SNKHSTIST10Y.FOR

```
0001    C  ************************************************
0002    C     THIS PROGRAM LISTS DATA FROM WD-01 HISTORY FILES
0003    C                  (OR SIMILAR FILES)
0004    C          IN USGS STYLE RECORDS BY WATER YEAR
0005    C
0006    C              WRITTEN USING VAX FORTRAN
0007    C
0008    C       ADAPTED TO UPPER SNAKE SEPTEMBER 1988 - RJS
0009    C
0010    C ************************************************
0011    C
0012          LOGICAL*1 FILES(40)
0013          BYTE CNTRL
0014          INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
0015          REAL MEAN,MAX,MIN,NOVALUE
0016          CHARACTER NAME*50,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
0017          CHARACTER TTL*50,BLANK*1,NNAME*50
0018          DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
0019          DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12)
0020          DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
0021          CNTRL='00'X
0022          NOVALUE=9999999.
0023          BLANK=' '
0024          OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
0025          OPEN(UNIT=8,FILE='SNKHSTLST.OUT',STATUS='NEW',RECL=133)
0026        500 PRINT *,'*****************************************************'
0027          PRINT *,'                                                     '
0028          PRINT *,'   YOU HAVE A CHOICE:                                '
0029          PRINT *,'                                                     '
0030          PRINT *,'   LIST GROUPS OF DIVERSIONS, FLOWS, EXCHANGE PUMPS, '
0031          PRINT *,'   DRAINS, OR RESERVOIRS  (STATION NUMBERS AND TITLES'
0032          PRINT *,'   READ IN FROM FILES)                               '
0033          PRINT *,'                                                     '
0034          PRINT *,'                             OR                      '
0035          PRINT *,'                                                     '
0036          PRINT *,'   LIST STATIONS INDIVIDUALLY                        '
0037          PRINT *,'   (YOU PROVIDE THE STATION NUMBER AND TITLE)        '
0038          PRINT *,'                                                     '
0039          PRINT *,'   TO LIST GROUPS , ENTER  G                         '
0040          PRINT *,'   TO LIST INDIVIDUALLY, ENTER  I                    '
0041          PRINT *,'   TO QUIT, ENTER  Q                                 '
0042          PRINT *,'                                                     '
0043          PRINT *,'*****************************************************'
0044          PRINT *,'                                                     '
0045          PRINT 904
0046        904 FORMAT('         ENTER CHOICE:'$)
0047          ACCEPT 902, IND
0048        902 FORMAT(A1)
0049          IF( IND .EQ. 'Q' ) GO TO 470
0050          IF( IND .EQ. 'I' ) GO TO 911
0051          PRINT *,'                                                     '
0052          PRINT *,'*****************************************************'
0053          PRINT *,'                                                     '
0054          PRINT *,'   YOU HAVE ANOTHER CHOICE:                          '
0055          PRINT *,'                                                     '
0056          PRINT *,'   TO LIST DIVERSIONS, ANSWER  D                     '
0057          PRINT *,'   TO LIST ALL RIVER FLOWS, ANSWER  F                '
```

```
0058          PRINT *,'   TO LIST ALL EXCHANGE PUMPS, ANSWER  E             '
0059          PRINT *,'   TO LIST ALL RESERVOIRS, ANSWER  R                 '
0060          PRINT *,'   TO LIST REACH GAINS, ANSWER  G                   '
0061          PRINT *,'   TO ENTER FILE NAME OF STATION TITLES, ANSWER T    '
0062          PRINT *,'   TO QUIT, ANSWER  Q  '
0063          PRINT *,'                                                    '
0064          PRINT *,'****************************************************'
0065          PRINT *,'                                                    '
0066          PRINT 904
0067          ACCEPT 902, IND
0068          IF( IND .EQ. 'Q' ) GO TO 470
0069          IF( IND .NE. 'D' ) GO TO 911
0070          PRINT *,'                                                    '
0071          PRINT *,'****************************************************'
0072          PRINT *,'                                                    '
0073          PRINT *,'   YOU HAVE STILL ANOTHER CHOICE:                   '
0074          PRINT *,'                                                    '
0075          PRINT *,'   TO LIST ALL 300 DIVERSIONS, ANSWER  D            '
0076          PRINT *,'   TO LIST ALL CANAL DIVERSIONS, ANSWER  C          '
0077          PRINT *,'   TO LIST ALL PUMP DIVERSIONS, ANSWER  P           '
0078          PRINT *,'   TO LIST DIVERSIONS TO LORENZO, ANSWER  L         '
0079          PRINT *,'   TO LIST HENRYS FORK DIVERSIONS, ANSWER  H        '
0080          PRINT *,'   TO LIST DIVERSIONS LORENZO TO BLACKFOOT, ANSWER  B'
0081          PRINT *,'   TO LIST DIVERSIONS BLACKFOOT TO MILNER, ANSWER M  '
0082          PRINT *,'   TO QUIT, ANSWER  Q  '
0083          PRINT *,'                                                    '
0084          PRINT *,'****************************************************'
0085          PRINT *,'                                                    '
0086          PRINT 904
0087          ACCEPT 902, INK
0088          IF( INK .EQ. 'Q' ) GO TO 470
0089      911 PRINT *,' '
0090          PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
0091          PRINT *,'   WATER YEAR FOR WHICH DATA IS TO BE LISTED -'
0092          PRINT *,' '
0093          PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR WATER YEAR  '
0094          PRINT *,'   1990, ENTER 89-90   '
0095          PRINT *,'               __-__   '
0096          PRINT 901
0097      901 FORMAT('+              '$)
0098          KY=0
0099          ACCEPT 900, NP,NP1
0100      900 FORMAT(I2,1X,I2)
0101          PRINT *,' '
0102          PRINT *,'   TO CREATE A FILE OF RECORDS WITH MONTHLY TOTALS IN'
0103          PRINT *,'   ACRE-FEET FOR EACH STATION - FORMAT(I6,I2,12F6.1),'
0104          PRINT 909
0105      909 FORMAT('    ANSWER Y/N: '$)
0106          ACCEPT 902, IMON
0107          PRINT *,' '
0108          IF( IMON .EQ. 'Y') THEN
0109                      OPEN(UNIT=9,FILE='SNKMON.OUT',STATUS='NEW',RECL=80)
0110                      PRINT *,'   MONTHLY ACRE FEET WILL BE WRITTEN TO '
0111                      PRINT *,'   FILE "SNKMON.OUT".                   '
0112                      PRINT *,' '
0113                      ENDIF
0114      C
```

```
0115    C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
0116    C
0117          NP=NP+1900
0118          NP1=NP1+1900
0119          IF(JMOD(NP,4).EQ.0) KY=1
0120          ID=(NP*1000)+274+KY
0121          KY=0
0122          IF(JMOD(NP1,4).EQ.0) KY=1
0123          ID1=(NP1*1000)+273+KY
0124    C
0125    C***** READ STATION NUMBERS FOR INDIVIDUALLY ENTERED STATIONS
0126    C
0127          IF( IND .EQ. 'I' ) THEN
0128                        OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
0129      1                 PRINT *,'   ENTER STATION NUMBER AND TITLE '
0130                        PRINT *,'   TO END   ENTER   99999999  '
0131                        PRINT *,'            ________          '
0132                        PRINT 903
0133    903                 FORMAT('+           '$)
0134                        ACCEPT 905, ISTA,TTL
0135    905                 FORMAT(I8,A50)
0136                        IF( ISTA .EQ. 99999999 ) GO TO 9
0137                        WRITE(3,906) ISTA,TTL
0138    906                 FORMAT(I8,A50)
0139                        GO TO 1
0140      9                 REWIND 3
0141                        GO TO 10
0142                        END IF
0143          IF( IND .NE. 'T' ) GO TO 8
0144          PRINT 907
0145      907 FORMAT('     ENTER FILE NAME:'$)
0146          ACCEPT 908, IC,(FILES(I),I=1,IC)
0147      908 FORMAT(Q,40A1)
0148          CALL ASSIGN(3,FILES)
0149          GO TO 2
0150        8 IF(IND.NE.'D') GO TO 5
0151          IF(INK.EQ.'D') OPEN(UNIT=3,FILE='SNKDIV.TTL',STATUS='OLD')
0152          IF(INK.EQ.'C') OPEN(UNIT=3,FILE='SNKCAN.TTL',STATUS='OLD')
0153          IF(INK.EQ.'P') OPEN(UNIT=3,FILE='SNKPMP.TTL',STATUS='OLD')
0154          IF(INK.EQ.'L') OPEN(UNIT=3,FILE='SNKTOL.TTL',STATUS='OLD')
0155          IF(INK.EQ.'H') OPEN(UNIT=3,FILE='SNKHEN.TTL',STATUS='OLD')
0156          IF(INK.EQ.'B') OPEN(UNIT=3,FILE='SNKLTB.TTL',STATUS='OLD')
0157          IF(INK.EQ.'M') OPEN(UNIT=3,FILE='SNKBTM.TTL',STATUS='OLD')
0158          GO TO 10
0159        5 IF(IND.EQ.'F') OPEN(UNIT=3,FILE='SNKFLO.TTL',STATUS='OLD')
0160          IF(IND.EQ.'E') OPEN(UNIT=3,FILE='SNKEXC.TTL',STATUS='OLD')
0161          IF(IND.EQ.'R') OPEN(UNIT=3,FILE='SNKRES.TTL',STATUS='OLD')
0162          IF(IND.EQ.'G') OPEN(UNIT=3,FILE='SNKRCHGN.TTL',STATUS='OLD')
0163        2 INSTA=0
0164          IEND=0
0165          STATION_NUM=0
0166       10 IF (INSTA.EQ.1) THEN
0167                           STA_ID=NSTA_ID
0168                           NAME=NNAME
0169                           INSTA=0
0170                           GO TO 15
0171                           ENDIF
```

```
0172          READ(3,301,END=450) STA_ID,NAME
0173        301 FORMAT(I8,A50)
0174      C
0175      C***** FILL ARRAY WITH MISSING VALUE INDICATORS
0176      C
0177         15 DO 20 I=1,12
0178            DO 20 J=1,31
0179         20 DV(I,J)=NOVALUE
0180      C
0181      C***** IF STATION NUMBERS MATCH THEN PROCESS DATA,
0182      C***** SKIP READ STATEMENT IF FIRST RECORD ALREADY READ
0183      C
0184          IF(STA_ID.EQ.STATION_NUM) GO TO 60
0185          IF(STATION_NUM.GT.STA_ID) GO TO 651
0186      C
0187      C***** READ ANOTHER RECORD FROM DATA FILE
0188      C
0189         25 READ(7,701,END=99) STATION_NUM,TYPE,YEAR,ND,VALUE
0190        701 FORMAT(I8,A1,I4,I3,F7.0)
0191          IF(STATION_NUM.LT.STA_ID) GO TO 25
0192          IF(STATION_NUM.EQ.STA_ID) GO TO 60
0193        651 IF(TD.NE.STA_ID) THEN
0194                             PRINT 601,STA_ID
0195        601                  FORMAT(1X,'STATION ',I8,' NOT FOUND')
0196                             GO TO 10
0197                             END IF
0198          READ(3,301,END=99) NSTA_ID,NNAME
0199          INSTA=1
0200         35 IF(STATION_NUM.LT.NSTA_ID) THEN
0201                    READ(7,701,END=99) STATION_NUM,TYPE,YEAR,ND,VALUE
0202                    GO TO 35
0203                    ENDIF
0204          GO TO 100
0205      C
0206      C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
0207      C
0208         60 IDATE=(YEAR*1000)+ND
0209            IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 25
0210            JD(2)=28
0211            IF( JMOD(YEAR,4) .EQ. 0 ) JD(2)=29
0212            LD=ND
0213            MON=0
0214         62 MON=MON+1
0215            IF(MON .GT. 12 ) GO TO 68
0216            LD=LD-JD(MON)
0217            IF( LD .GT. 0 ) GO TO 62
0218            LD=LD+JD(MON)
0219            DV(MON,LD)=VALUE
0220      C
0221      C***** SET VARIABLES TO PRESENT VALUES IN PREPARATION FOR
0222      C***** READING NEXT RECORD
0223      C
0224         68 CONTINUE
0225            TD=STATION_NUM
0226            P_YEAR=YEAR
0227            P_TYPE=TYPE
0228            GO TO 25
```

```
0229     C
0230     C***** ENTIRE YEAR FOR ONE STATION READ IN -
0231     C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
0232     C
0233        99 IEND=1
0234       100 LEAP=0
0235           IF(JMOD(NP1,4).EQ.0) LEAP=1
0236           IF( P_TYPE .NE. 'R' ) GO TO 120
0237               DELTA_S(1)=DV(1,31)-DV(12,31)
0238               DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
0239               DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
0240               DELTA_S(4)=DV(4,30)-DV(3,31)
0241               DELTA_S(5)=DV(5,31)-DV(4,30)
0242               DELTA_S(6)=DV(6,30)-DV(5,31)
0243               DELTA_S(7)=DV(7,31)-DV(6,30)
0244               DELTA_S(8)=DV(8,31)-DV(7,31)
0245               DELTA_S(9)=DV(9,30)-DV(8,31)
0246               DELTA_S(10)=NOVALUE
0247               DELTA_S(11)=DV(11,30)-DV(10,31)
0248               DELTA_S(12)=DV(12,31)-DV(11,30)
0249             DO 119 I=1,12
0250       119 IF( ABS(DELTA_S(I)) .GE. 8000000. ) DELTA_S(I)=NOVALUE
0251       120 DO 135 I=1,12
0252           D=31
0253           IF( I .EQ. 2 ) D=28+LEAP
0254           IF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.9.OR.I.EQ.11) D=30
0255           CFSD(I)=DV(I,1)
0256           D_ADJ=0
0257           IF( DV(I,1) .EQ. NOVALUE )THEN
0258                                       CFSD(I)=0.0
0259                                       D_ADJ=1
0260                                       END IF
0261           MAX(I)=DV(I,1)
0262           IF( DV(I,1) .EQ. NOVALUE ) MAX(I)=-1.0
0263           MIN(I)=DV(I,1)
0264           DO 130 J=2,D
0265           DZ=DV(I,J)
0266           IF( DV(I,J) .EQ. NOVALUE ) THEN
0267                                       DZ=0.0
0268                                       D_ADJ=D_ADJ+1
0269                                       END IF
0270           CFSD(I)=CFSD(I)+DZ
0271           IF( DV(I,J) .EQ. NOVALUE ) THEN
0272                                       GO TO 129
0273                                       END IF
0274           IF( DV(I,J) .GT. MAX(I) ) MAX(I)=DV(I,J)
0275       129 IF( DV(I,J) .LT. MIN(I) ) MIN(I)=DV(I,J)
0276       130 CONTINUE
0277           IF( D_ADJ .GE. D ) THEN
0278                              CFSD(I)=NOVALUE
0279                              DELTA_S(I)=NOVALUE
0280                              MEAN(I)=NOVALUE
0281                              GO TO 133
0282                              END IF
0283           MEAN(I)=CFSD(I)/(D-D_ADJ)
0284       133 ACFT(I)=CFSD(I)*1.9835
0285       135 CONTINUE
```

```
0286          CALL DATE(RUNDATE)
0287          WRITE(8,801) RUNDATE
0288      801 FORMAT('1',/,120X,A9)
0289          IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
0290      800 FORMAT(1H+,'PRELIMINARY DATA')
0291          WRITE(8,802) TD,NAME
0292      802 FORMAT(//,40X,I8,6X,A50)
0293          IF( P_TYPE .EQ. 'R' ) THEN
0294               WRITE(8,803) NP,NP1
0295      803      FORMAT(33X,'CONTENTS IN ACRE FEET, WATER YEAR OCTOBER',
0296         *     I5,' TO SEPTEMBER',I5)
0297               GO TO 160
0298                               ELSE
0299                               WRITE(8,804) NP,NP1
0300      804 FORMAT(33X,'DISCHARGE, CUBIC FEET PER SECOND, WATER YEAR',
0301         *          ' OCTOBER',I5,' TO SEPTEMBER',I5)
0302                               WRITE(8,805)
0303      805 FORMAT(59X,'MEAN VALUES')
0304                               END IF
0305      160 WRITE(8,806)
0306      806 FORMAT(//,1X,'DAY',11X,'OCT',7X,'NOV',7X,'DEC',7X,'JAN',7X,
0307         *          'FEB',7X,'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',
0308         *          7X,'AUG',7X,'SEP')
0309          WRITE(8,807)
0310      807 FORMAT(' ')
0311          DO 180 J=1,31
0312          IF( J .EQ.6 .OR. J .EQ. 11 .OR. J .EQ. 16 .OR. J .EQ. 21
0313         *       .OR. J .EQ. 26 ) THEN
0314                                WRITE(8,807)
0315                                  END IF
0316          DO 170 I=1,12
0317          II=I-3
0318          IF(II.LE.0) II=II+12
0319          IF( II .EQ. 10 ) WRITE(8,807)
0320          IF( II .EQ. 10 ) WRITE(8,811) CNTRL,J
0321      811                  FORMAT(A1,I3,'    ')
0322          IF( DV(II,J) .EQ. NOVALUE ) THEN
0323                                 WRITE(8,809) CNTRL
0324      809                        FORMAT(A1,'       ---')
0325                                 GO TO 170
0326                                 END IF
0327          IF(ABS(DV(II,J)).LT.10.0) THEN
0328                                WRITE(8,808) CNTRL,DV(II,J)
0329      808                       FORMAT(A1,F10.1)
0330                                GO TO 170
0331                                END IF
0332          IDV=JNINT(DV(II,J))
0333          WRITE(8,810) CNTRL,IDV
0334      810 FORMAT(A1,I10)
0335      170 CONTINUE
0336      180 CONTINUE
0337          IF( P_TYPE .EQ. 'R' ) THEN
0338                               WRITE(8,812)
0339      812                      FORMAT('0')
0340                               WRITE(8,813) CNTRL
0341      813                      FORMAT(A1,2X,'MAX  ')
0342                               DO 190 I=1,12
```

```
0343                                  II=I-3
0344                                  IF(II.LE.0) II=II+12
0345                                  IF( MAX(II) .EQ. -1.0 ) THEN
0346                                            WRITE(8,860) CNTRL
0347      860                                   FORMAT(A1,'          ')
0348                                            GO TO 190
0349                                            END IF
0350                                  IDV=JNINT(MAX(II))
0351                                  WRITE(8,810) CNTRL,IDV
0352      190                         CONTINUE
0353                                  WRITE(8,807)
0354                                  WRITE(8,814) CNTRL
0355      814                         FORMAT(A1,2X,'MIN  ')
0356                                  DO 200 I=1,12
0357                                  II=I-3
0358                                  IF(II.LE.0) II=II+12
0359                                  IF( MIN(II) .EQ. NOVALUE ) THEN
0360                                                    WRITE(8,860) CNTRL
0361                                                    GO TO 200
0362                                                    END IF
0363                                  IDV=JNINT(MIN(II))
0364                                  WRITE(8,810) CNTRL,IDV
0365      200                         CONTINUE
0366                                  WRITE(8,807)
0367                                  WRITE(8,815) CNTRL
0368      815                         FORMAT(A1,1X,'CHNG           ')
0369                                  DO 210 I=2,12
0370                                  II=I-3
0371                                  IF(II.LE.0) II=II+12
0372                                  IDV=JNINT(DELTA_S(II))
0373                                  IF( DELTA_S(II) .EQ. NOVALUE )THEN
0374                                                  WRITE(8,860) CNTRL
0375                                                  GO TO 210
0376                                                  END IF
0377                                  WRITE(8,810) CNTRL,IDV
0378      210                         CONTINUE
0379                                  GO TO 10
0380                                  END IF
0381          WRITE(8,812)
0382          WRITE(8,816) CNTRL
0383      816 FORMAT(A1,'TOTAL  ')
0384          DO 220 I=1,12
0385          II=I-3
0386          IF(II.LE.0) II=II+12
0387          IF( CFSD(II) .GE. NOVALUE ) THEN
0388                                        WRITE(8,860) CNTRL
0389                                        GO TO 220
0390                                        END IF
0391          IDV=JNINT(CFSD(II))
0392          WRITE(8,810) CNTRL,IDV
0393      220 CONTINUE
0394          WRITE(8,807)
0395          WRITE(8,818) CNTRL
0396      818 FORMAT(A1,' MEAN  ')
0397          DO 230 I=1,12
0398          II=I-3
0399          IF(II.LE.0) II=II+12
```

```
0400          IF( MEAN(II) .GE. NOVALUE ) THEN
0401                                        WRITE(8,860) CNTRL
0402                                        GO TO 230
0403                                        END IF
0404          IF(ABS(MEAN(II)).LT.10.0) THEN
0405                                       WRITE(8,808) CNTRL,MEAN(II)
0406                                       GO TO 230
0407                                       END IF
0408          IDV=JNINT(MEAN(II))
0409          WRITE(8,810) CNTRL,IDV
0410      230 CONTINUE
0411          WRITE(8,807)
0412          WRITE(8,819) CNTRL
0413      819 FORMAT(A1,'  MAX  ')
0414          DO 240 I=1,12
0415          II=I-3
0416          IF(II.LE.0) II=II+12
0417          IF( MAX(II) .EQ. -1.0 ) THEN
0418                                      WRITE(8,860) CNTRL
0419                                      GO TO 240
0420                                      END IF
0421          IF(ABS(MAX(II)).LT.10.) THEN
0422                                      WRITE(8,808) CNTRL,MAX(II)
0423                                      GO TO 240
0424                                      END IF
0425          IDV=JNINT(MAX(II))
0426          WRITE(8,810) CNTRL,IDV
0427      240 CONTINUE
0428          WRITE(8,807)
0429          WRITE(8,820) CNTRL
0430      820 FORMAT(A1,'  MIN  ')
0431          DO 250 I=1,12
0432          II=I-3
0433          IF(II.LE.0) II=II+12
0434          IF( MIN(II) .GE. NOVALUE ) THEN
0435                                       WRITE(8,860) CNTRL
0436                                       GO TO 250
0437                                       END IF
0438          IF(ABS(MIN(II)).LT.10.) THEN
0439                                      WRITE(8,808) CNTRL,MIN(II)
0440                                      GO TO 250
0441                                      END IF
0442          IDV=JNINT(MIN(II))
0443          WRITE(8,810) CNTRL,IDV
0444      250 CONTINUE
0445      C
0446      C         DO 260 I=1,12
0447      C         IFT(I)=JNINT(ACFT(I)/100.)
0448      C         IFT(I)=IFT(I)*100
0449      C260      CONTINUE
0450      C
0451          WRITE(8,807)
0452          WRITE(8,821) CNTRL
0453        821 FORMAT(A1,'AC-FT  ')
0454          DO 270 I=1,12
0455          II=I-3
0456          IF(II.LE.0) II=II+12
```

```
0457          IF( ACFT(II) .GE. 19834900. ) THEN
0458                                          WRITE(8,860) CNTRL
0459                                          GO TO 270
0460                                          END IF
0461     C       IF( ACFT(II) .LE. 999. ) THEN
0462     C                                  IFT(II)=JNINT(ACFT(II))
0463     C                                   WRITE(8,810) CNTRL,IFT(II)
0464     C                                  GO TO 270
0465     C                                  END IF
0466          IFT(II)=JNINT(ACFT(II))
0467          WRITE(8,810) CNTRL,IFT(II)
0468        270 CONTINUE
0469          YT=0.0
0470          DO 280 I=1,12
0471          IF( CFSD(I) .EQ. NOVALUE ) CFSD(I)=0.0
0472        280 YT=YT + CFSD(I)
0473          IYT=JNINT(YT)
0474          IYM=JNINT(YT/(365+LEAP))
0475     C       NT=JNINT(YT*1.9835/100. )
0476     C       IYA=NT*100
0477     C       IF( IYA .LT. 9999 ) IYA=1.9835*YT
0478          IYA=1.9835*YT
0479          WRITE(8,807)
0480          WRITE(8,831) NP1,IYT,IYM,IYA
0481        831 FORMAT(11X,'WATER YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
0482         1          I6,4X,'AC-FT',I8)
0483          IF(IMON.NE.'Y') GO TO 98
0484          TD=TD-13000000
0485          IYR=NP1-1900
0486          DO 3 I=1,12
0487          II=I-3
0488          IF(II.LE.0) II=II+12
0489          IF (ACFT(II).GT.19834900) ACFT(II)=0.0
0490          ACFT(II)=ACFT(II)/100.
0491        3 IFT(I)=JNINT(ACFT(II))
0492          WRITE(9,4) TD,IYR,(IFT(I),I=1,12)
0493        4 FORMAT(I6.6,I2,12I6)
0494     C
0495     C***** IF END OF DATA HAS NOT BEEN REACHED, RETURN TO READ NEXT STATION
0496     C
0497       98 IF(IEND.EQ.1) GO TO 450
0498          GO TO 10
0499        450 REWIND 7
0500          CLOSE (UNIT=3)
0501          PRINT *,'  '
0502          PRINT *,'  '
0503          PRINT *,'  REQUESTED OF STATIONS COMPLETE - RETURN TO START'
0504          PRINT *,'  '
0505          PRINT *,'  '
0506          GO TO 500
0507        470 PRINT *,'                                        '
0508          PRINT *,'  OUTPUT IS  THE LATEST VERSION OF SNKHSTLST.OUT'
0509          STOP
0510          END
```

