSWER G '
      PRINT *,'   TO QUIT, ANSWER  Q  '
      PRINT *,'                                          '
      PRINT 905
  905 FORMAT('        ENTER CHOICE:'$)
      ACCEPT 4, IND
      IF(IND.EQ.'Q') GO TO 470
   11 PRINT *,' '
      PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
      PRINT *,'   WATER YEAR FOR WHICH DATA IS TO BE LISTED - '
      PRINT *,' '
      PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR WATER YEAR  '
      PRINT *,'   1990, ENTER 89-90   '
      PRINT *,'                __-__   '
      PRINT 901
  901 FORMAT('+               '$)
      KY=0
      ACCEPT 900, NP,NP1
  900 FORMAT(I2,1X,I2)
C
C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
C
      NP=NP+1900
      NP1=NP1+1900
      IF(JMOD(NP,4).EQ.0) KY=1
      ID=(NP*1000)+274+KY
      KY=0
      IF(JMOD(NP1,4).EQ.0) KY=1
      ID1=(NP1*1000)+273+KY
C
C***** READ IN TITLES OF DATA TO BE SUMMED
C
      IF(SEL.NE.'Y') GO TO 2
      IF(IND.EQ.'I') THEN
                  OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
    1             PRINT *,'   ENTER STATION NUMBER AND TITLE '
                  PRINT *,'   TO END   ENTER   99999999  '
                  PRINT *,'           ________ '
                  PRINT 903
  903             FORMAT('+           '$)
                  ACCEPT 301, ISTA,TTL
                  IF(ISTA .EQ. 99999999) GO TO 9
                  WRITE(3,301) ISTA,TTL
                  GO TO 1
    9             REWIND 3
                  GO TO 2
                  END IF
      IF(IND.NE.'G') GO TO 8
      PRINT 907
```

```
  907 FORMAT(' ENTER FILE NAME OF TITLES OF DATA TO BE SUMMED: '$)
      ACCEPT 908, IC,(FILES(I),I=1,IC)
  908 FORMAT(Q,40A1)
      CALL ASSIGN(3,FILES)
      GO TO 2
    8 IF(IND.EQ.'D') OPEN(UNIT=3,FILE='BOIDIVSUM.TTL',STATUS='OLD')
      IF(IND.EQ.'F') OPEN(UNIT=3,FILE='BOIFLO.TTL',STATUS='OLD')
      IF(IND.EQ.'W') OPEN(UNIT=3,FILE='BOIPMP.TTL',STATUS='OLD')
      IF(IND.EQ.'R') OPEN(UNIT=3,FILE='BOIRES.TTL',STATUS='OLD')
    2 PRINT 3
    3 FORMAT(' TO LIST STATIONS BEING SUMMED ENTER Y/N: '$)
      ACCEPT 4,PIS
      PRINT 7
    7 FORMAT(' TO MAKE A SEPARATE FILE OF SUMS ENTER Y/N: '$)
      ACCEPT 4,FSUM
      IF(FSUM.EQ.'Y') OPEN(UNIT=9,FILE='BOIHSTSUM.DAT',STATUS='NEW',
     1CARRIAGECONTROL='LIST')
      INREC=0
      INSTA=0
      IEND=0
      ISR=1
      STATION_NUM=0
    5 ISUM=0
      DO 40 I=1,12
      DO 40 J=1,31
   40 SDV(I,J)=NOVALUE
C
C***** IF NOT SELECTING SPECIFIC DATA FROM DATA SET TO BE SUMMED
C***** SKIP SEARCH FOR STATION ROUTINE AND STATION READ STATEMENT
C
   10 IF(SEL.NE.'Y') GO TO 15
      IF(INSTA.EQ.1) THEN
                     STA_ID=NSTA_ID
                     NAME=NNAME
                     INSTA=0
                     GO TO 15
                     ENDIF
      READ(3,301,END=450) STA_ID,NAME
  301 FORMAT(I8,A70)
C
C***** FILL ARRAY WITH MISSING VALUE INDICATORS
C
   15 DO 20 I=1,12
      DO 20 J=1,31
   20 DV(I,J)=NOVALUE
      IF(SEL.NE.'Y') GO TO 13
C
C***** SKIP READ STATEMENT IF FIRST RECORD PREVIOUSLY READ
C
      IF(STATION_NUM.EQ.STA_ID) GO TO 60
      IF(STATION_NUM.GT.STA_ID) GO TO 651
C
C***** READ A RECORD (STATION SEARCH ROUTINE)
C
   26 IF(SEL.NE.'Y') GO TO 13
   25 READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
  701 FORMAT(I8,A1,I4,I3,F7.0)
```

```
          IF(STATION_NUM.LT.STA_ID) GO TO 25
          IF(STATION_NUM.EQ.STA_ID) GO TO 60
      651 IF(TD.NE.STA_ID) THEN
            PRINT 601,STA_ID
      601   FORMAT(1X,'STATION ',I8,' NOT FOUND')
            GO TO 10
            END IF
          READ(3,301,END=99) NSTA_ID,NNAME
          INSTA=1
       35 IF(STATION_NUM.LT.NSTA_ID) THEN
            READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
            GO TO 35
            END IF
          GO TO 100
C
C***** READ A RECORD (NO STATION SEARCH ROUTINE), WHEN THE STATION
C***** NUMBER CHANGES, GO TO LISTING AND/OR SUMMING ROUTINE
C
       13 IF(INSTA.EQ.1) GO TO 12
          READ(7,701,END=98) STATION_NUM,TYPE,YEAR,ND,VALUE
          IF(INREC.EQ.0) THEN
                         INREC=1
                         GO TO 12
                         ENDIF
          IF(STATION_NUM.LT.TD) GO TO 13
          IF(STATION_NUM.GT.TD) THEN
                                INSTA=1
                                GO TO 100
                                ENDIF
       12 STA_ID=STATION_NUM
          INSTA=0
C
C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
C
       60 IDATE=(YEAR*1000)+ND
          IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 26
          JD(2)=28
          IF(JMOD(YEAR,4) .EQ. 0) JD(2)=29
          LD=ND
          MON=0
       62 MON=MON+1
          IF(MON.GT.12) GO TO 68
          LD=LD-JD(MON)
          IF(LD.GT.0) GO TO 62
          LD=LD+JD(MON)
          DV(MON,LD)=VALUE
          JDATE(MON,LD)=IDATE
C
C***** SAVE VARIABLE VALUES IN PREPARATION FOR READING NEXT RECORD
C***** SO THAT IF STATION CHANGES, WE HAVE PRESERVED DATA FOR
C***** STATION TO BE LISTED.
C
       68 CONTINUE
          TD=STATION_NUM
          P_YEAR=YEAR
          P_TYPE=TYPE
          IF(SEL.NE.'Y') GO TO 13
```