I6

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	6487	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	2951	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	3284	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	12722	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTLSTWY$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	CHAR	BLANK		17 23=
**	L*1	CNTRL		13 21= 320 323 328 333 340 346 351 354 360 367 374 377 382 388 395 401 405 409 412 422 426 429 435 439 452 458 467
**	R*4	D		252= 253= 254= 264 277 283
**	R*4	DZ		265= 267= 270
**	I*4	D_ADJ		14 256= 259= 268(2)= 277 283
**	I*4	I		146(2)= 177= 179 249= 250(2) 251= 253 254(4) 255(2) 257 258 262(2) 263(2) 265 266 270(2) 274(4) 275(4) 278 279 280 284(2) 316= 317 342= 343 357 369= 370 384= 385 398 414= 415 431= 432 455 470= 471(2) 472 486= 491 492(2)=
2-00000848	I*4	IC		146(2)=
2-00000840	I*4	ID		120= 209
**	I*4	ID1		123= 209
**	I*4	IDATE		208= 209(2)
**	I*4	IDV		332= 333 350= 351 363= 364 372= 377 391= 392 408= 425= 426 442= 443
2-00000850	I*4	IEND		164= 233= 497
**	I*4	II		317= 318(3)= 319 320 322 327 328 332 343= 344(3)= 345 357= 358(3)= 359 363 370= 372 373 385= 386(3)= 387 398= 399(3)= 400 404 405 415= 416(3)= 417 421 422

```
                                                 432=      433(3)=   434       438       439
                                                 455=      456(3)=   457       466(2)    467
                                                 488(3)=   489(2)    490(2)    491
  2-0000083C  I*4   IMON                          106=      108       483
  2-0000082C  I*4   IND                            47=       49        50        67=       68        69
                                                 127       143       150       159       160
                                                 162       289(2)

  2-00000830  I*4   INK                           87=       88        151       152       153       154
                                                 155       156       157
  2-0000084C  I*4   INSTA                        163=      166       169=      199=
  2-00000844  I*4   ISTA                         134=      136       137
      **      I*4   IYA                          478=      480
      **      I*4   IYM                          474=      480

      **      I*4   IYR                          485=      492
      **      I*4   IYT                          473=      480
      **      I*4   J                            178=      179       264=      265       266       271
                                                 274(2)    275(2)    311=      312(5)    320
                                                 327       328       332
      **      I*4   KY                           98=       119=      120       121=      122=      123
      **      I*4   LD                          212=      216(2)=   217       218(2)=   219

      **      I*4   LEAP                        234=      235=      238       239       253       474
      **      I*4   MON                         213=      214(2)=   215       216       218       219
  2-00000778  CHAR  NAME                         16       168=      172=      291
  2-00000858  I*4   ND                          189=      201=      208       212
  2-000007E7  CHAR  NNAME                        17       168       198=

      **      R*4   NOVALUE                      15        22=      179       246       250       257
                                                 262       266       271       278       279
                                                 322       359       373       387       400
                                                 471
  2-00000834  I*4   NP                           99=      117(2)=   119       120       294       299
  2-00000838  I*4   NP1                          99=      118(2)=   122       123       235       294
                                                 299       480       485
  2-00000854  I*4   NSTA_ID                     167       198=      200
  2-000007AB  CHAR  P_TYPE                       16       227=      236       293       337

      **      I*4   P_YEAR                       14       226=
  2-000007AC  CHAR  RUNDATE                      16       286A      287
  2-00000824  I*4   STATION_NUM                  14       165=      184       185       189=      191
                                                 192       200       201=      225
  2-00000820  I*4   STA_ID                       14       167=      172=      184       185       191
                                                 192       193       194
  2-0000081C  I*4   TD                           14       193       225=      291       484(2)=   492

      **      CHAR  TITLE                        16
  2-000007B5  CHAR  TTL                          17       134=      137
  2-000007AA  CHAR  TYPE                         16       189=      201=      227
  2-0000085C  R*4   VALUE                       189=      201=      219
  2-00000828  I*4   YEAR                         14       189=      201=      208       211       226

      **      R*4   YT                          469=      472(2)=   473       474       478
      **      CHAR  ZLCH                         16
```

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-000006F0	R*4	ACFT		48	(12)	19 284= 457 466 489(2)= 490(2)= 491
2-00000690	R*4	CFSD		48	(12)	18 255= 258= 270(2)= 278= 283 284 387 391 472
2-000006C0	R*4	DELTA_S		48	(12)	19 237= 238= 239= 240= 241= 242= 243= 244= 246= 247= 248= 250(372 373
2-000000C0	R*4	DV		1488	(12, 31)	18 179= 219= 237(2) 238(2) 239(2) 240(2) 241(2) 242(244(2) 245(2) 247(2) 248(257 261 262 263 266 271 274(2) 275(327 328 332
2-00000750	L*1	FILES		40	(40)	12 146= 148A
2-00000720	I*4	IFT		48	(12)	19 466= 467 491= 492
2-00000090	I*4	JD		48	(12)	18 20D 210= 211= 216 218
2-00000030	R*4	MAX		48	(12)	15 18 261= 262= 274(2)= 345 350 417 421 425
2-00000000	R*4	MEAN		48	(12)	15 19 280= 283= 400 404 405 408
2-00000060	R*4	MIN		48	(12)	15 18 263= 275(2)= 359 363 434 438 439

LABELS

Address	Label	References
0-00000910	1	129# 139
0-00000B38	2	149 163#
**	3	486 491#
1-00000B7D	4'	492 493#
0-00000AEC	5	150 159#
0-00000A58	8	143 150#
0-000009E0	9	136 140#
0-00000B41	10	141 158 166# 196 379 498
0-00000B8C	15	170 177#
**	20	177 178 179#
0-00000C27	25	189# 191 209 228
0-00000CE0	35	200# 202
0-00000BC4	60	184 192 208#
0-00000BFC	62	214# 217
0-00000C1E	68	215 224#
0-00001868	98	483 497#

```
  0-00000D3C  99                189       198       201       233#
  0-00000D40  100               204       234#
      **      119               249       250#
  0-00000DFB  120               236       251#

  0-00000EA0  129               272       275#
      **      130               264       276#
  0-00000EE1  133               281       284#
      **      135               251       285#
  0-00000FDF  160               297       305#

  0-0000113B  170               316       325       330       335#
      **      180               311       336#
  0-000011F3  190               342       348       352#
  0-00001297  200               356       361       365#
  0-0000133D  210               369       375       378#

  0-000013E7  220               384       389       393#
  0-000014CB  230               397       402       406       410#
  0-000015B3  240               414       419       423       427#
  0-0000169B  250               431       436       440       444#
  0-00001749  270               454       459       468#

      **      280               470       472#
  1-00000952  301'              172       173#      198
  0-00001874  450               172       497       499#
  0-00001918  470                49        68        88       507#
  0-00000020  500                26#      506

  1-00000963  601'              194       195#
  0-00000C82  651               185       193#
  1-00000957  701'              189       190#      201
  1-00000987  800'              289       290#
  1-0000097E  801'              287       288#

  1-0000099D  802'              291       292#
  1-000009A8  803'              294       295#
  1-000009E9  804'              299       300#
  1-00000A37  805'              302       303#
  1-00000A47  806'              305       306#

  1-00000AA5  807'              309       310#      314       319       353       366
                                                394       411       428       451       479
  1-00000AC3  808'              328       329#      405       422       439
  1-00000AB4  809'              323       324#
  1-00000AC9  810'              333       334#      351       364       377       392
                                                409       426       443       467
  1-00000AA9  811'              320       321#

  1-00000ACE  812'              338       339#      381
  1-00000AD2  813'              340       341#
  1-00000AED  814'              354       355#
  1-00000AF9  815'              367       368#
  1-00000B10  816'              382       383#

  1-00000B1C  818'              395       396#
  1-00000B28  819'              412       413#
```

```
  1-00000B34  820'          429       430#
  1-00000B40  821'          452       453#
  1-00000B4C  831'          480       481#

  1-00000ADE  860'          346       347#      360       374       388       401
                                      418       435       458
  1-000008FE  900'           99       100#
  1-000008EA  901'           96        97#
  1-000008E7  902'           47        48#       67        87       106
  1-00000919  903'          132       133#

  1-000008CE  904'           45        46#       66        86
  1-00000929  905'          134       135#
  1-0000092E  906'          137       138#
  1-00000933  907'          144       145#
  1-0000094C  908'          146       147#

  1-00000905  909'          104       105#
  0-00000681  911            50        69        89#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                      References

        ASSIGN                    148
        FOR$CLOSE                 500
        FOR$DATE_T_DS             286
        FOR$OPEN                   24        25       109       128       151       152
                                            153       154       155       156       157
                                            160       161       162
```

```
+-----------------------------------------------+
|           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 SNKHSTLSTWY

  /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.SNAKE]SNKHSTLSTWY.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTLSTWY.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           19.80 seconds
  Elapsed Time:       30.87 seconds
  Page Faults:        1501
  Dynamic Memory:     977 pages
```

```
0001    C  ************************************************
0002    C     THIS PROGRAM LISTS DATA FROM WD-01 HISTORY FILES
0003    C                (OR SIMILAR FILES)
0004    C       IN USGS STYLE RECORDS BY IRRIGATION YEAR
0005    C
0006    C              WRITTEN USING VAX FORTRAN
0007    C
0008    C       ADAPTED TO UPPER SNAKE SEPTEMBER 1988 - RJS
0009    C
0010    C ************************************************
0011    C
0012          LOGICAL*1 FILES(40)
0013          BYTE CNTRL
0014          INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
0015          REAL MEAN,MAX,MIN,NOVALUE
0016          DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
0017          DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12),PRMTR(7)
0018          CHARACTER NAME*50,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
0019          CHARACTER TTL*50,BLANK*1,PRMTR*32
0020          DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
0021          DATA PRMTR/'DISCHARGE, CUBIC FEET PER SECOND',
0022         1           'CONTENTS IN ACRE FEET AT HR 2400',
0023         2           'GAGE HEIGHT IN FEET ABOVE DATUM ',
0024         3           'OBSERVED SHIFT IN FEET          ',
0025         4           'INTERPOLATED SHIFT IN FEET      ',
0026         5           'EVAPORATION OR ET IN INCHES     ',
0027         6           'PRECIPITATION IN INCHES         '/
0028          CNTRL='00'X
0029          NOVALUE=9999999.
0030          BLANK=' '
0031          OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
0032          OPEN(UNIT=8,FILE='SNKHSTLST.OUT',STATUS='NEW',RECL=133)
0033      500 PRINT *,'*****************************************************'
0034          PRINT *,'                                                     '
0035          PRINT *,'   YOU HAVE A CHOICE:                                '
0036          PRINT *,'                                                     '
0037          PRINT *,'   LIST GROUPS OF DIVERSIONS, FLOWS, EXCHANGE PUMPS, '
0038          PRINT *,'   DRAINS, OR RESERVOIRS  (STATION NUMBERS AND TITLES'
0039          PRINT *,'   READ IN FROM FILES)                               '
0040          PRINT *,'                                                     '
0041          PRINT *,'                            OR                       '
0042          PRINT *,'                                                     '
0043          PRINT *,'   LIST STATIONS INDIVIDUALLY                        '
0044          PRINT *,'   (YOU PROVIDE THE STATION NUMBER AND TITLE)        '
0045          PRINT *,'                                                     '
0046          PRINT *,'   TO LIST GROUPS , ENTER  G                         '
0047          PRINT *,'   TO LIST INDIVIDUALLY, ENTER  I                    '
0048          PRINT *,'   TO QUIT, ENTER  Q                                 '
0049          PRINT *,'                                                     '
0050          PRINT *,'*****************************************************'
0051          PRINT *,'                                                     '
0052          PRINT 904
0053      904 FORMAT('         ENTER CHOICE:'$)
0054          ACCEPT 902, IND
0055      902 FORMAT(A1)
0056          IF(IND.EQ.'Q') GO TO 470
0057          IF(IND.EQ.'I') GO TO 911
```

```
0058          PRINT *,'                                                  '
0059          PRINT *,'**********************************************************'
0060          PRINT *,'                                                  '
0061          PRINT *,'   YOU HAVE ANOTHER CHOICE:                       '
0062          PRINT *,'                                                  '
0063          PRINT *,'   TO LIST DIVERSIONS, ANSWER  D                  '
0064          PRINT *,'   TO LIST ALL RIVER FLOWS, ANSWER  F             '
0065          PRINT *,'   TO LIST ALL EXCHANGE PUMPS, ANSWER  E          '
0066          PRINT *,'   TO LIST ALL RESERVOIRS, ANSWER  R              '
0067          PRINT *,'   TO LIST REACH GAINS, ANSWER  G                 '
0068          PRINT *,'   TO ENTER FILE NAME OF STATION TITLES, ANSWER T  '
0069          PRINT *,'   TO QUIT, ANSWER  Q  '
0070          PRINT *,'                                                  '
0071          PRINT *,'**********************************************************'
0072          PRINT *,'                                                  '
0073          PRINT 904
0074          ACCEPT 902, IND
0075          IF(IND.EQ.'Q') GO TO 470
0076          IF(IND.NE.'D') GO TO 911
0077          PRINT *,'                                                  '
0078          PRINT *,'**********************************************************'
0079          PRINT *,'                                                  '
0080          PRINT *,'   YOU HAVE STILL ANOTHER CHOICE:                 '
0081          PRINT *,'                                                  '
0082          PRINT *,'   TO LIST ALL 300 DIVERSIONS, ANSWER  D          '
0083          PRINT *,'   TO LIST ALL CANAL DIVERSIONS, ANSWER  C        '
0084          PRINT *,'   TO LIST ALL PUMP DIVERSIONS, ANSWER  P         '
0085          PRINT *,'   TO LIST DIVERSIONS TO LORENZO, ANSWER  L       '
0086          PRINT *,'   TO LIST HENRYS FORK DIVERSIONS, ANSWER  H      '
0087          PRINT *,'   TO LIST DIVERSIONS LORENZO TO BLACKFOOT, ANSWER  B'
0088          PRINT *,'   TO LIST DIVERSIONS BLACKFOOT TO MILNER, ANSWER M  '
0089          PRINT *,'   TO QUIT, ANSWER  Q  '
0090          PRINT *,'                                                  '
0091          PRINT *,'**********************************************************'
0092          PRINT *,'                                                  '
0093          PRINT 904
0094          ACCEPT 902, INK
0095          IF(INK.EQ.'Q') GO TO 470
0096        911 PRINT *,' '
0097          PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
0098          PRINT *,'   IRRIGATION YEAR FOR WHICH DATA IS TO BE LISTED -'
0099          PRINT *,' '
0100          PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR IRRIGATION YEAR  '
0101          PRINT *,'   1990, ENTER 89-90    '
0102          PRINT *,'                __-__   '
0103          PRINT 901
0104        901 FORMAT('+               '$)
0105          KY=0
0106          ACCEPT 900, NP,NP1
0107        900 FORMAT(I2,1X,I2)
0108      C
0109      C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
0110      C
0111          NP=NP+1900
0112          NP1=NP1+1900
0113          IF(JMOD(NP,4).EQ.0) KY=1
0114          ID=(NP*1000)+305+KY
```

```
0115          KY=0
0116          IF(JMOD(NP1,4).EQ.0) KY=1
0117          ID1=(NP1*1000)+304+KY
0118      C
0119          IF(IND.EQ.'I') THEN
0120                         OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
0121        1                PRINT *,'   ENTER STATION NUMBER AND TITLE '
0122                         PRINT *,'   TO END   ENTER   99999999  '
0123                         PRINT *,'                    ________ '
0124                         PRINT 903
0125        903              FORMAT('+                   '$)
0126                         ACCEPT 905, ISTA,TTL
0127        905              FORMAT(I8,A50)
0128                         IF(ISTA.EQ.99999999) GO TO 9
0129                         WRITE(3,906) ISTA,TTL
0130        906              FORMAT(I8,A50)
0131                         GO TO 1
0132          9              REWIND 3
0133                         GO TO 11
0134                         END IF
0135          IF(IND.NE.'T') GO TO 8
0136          PRINT 907
0137        907 FORMAT('     ENTER FILE NAME:'$)
0138          ACCEPT 908, IC,(FILES(I),I=1,IC)
0139        908 FORMAT(Q,40A1)
0140          CALL ASSIGN(3,FILES)
0141          GO TO 11
0142          8 IF(IND.NE.'D') GO TO 2
0143            IF(INK.EQ.'D') OPEN(UNIT=3,FILE='SNKDIV.TTL',STATUS='OLD')
0144            IF(INK.EQ.'C') OPEN(UNIT=3,FILE='SNKCAN.TTL',STATUS='OLD')
0145            IF(INK.EQ.'P') OPEN(UNIT=3,FILE='SNKPMP.TTL',STATUS='OLD')
0146            IF(INK.EQ.'L') OPEN(UNIT=3,FILE='SNKTOL.TTL',STATUS='OLD')
0147            IF(INK.EQ.'H') OPEN(UNIT=3,FILE='SNKHEN.TTL',STATUS='OLD')
0148            IF(INK.EQ.'B') OPEN(UNIT=3,FILE='SNKLTB.TTL',STATUS='OLD')
0149            IF(INK.EQ.'M') OPEN(UNIT=3,FILE='SNKBTM.TTL',STATUS='OLD')
0150            GO TO 11
0151          2 IF(IND.EQ.'F') OPEN(UNIT=3,FILE='SNKFLO.TTL',STATUS='OLD')
0152            IF(IND.EQ.'E') OPEN(UNIT=3,FILE='SNKEXC.TTL',STATUS='OLD')
0153            IF(IND.EQ.'R') OPEN(UNIT=3,FILE='SNKRES.TTL',STATUS='OLD')
0154            IF(IND.EQ.'G') OPEN(UNIT=3,FILE='SNKRCHGN.TTL',STATUS='OLD')
0155            IF(IND.EQ.'G') THEN
0156                           INPCODE=1
0157                           GO TO 10
0158                           END IF
0159      C
0160      C***** CHOOSE FIELD (PARAMETER) TO BE LISTED
0161      C
0162         11 PRINT *,'                                                    '
0163            PRINT *,'****************************************************'
0164            PRINT *,'                                                    '
0165            PRINT *,'   YOU MUST NOW CHOOSE PARAMETER TO BE LISTED       '
0166            PRINT *,'                                                    '
0167            PRINT *,'   TO LIST DISCHARGE IN CFS, ANSWER  D              '
0168            PRINT *,'            CONTENT IN ACRE-FEET, ANSWER  C         '
0169            PRINT *,'            GAGE HEIGHT, ANSWER  G                  '
0170            PRINT *,'            OBSERVED SHIFT, ANSWER  O               '
0171            PRINT *,'            INTERPOLATED SHIFT, ANSWER  I           '
```

```
0172          PRINT *,'           EVAPORATION OR ET, ANSWER E              '
0173          PRINT *,'           PRECIPITATION OR , ANSWER P              '
0174          PRINT *,'   TO QUIT, ANSWER  Q  '
0175          PRINT *,'                                                    '
0176          PRINT *,'****************************************************'
0177          PRINT *,'                                                    '
0178          PRINT 904
0179          ACCEPT 902, INP
0180          IF(INP.EQ.'Q') GO TO 470
0181          IF(INP.EQ.'D') INPCODE=1
0182          IF(INP.EQ.'C') INPCODE=2
0183          IF(INP.EQ.'G') INPCODE=3
0184          IF(INP.EQ.'O') INPCODE=4
0185          IF(INP.EQ.'I') INPCODE=5
0186          IF(INP.EQ.'E') INPCODE=6
0187          IF(INP.EQ.'P') INPCODE=7
0188          STATION_NUM=0
0189       10 READ(3,301,END=450) STA_ID,NAME