```
      GO TO 25
C
C***** ENTIRE YEAR FOR ONE STATION READ IN -
C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
C***** IF LAST STATION TITLE HAS BEEN READ, SET NEXT STATION
C***** NUMBER = 99999999 SO THAT SUMS WILL BE PRINTED.
C
   98 IEND=1
   99 STATION_NUM=99999999
  100 IF(ISUM.NE.1.AND.PIS.NE.'Y') GO TO 6
      LEAP=0
      IF(JMOD(NP1,4).EQ.0) LEAP=1
      IF(P_TYPE.NE.'R') GO TO 120
      DELTA_S(1)=DV(1,31)-DV(12,31)
      DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
      DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
      DELTA_S(4)=DV(4,30)-DV(3,31)
      DELTA_S(5)=DV(5,31)-DV(4,30)
      DELTA_S(6)=DV(6,30)-DV(5,31)
      DELTA_S(7)=DV(7,31)-DV(6,30)
      DELTA_S(8)=DV(8,31)-DV(7,31)
      DELTA_S(9)=DV(9,30)-DV(8,31)
      DELTA_S(10)=NOVALUE
      DELTA_S(11)=DV(11,30)-DV(10,31)
      DELTA_S(12)=DV(12,31)-DV(11,30)
      DO 119 I=2,12
  119 IF(ABS(DELTA_S(I)).GE.8000000.) DELTA_S(I)=NOVALUE
  120 DO 135 I=1,12
      D=31
      IF(I.EQ.2) D=28+LEAP
      IF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.9) D=30
      CFSD(I)=DV(I,1)
      D_ADJ=0
      IF(DV(I,1).EQ.NOVALUE) THEN
                             CFSD(I)=0.0
                             D_ADJ=1
                             END IF
      MAX(I)=DV(I,1)
      IF(DV(I,1).EQ.NOVALUE) MAX(I)=-1.0
      MIN(I)=DV(I,1)
      DO 130 J=2,D
      DZ=DV(I,J)
      IF(DV(I,J).EQ.NOVALUE) THEN
                             DZ=0.0
                             D_ADJ=D_ADJ+1
                             END IF
      CFSD(I)=CFSD(I)+DZ
      IF(DV(I,J).EQ.NOVALUE) THEN
                             GO TO 129
                             END IF
      IF(DV(I,J).GT.MAX(I)) MAX(I)=DV(I,J)
  129 IF(DV(I,J).LT.MIN(I)) MIN(I)=DV(I,J)
  130 CONTINUE
      IF(D_ADJ.GE.D) THEN
                     CFSD(I)=NOVALUE
                     DELTA_S(I)=NOVALUE
                     MEAN(I)=NOVALUE
```

```
                        GO TO 133
                        END IF
        MEAN(I)=CFSD(I)/(D-D_ADJ)
    133 ACFT(I)=CFSD(I)*1.9835
    135 CONTINUE
        CALL DATE(RUNDATE)
        WRITE(8,801) RUNDATE
    801 FORMAT('1',/,120X,A9)
        IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
    800 FORMAT(1H+,'PRELIMINARY DATA')
        WRITE(8,802) STA_ID,NAME
    802 FORMAT(//37X,I8,3X,A70)
        IF(P_TYPE .EQ.'R') THEN
                            WRITE(8,803) NP,NP1
    803 FORMAT(30X,'CONTENTS IN ACRE FEET, WATER YEAR OCTOBER',
       *         I5,' TO SEPTEMBER',I5)
                            GO TO 160
                            ELSE
                            WRITE(8,804) NP,NP1
    804 FORMAT(30X,'DISCHARGE, CUBIC FEET PER SECOND, WATER YEAR',
       *         ' OCTOBER',I5,' TO SEPTEMBER',I5)
                            WRITE(8,805)
    805 FORMAT(59X,'MEAN VALUES')
                            END IF
    160 WRITE(8,806)
    806 FORMAT(//,1X,'DAY',11X,'OCT',7X,'NOV',7X,'DEC',7X,'JAN',7X,
       *         'FEB',7X,'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',
       *         7X,'AUG',7X,'SEP')
        WRITE(8,807)
    807 FORMAT(' ')
        DO 180 J=1,31
        IF(J.EQ.6.OR.J.EQ.11.OR.J.EQ.16.OR.J.EQ.21.OR.J.EQ.26) THEN
                                WRITE(8,807)
                                END IF
        DO 170 I=1,12
        II=I-3
        IF(I.EQ.1) II=10
        IF(I.EQ.2) II=11
        IF(I.EQ.3) II=12
        IF(II.EQ.10) WRITE(8,807)
        IF(II.EQ.10) WRITE(8,811) CNTRL,J
    811 FORMAT(A1,I3,'    ')
        IF(DV(II,J).EQ.NOVALUE) THEN
                                WRITE(8,809) CNTRL
                                GO TO 170
                                END IF
        IF(DV(II,J).GE.10.0) THEN
                            IDV=JNINT(DV(II,J))
                            WRITE(8,810) CNTRL,IDV
    810                     FORMAT(A1,I10)
                            GO TO 170
                            END IF
        IF(DV(II,J).LT.10.0) THEN
                            WRITE(8,808) CNTRL,DV(II,J)
    808                     FORMAT(A1,F10.1)
                            GO TO 170
                            END IF
```