0190      301 FORMAT(I8,A50)
0191      C
0192      C***** FILL ARRAY WITH MISSING VALUE INDICATORS
0193      C
0194          DO 20 I=1,12
0195          DO 20 J=1,31
0196       20 DV(I,J)=NOVALUE
0197          IF(STA_ID.EQ.STATION_NUM) GO TO 60
0198          IF(STA_ID.LT.STATION_NUM) GO TO 10
0199       25 IF(INPCODE.LE.2)
0200         1READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0201      701 FORMAT(I8,A1,I4,I3,F7.0)
0202          IF(INPCODE.EQ.3)
0203         1READ(7,702,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0204      702 FORMAT(I8,A1,I4,I3,7X,F7.0)
0205          IF(INPCODE.EQ.4.OR.INPCODE.EQ.6)
0206         1READ(7,703,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0207      703 FORMAT(I8,A1,I4,I3,14X,F7.0)
0208          IF(INPCODE.EQ.5.OR.INPCODE.EQ.7)
0209         1READ(7,704,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0210      704 FORMAT(I8,A1,I4,I3,21X,F7.0)
0211          IF(STA_ID.LT.STATION_NUM) THEN
0212                                    IF(TD.NE.STA_ID) THEN
0213                                                     PRINT 601,STA_ID
0214      601 FORMAT(1X,'STATION ',I8,' NOT FOUND')
0215                                                     GO TO 10
0216                                                     END IF
0217                                    GO TO 30
0218                                    END IF
0219          IF(STA_ID.EQ.STATION_NUM) GO TO 60
0220          IF(STA_ID.GT.STATION_NUM) GO TO 25
0221       30 IF(STA_ID.NE.STATION_NUM.AND.TD.EQ.STA_ID) GO TO 100
0222      C
0223      C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
0224      C
0225         60 IDATE=(YEAR*1000)+ND
0226          IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 25
0227          JD(2)=28
0228          IF(JMOD(YEAR,4).EQ.0) JD(2)=29
```

```
0229          LD=ND
0230          MON=0
0231       62 MON=MON+1
0232          IF(MON.GT.12) GO TO 68
0233          LD=LD-JD(MON)
0234          IF(LD.GT.0) GO TO 62
0235          LD=LD+JD(MON)
0236          DV(MON,LD)=VALUE
0237    C
0238    C***** SET VARIABLES TO PRESENT VALUES IN PREPARATION FOR
0239    C***** READING NEXT RECORD
0240    C
0241       68 CONTINUE
0242          TD=STATION_NUM
0243          P_YEAR=YEAR
0244          P_TYPE=TYPE
0245          GO TO 25
0246    C
0247    C***** ENTIRE YEAR FOR ONE STATION READ IN -
0248    C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
0249    C
0250      100 LEAP=0
0251          IF(JMOD(NP1,4).EQ.0) LEAP=1
0252          IF(INP.NE.'C'.AND.P_TYPE.NE.'R') GO TO 120
0253          DELTA_S(1)=DV(1,31)-DV(12,31)
0254          DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
0255          DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
0256          DELTA_S(4)=DV(4,30)-DV(3,31)
0257          DELTA_S(5)=DV(5,31)-DV(4,30)
0258          DELTA_S(6)=DV(6,30)-DV(5,31)
0259          DELTA_S(7)=DV(7,31)-DV(6,30)
0260          DELTA_S(8)=DV(8,31)-DV(7,31)
0261          DELTA_S(9)=DV(9,30)-DV(8,31)
0262          DELTA_S(10)=DV(10,31)-DV(9,30)
0263          DELTA_S(11)=NOVALUE
0264          DELTA_S(12)=DV(12,31)-DV(11,30)
0265          DO 119 I=2,12
0266      119 IF(ABS(DELTA_S(I)).GE.8000000.) DELTA_S(I)=NOVALUE
0267      120 DO 135 I=1,12
0268          D=31
0269          IF(I.EQ.2) D=28+LEAP
0270          IF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.9) D=30
0271          CFSD(I)=DV(I,1)
0272          D_ADJ=0
0273          IF(DV(I,1).EQ.NOVALUE) THEN
0274                                 CFSD(I)=0.0
0275                                 D_ADJ=1
0276                                 END IF
0277          MAX(I)=DV(I,1)
0278          IF(DV(I,1).EQ.NOVALUE) MAX(I)=-1.0
0279          MIN(I)=DV(I,1)
0280          DO 130 J=2,D
0281          DZ=DV(I,J)
0282          IF(DV(I,J).EQ.NOVALUE) THEN
0283                                     DZ=0.0
0284                                     D_ADJ=D_ADJ+1
0285                                     END IF
```

```
0286          CFSD(I)=CFSD(I)+DZ
0287          IF(DV(I,J).EQ.NOVALUE) THEN
0288                                       GO TO 129
0289                                       END IF
0290          IF(DV(I,J).GT.MAX(I)) MAX(I)=DV(I,J)
0291      129 IF(DV(I,J).LT.MIN(I)) MIN(I)=DV(I,J)
0292      130 CONTINUE
0293          IF(D_ADJ.GE.D) THEN
0294                              CFSD(I)=NOVALUE
0295                              DELTA_S(I)=NOVALUE
0296                              MEAN(I)=NOVALUE
0297                              GO TO 133
0298                              END IF
0299          MEAN(I)=CFSD(I)/(D-D_ADJ)
0300      133 ACFT(I)=CFSD(I)*1.9835
0301      135 CONTINUE
0302          CALL DATE(RUNDATE)
0303          WRITE(8,801) RUNDATE
0304      801 FORMAT('1',/,120X,A9)
0305          IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
0306      800 FORMAT(1H+,'PRELIMINARY DATA')
0307          WRITE(8,802) TD,NAME
0308      802 FORMAT(//,40X,I8,6X,A50)
0309          WRITE(8,804) PRMTR(INPCODE),NP,NP1
0310      804 FORMAT(30X,A32,', IRRIGATION YEAR',
0311         *          ' NOVEMBER',I5,' TO OCTOBER',I5)
0312          IF(INPCODE.EQ.1.AND.P_TYPE.NE.'R') WRITE(8,805)
0313          IF(INPCODE.GT.5) WRITE(8,805)
0314      805 FORMAT(59X,'MEAN VALUES')
0315      160 WRITE(8,806)
0316      806 FORMAT(//,1X,'DAY',11X,'NOV',7X,'DEC',7X,'JAN',7X,'FEB',7X,
0317         *          'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',7X,'AUG',
0318         *          7X,'SEP',7X,'OCT')
0319          WRITE(8,807)
0320      807 FORMAT(' ')
0321          DO 180 J=1,31
0322          IF(J.EQ.6.OR.J.EQ.11.OR.J.EQ.16.OR.J.EQ.21
0323         *     .OR.J.EQ.26) THEN
0324                                  WRITE(8,807)
0325                                    END IF
0326          DO 170 I=1,12
0327          II=I - 2
0328          IF(I.EQ.1) II=11
0329          IF(I.EQ.2) II=12
0330          IF(II.EQ.11) WRITE(8,807)
0331          IF(II.EQ.11) WRITE(8,811) CNTRL,J
0332      811                       FORMAT(A1,I3,'    ')
0333          IF(DV(II,J).EQ.NOVALUE) THEN
0334                                   WRITE(8,809) CNTRL
0335      809                          FORMAT(A1,'       ---')
0336                                   GO TO 170
0337                                   END IF
0338          IF(ABS(DV(II,J)).LT.10.AND.INPCODE.EQ.1.AND.
0339         * P_TYPE.NE.'R') THEN
0340                           WRITE(8,808) CNTRL,DV(II,J)
0341      808                  FORMAT(A1,F10.1)
0342                           GO TO 170
```

```
0343                          END IF
0344          IF(INPCODE.EQ.4) THEN
0345                 IF(ABS(DV(II,J)).LT.0.01.AND.DV(II,J).NE.0.0) THEN
0346                          WRITE(8,822) CNTRL,DV(II,J)
0347      822                 FORMAT(A1,F10.3)
0348                          GO TO 170
0349                          END IF
0350                 END IF
0351          IF(INPCODE.GT.2) THEN
0352                          WRITE(8,817) CNTRL,DV(II,J)
0353      817                 FORMAT(A1,F10.2)
0354                          GO TO 170
0355                          END IF
0356          IDV=JNINT(DV(II,J))
0357          WRITE(8,810) CNTRL,IDV
0358      810 FORMAT(A1,I10)
0359      170 CONTINUE
0360      180 CONTINUE
0361          IF(INP.EQ.'C'.OR.P_TYPE.EQ.'R') THEN
0362                        WRITE(8,812)
0363      812               FORMAT('0')
0364                        WRITE(8,813) CNTRL
0365      813               FORMAT(A1,2X,'MAX  ')
0366                        DO 190 I=1,12
0367                        II=I-2
0368                        IF(I.EQ.1) II=11
0369                        IF(I.EQ.2) II=12
0370                        IF(MAX(II).EQ.-1.0) THEN
0371                                  WRITE(8,860) CNTRL
0372      860                         FORMAT(A1,'          ')
0373                                  GO TO 190
0374                                  END IF
0375                        IDV=JNINT(MAX(II))
0376                        WRITE(8,810) CNTRL,IDV
0377      190               CONTINUE
0378                        WRITE(8,807)
0379                        WRITE(8,814) CNTRL
0380      814               FORMAT(A1,2X,'MIN  ')
0381                        DO 200 I=1,12
0382                        II=I-2
0383                        IF(I.EQ.1) II=11
0384                        IF(I.EQ.2) II=12
0385                        IF(MIN(II).EQ.NOVALUE) THEN
0386                                              WRITE(8,860) CNTRL
0387                                              GO TO 200
0388                                              END IF
0389                        IDV=JNINT(MIN(II))
0390                        WRITE(8,810) CNTRL,IDV
0391      200               CONTINUE
0392                        WRITE(8,807)
0393                        WRITE(8,815) CNTRL
0394      815               FORMAT(A1,1X,'CHNG           ')
0395                        DO 210 I=2,12
0396                        II=I-2
0397                        IF(I.EQ.2) II=12
0398                        IDV=JNINT(DELTA_S(II))
0399                        IF(DELTA_S(II).EQ.NOVALUE)THEN
```

```
0400                                                WRITE(8,860) CNTRL
0401                                                GO TO 210
0402                                                END IF
0403                          WRITE(8,810) CNTRL,IDV
0404          210             CONTINUE
0405                          GO TO 10
0406                          END IF
0407              IF(INPCODE.GT.2.AND.INPCODE.LT.6) GO TO 750
0408              WRITE(8,812)
0409              WRITE(8,816) CNTRL
0410          816 FORMAT(A1,'TOTAL  ')
0411              DO 220 I=1,12
0412              II=I-2
0413              IF(I.EQ.1) II=11
0414              IF(I.EQ.2) II=12
0415              IF(CFSD(II).GE.NOVALUE) THEN
0416                                      WRITE(8,860) CNTRL
0417                                      GO TO 220
0418                                      END IF
0419              IF(ABS(CFSD(II)).LT.10.AND.INPCODE.EQ.1) THEN
0420                                      WRITE(8,808) CNTRL,CFSD(II)
0421                                      GO TO 220
0422                                      END IF
0423              IF(INPCODE.GT.2) THEN
0424                                WRITE(8,817) CNTRL,CFSD(II)
0425                                GO TO 220
0426                                END IF
0427              IDV=JNINT(CFSD(II))
0428              WRITE(8,810) CNTRL,IDV
0429          220 CONTINUE
0430          750 WRITE(8,807)
0431              WRITE(8,807)
0432              WRITE(8,818) CNTRL
0433          818 FORMAT(A1,' MEAN  ')
0434              DO 230 I=1,12
0435              II=I-2
0436              IF(I.EQ.1) II=11
0437              IF(I.EQ.2) II=12
0438              IF(MEAN(II).GE.NOVALUE) THEN
0439                                      WRITE(8,860) CNTRL
0440                                      GO TO 230
0441                                      END IF
0442              IF(ABS(MEAN(II)).LT.10.AND.INPCODE.EQ.1) THEN
0443                                      WRITE(8,808) CNTRL,MEAN(II)
0444                                      GO TO 230
0445                                      END IF
0446              IF(INPCODE.GT.2) THEN
0447                                WRITE(8,817) CNTRL,MEAN(II)
0448                                GO TO 230
0449                                END IF
0450              IDV=JNINT(MEAN(II))
0451              WRITE(8,810) CNTRL,IDV
0452          230 CONTINUE
0453              WRITE(8,807)
0454              WRITE(8,819) CNTRL
0455          819 FORMAT(A1,'  MAX  ')
0456              DO 240 I=1,12
```

```
0457          II=I-2
0458          IF(I.EQ.1) II=11
0459          IF(I.EQ.2) II=12
0460          IF(MAX(II).EQ.-1.0) THEN
0461                              WRITE(8,860) CNTRL
0462                              GO TO 240
0463                              END IF
0464          IF(ABS(MAX(II)).LT.10.AND.INPCODE.EQ.1) THEN
0465                              WRITE(8,808) CNTRL,MAX(II)
0466                              GO TO 240
0467                              END IF
0468          IF(INPCODE.GT.2) THEN
0469                           WRITE(8,817) CNTRL,MAX(II)
0470                           GO TO 240
0471                           END IF
0472          IDV=JNINT(MAX(II))
0473          WRITE(8,810) CNTRL,IDV
0474      240 CONTINUE
0475          WRITE(8,807)
0476          WRITE(8,820) CNTRL
0477      820 FORMAT(A1,'  MIN  ')
0478          DO 250 I=1,12
0479          II=I-2
0480          IF(I.EQ.1) II=11
0481          IF(I.EQ.2) II=12
0482          IF(MIN(II).GE.NOVALUE) THEN
0483                                 WRITE(8,860) CNTRL
0484                                 GO TO 250
0485                                 END IF
0486          IF(ABS(MIN(II)).LT.10.AND.INPCODE.EQ.1) THEN
0487                                 WRITE(8,808) CNTRL,MIN(II)
0488                                 GO TO 250
0489                                 END IF
0490          IF(INPCODE.GT.2) THEN
0491                           WRITE(8,817) CNTRL,MIN(II)
0492                           GO TO 250
0493                           END IF
0494          IDV=JNINT(MIN(II))
0495          WRITE(8,810) CNTRL,IDV
0496      250 CONTINUE
0497          IF(INPCODE.GT.1) GO TO 10
0498          WRITE(8,807)
0499          WRITE(8,821) CNTRL
0500      821 FORMAT(A1,'AC-FT  ')
0501          DO 270 I=1,12
0502          II=I-2
0503          IF(I.EQ.1) II=11
0504          IF(I.EQ.2) II=12
0505          IF(ACFT(II).GE.19834900.) THEN
0506                                    WRITE(8,860) CNTRL
0507                                    GO TO 270
0508                                    END IF
0509          IFT(II)=JNINT(ACFT(II))
0510          WRITE(8,810) CNTRL,IFT(II)
0511      270 CONTINUE
0512          YT=0.0
0513          DO 280 I=1,12
```

```
0514          IF(CFSD(I).EQ.NOVALUE) CFSD(I)=0.0
0515       280 YT=YT + CFSD(I)
0516          IYT=JNINT(YT)
0517          IYM=JNINT(YT/(365+LEAP))
0518     C     NT=JNINT(YT*1.9835/100.)
0519     C     IYA=NT*100
0520     C     IF(IYA.LT.9999) IYA=1.9835*YT
0521          IYA=1.9835*YT
0522          WRITE(8,807)
0523          WRITE(8,831) NP1,IYT,IYM,IYA
0524       831 FORMAT(11X,'IRRIGATION YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
0525         *         I6,4X,'AC-FT',I8)
0526          GO TO 10      ! RETURN TO START ANOTHER STATION
0527       450 REWIND 7
0528          CLOSE (UNIT=3)
0529          PRINT *,'  '
0530          PRINT *,'  '
0531          PRINT *,'  REQUESTED OF STATIONS COMPLETE - RETURN TO START'
0532          PRINT *,'  '
0533          PRINT *,'  '
0534          GO TO 500
0535       470 PRINT *,'                                        '
0536          PRINT *,'  OUTPUT IS  THE LATEST VERSION OF SNKHSTLST.OUT'
0537          STOP
0538          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	7391	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	3125	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	3488	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	14004	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTLST$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	CHAR	BLANK		19 30=
**	L*1	CNTRL		13 28= 331 334 340 346 352 357 364 371 376 386 390 393 400 403 416 420 424 428 432 443 447 451 454 461

```
                                                            469       473       476       483       487
                                                            495       499       506       510
      **       R*4  D                      268=      269=      270=      280       293       299
      **       R*4  DZ                     281=      283=      286
      **       I*4  D_ADJ                   14       272=      275=      284(2)=   293       299

      **       I*4  I                      138(2)=   194=      196       265=      266(2)    267=
                                                            269       270(3)    271(2)    273       274
                                                            278(2)    279(2)    281       282       286(2)
                                                            290(4)    291(4)    294       295       296
                                                            300(2)    326=      327       328       329
                                                            367       368       369       381=      382
                                                            384       395=      396       397       411=
                                                            413       414       434=      435       436
                                                            456=      457       458       459       478=
                                                            480       481       501=      502       503
                                                            513=      514(2)    515
  2-000008F4   I*4  IC                     138(2)=
  2-000008E8   I*4  ID                     114=      226
  2-000008EC   I*4  ID1                    117=      226
      **       I*4  IDATE                  225=      226(2)

      **       I*4  IDV                    356=      357       375=      376       389=      390
                                                            398=      403       427=      428       450=
                                                            472=      473       494=      495
      **       I*4  II                     327=      328=      329=      330       331       333
                                                            338       340       345(2)    346       352
                                                            367=      368=      369=      370       375
                                                            383=      384=      385       389       396=
                                                            398       399       412=      413=      414=
                                                            419       420       424       427       435=
                                                            437=      438       442       443       447
                                                            457=      458=      459=      460       464
                                                            469       472       479=      480=      481=
                                                            486       487       491       494       502=
                                                            504=      505       509(2)    510
  2-000008D8   I*4  IND                     54=       56        57        74=       75        76
                                                            119       135       142       151       152
                                                            154       155       305(2)
  2-000008DC   I*4  INK                     94=       95       143       144       145       146
                                                            147       148       149
  2-000008FC   I*4  INP                    179=      180       181       182       183       184
                                                            185       186       187       252       361

  2-000008F8   I*4  INPCODE                156=      181=      182=      183=      184=      185=
                                                            186=      187=      199       202       205(2)
                                                            309       312       313       338       344
                                                            407(2)    419       423       442       446
                                                            468       486       490       497
  2-000008F0   I*4  ISTA                   126=      128       129
      **       I*4  IYA                    521=      523
      **       I*4  IYM                    517=      523
      **       I*4  IYT                    516=      523

      **       I*4  J                      195=      196       280=      281       282       287
                                                            290(2)    291(2)    321=      322(5)    331
                                                            338       340       345(2)    346       352
```

```
      **      I*4  KY                          105=      113=      114       115=      116=      117
      **      I*4  LD                          229=      233(2)=   234       235(2)=   236
      **      I*4  LEAP                        250=      251=      254       255       269       517
      **      I*4  MON                         230=      231(2)=   232       233       235       236

  2-00000858  CHAR NAME                         18       189=      307
  2-00000900  I*4  ND                          199=      202=      205=      208=      225       229
      **      R*4  NOVALUE                     15        29=      196       263       266       273
                                                          278       282       287       294       295
                                                          333       385       399       415       438
                                                          514
  2-000008E0  I*4  NP                          106=      111(2)=   113       114       309
  2-000008E4  I*4  NP1                         106=      112(2)=   116       117       251       309
                                                          523

  2-0000088B  CHAR P_TYPE                       18       244=      252       312       338       361
      **      I*4  P_YEAR                       14       243=
  2-0000088C  CHAR RUNDATE                      18       302A      303
  2-000008D0  I*4  STATION_NUM                  14       188=      197       198       199=      202=
                                                          205=      208=      211       219       220
                                                          242
  2-000008CC  I*4  STA_ID                       14       189=      197       198       211       212
                                                          213       219       220       221(2)

  2-000008C8  I*4  TD                           14       212       221       242=      307
      **      CHAR TITLE                        18
  2-00000895  CHAR TTL                          19       126=      129
  2-0000088A  CHAR TYPE                         18       199=      202=      205=      208=      244
  2-00000904  R*4  VALUE                       199=      202=      205=      208=      236

  2-000008D4  I*4  YEAR                         14       199=      202=      205=      208=      225
                                                          228       243
      **      R*4  YT                          512=      515(2)=   516       517       521
      **      CHAR ZLCH                         18
```

ARRAYS

```
    Address  Type  Name          Attributes      Bytes  Dimensions     References

  2-000006F0  R*4  ACFT                           48  (12)             17       300=      505       509
  2-00000690  R*4  CFSD                           48  (12)             16       271=      274=      286(2)=   294=
                                                                                          299       300       415       419
                                                                                          424       427       514(2)=   515
  2-000006C0  R*4  DELTA_S                       48  (12)             17       253=      254=      255=      256=
                                                                                          257=      258=      259=      260=
                                                                                          262=      263=      264=      266(
                                                                                          398       399
  2-000000C0  R*4  DV                          1488  (12, 31)         16       196=      236=      253(2)    254(2)
                                                                                          255(2)    256(2)    257(2)    258(
                                                                                          260(2)    261(2)    262(2)    264(
                                                                                          273       277       278       279
                                                                                          282       287       290(2)    291(
                                                                                          338       340       345(2)    346
                                                                                          356
  2-00000750  L*1  FILES                         40  (40)             12       138=      140A
```

```
  2-00000720  I*4   IFT                          48  (12)          17       509=      510
  2-00000090  I*4   JD                           48  (12)          16        20D      227=       228=       233
                                                                                        235
  2-00000030  R*4   MAX                          48  (12)          15        16       277=       278=       290(2)=
                                                                                        370        375        460        464
                                                                                        469        472
  2-00000000  R*4   MEAN                         48  (12)          15        17       296=       299=       438
                                                                                        442        443        447        450
  2-00000060  R*4   MIN                          48  (12)          15        16       279=       291(2)=    385
                                                                                        389        482        486        487
                                                                                        494

  2-00000778  CHAR  PRMTR                       224  (7)           17        19        21D       309


LABELS

    Address   Label                                       References

  0-0000080C  1                                             121#       131
  0-000009E8  2                                             142        151#
  0-00000954  8                                             135        142#
  0-000008DC  9                                             128        132#
  0-00000CA3  10                                            157        189#       198        215        405        497
                                                                            526

  0-00000A40  11                                            133        141        150        162#
      **      20                                            194        195        196#
  0-00000D67  25                                            199#       220        226        245
  0-00000F03  30                                            217        221#
  0-00000D04  60                                            197        219        225#

  0-00000D3C  62                                            231#       234
  0-00000D5E  68                                            232        241#
  0-00000F16  100                                           199        202        205        208        221        250#
      **      119                                           265        266#
  0-00000FDB  120                                           252        267#

  0-0000107C  129                                           288        291#
      **      130                                           280        292#
  0-000010BD  133                                           297        300#
      **      135                                           267        301#
      **      160                                           315#

  0-000013B3  170                                           326        336        342        348        354        359#
      **      180                                           321        360#
  0-0000147F  190                                           366        373        377#
  0-0000152F  200                                           381        387        391#
  0-000015D5  210                                           395        401        404#

  0-0000170F  220                                           411        417        421        425        429#
  0-0000184F  230                                           434        440        444        448        452#
  0-0000197B  240                                           456        462        466        470        474#
  0-00001AA7  250                                           478        484        488        492        496#
  0-00001B69  270                                           501        507        511#

      **      280                                           513        515#
```

```
  1-00000A2A  301'    189       190#
  0-00001BFC  450     189       527#
  0-00001CA0  470      56        75         95        180        535#
  0-00000024  500      33#       534

  1-00000A65  601'    213       214#
  1-00000A2F  701'    199       201#
  1-00000A3B  702'    202       204#
  1-00000A49  703'    205       207#
  1-00000A57  704'    208       210#

  0-00001715  750     407       430#
  1-00000A89  800'    305       306#
  1-00000A80  801'    303       304#
  1-00000A9F  802'    307       308#
  1-00000AAA  804'    309       310#

  1-00000ADE  805'    312       313        314#
  1-00000AEE  806'    315       316#
  1-00000B4C  807'    319       320#       324        330        378        392
                                        430        431        453        475        498
  1-00000B6A  808'    340       341#       420        443        465        487
  1-00000B5B  809'    334       335#

  1-00000B7C  810'    357       358#       376        390        403        428
                                        451        473        495        510
  1-00000B50  811'    331       332#
  1-00000B81  812'    362       363#       408
  1-00000B85  813'    364       365#
  1-00000BA0  814'    379       380#

  1-00000BAC  815'    393       394#
  1-00000BC3  816'    409       410#
  1-00000B76  817'    352       353#       424        447        469        491
  1-00000BCF  818'    432       433#
  1-00000BDB  819'    454       455#

  1-00000BE7  820'    476       477#
  1-00000BF3  821'    499       500#
  1-00000B70  822'    346       347#
  1-00000BFF  831'    523       524#
  1-00000B91  860'    371       372#       386        400        416        439
                                        461        483        506

  1-000009EA  900'    106       107#
  1-000009D6  901'    103       104#
  1-000009D3  902'     54        55#        74         94        179
  1-000009F1  903'    124       125#
  1-000009BA  904'     52        53#        73         93        178

  1-00000A01  905'    126       127#
  1-00000A06  906'    129       130#
  1-00000A0B  907'    136       137#
  1-00000A24  908'    138       139#
  0-00000689  911      57        76         96#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	140
	FOR$CLOSE	528
	FOR$DATE_T_DS	302
	FOR$OPEN	31 32 120 143 144 145 146 147 148 149 151 153 154

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 SNKHSTLST

/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.SNAKE]SNKHSTLST.LIS;3
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTLST.OBJ;1