```
809                     FORMAT(A1,'       ---')
170 CONTINUE
180 CONTINUE
    IF(P_TYPE.EQ.'R') THEN
                    WRITE(8,812)
812                 FORMAT('0')
                    WRITE(8,813) CNTRL
813                 FORMAT(A1,2X,'MAX  ')
                    DO 190 I=1,12
                    II=I-3
                    IF(I.EQ.1) II=10
                    IF(I.EQ.2) II=11
                    IF(I.EQ.3) II=12
                    IF(MAX(II).EQ.-1.0) THEN
                              WRITE(8,860) CNTRL
860                           FORMAT(A1,'          ')
                              GO TO 190
                              END IF
                    IDV=JNINT(MAX(II))
                    WRITE(8,810) CNTRL,IDV
190                 CONTINUE
                    WRITE(8,807)
                    WRITE(8,814) CNTRL
814                 FORMAT(A1,2X,'MIN  ')
                    DO 200 I=1,12
                    II=I-3
                    IF(I.EQ.1) II=10
                    IF(I.EQ.2) II=11
                    IF(I.EQ.3) II=12
                    IF(MIN(II).EQ.NOVALUE) THEN
                                           WRITE(8,860) CNTRL
                                           GO TO 200
                                           END IF
                    IDV=JNINT(MIN(II))
                    WRITE(8,810) CNTRL,IDV
200                 CONTINUE
                    WRITE(8,807)
                    WRITE(8,815) CNTRL
815                 FORMAT(A1,1X,'CHNG            ')
                    DO 210 I=2,12
                    II=I-3
                    IF(I.EQ.2) II=11
                    IF(I.EQ.3) II=12
                    IDV=JNINT(DELTA_S(II))
                    IF(DELTA_S(II).EQ.NOVALUE) THEN
                                           WRITE(8,860) CNTRL
                                           GO TO 210
                                           END IF
                    WRITE(8,810) CNTRL,IDV
210                 CONTINUE
                    GO TO 10
                    END IF
    WRITE(8,812)
    WRITE(8,816) CNTRL
816 FORMAT(A1,'TOTAL  ')
    DO 220 I=1,12
    II=I-3
```

```
      IF(I.EQ.1) II=10
      IF(I.EQ.2) II=11
      IF(I.EQ.3) II=12
      IF(CFSD(II).GE.NOVALUE) THEN
                              WRITE(8,860) CNTRL
                              GO TO 220
                              END IF
      IDV=JNINT(CFSD(II))
      WRITE(8,810) CNTRL,IDV
  220 CONTINUE
      WRITE(8,807)
      WRITE(8,818) CNTRL
  818 FORMAT(A1,' MEAN  ')
      DO 230 I=1,12
      II=I-3
      IF(I.EQ.1) II=10
      IF(I.EQ.2) II=11
      IF(I.EQ.3) II=12
      IF(MEAN(II).GE. NOVALUE) THEN
                               WRITE(8,860) CNTRL
                               GO TO 230
                               END IF
      IF(MEAN(II).LT.10.0) THEN
                           WRITE(8,808) CNTRL,MEAN(II)
                           GO TO 230
                           END IF
      IDV=JNINT(MEAN(II))
      WRITE(8,810) CNTRL,IDV
  230 CONTINUE
      WRITE(8,807)
      WRITE(8,819) CNTRL
  819 FORMAT(A1,'  MAX  ')
      DO 240 I=1,12
      II=I-3
      IF(I.EQ.1) II=10
      IF(I.EQ.2) II=11
      IF(I.EQ.3) II=12
      IF(MAX(II).EQ.-1.0) THEN
                          WRITE(8,860) CNTRL
                          GO TO 240
                          END IF
      IF(MAX(II).LT.10.0) THEN
                          WRITE(8,808) CNTRL,MAX(II)
                          GO TO 240
                          END IF
      IDV=JNINT(MAX(II))
      WRITE(8,810) CNTRL,IDV
  240 CONTINUE
      WRITE(8,807)
      WRITE(8,820) CNTRL
  820 FORMAT(A1,'  MIN  ')
      DO 250 I=1,12
      II=I-3
      IF(I.EQ.1) II=10
      IF(I.EQ.2) II=11
      IF(I.EQ.3) II=12
      IF(MIN(II).GE.NOVALUE) THEN
```

```
                                WRITE(8,860) CNTRL
                                GO TO 250
                                END IF
          IF(MIN(II).LT.10.0) THEN
                             WRITE(8,808) CNTRL,MIN(II)
                             GO TO 250
                             END IF
          IDV=JNINT(MIN(II))
          WRITE(8,810) CNTRL,IDV
      250 CONTINUE
C            DO 260 I=1,12
C            IFT(I)=JNINT(ACFT(I)/100.)
C            IFT(I)=IFT(I)*100
C260         CONTINUE
          WRITE(8,807)
          WRITE(8,821) CNTRL
      821 FORMAT(A1,'AC-FT  ')
          DO 270 I=1,12
          II=I-3
          IF(I.EQ.1) II=10
          IF(I.EQ.2) II=11
          IF(I.EQ.3) II=12
          IF(ACFT(II).GE.19834900.0) THEN
                                  WRITE(8,860) CNTRL
                                  GO TO 270
                                  END IF
C            IF(ACFT(II).LE.999.) THEN
C                                   IFT(II)=JNINT(ACFT(II))
C                                 WRITE(8,810) CNTRL,IFT(II)
C                                   GO TO 270
C                                   END IF
          IFT(II)=JNINT(ACFT(II))
          WRITE(8,810) CNTRL,IFT(II)
      270 CONTINUE
          YT=0.0
          DO 280 I=1,12
          IF(CFSD(I) .EQ. NOVALUE) CFSD(I)=0.0
      280 YT=YT + CFSD(I)
          IYT=JNINT(YT)
          IYM=JNINT(YT/(365+LEAP))
C            NT=JNINT(YT*1.9835/100.)
C            IYA=NT*100
C            IF(IYA.LT.9999) IYA=1.9835*YT
          IYA=1.9835*YT
          WRITE(8,807)
          WRITE(8,831) NP1,IYT,IYM,IYA
      831 FORMAT(11X,'WATER YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
     *         I6,4X,'AC-FT',I8)
C
C***** ENTER SUMMING ROUTINE UNLESS END OF DATA OR SUM HAS JUST
C***** BEEN PRINTED.
C
          IF(IEND.EQ.1) GO TO 450
          IF(ISUM.EQ.1) GO TO 5
        6 DO 460 I=1,12
          DO 460 J=1,31
          IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).EQ.NOVALUE) GO TO 460
```