COMPILATION STATISTICS

Run Time:	23.04 seconds
Elapsed Time:	32.30 seconds
Page Faults:	1797
Dynamic Memory:	1081 pages

```
0001    C          DATA SET WRDBILL    AT LEVEL 084 AS OF 06/20/85
0002    C                                                              C
0003    CC    WATER DISTRICT 01 ANNUAL BILLING PROGRAM (WRDBILL)       CC
0004    CCC                                                           CCC
0005    CCCC  OBJECTIVES:                                            CCCC
0006    CCCCC     (1) TO DISTRIBUTE EQUITABLY THE COST OF MANAGING THE WATER CCC
0007    CCCCCC        DISTRIBUTION SYSTEM                             CC
0008    CCCCC     (2) TO WRITE THE BILL OF EACH WATER USER             C
0009    CCCC                                                           C
0010    CCC  PROGRAM AND SUBPROGRAMS REQUIRED:                        CC
0011    CC                                                           CCC
0012    CCC  DEFINITIONS OF VARIABLES AND PARAMETERS                CCCC
0013    CC         IYR    - BILLING YEAR OR MANAGEMENT YEAR        CCCCC
0014    CCC        BSCOST - BASE OR MINIMUM CHARGE TO EACH WATER USER  CCCC
0015    CCCC       BSUSE  - BASE OR MAXIMUM WATER USE ALLOCATED FOR THE BASE  CCC
0016    CCCCCCCC            MINIMUM CHARGE TO EACH WATER USER.         CC
0017    CCCCCCCC   TOTBGT - TOTAL BUDGET OF THE DISTRICT FOR THE YEAR   C
0018    CCCCCCCC   C9COST - EXPENDITURES INCURRED BY THE COMMITTEE OF NINE  CC
0019    CCCCCCC    EXCFND - EXCESS FUND REBATE                         CCC
0020    CCCCCC     SFDA   - AVERAGE WATER USE RATE IN CFS PER ACRE     CCCC
0021    CCCCC      ILOW   - IDENTIFICATION CODE NUMBER SEPARATING THE LOWER   CCC
0022    C                   VALLEY CANALS FROM THE UPPER VALLEY CANALS.
0023    CC         NC     - TOTAL NUMBER OF MAJOR USER GROUPINGS/COUNTIES
0024    CCC        BILLYR - TOTAL WATER USE COST OR BILL OF THE SYSTEM 4 THE YR
0025    CCCC       GTWAT  - TOTAL WATER USE OF ALL THE INDIVIDUAL DIVERSION
0026    CCCCC               GREATER THAN THE MAXIMUM ALLOCATED USE FOR THE
0027    CCCCCC              MINIMUM CHARGE
0028    CCCCC      EBWAT  - GTWAT LESS THE TOTAL MAXIMUM ALLOCATED USE FOR THE
0029    CCCC                MINIMUM CHARGE OF ALL USERS WITH DIVERSION GREATER
0030    CCC                 THAN THAT OF THE MINIMUM CHARGE
0031    CC         LOCOST - COST FACTOR OF WATER USE IN $/CFS-DAY BY THE LOWER
0032    C                   VALLEY CANALS
0033    CC         C9FAC  - COST FACTOR OF WATER USE IN $/CFS-DAY BY THE
0034    CCC                 COMMITTEE OF NINE
0035    CCCC       UPCOST - COST FACTOR OF WATER USE IN $/CFS-DAY BY THE UPPER
0036    CCCCC               VALLEY CANALS, THIS IS = LOCOST+C9FAC ACTUALLY.
0037    CCCCCC     QNEW   - FINAL CHOICE OF THE BASE OR MAXIMUM WATER USE
0038    CCCCC               ALLOCATED FOR THE MINIMUM CHARGE.
0039    CCCC
0040    CCC    NOTE: DEFINITIONS OF OTHER PARAMETERS & VARIABLES ARE MADE AS
0041    CC           THEY ARE READ OR COMPUTED
0042    C
0043    CC    REMARKS:
0044    CCC          THIS PROGRAM WAS WRITTEN BY THE STAFF OF THE HYDROLOGY
0045    CCCC         SECTION, IDWR, STATE OF IDAHO AND INITIATED BY THE STAFF
0046    CCCCC*****   OF THE EASTERN DISTRICT, IDWR REGIONAL OFFICE, IDAHO FALLS.
0047    CCCC         THE GENERAL ALGORITHM OF THE ITERATION PROCESS WAS ALSO
0048    CCC          DEVELOPED BY THE STAFF OF THE EASTERN DISTRICT REGIONAL
0049    CC           OFFICE, IDWR, STATE OF IDAHO.
0050    C
0051          DIMENSION IGN(20),IDGN(20,20),GNAME(20,20,10),ITY(5000),ITX(5000)
0052          DIMENSION IDEN(5000),AREA(5000),SFC(5000),PYRCST(5000),USE(5000)
0053          DIMENSION INAME(5000,10),IDDRES(5000,20),DESCRP(5000,50),TOWN(9)
0054          DIMENSION WACOST(5000),TAB(20,20,10),COUNTY(20,5),DIV(5000)
0055          DIMENSION COSTYR(5000),TC(20,4),XCOUNT(20,20,2),PNO(5000)
0056          DOUBLE PRECISION UPCOST,C9FAC
0057          REAL*8 LOCOST
```

```
0058    C                                                               CCCC
0059          CALL ASSIGN(5,'BILLIN')
0060          CALL ASSIGN(6,'BILLOUT')
0061    C
0062    CCCCC**READ BILLING PARAMETERS*******************************CCCCCCCCC
0063    C                                                               CCCC
0064          READ(5,10) IYR,BSCOST,BSUSE,TOTBGT,C9COST,EXCFND,SFDA,ILOW,NC,IP
0065       10 FORMAT(2X,I4,F8.2,F6.0,3F10.2,F6.1,2I4,I3)
0066    C        IP   - INDICATOR TO TRIGGER PRINTING OF USER'S BILL
0067    CCCC              IF IP=0, NO PRINTING OF USER'S BILL IS PERFORMED
0068    CCCC
0069    C
0070          BILLYR=TOTBGT-EXCFND-C9COST
0071    C
0072    CC     READ PARAMETER CODES, IDENTIFICATIONS, AREAS, COST/CFS-DAY, USE,
0073    CCC         PREVIOUS YEARS' BILL ADJUSTMENT, USER'S NAME, ADDRESS,
0074    CCCC        AND LOCATION OF IRRIGATED ACCREAGE OR DESCRIPTION OF AREA.
0075    CCCCC
0076    CCCCCC  READ INPUTS AND PERFORM CALCULATIONS IN EVERY PASS.  AFTER
0077    CCCCCC  ONE COMPLETE PASS, THE READING OF INPUTS ARE TERMINATED
0078    CCCCC   AND THE COMPUTATIONS ARE CONTINUED UNTIL THE NEAREST VALUE
0079    CCCC    OF THE BASE MAXIMUM WATER USE FOR THE MINIMUM CHARGE IS
0080    CCC     DETERMINED.  IN NO EVENT SHALL THE NUMBER OF COMPLETE PASSES
0081    CC      EXCEEDS ONE HUNDRED TIMES (ACTUALLY, IT MAY TAKE ONLY 4 TIMES).
0082    C
0083          DO 30 IC=1,NC
0084          READ(5,20) IGN(IC),(COUNTY(IC,J),J=1,5)
0085       20 FORMAT(I5,10A4)
0086          NGO=IGN(IC)
0087          DO 30 IG=1,NGO
0088          DO 25 J=1,9
0089       25 TAB(IC,IG,J)=0.0
0090       30 READ(5,20) IDGN(IC,IG),(GNAME(IC,IG,J),J=1,10)
0091    C
0092          ITER=0
0093    C        IGN(K)   - NUMBER OF SUBGROUPS IN THE K'TH MAIN GROUP
0094    CC       COUNTY(K,5) - THE NAME OF K'TH MAIN GROUP- NOT MORE THAN 20 CHA
0095    CCC      IDGN(K,IG)  - NUMBER OF INDIVIDUAL USERS OR COMPANIES BELONGING
0096    CCCC                   TO THE K'TH MAIN GROUP AND IG'TH SUBGROUP.
0097    CCCCC                  THE PROGRAM IS WRITTEN SUCH THAT THIS NUMBER IS
0098    CCCCCC                 NOT REALLY REQUIRED.
0099    CCCCCC   GNAME(K,IG,10) - THE NAME OF THE IG'TH SUBGROUP OF THE K'TH
0100    CCCCC                  MAIN GROUP.  THERE IS SPACE FOR 40 COLUMNS OF
0101    CCCC                   CHARACTER, HOWEVER, ONLY 20 CHARACTERS ARE
0102    CCC                    ACTUALLY PRINTED IN THE SUMMARY OUTPUT.
0103    CC
0104    C
0105          EXLOW=0.0
0106          GTWAT=0.0
0107    C
0108          NB=0
0109          ENJ=0.0
0110          I=0
0111    C
0112    C        READ INPUT FILE.  THERE ARE FOUR  LINES OF 80 CHARACTERS EACH
0113    C             FOR EACH USER.  THE WATERUSERS ARE CLASSIFIED INTO
0114    C             MAIN GROUPS AND SUBGROUPS.  THE MAIN GROUPS COULD BE BY
```

```
0115    C             COUNTIES BUT IT DOESN'T HAVE TO BE AS IT COULD BE BY
0116    C             USAGE OR BY LOWER VALLEY OR UPPER VALLEY.  THE SUBGROUPS
0117    C             COULD BE BY COUNTIES, BY IRRIGATION COMPANIES, BY USAGE
0118    C             OR BY MAILING CONVENIENCE.  THE USER OF THE PROGRAM
0119    C             WOULD HAVE TO DECIDE HOW HE SHOULD GROUP THE INDIVIDUAL
0120    C             WATER USERS FOR CONVENIENCE OF MAILING OR BY COUNTY
0121    C             LAW REQUIREMENT.
0122    C
0123    C
0124    C
0125    C
0126    C
0127       50 I=I+1
0128          READ(5,60) ITY(I),ITX(I),IDEN(I),AREA(I),SFC(I),DIV(I),PYRCST(I),
0129         1PNO(I),(INAME(I,J),J=1,10)
0130       60 FORMAT(2I2,I6,F7.2,F4.1,F8.1,F8.2,F3.1,10A4)
0131          READ(5,61) (IDDRES(I,J),J=1,10),(DESCRP(I,J),J=1,50)
0132       61 FORMAT(20A4)
0133    C      ITY(I)          - MAIN GROUP CODE NUMBER - IST SUBSCRIPT OF TAB
0134    C      ITX(I)          - SUBGROUP CODE NUMBER - 2ND SUBSCRIPT OF TAB
0135    C      IDEN(I)         - USER'S IDENTIFICATION NO. OR ACCOUNT NO.
0136    C      AREA(I)         - USER'S IRRIGATED AREA IN ACRES
0137    C      SFC(I)          - CFS-DAYS(SEC-FT DAYS)/ACRE, WATER USE RATE
0138    C      DIV(I)          - USER'S DIVERSION IN CFS-DAYS IF SFC=ZERO.
0139    C      PYRCST(I)       - USER'S ADJUSTED BILL OF THE PREVIOUS YEAR
0140    C      PNO(I)          - NO. OF PUMPS BEING USED BY THE USER
0141    C      INAME(I,J)      - 40 CHARACTERS OF WATERUSER'S NAME
0142    C      IDDRES(I,J)     - 40 CHARACTERS OF WATERUSER'S ADDRESS
0143    C      DESCRP(I,J)     - 200 CHRCTRS OF THE LOCATION OF IRRIGATED AREA
0144    C
0145       70 K=ITY(I)
0146          L=ITX(I)
0147          IF(K.EQ.0.OR.K.GT.NC) GO TO 1000
0148          IF(L.LE.0) GO TO 1000
0149          IF(K.EQ.NC.AND.L.EQ.99) GO TO 80
0150          IF(L.GT.20) GO TO 1000
0151          EN=I
0152          PN=PNO(I)
0153          DVN=SFDA*AREA(I)
0154          IF(SFC(I).GT.0.0) DVN=SFC(I)*AREA(I)
0155          DDN=DIV(I)
0156          IF(SFC(I).EQ.0.0.AND.DDN.GT.0.0) DVN=DDN
0157          IF(DVN.GT.0.0.AND.PN.EQ.0.0) PN=1.0
0158          TAB(K,L,2)=TAB(K,L,2)+PN
0159          USE(I)=DVN
0160          TAB(K,L,1)=TAB(K,L,1)+DVN
0161          ENJ=ENJ+PN
0162          GO TO 50
0163    C
0164    C       ESTIMATE TOTAL WATER USAGE ABOVE BASE OF LOWER AND UPPER VALLEYS
0165    C
0166       80 CONTINUE
0167          NT=EN
0168          DO 90 I=1,NT
0169          K=ITY(I)
0170          L=ITX(I)
0171          DVN=USE(I)
```

```
0172          PN=PNO(I)
0173          IF(DVN.GT.0.0.AND.PN.EQ.0.0) PN=1.0
0174          TER=DVN - BSUSE*PN
0175          IF(TER.LE.0.0) GO TO 90
0176          GTWAT=GTWAT+DVN
0177          IF(K.EQ.ILOW) EXLOW=EXLOW+TER
0178          NPLO=PN
0179          NB=NB+NPLO
0180    C
0181          TAB(K,L,4)=TAB(K,L,4)+PN
0182    C
0183          TAB(K,L,5)=TAB(K,L,5)+USE(I)
0184    C
0185       90 CONTINUE
0186    C
0187    C       CALCULATE ANNUAL COST PARAMETERS
0188    C
0189    C
0190          BILMIN=ENJ*BSCOST
0191          BILL=BILLYR-BILMIN
0192          EB=NB
0193          EBWAT=GTWAT-EB*BSUSE
0194          LOCOST=BILL/EBWAT
0195          C9FAC=C9COST/(EBWAT-EXLOW)
0196          UPCOST=LOCOST+C9FAC
0197          QNEW=BSCOST/UPCOST
0198          QCH=BSUSE-QNEW
0199    C
0200          IF(ITER.GT.100) GO TO 2000
0201          IF(ABS(QCH).GT.0.0001) GO TO 210
0202    C
0203    C
0204          DO 100 K=1,NC
0205          NGO=IGN(K)
0206          DO 100 L=1,NGO
0207          TAB(K,L,3)=TAB(K,L,2)*BSCOST
0208          TAB(K,L,6)=TAB(K,L,4)*BSUSE
0209          TAB(K,L,7)=TAB(K,L,5)-TAB(K,L,6)
0210          TAB(K,L,8)=TAB(K,L,7)*UPCOST
0211          IF(K.EQ.ILOW) TAB(K,L,8)=TAB(K,L,7)*LOCOST
0212          TAB(K,L,9)=TAB(K,L,3)+TAB(K,L,8)
0213      100 CONTINUE
0214    C         PRINT RESULTS OF COMPUTATION ON SUMMARY TABLE
0215    C
0216          WRITE(6,110) IYR
0217      110 FORMAT(1H1//40X,I5,' ESTIMATED BILLING SUMMARY'/45X,'WATER DISTRIC
0218         1T NO. 1'/)
0219          NBS=BSCOST
0220          UBN=BSUSE + 0.5
0221          NBU=UBN
0222          WRITE(6,120) NBS,NBU,NBU,NBU,NBU
0223      120 FORMAT(35X,'TOTAL 24-HR NUMBER BASE COST NO. OF 24-HR CFS    COL.
0224         14   COL.5-COL.6   COST      TOTAL'/35X,'       CFS  OF DIV  $',
0225         2  I2,'/DIV. DIV>',I3,'  DIV>',I3,5X,'X ',I3,15X,'DIV>',I3,5X,
0226         3   'COST'/37X,'   COL. 1 COL. 2   COL. 3  COL. 4    COL. 5    COL.
0227         4 6     COL. 7     COL. 8     COL. 9'/)
0228    C
```

```
0229          DO 180 IC=1,NC
0230          NGO=IGN(IC)
0231          WRITE(6,130) (COUNTY(IC,J),J=1,5)
0232        130 FORMAT(10X,5A4)
0233          DO 150 IG=1,NGO
0234          WRITE(6,140) (GNAME(IC,IG,J),J=1,5),(TAB(IC,IG,J),J=1,9)
0235        140 FORMAT(14X,5A4,F12.1,F6.1,F10.2,F6.1,F12.1,F10.1,F11.1,2F11.2)
0236      C
0237        150 CONTINUE
0238          IF(NGO.EQ.1) GO TO 180
0239          DO 160 K=1,9
0240        160 TOWN(K)=0.0
0241          DO 170 K=1,9
0242          DO 170 IG=1,NGO
0243        170 TOWN(K)=TOWN(K)+TAB(IC,IG,K)
0244          WRITE(6,140) (COUNTY(IC,J),J=1,5),(TOWN(K),K=1,9)
0245      C
0246      C
0247        180 CONTINUE
0248      C
0249      C         COMPUTE GRAND TOTAL AND PRINT
0250      C
0251          DO 190 J=1,9
0252          TOWN(J)=0.0
0253          DO 190 IC=1,NC
0254          NGO=IGN(IC)
0255          DO 190 IG=1,NGO
0256        190 TOWN(J)=TOWN(J)+TAB(IC,IG,J)
0257          WRITE(6,200) (TOWN(J),J=1,9)
0258        200 FORMAT(/9X,'GRAND TOTAL',14X,F12.1,F6.1,F10.2,F6.1,F12.1,F10.1,
0259           1F11.1,2F11.2)
0260      C
0261          WRITE(6,203) LOCOST,C9FAC,UPCOST,BSCOST,BSUSE,TOTBGT,EXCFND,BILL
0262        203 FORMAT(//10X,
0263           * 'RATE, CANALS BELOW AMERICAN FALLS  =$',F12.10,'/CFS-DAY'/10X,
0264           1 'RATE FACTOR, COMM. OF NINE EXPENSE =$',F12.10,'/CFS-DAY'/10X,
0265           2 'RATE, CANALS ABOVE AMERICAN FALLS  =$',F12.10,'/CFS-DAY'/10X,
0266           3 'MINIMUM CHARGE PER WATERUSER       =$',6X,F6.2/10X,
0267           4 'ALLOCATED USE FOR MIN CHARGE       = ',6X,F6.1, ' CFS-DAYS'/10X,
0268           5 'TOTAL BUDGET OF THE DISTRICT       =$',2X,F10.2/10X,
0269           6 'EXCESS FUND REBATE                 =$',4X,F8.2/10X,
0270           7 'BUDGET - EXCESS - C9COST - MIN CHRG=$',2X,F10.2/)
0271          WRITE(6,204) GTWAT,EBWAT,EXLOW,C9COST
0272        204 FORMAT(//10X,