```
      IF(DV(I,J).GE.0.0.AND.SDV(I,J).EQ.NOVALUE) SDV(I,J)=0.0
      IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).GT.0.0) DV(I,J)=0.0
      SDV(I,J)=SDV(I,J)+DV(I,J)
  460 CONTINUE
      IF(ISR.EQ.0) GO TO 480
      READ(4,301,END=10) IDSUM,SNAME
      ISR=0
  480 IF(STATION_NUM.LT.IDSUM) GO TO 10 ! GO TO NEXT STATION
  485 STA_ID=IDSUM
      NAME=SNAME
      DO 465 I=1,12
      DO 465 J=1,31
  465 DV(I,J)=SDV(I,J)
      ISUM=1
      ISR=1
C
C***** IF FSUM EQUALS 'Y' MAKE A FILE OF SUMMED DATA
C
      IF(FSUM.NE.'Y') GO TO 100
      DO 885 K=1,12
      I=K+9
      IF(I.GT.12) I=I-12
      JMD=JD(I)
      DO 885 J=1,JMD
  885 IF(JDATE(I,J).GT.0.AND.SDV(I,J).NE.NOVALUE)
     1                  WRITE(9,880) IDSUM,JDATE(I,J),SDV(I,J)
  880 FORMAT(I8,'S',I7,F7.0)
      GO TO 100  ! ORGANIZE AND PRINT SUMS
  450 CONTINUE
      IF(ISR.EQ.0) GO TO 485
      PRINT *,'  '
      PRINT *,'  '
      PRINT *,'  REQUESTED STATION LISTINGS AND SUMS COMPLETE '
      PRINT *,'  '
      PRINT *,'  '
  470 PRINT *,'                                    '
      PRINT *,'  OUTPUT IS  THE LATEST VERSION OF BOIHSTSUM.OUT'
      PRINT *,'  '
      STOP
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	6113	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	2229	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	5976	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	14318	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIHSTSUMW$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	CHAR	BLANK		16 23=
**	L*1	CNTRL		12 21= 326 329 334 339 349 357 362 365 373 377 380 388 391 396 404 408 411 419 423 427 430 438 442 446 449 457 461 465 472 480 489
**	R*4	D		257= 258= 259= 269 282 288
**	R*4	DZ		270= 272= 275
**	I*4	D_ADJ		13 261= 264= 273(2)= 282 288
2-00001468	R*4	FSUM		129= 130 532
**	I*4	I		116(2)= 138= 140 157= 159 254= 255(2) 256= 258 259(3) 260(2) 262 263 266(2) 267(2) 268(2) 270 271 275(2) 276 279(4) 280(4) 283 284 285 288(2) 289(2) 320= 321 322 323 324 351= 352 353 354 355 367= 368 369 370 371 382= 383 384 385 398= 399 400 401 402 413= 414 415 416 417 432= 433 434 435 436 451= 452 453 454 455 474= 475 476 477 478 492= 493(2) 494 511= 513(2) 514(3) 515(3) 516(3) 524= 526(2) 534= 535(3)= 536 538(4)
2-00001460	I*4	IC		116(2)=
**	I*4	ID		91= 207
**	I*4	ID1		94= 207
**	I*4	IDATE		206= 207(2) 218
2-00001484	I*4	IDSUM		519= 521 522 538
**	I*4	IDV		333= 334 361= 362 376= 377 386= 391 407= 408 426= 427

```
                                                445=      446       464=      465
  2-00001470  I*4  IEND                   134=      236=      509
      **      I*4  II                     321=      322=      323=      324=      325       326
                                                328       332       333       338       339       352=
                                                353=      354=      355=      356       361       368=
                                                369=      370=      371=      372       376       383=
                                                384=      385=      386       387       399=      400=
                                                401=      402=      403       407       414=      415=
                                                416=      417=      418       422       423       426
                                                433=      434=      435=      436=      437       441
                                                442       445       452=      453=      454=      455=
                                                456       460       461       464       475=      476=
                                                477=      478=      479       488(2)    489

  2-00001450  I*4  IND                     71=       72        99        113       120       121
                                                122       123       294(2)
  2-0000146C  I*4  INREC                  132=      192       193=
      **      I*4  INSTA                  133=      146       149=      180=      190       198=
                                                202=
      **      I*4  ISR                    135=      518       520=      528=      543
  2-0000145C  I*4  ISTA                   106=      107       108

  2-00001474  I*4  ISUM                   137=      238       510       527=
      **      I*4  IYA                    500=      502
      **      I*4  IYM                    496=      502
      **      I*4  IYT                    495=      502
      **      I*4  J                      139=      140       158=      159       269=      270
                                                271       276       279(2)    280(2)    316=      317(5)
                                                326       328       332       333       338       339
                                                512=      513(2)    514(3)    515(3)    516(3)    525=
                                                526(2)    537=      538(4)

      **      I*4  JMD                    536=      537
      **      I*4  K                      533=      534
      **      I*4  KY                      82=       90=       91        92=       93=       94
      **      I*4  LD                     210=      214(2)=   215       216(2)=   217       218
      **      I*4  LEAP                   239=      240=      243       244       258       496

      **      I*4  MON                    211=      212(2)=   213       214       216       217
                                                218
  2-00001318  CHAR NAME                    15       148=      152=      296       523=
  2-0000147C  I*4  ND                     170=      182=      191=      206       210
  2-000013F5  CHAR NNAME                   16       148       179=
      **      R*4  NOVALUE                 14        22=      140       159       251       255
                                                262       267       271       276       283       284
                                                285       328       372       387       403       418
                                                456       493       513(2)    514       515       538

  2-00001454  I*4  NP                      83=       88(2)=    90        91        299       304
  2-00001458  I*4  NP1                     83=       89(2)=    93        94        240       299
                                                304       502
  2-00001478  I*4  NSTA_ID                147       179=      181
  2-00001464  R*4  PIS                    126=      238
  2-0000135F  CHAR P_TYPE                  15       227=      241       298       346

      **      I*4  P_YEAR                  13       226=
  2-00001360  CHAR RUNDATE                 15       291A      292
```

```
  2-0000144C  R*4  SEL                          40=      42       98      145      160      169
                                                            228
  2-000013AF  CHAR SNAME                         16     519=     523
  2-00001444  I*4  STATION_NUM                   13     136=     164      165      170=     172
                                                            173      181      182=     191=     196      197
                                                            201      225      237=     521

  2-00001440  I*4  STA_ID                        13     147=     152=     164      165      172
                                                            173      174      175      201=     296      522=
  2-0000143C  I*4  TD                            13     174      196      197      225=
      **      CHAR TITLE                         15
  2-00001369  CHAR TTL                           16     106=     108
  2-0000135E  CHAR TYPE                          15     170=     182=     191=     227

  2-00001480  R*4  VALUE                        170=    182=     191=     217
  2-00001448  I*4  YEAR                          13     170=     182=     191=     206      209
                                                            226
      **      R*4  YT                           491=    494(2)=  495      496      500
      **      CHAR ZLCH                          15
```

ARRAYS

```
    Address  Type  Name          Attributes        Bytes  Dimensions     References

  2-000006F0  R*4  ACFT                             48  (12)              18      289=     479      488
  2-00000690  R*4  CFSD                             48  (12)              17      260=     263=     275(2)=  283=
                                                                                         288      289      403      407      493(
                                                                                         494
  2-000006C0  R*4  DELTA_S                          48  (12)              18      242=     243=     244=     245=
                                                                                         246=     247=     248=     249=     250=
                                                                                         251=     252=     253=     255(2)=  284=
                                                                                         386      387
  2-000000C0  R*4  DV                             1488  (12, 31)          17      159=     217=     242(2)   243(2)
                                                                                         244(2)   245(2)   246(2)   247(2)   248(
                                                                                         249(2)   250(2)   252(2)   253(2)   260
                                                                                         262      266      267      268      270
                                                                                         271      276      279(2)   280(2)   328
                                                                                         332      333      338      339      513
                                                                                         514      515(2)=  516      526=
  2-000012F0  L*1  FILES                            40  (40)              11      116=     118A

  2-00000720  I*4  IFT                              48  (12)              18      488=     489
  2-00000090  I*4  JD                               48  (12)              17       20D     208=     209=     214
                                                                                         216      536
  2-00000D20  I*4  JDATE                          1488  (12, 31)          19      218=     538(2)
  2-00000030  R*4  MAX                              48  (12)              14       17      266=     267=     279(2)=
                                                                                         356      361      437      441      442
                                                                                         445
  2-00000000  R*4  MEAN                             48  (12)              14       18      285=     288=     418
                                                                                         422      423      426

  2-00000060  R*4  MIN                              48  (12)              14       17      268=     280(2)=  372
                                                                                         376      456      460      461      464
  2-00000750  R*4  SDV                            1488  (12, 31)          18      140=     513      514(2)=  515
                                                                                         516(2)=  526      538(2)
```