0273           1 'TOTAL DIVERSIONS GREATER ALLOCATED = ',2X,F10.2/10X,
0274           2 'TOTAL DIVERSIONS > ALLCTD - ALLCTD = ',2X,F10.2/10X,
0275           3 'TOT DIV LWR VALL > ALLCTD - ALLCTD = ',2X,F10.2/10X,
0276           4 'COMMITTEE OF NINE EXPENDITURES     = ',2X,F10.2/)
0277      CC****REMINDER: C9FAC=C9COST/(EBWAT-EXLOW)****************************
0278          GO TO 230
0279        210 CONTINUE
0280          ITER=ITER+1
0281          NB=0
0282          BSUSE=QNEW
0283      C
0284          EXLOW=0.0
0285          GTWAT=0.0
```

```
0286    C
0287          DO 220 IC=1,NC
0288          NGO=IGN(IC)
0289          DO 220 IG=1,NGO
0290          DO 220 J=3,9,1
0291      220 TAB(IC,IG,J)=0.0
0292    C
0293          GO TO 80
0294    C
0295      230 CONTINUE
0296    C
0297    C         PRINT USER'S BILL......INSERT ROUTINE TO WRITE DESIRED
0298    C                                OUTPUT CONFIGURATIONS..........
0299    C
0300          TOTUSE=0.0
0301          TOTCST=0.0
0302          IF(IP.EQ.0) GO TO 2000
0303          I=0
0304      240 I=I+1
0305          PN=PNO(I)
0306          IF(USE(I).GT.0.0.AND.PN.EQ.0.0) PN=1.0
0307          TER=USE(I)-BSUSE*PN
0308          IF(TER.LE.0.0) GO TO 260
0309          ID=ITY(I)
0310          IF(ID.EQ.ILOW) GO TO 250
0311          WACOST(I)=BSCOST*PN+TER*UPCOST
0312          GO TO 270
0313      250 WACOST(I)=BSCOST*PN+TER*LOCOST
0314          GO TO 270
0315      260 WACOST(I)=BSCOST*PN
0316          IF(USE(I).EQ.0.0) WACOST(I)=0.0
0317      270 COSTYR(I)=WACOST(I)+PYRCST(I)
0318          TOTUSE=TOTUSE+USE(I)
0319          TOTCST=TOTCST+COSTYR(I)
0320          IF(I.LT.NT) GO TO 240
0321          IPYR=IYR-1
0322          WRITE(6,280) IYR,IYR,IPYR,IYR
0323      280 FORMAT(1H1/45X,'ESTIMATED BILLING -',I5,' IRRIGATION SEASON'//11X,
0324         1' USERS NAME AND ',64X,'PMP   WATER USE  ',I4,5X,I4,5X,I4/3X,
0325         2 'TAX NO.  ADDRESS',43X,'DESCRIPTION',17X,' NO.  (CFS-DAYS)  CHARG
0326         3E   ADJST    BILL'/)
0327          DO 390 IC=1,NC
0328          NGO=IGN(IC)
0329          DO 285 J=1,4
0330      285 TC(IC,J)=0.0
0331          WRITE(6,290) (COUNTY(IC,J),J=1,5)
0332      290 FORMAT(//40X,5A4/)
0333          DO 350 IG=1,NGO
0334          I=0
0335          DO 295 J=1,2
0336      295 XCOUNT(IC,IG,J)=0.0
0337          WRITE(6,300) IC,IG,(GNAME(IC,IG,J),J=1,10)
0338      300 FORMAT(/20X,2I4,2X,10A4//)
0339      310 I=I+1
0340          K=ITY(I)
0341          L=ITX(I)
0342          IPN=PNO(I)
```

```
0343          IF(K.EQ.IC.AND.L.EQ.IG) GO TO 315
0344          IF(I.LT.NT) GO TO 310
0345          GO TO 330
0346        315 WRITE(6,320) IDEN(I),(INAME(I,J),J=1,10),(DESCRP(I,J),J=1,10),
0347         1   IPN,USE(I),WACOST(I),PYRCST(I),COSTYR(I)
0348        320 FORMAT(1X,I6,2(1X,10A4),  2X,I2,F9.1,F9.2,F10.2,F9.2)
0349          WRITE(6,325) (IDDRES(I,J),J=1,10),(DESCRP(I,J),J=11,20),(DESCRP(I,
0350         1J),J=21,50)
0351        325 FORMAT( 9X,10A4,1X,10A4/51X,10A4/51X,10A4/51X,10A4//)
0352      C
0353          XCOUNT(IC,IG,1)=XCOUNT(IC,IG,1)+PYRCST(I)
0354          XCOUNT(IC,IG,2)=XCOUNT(IC,IG,2)+COSTYR(I)
0355      C
0356          IF(I.GE.NT) GO TO 330
0357          GO TO 310
0358        330 CONTINUE
0359      C
0360          T1=TAB(IC,IG,1)
0361          T2=TAB(IC,IG,9)
0362          T3=XCOUNT(IC,IG,1)
0363          T4=XCOUNT(IC,IG,2)
0364      C
0365          WRITE(6,340)
0366        340 FORMAT(6X,'_____
0367         1_____
0368         2_____')
0369      C
0370          WRITE(6,360) (GNAME(IC,IG,J),J=1,10),T1,T2,T3,T4
0371      C
0372          WRITE(6,340)
0373          TC(IC,1)=TC(IC,1)+T1
0374          TC(IC,2)=TC(IC,2)+T2
0375          TC(IC,3)=TC(IC,3)+T3
0376          TC(IC,4)=TC(IC,4)+T4
0377        350 CONTINUE
0378          KG=NGO-1
0379          IF(KG.EQ.0) GO TO 390
0380      C
0381          DO 370 IG=1,KG
0382          T1=TAB(IC,IG,1)
0383          T2=TAB(IC,IG,9)
0384          T3=XCOUNT(IC,IG,1)
0385          T4=XCOUNT(IC,IG,2)
0386          WRITE(6,360) (GNAME(IC,IG,J),J=1,10),T1,T2,T3,T4
0387        360 FORMAT(/47X,'TOTAL  ',10A4,F8.0,F10.2,2F10.2/)
0388        370 CONTINUE
0389          WRITE(6,340)
0390          WRITE(6,380) (COUNTY(IC,J),J=1,5),(TC(IC,J),J=1,4)
0391        380 FORMAT(47X,'TOTAL  ',5A4,20X,F8.0,F10.2,2F10.2//)
0392          WRITE(6,340)
0393        390 CONTINUE
0394      C
0395      C           SUMMARIZE WATER USE AND WATER BILL OF THE SYSTEM
0396      C
0397          USEDIR=TAB(4,3,1)
0398          CSTDIR=TAB(4,3,9)
0399          ADJCST=XCOUNT(4,3,1)
```

```
0400          FNLCST=XCOUNT(4,3,2)
0401     C
0402          DO 395 K=5,NC
0403          USEDIR=USEDIR+TC(K,1)
0404          CSTDIR=CSTDIR+TC(K,2)
0405          ADJCST=ADJCST+TC(K,3)
0406        395 FNLCST=FNLCST+TC(K,4)
0407     C
0408          WRITE(6,400) USEDIR,CSTDIR,ADJCST,FNLCST
0409        400 FORMAT(58X,'TOTAL OWNER-USERS DIRECT CHARGE',5X,F8.0,3F10.2/)
0410          WRITE(6,340)
0411     C
0412     C
0413          WRITE(6,410) IPYR
0414        410 FORMAT(1H1///45X,'SUMMARY OF BILLING'/43X,'WATER DISTRICT NO. 1'/3
0415           15X,'WATER YEAR ENDING SEPTEMBER 30,',I5//65X,
0416           2'   TOTAL    REGULAR'/65X,
0417           3'   24-HR    COST OF'/65X,
0418           4'  SEC-FT    DELIVERY'/)
0419     C
0420          TTU=0.0
0421          TTC=0.0
0422          DO 450 IC=1,NC
0423          TAU=TC(IC,1)
0424          TAC=TC(IC,2)
0425          IF(IC.NE.4) GO TO 420
0426          TAU=TC(4,1)-TAB(4,3,1)
0427          TAC=TC(4,2)-TAB(4,3,9)
0428        420 TTU=TTU+TAU
0429          TTC=TTC+TAC
0430          IF(IC.EQ.4) WRITE(6,430) (COUNTY(IC,J),J=1,5),TAU,TAC
0431        430 FORMAT(20X,5A4,' INDIRECT ',11X,F12.1,F12.2/)
0432          IF(IC.NE.4) WRITE(6,440) (COUNTY(IC,J),J=1,5),TAU,TAC
0433        440 FORMAT(20X,5A4,21X,F12.1,F12.2/)
0434          IF(IC.NE.4) GO TO 450
0435          WRITE(6,460) TTU,TTC
0436          WRITE(6,445) (COUNTY(IC,J),J=1,5),TAB(4,3,1),TAB(4,3,9)
0437        445 FORMAT(/20X,5A4,' DIRECT CHARGE ',6X,F12.1,F12.2/)
0438        450 CONTINUE
0439        460 FORMAT(30X,'INDIRECT CHARGE',16X,F12.1,F12.2/)
0440          WRITE(6,470) USEDIR,CSTDIR
0441        470 FORMAT(30X,'DIRECT CHARGE',18X,F12.1,F12.2/)
0442          WRITE(6,480) TOTUSE,TOTCST
0443        480 FORMAT(24X,'GRAND TOTAL USE & GENERAL EXPENSE',4X,F12.1,F12.2///)
0444          GO TO 2000
0445       1000 L=I-1
0446          K=ITY(L)
0447     C
0448          WRITE(6,1010) ITY(L),ITX(L),IDEN(L),IGN(K),(INAME(L,J),J=1,10)
0449     C
0450       1010 FORMAT(//2X,'CHECK USERS FILE AFTER THIS:  ',2I3,2I6,2X,10A4//)
0451       2000 WRITE(6,2010) ITER
0452       2010 FORMAT(//2X,I5,' ITERATIONS WERE PERFORMED'/)
0453          STOP
0454          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	4230	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	2057	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	1857840	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1864127	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKBILL$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	ADJCST		399= 405(2)= 408
**	R*4	BILL		191= 194 261
**	R*4	BILLYR		70= 191
**	R*4	BILMIN		190= 191
2-001C5868	R*4	BSCOST		64= 190 197 207 219 261 311 313 315
2-001C586C	R*4	BSUSE		64= 174 193 198 208 220 261 282= 307
2-001C5874	R*4	C9COST		64= 70 195 271
**	R*8	C9FAC		56 195= 196 261
2-001C58D4	R*4	CSTDIR		398= 404(2)= 408 440
**	R*4	DDN		155= 156(2)
**	R*4	DVN		153= 154= 156= 157 159 160 171= 173 174 176
**	R*4	EB		192= 193
**	R*4	EBWAT		193= 194 195 271
2-001C58A8	R*4	EN		151= 167
**	R*4	ENJ		109= 161(2)= 190
2-001C5878	R*4	EXCFND		64= 70 261
2-001C5898	R*4	EXLOW		105= 177(2)= 195 271 284=
**	R*4	FNLCST		400= 406(2)= 408
2-001C589C	R*4	GTWAT		106= 176(2)= 193 271 285=
2-001C58A4	I*4	I		110= 127(2)= 128(9) 131(2) 145 146 151 152 153 154(3) 155 156 159 168= 169 170 171 172 183 303= 304(2)= 305 306 307 309 311 313 315 316(2) 317(3) 318 319 320 334= 339(2)= 340 341 342 344 346(349(3) 353 354 356 445
2-001C588C	I*4	IC		83= 84(2) 86 89 90(2) 229= 230 231 234(2) 243 244 253= 254 256 287= 288 291 327= 328 330 331 336 337(2) 343

				353(2)	354(2)	360	361	362	363	370	373(
				374(2)	375(2)	376(2)	382	383	384	385	386
				390(2)	422=	423	424	425	430(2)	432(2)	434
				436							
**	I*4	ID	309=	310							
**	I*4	IG	87=	89	90(2)	233=	234(2)	242=	243	255=	
				256	289=	291	333=	336	337(2)	343	353(
				354(2)	360	361	362	363	370	381=	382
				383	384	385	386				
2-001C5880	I*4	ILOW	64=	177	211	310					
2-001C5888	I*4	IP	64=	302							
**	I*4	IPN	342=	346							
**	I*4	IPYR	321=	322	413						
2-001C5894	I*4	ITER	92=	200	280(2)=	451					
2-001C5864	I*4	IYR	64=	216	321	322(3)					
**	I*4	J	84(2)=	88=	89	90(2)=	128(2)=	131(4)=	231(2)=	234(4)=	
				244(2)=	251=	252	256(3)	257(2)=	290=	291	329=
				330	331(2)=	335=	336	337(2)=	346(4)=	349(6)=	370(
				386(2)=	390(4)=	430(2)=	432(2)=	436(2)=	448(2)=		
**	I*4	K	145=	147(2)	149	158(2)	160(2)	169=	177	181(2)	
				183(2)	204=	205	207(2)	208(2)	209(3)	210(2)	211(
				212(3)	239=	240	241=	243(3)	244(2)=	340=	343
				402=	403	404	405	406	446=	448	
**	I*4	KG	378=	379	381						
**	I*4	L	146=	148	149	150	158(2)	160(2)	170=	181(2)	
				183(2)	206=	207(2)	208(2)	209(3)	210(2)	211(2)	212(
				341=	343	445=	446	448(4)			
**	R*8	LOCOST	57	194=	196	211	261	313			
2-001C58A0	I*4	NB	108=	179(2)=	192	281=					
**	I*4	NBS	219=	222							
**	I*4	NBU	221=	222(4)							
2-001C5884	I*4	NC	64=	83	147	149	204	229	253	287	
				327	402	422					
**	I*4	NGO	86=	87	205=	206	230=	233	238	242	
				254=	255	288=	289	328=	333	378	
**	I*4	NPLO	178=	179							
**	I*4	NT	167=	168	320	344	356				
**	R*4	PN	152=	157(2)=	158	161	172=	173(2)=	174	178	
				181	305=	306(2)=	307	311	313	315	
**	R*4	QCH	198=	201							
**	R*4	QNEW	197=	198	282						
2-001C587C	R*4	SFDA	64=	153							
**	R*4	T1	360=	370	373	382=	386				
**	R*4	T2	361=	370	374	383=	386				
**	R*4	T3	362=	370	375	384=	386				
**	R*4	T4	363=	370	376	385=	386				
**	R*4	TAC	424=	427=	429	430	432				
**	R*4	TAU	423=	426=	428	430	432				
**	R*4	TER	174=	175	177	307=	308	311	313		
2-001C5870	R*4	TOTBGT	64=	70	261						
2-001C58C8	R*4	TOTCST	301=	319(2)=	442						

2-001C58C4	R*4	TOTUSE	300=	318(2)=	442					
**	R*4	TTC	421=	429(2)=	435					
**	R*4	TTU	420=	428(2)=	435					
**	R*4	UBN	220=	221						
**	R*8	UPCOST	56	196=	197	210	261	311		
**	R*4	USEDIR	397=	403(2)=	408	440				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References					
2-00012F70	R*4	AREA		20000	(5000)	52	128=	153	154		
2-001BAE64	R*4	COSTYR		20000	(5000)	55	317=	319	346	354	
2-001B5EB4	R*4	COUNTY		400	(20, 5)	54	84=	231	244	331	
								390	430	432	43
2-000B8FB0	R*4	DESCRP		1000000	(5000, 50)	53	131=	346	349(2)		
2-001B6044	R*4	DIV		20000	(5000)	54	128=	155			
2-00000690	R*4	GNAME		16000	(20, 20, 10)	51	90=	234	337	370	
								386			
2-00057530	I*4	IDDRES		400000	(5000, 20)	53	131=	349			
2-0000E150	I*4	IDEN		20000	(5000)	52	128=	346	448		
2-00000050	I*4	IDGN		1600	(20, 20)	51	90=				
2-00000000	I*4	IGN		80	(20)	51	84=	86	205	230	
								254	288	328	44
2-000267F0	I*4	INAME		200000	(5000, 10)	53	128=	346	448		
2-00009330	I*4	ITX		20000	(5000)	51	128=	146	170	341	
								448			
2-00004510	I*4	ITY		20000	(5000)	51	128=	145	169	309	
								340	446	448	
2-001C0A44	R*4	PNO		20000	(5000)	55	128=	152	172	305	
								342			
2-0001CBB0	R*4	PYRCST		20000	(5000)	52	128=	317	346	353	
2-00017D90	R*4	SFC		20000	(5000)	52	128=	154(2)	156		
2-001B2034	R*4	TAB		16000	(20, 20, 10)	54	89=	158(2)=	160(2)=	181(2)=	
								183(2)=	207(2)=	208(2)=	20
								211(2)=	212(3)=	234	24
								291=	360	361	38
								397	398	426	42
2-001BFC84	R*4	TC		320	(20, 4)	55	330=	373(2)=	374(2)=	375(2)=	
								376(2)=	390	403	40
								406	423	424	42
2-001AD1F0	R*4	TOWN		36	(9)	53	240=	243(2)=	244	252=	
								256(2)=	257		
2-000219D0	R*4	USE		20000	(5000)	52	159=	171	183	306	
								307	316	318	34
2-001AD214	R*4	WACOST		20000	(5000)	54	311=	313=	315=	316=	
								317	346		
2-001BFDC4	R*4	XCOUNT		3200	(20, 20, 2)	55	336=	353(2)=	354(2)=	362	
								363	384	385	39

LABELS

Address	Label	References					
1-00000017	10'	64	65#				
1-00000030	20'	84	85#	90			
**	25	88	89#				
**	30	83	87	90#			
0-00000184	50	127#	162				
1-00000037	60'	128	130#				
1-00000051	61'	131	132#				
**	70	145#					
0-00000344	80	149	166#	293			
0-000003C5	90	168	175	185#			
**	100	204	206	213#			
1-00000056	110'	216	217#				
1-00000096	120'	222	223#				
1-0000019C	130'	231	232#				
1-000001A3	140'	234	235#	244			
**	150	233	237#				
**	160	239	240#				
**	170	241	242	243#			
0-0000066D	180	229	238	247#			
**	190	251	253	255	256#		
1-000001C4	200'	257	258#				
1-000001F1	203'	261	262#				
1-0000038F	204'	271	272#				
0-00000F84	210	201	279#				
**	220	287	289	290	291#		
**	230	278	295#				
0-000007B0	240	304#	320				
0-00000804	250	310	313#				
0-00000820	260	308	315#				
0-0000083C	270	312	314	317#			
1-0000044E	280'	322	323#				
**	285	329	330#				
1-00000506	290'	331	332#				
**	295	335	336#				
1-00000510	300'	337	338#				
0-000009B0	310	339#	344	357			
0-000009DC	315	343	346#				
1-00000520	320'	346	348#				
1-00000540	325'	349	351#				
0-00000B14	330	345	356	358#			
1-00000564	340'	365	366#	372	389	392	410
**	350	333	377#				
1-000005E7	360'	370	386	387#			
**	370	381	388#				
1-00000604	380'	390	391#				