LABELS

Address	Label	References					
0-00000534	1	101#	109				
0-000006C8	2	98	111	119	124#		
1-00000621	3'	124	125#				
1-0000059F	4'	40	41#	71	126	129	
0-0000075C	5	137#	510				
0-0000168C	6	238	511#				
1-0000064F	7'	127	128#				
0-0000067C	8	113	120#				
0-00000604	9	107	110#				
**	10	145#	177	393	519	521	
0-000003A5	11	42	73#				
0-00000894	12	190	194	201#			
0-00000818	13	160	169	190#	196	228	
0-000007D8	15	145	150	157#			
**	20	157	158	159#			
0-00000928	25	170#	172	229			
0-000008B7	26	169#	207				
0-000009E0	35	181#	183				
**	40	138	139	140#			
0-0000089C	60	164	173	206#			
0-000008E8	62	212#	215				
0-00000910	68	213	224#				
0-0000088C	98	191	236#				
0-00000A3C	99	179	237#				
0-00000A44	100	170	182	185	199	238#	532
			541				
**	119	254	255#				
0-00000B13	120	241	256#				
0-00000BB4	129	277	280#				
**	130	269	281#				
0-00000BF5	133	286	289#				
**	135	256	290#				
0-00000CF8	160	302	310#				
0-00000E6C	170	320	330	336	341	344#	
**	180	316	345#				
0-00000F37	190	351	359	363#			
0-00000FEF	200	367	374	378#			
0-0000109D	210	382	389	392#			
0-0000115B	220	398	405	409#			
0-0000124B	230	413	420	424	428#		
0-0000133F	240	432	439	443	447#		
0-00001433	250	451	458	462	466#		
0-000014F5	270	474	481	490#			
**	280	492	494#				
1-0000067F	301'	106	108	152	153#	179	519

```
  0-0000159C  450                                   152       509       542#

  0-000016D9  460                                   511       512       513       517#
      **      465                                   524       525       526#
  0-0000162F  470                                    72       549#
  0-0000171A  480                                   518       521#
  0-00001724  485                                   522#      543

      **      500                                    32#
  1-00000690  601'                                  175       176#
  0-00000982  651                                   165       174#
  1-00000684  701'                                  170       171#      182       191
  1-000006B4  800'                                  294       295#

  1-000006AB  801'                                  292       293#
  1-000006CA  802'                                  296       297#
  1-000006D5  803'                                  299       300#
  1-00000716  804'                                  304       305#
  1-00000764  805'                                  307       308#

  1-00000774  806'                                  310       311#
  1-000007D2  807'                                  314       315#      318       325       364       379
                                                                  410       429       448       471       501
  1-000007E6  808'                                  339       340#      423       442       461
  1-000007EC  809'                                  329       343#
  1-000007E1  810'                                  334       335#      362       377       391       408
                                                                  427       446       465       489

  1-000007D6  811'                                  326       327#
  1-000007FB  812'                                  347       348#      395
  1-000007FF  813'                                  349       350#
  1-0000081A  814'                                  365       366#
  1-00000826  815'                                  380       381#

  1-0000083D  816'                                  396       397#
  1-00000849  818'                                  411       412#
  1-00000855  819'                                  430       431#
  1-00000861  820'                                  449       450#
  1-0000086D  821'                                  472       473#

  1-00000879  831'                                  502       503#
  1-0000080B  860'                                  357       358#      373       388       404       419
                                                                  438       457       480
  1-000008AA  880'                                  538       540#
      **      885                                   533       537       538#
  1-000005CF  900'                                   83        84#

  1-000005BB  901'                                   80        81#
  1-000005D6  903'                                  104       105#
  1-0000056B  904'                                   38        39#
  1-000005A2  905'                                   69        70#
  1-000005E6  907'                                  114       115#

  1-0000061B  908'                                  116       117#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	118
	FOR$DATE_T_DS	291
	FOR$OPEN	24 25 26 100 120 121 122 123 130

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|    =   - Value Modified                       |
|    #   - Defining Reference                   |
|    A   - Actual Argument, possibly modified   |
|    D   - Data Initialization                  |
|   (n)  - Number of occurrences on line        |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
FORTRAN/LIST/CROSS_REFERENCE BOIHSTSUMW

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/SHOW=(NODICTIONARY,NOINCLUDE,MAP,NOPREPROCESSOR,SINGLE)
/STANDARD=(NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
/WARNINGS=(NODECLARATIONS,GENERAL,NOULTRIX,NOVAXELN)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE
/F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOPARALLEL
/NOANALYSIS_DATA
/NODIAGNOSTICS
/LIST=HYDRO:[SUTTER.BOISE]BOIHSTSUMW.LIS;1
/OBJECT=HYDRO:[SUTTER.BOISE]BOIHSTSUMW.OBJ;1
```

COMPILATION STATISTICS

```
Run Time:           21.06 seconds
Elapsed Time:       24.67 seconds
Page Faults:        1621
Dynamic Memory:     1065 pages
```

BOIGEISU. LIS

```
C     ***************************************************************
C
C             PROGRAM BOIGETSU - STORAGE USE SELECT
C
C       THIS PROGRAM SELECTS FROM THE ALLOCATIONS FILE THE STOARGE
C       USED BY EACH USER ENTITY ON THE SELECTED DATE.  THESE DATA
C       ARE PLACED IN A FILE CALLED BOISTO.USE FOR INPUT TO
C       STORAGE ACCOUNTING PROGRAM.
C
C                       BOB SUTTER  SEPTEMBER 1987
C
C     ***************************************************************
C
C
      DIMENSION OM(12),JD(12)
      DATA OM/'OCT','NOV','DEC','JAN','FEB','MAR','APR','MAY','JUN',
     !  'JUL','AUG','SEP'/
      DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
      DATA BLANK/' '/
      CALL ASSIGN(1,'BOIWRA.ALC')
      OPEN (UNIT=2,NAME='BOISTO.USE',TYPE='NEW',CARRIAGECONTROL='LIST')
      PRINT 1
    1 FORMAT (' ')
      PRINT *,' ******************************************************'
      PRINT *,' '
      PRINT *,'  THIS PROCEDURE SELECTS THE STORAGE USE DATA FROM THE  '
      PRINT *,'  LATEST ALLOCATION FILE FOR INPUT TO THE STORAGE '
      PRINT *,'  ACCOUNTING PROGRAM.  ENTER THE DATE AND ANSWER THE    '
      PRINT *,'  FOLLOWING QUESTIONS USING Y FOR YES, N FOR NO, AND C  '
      PRINT *,'  IF YOU WISH TO CANCEL ALL.  '
      PRINT *,' '
      PRINT *,' ******************************************************'
      PRINT 1
C
    5 PRINT *,'  ENTER THE MONTH, DAY, AND YEAR FOR THE DAY YOU WANT TO'
      PRINT *,'  RETRIEVE THE STORED WATER USES.  '
      PRINT *,'  FOR EXAMPLE : 07-25-1995  '
      PRINT 1
      PRINT *,'                ??????????  '
      PRINT 35
   35 FORMAT('+                '$)
      ACCEPT 2, IM,ID,IY
    2 FORMAT(I2,1X,I2,1X,I4)
      PRINT 1
      PRINT 3, IM,ID,IY
    3 FORMAT (1X,' THE STORAGE USE DATE IS ',I2,'-',I2,'-',I4/
     !   2X,'IS THIS CORRECT?  ANSWER Y/N/C.  ')
      PRINT 36
   36 FORMAT(28X,' '$)
      ACCEPT 4,IANSWR
    4 FORMAT (1A1)
      PRINT 1
      IF (IANSWR.EQ.'C') GO TO 10
      IF (IANSWR.EQ.'N') GO TO 5
      ZERO=0.0
C
C***** CONVERT DATE TO SEVEN DIGIT DATE YYYYDDD,
```

```
C***** WHERE YYYY IS THE YEAR AND DDD IS THE NUMERICAL DAY OF THE YEAR.
C
      FEB=IY/4.0
      IFEB=FEB
      FEB=FEB-IFEB
      NDY=0
      DO 11 I=1,IM
      IF(I.EQ.IM) GO TO 12
      IF(I.EQ.2.AND.FEB.LT.0.001) JD(2)=29
   11 NDY=NDY+JD(I)
   12 JD(2)=28
      NDY=NDY+ID
      IYD=(IY*1000)+NDY
      PRINT 16, IYD
   16 FORMAT(3X,'ALL STORAGE USE DATA FROM DAY ',I8,' IS BEING RETRIEVED
     1 FROM'/3X,'THE THE LATEST ALLOCATIONS FILE, SO PLEASE HANG ON FOR
     2A WHILE. ')
      PRINT 1
C
C***** SELECT ALL STORAGE USE DATA FROM ALLOCATIONS FILE FOR DAY IYD AND
C***** WRITE THESE DATA TO FILE BOISTO.USE.
C
      IALK=0
      IERR=0
   29 READ(1,27,END=28,ERR=34) ID,A,IDATE,USED
   27 FORMAT(I8,A1,I7,20X,F10.0)
      IF(IDATE.NE.IYD) GO TO 29
      IALK=1
      IF(A.EQ.'D') GO TO 31
      GO TO 29
   34 IERR=IERR+1
      PRINT *,'  AN ERROR HAS OCCURRED DURING ALLOCATIONS FILE READ.'
      PRINT *,'  EXECUTION CONTINUING.'
      IF(IERR.LE.10) GO TO 29
      PRINT *,'  THE NUMBER OF ERRORS IN READING THE ALLOCATIONS FILE'
      PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
      GO TO 10
   31 WRITE(2,30) ID,A,IDATE,USED
   30 FORMAT(I8,A1,I7,F10.1)
      GO TO 29
   28 IF(IALK.EQ.0) GO TO 45
      PRINT 1
      PRINT *,'  '
      PRINT *,'  STORAGE USE DATA FROM ALLOCATIONS FILE HAVE BEEN '
      PRINT *,'  RETRIEVED AND ARE IN FILE BOISTO.USE '
      PRINT *,'  '
   10 STOP
   45 PRINT 46, IYD
   46 FORMAT(/3X,'  NO DATA FOR DAY ',I8,' WAS FOUND IN ALLOCATIONS')
```

```
00107          STOP
00108          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1335	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	1085	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	304	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	2724	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BOIGETSU$MAIN	

VARIABLES

Address	Type	Name	Attributes	References				
2-00000044	R*4	A		82=	86	95		
**	R*4	BLANK		19D				
**	R*4	FEB		60=	61	62(2)=	66	
**	I*4	I		64=	65	66	67	
**	I*4	IALK		80=	85=	98		
2-0000003C	I*4	IANSWR		50=	53	54		
2-00000034	I*4	ID		42=	45	69	82=	95
2-00000048	I*4	IDATE		82=	84	95		
**	I*4	IERR		81=	88(2)=	91		
**	I*4	IFEB		61=	62			
2-00000030	I*4	IM		42=	45	64	65	
2-00000038	I*4	IY		42=	45	60	70	
**	I*4	IYD		70=	71	84	105	
**	I*4	NDY		63=	67(2)=	69(2)=	70	
2-0000004C	R*4	USED		82=	95			
**	R*4	ZERO		55=				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000000	I*4	JD		48	(12)	15	18D	66=	67	68=
**	R*4	OM		48	(12)	15	16D			

LABELS

Address	Label	References							
1-00000000	1'	22	23#	33	38	44	52	75	99
1-00000019	2'	42	43#						
1-00000024	3'	45	46#						
1-0000007B	4'	50	51#						
0-00000140	5	35#	54						
0-00000530	10	53	94	104#					
**	11	64	67#						
0-00000304	12	65	68#						
1-0000007E	16'	71	72#						
1-00000102	27'	82	83#						
0-00000478	28	82	98#						
0-00000358	29	82#	84	87	91	97			
1-0000010E	30'	95	96#						
0-000003B4	31	86	95#						
0-000003F8	34	82	88#						
1-00000004	35'	40	41#						
1-00000074	36'	48	49#						
0-00000508	45	98	105#						
1-00000118	46'	105	106#						

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	20

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|    =    - Value Modified                      |
|    #    - Defining Reference                  |
|    A    - Actual Argument, possibly modified  |
|    D    - Data Initialization                 |
|   (n)   - Number of occurrences on line       |
+-----------------------------------------------+
```

```
COMMAND QUALIFIERS

  FORTRAN/LIST/CROSS_REFERENCE BOIGETSU

  /ASSUME=(ACCURACY_SENSITIVE,NODUMMY_ALIASES,NOSOURCE_INCLUDE)
  /BLAS=(INLINE,MAPPED)
  /CHECK=(NOALIGNMENT,NOASSERTIONS,NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(PARAMETERS=USED,NOSYMBOLS,TRACEBACK)
  /DESIGN=(NOCOMMENTS,NOPLACEHOLDERS)
  /DIRECTIVES=(DEPENDENCE)
  /MATH_LIBRARY=(ACCURATE,NOV5)
  /PARALLEL=(NOAUTOMATIC,NOMANUAL)
  /SHOW=(NODATA_DEPENDENCES,NODICTIONARY,NOINCLUDE,NOLOOPS,MAP,NOPREPROCESSOR,SINGLE)
  /STANDARD=(NOMIA,NOSEMANTIC,NOSOURCE_FORM,NOSYNTAX)
  /WARNINGS=(NOALIGNMENT,NOAlpha_AXP,NODECLARATIONS,GENERAL,INFORMATIONAL,NOINLINE,NOTRUNCATED_SOURCE,
             NOULTRIX,UNCALLED,UNINITIALIZED,UNUSED,USAGE,NOVAXELN)
  /CONVERT=NATIVE  /CROSS_REFERENCE  /NOD_LINES  /ERROR_LIMIT=30  /NOEXTEND_SOURCE
  /F77  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE=LEVEL=3
  /NORECURSIVE  /NOSYNCHRONOUS_EXCEPTIONS  /TERMINAL=NOSTATISTICS  /NOVECTOR
  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /LIST=HYDRO:[BSUTTER.BOISE]BOIGETSU.LIS;3
  /OBJECT=HYDRO:[BSUTTER.BOISE]BOIGETSU.OBJ;1


COMPILER:  DEC Fortran V6.3-141

COMPILATION STATISTICS

  Run Time:           0.33 seconds
  Elapsed Time:       0.97 seconds
  Page Faults:        795
  Dynamic Memory:     768 pages
```

BOISTO. SPA

13190000	SANDERSON RANCH	0.0	0.0	0.0
13194000	SARROWROCK	0.0	0.0	0.0
13201050	LUCKY PEAK NURSERY	200.0	0.0	0.0
13201500	SLUCKY PEAK	0.0	0.0	0.0
13202995	PENITENTIARY	0.0	0.0	0.0
13203000	NEW YORK	232871.0	359662.0	0.0
13203500	SLAKE LOWELL	0.0	0.0	0.0
13203527	SURPRISE VLY FARMS	0.0	0.0	0.0
13203760	RIDENBAUGH	3832.0	15137.0	35000.0
13204005	BUBB	0.0	543.0	500.0
13204020	MEEVES	0.0	0.0	0.0
13204060	ROSSI MILL	0.0	0.0	700.0
13204190	BOISE CITY	0.0	0.0	1000.0
13205515	SETTLERS	1778.0	6082.0	10000.0
13205517	DAVIS	0.0	0.0	1500.0
13205613	BOISE CITY PARKS	0.0	0.0	0.0
13205622	THURMAN MILL	0.0	0.0	800.0
13205638	BOISE WATER CORP	0.0	1000.0	0.0
13205640	FARMERS UNION	2874.0	5727.0	10000.0
13205641	BOISE VALLEY	0.0	961.0	2500.0
13205643	CAPITOL VIEW	0.0	460.0	300.0
13206090	NEW DRY CREEK	0.0	1296.0	3000.0
13206092	NEW UNION	0.0	0.0	1400.0
13206205	ELEMP	0.0	0.0	0.0
13206220	EWARM SPRINGS	0.0	0.0	0.0
13206260	EGRAHAM-GILBERT	0.0	0.0	0.0
13206265	BALLENTYNE	0.0	376.0	1300.0
13206270	ECONWAY-HAMMING	0.0	0.0	0.0
13206274	EAGLE ISLAND PARK	0.0	0.0	0.0
13206290	ETHOMAS AIKEN	0.0	0.0	0.0
13206292	EMACE-CATLIN	0.0	0.0	0.0
13206295	EMACE-MACE	0.0	0.0	0.0
13208450	EHART-DAVIS	0.0	0.0	0.0
13208710	MIDDLETON	0.0	0.0	11000.0
13208738	BARBER	0.0	0.0	0.0
13208740	ESEVEN SUCKERS	0.0	0.0	0.0
13209480	PHYLLIS	21018.0	25582.0	16000.0
13209482	EUREKA #1	0.0	0.0	2800.0
13209630	LITTLE PIONEER	0.0	2174.0	500.0
13209990	CANYON COUNTY	0.0	0.0	6000.0
13210005	CALDWELL HIGHLINE	0.0	0.0	0.0
13210049	SMIDDLETON FLOW	0.0	0.0	0.0
13210984	SRIVERSIDE	0.0	0.0	0.0
13210989	SPIONEER DIXIE	0.0	0.0	0.0
13210992	SEBREE	1227.0	0.0	0.0
13210993	SCAMPBELL	0.0	0.0	0.0
13210994	SSIEBENBERG	0.0	0.0	0.0
13210999	SCALDWELL FLOW	0.0	0.0	0.0
13211001	SSHIPLEY	0.0	0.0	0.0
13211003	SWAGNER	0.0	0.0	0.0
13211603	SSIMPLOT	0.0	0.0	0.0
13211725	SEUREKA #2	0.0	0.0	0.0
13211735	SUPPER CENTER POINT	0.0	0.0	0.0
13211745	SMCMANUS AND TEATER	0.0	0.0	0.0
13211825	SLOWER CENTER POINT	0.0	0.0	0.0
13212499	SNOTUS FLOW	0.0	0.0	0.0
13212548	SBOWMAN AND SWISHER	0.0	0.0	0.0
13212645	SBAXTER	0.0	0.0	0.0
13212832	SANDREWS	0.0	0.0	0.0
13212896	SMAMMON	0.0	0.0	0.0
13212938	SHAAS	0.0	0.0	0.0
13212954	SPARMA	0.0	0.0	0.0
13212966	SISLAND HIGHLINE	0.0	0.0	0.0
13212992	SCRAWFORTH	0.0	0.0	0.0
13212994	SMCCONNEL ISLAND	0.0	0.0	0.0
99999050	MANDERSON DAM POWER	0.0	4200.0	0.0
99999100	MEAGLE IS CANALS	0.0	0.0	7650.0
99999150	MIDAHO FISH & GAME	0.0	0.0	50000.0
99999200	MARROWROCK FLOW	22800.0	0.0	0.0
99999250	MLUCKY PEAK FLOW	0.0	0.0	102300.0