```
  0-00000D15  390                         327       379       393#
      **      395                         402       406#
  1-00000623  400'                        408       409#
  1-00000652  410'                        413       414#
  0-00000E08  420                         425       428#

  1-000006F9  430'                        430       431#
  1-00000715  440'                        432       433#
  1-00000725  445'                        436       437#
  0-00000F25  450                         422       434       438#
  1-00000747  460'                        435       439#

  1-00000764  470'                        440       441#
  1-0000077F  480'                        442       443#
  0-00000FE4  1000                        147       148       150       445#
  1-000007B0  1010'                       448       450#
  0-0000105F  2000                        200       302       444       451#

  1-000007E5  2010'                       451       452#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ASSIGN	59	60

```
+-----------------------------------------------+
|           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 SNKBILL

  /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
```

SNKBILL$MAIN

3-Feb-1989 16:15:37 VAX FORTRAN V4.8-276
3-Feb-1989 16:14:41 HYDRO:[SUTTER]SNKBILL.FOR;

COMPILATION STATISTICS

Run Time:	23.96 seconds
Elapsed Time:	29.32 seconds
Page Faults:	2002
Dynamic Memory:	772 pages

SNKCOR NOT USED
(DUPLICATE DATA
ELIMINATED BY USING
/STABLE AND THEN
IN UPDATE PROGRAM)

```
0001    C
0002    C        **** PROGRAM TO ENTER RESULTS FROM WATER RIGHT ACCOUNTING TO
0003    C        PERMANENT ALLOCATIONS FILE.  IF DATA ALREADY EXISTS FOR STATION
0004    C        AND DAY BEING ENTERED, OLD DATA WILL BE REPLACED WITH NEW
0005    C        DATA. BOB SUTTER - OCTOBER 1986.
0006    C
0007    C       MODIFIED FOR WATER DISTRICT 01  -  AUGUST 1988
0008    C
0009           REAL*8 VAOLD(5),VANEW(5)
0010           INTEGER*4 ISTA,JSTA,IDATE,JDATE
0011    C
0012           CALL ASSIGN(1,'INPUT')
0013           CALL ASSIGN(2,'OLDALC')
0014           OPEN(UNIT=3,NAME='NEWALC',TYPE='NEW',
0015          1CARRIAGECONTROL='LIST')
0016    C
0017           JSKIP=0
0018           IREP=0
0019           INEW=0
0020           ISAM=0
0021           IN=0
0022           IR=0
0023           KW=1
0024    C
0025    C***** READ ONE RECORD FROM FROM INPUT FILE AND ONE RECORD
0026    C***** FROM OLD ALLOCATIONS FILE.  SKIP READING DATA FROM OLD
0027    C***** ALLOCATIONS FILE IF LAST INPUT DATA HAS BEEN MERGED WITH
0028    C***** DATA FROM OLD ALLOCATIONS FILE.
0029    C
0030        2 READ(1,1,END=10)ISTA,S,IDATE,(VANEW(I),I=1,5)
0031        1 FORMAT(I8,A1,I7,5F10.0)
0032        3 IF(JSKIP.EQ.0) READ(2,1,END=20)JSTA,T,JDATE,(VAOLD(I),I=1,5)
0033          JSKIP=0
0034    C
0035    C***** CHECK RELATIVE VALUE OF STATION NUMBERS OF OLD
0036    C***** AND NEW DATA.
0037    C
0038           IF(JSTA-ISTA) 4,5,6
0039    C
0040    C***** CHECK RELATIVE VALUE OF DATES OF OLD AND NEW
0041    C***** INPUT DATA.
0042    C
0043        5 IF(JDATE-IDATE) 4,7,6
0044    C
0045    C***** WRITE DATA FROM INPUT FILE TO NEW ALLOCATIONS FILE TO
0046    C***** EITHER REPLACE OR BE MERGED WITH OLD DATA.
0047    C
0048        6 JSKIP=1
0049          INEW=INEW+1
0050          IN=1
0051          GO TO 9
0052        7 IREP=IREP+1
0053          IR=1
0054        9 IF(S.EQ.'D') THEN
0055              KW=0
0056              DO 12 I=1,4
0057       12     IF(VANEW(I).GT.0.0) KW=1
```

```
0058              IF(KW.EQ.1) GO TO 23
0059              IF(IN.EQ.1) INEW=INEW-1
0060              IF(IR.EQ.1) IREP=IREP-1
0061              GO TO 23
0062              END IF
0063           IF(S.EQ.'F') THEN
0064              IVANEW5=VANEW(5)
0065              WRITE(3,18) ISTA,S,IDATE,(VANEW(I),I=1,4),IVANEW5
0066        18    FORMAT(I8,A1,I7,4F10.1,I10)
0067              GO TO 19
0068              END IF
0069        23 IF(KW.GT.0) WRITE(3,21) ISTA,S,IDATE,(VANEW(I),I=1,5)
0070        19 IN=0
0071           IR=0
0072           KW=1
0073           GO TO 2
0074     C
0075     C***** WRITE DATA RECORDS FROM OLD ALLOCATIONS FILE TO NEW
0076     C***** ALLOCATIONS FILE UNTIL DATA FROM INPUT FILE IS TO
0077     C***** REPLACE OLD DATA OR BE MERGED WITH OLD DATA.
0078     C
0079         4 ISAM=ISAM+1
0080           IF(S.EQ.'F') THEN
0081              IVAOLD5=VAOLD(5)
0082              WRITE(3,18) JSTA,T,JDATE,(VAOLD(I),I=1,4),IVAOLD5
0083              GO TO 3
0084              END IF
0085         8 WRITE(3,21) JSTA,T,JDATE,(VAOLD(I),I=1,5)
0086        21 FORMAT(I8,A1,I7,5F10.1)
0087           GO TO 3
0088     C
0089     C***** WRITE REMAINDER OF OLD ALLOCATIONS FILE TO NEW FILE
0090     C***** AFTER DATA INPUT FILE HAS BEEN EXHAUSTED.
0091     C
0092        10 IF(JSKIP.EQ.1) GO TO 14
0093        15 READ(2,1,END=100)JSTA,T,JDATE,(VAOLD(I),I=1,5)
0094        14 ISAM=ISAM+1
0095           IF(S.EQ.'F') THEN
0096              IVAOLD5=VAOLD(5)
0097              WRITE(3,18) JSTA,T,JDATE,(VAOLD(I),I=1,4),IVAOLD5
0098              GO TO 15
0099              END IF
0100        13 WRITE(3,21) JSTA,T,JDATE,(VAOLD(I),I=1,5)
0101           GO TO 15
0102     C
0103     C***** WRITE REMAINDER OF DATA INPUT FILE TO NEW ALLOCATIONS
0104     C***** FILE AFTER OLD ALLOCATIONS FILE HAS BEEN EXHAUSTED.
0105     C
0106        22 READ(1,1,END=100)ISTA,S,IDATE,(VANEW(I),I=1,5)
0107           KW=1
0108        20 INEW=INEW+1
0109           IF(S.EQ.'D') THEN
0110              KW=0
0111              DO 17 I=1,4
0112        17    IF(VANEW(I).GT.0.0) KW=1
0113              IF(KW.EQ.0) INEW=INEW-1
0114              GO TO 16
```

```
0115             END IF
0116          IF(S.EQ.'F') THEN
0117             IVANEW5=VANEW(5)
0118             WRITE(3,18) ISTA,S,IDATE,(VANEW(I),I=1,4),IVANEW5
0119             GO TO 22
0120             END IF
0121       16 IF(KW.GT.0) WRITE(3,21) ISTA,S,IDATE,(VANEW(I),I=1,5)
0122          GO TO 22
0123     C
0124     C***** ALL DATA HAS BEEN ADDED OR TRANSFERRED TO NEW ALLOCATIONS FILE.
0125     C
0126       100 TYPE *, ' ALLOCATIONS FILE HAS NOW BEEN UPDATED '
0127           ITOT=IREP+ISAM+INEW
0128           TYPE *, ' '
0129           TYPE 11, IREP,INEW,ISAM,ITOT
0130        11 FORMAT(/2X,'UPDATED FILE HAS',I10,3X,'RECORDS REPLACED'/
0131          1 18X,I10,3X,'NEW RECORDS'/18X,I10,3X,'RECORDS TRANSFERRED'/
0132          2 4X,'FOR A TOTAL OF',I10,3X,'ALLOCATIONS FILE RECORDS')
0133           TYPE *, ' '
0134       500 STOP
0135           END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1369	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	247	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	232	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	1848	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKALCUP$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(
2-00000058	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	11	
								121				
2-00000000	R*8	VAOLD		40	(5)	9	32=	81	82	85		
								93=	96	97	10	

LABELS

Address	Label	References				
1-00000044	1'	30	31#	32	93	106
0-00000034	2	30#	73			
0-00000084	3	32#	83	87		
0-00000318	4	38	43	79#		
**	5	38	43#			
0-00000220	6	38	43	48#		
0-00000217	7	43	52#			
**	8	85#				
0-00000228	9	51	54#			
0-000003C8	10	30	92#			
1-0000006A	11'	129	130#			
**	12	56	57#			
**	13	100#				
0-000003D0	14	92	94#			
0-00000473	15	93#	98	101		
0-00000168	16	114	121#			
**	17	111	112#			
1-00000050	18'	65	66#	82	97	118
0-0000030C	19	67	70#			
0-000000DC	20	32	108#			
1-0000005E	21'	69	85	86#	100	121
0-000001AE	22	106#	119	122		
0-000002C0	23	58	61	69#		

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

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 1ine       |
  +-----------------------------------------------+
```

COMMAND QUALIFIERS

```
  FORTRAN/LIST/CROSS_REFERENCE SNKALCUP

  /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:           7.37 seconds
  Elapsed Time:       9.03 seconds
  Page Faults:        818
  Dynamic Memory:     460 pages
```

```
00001   C     ************************************************************
00002   C
00003   C             PROGRAM SNKSEL - WATER RIGHT ACCOUNTING SELECT
00004   C
00005   C       THIS PROGRAM SELECTS FROM THE HISTORY FILE THE HYDROLOGIC
00006   C       DATA NEEDED FOR THE WATER RIGHT ACCOUNTING OVER A SPECIFIED
00007   C       PERIOD.  THESE DATA ARE BROKEN INTO SEPARATE FILES ACCORDING
00008   C       TO DATA TYPE (DIVERSION, FLOW, RESERVOIR) IN SORTED FORM
00009   C       SUCH THAT THEY CAN BE DIRECTLY READ BY THE ACCOUNTING PROGRAM.
00010   C
00011   C                       BOB SUTTER  -  JULY 1988
00012   C
00013   C     ************************************************************
00014   C
00015   C
00016         DIMENSION OM(12),JD(12),IYD(4),A(25),D(4)
00017         CHARACTER*7 D
00018         DATA OM/'OCT','NOV','DEC','JAN','FEB','MAR','APR','MAY','JUN',
00019        !  'JUL','AUG','SEP'/
00020         DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
00021         CALL ASSIGN(1,'HISTORY')
00022         OPEN (UNIT=2,NAME='HISTDIV',TYPE='NEW',CARRIAGECONTROL='LIST')
00023         OPEN (UNIT=3,NAME='HISTFLO',TYPE='NEW',CARRIAGECONTROL='LIST')
00024         OPEN (UNIT=4,NAME='HISTRVR',TYPE='NEW',CARRIAGECONTROL='LIST')
00025         OPEN (UNIT=5,NAME='HISTEXC',TYPE='NEW',CARRIAGECONTROL='LIST')
00026         OPEN (UNIT=7,NAME='ALOCDIV',TYPE='NEW',CARRIAGECONTROL='LIST')
00027         OPEN (UNIT=8,NAME='ALOCRVR',TYPE='NEW',CARRIAGECONTROL='LIST')
00028         OPEN (UNIT=9,NAME='ALOCMIS',TYPE='NEW',CARRIAGECONTROL='LIST')
00029         CALL ASSIGN(10,'ALOCTNS')
00030   C
00031         PRINT 1
00032       1 FORMAT (' ')
00033         PRINT *,' ********************************************************'
00034         PRINT *,' '
00035         PRINT *,'  THIS PROCEDURE SELECTS THE HYDROLOGIC DATA FROM THE  '
00036         PRINT *,'  HISTORY REQUIRED TO PERFORM THE WATER RIGHT ACCOUNTING'
00037         PRINT *,'  OVER A SELECTED PERIOD SPECIFIED BY A BEGINNING AND '
00038         PRINT *,'  ENDING DATE.  ANSWER FOLLOWING QUESTIONS USING Y FOR '
00039         PRINT *,'  YES, N FOR NO, AND C IF YOU WISH TO CANCEL ALL.  '
00040         PRINT *,' '
00041         PRINT *,' ********************************************************'
00042         PRINT 1
00043   C
00044       5 PRINT *,'  ENTER THE MONTH, DAY, AND YEAR FOR THE FIRST DAY YOU'
00045         PRINT *,'  WISH TO PERFORM THE ACCOUNTING. '
00046         PRINT *,'  FOR EXAMPLE : 06-12-1989  '
00047         PRINT 1
00048         PRINT *,'                ??????????  '
00049         PRINT 35
00050      35 FORMAT('+                '$)
00051         ACCEPT 2, IM1,ID1,IY1
00052       2 FORMAT(I2,1X,I2,1X,I4)
00053         PRINT 1
00054         PRINT 3, IM1,ID1,IY1
00055       3 FORMAT (1X,' THE FIRST ACCOUNTING DAY IS ',I2,'-',I2,'-',I4/
00056        !   2X,'IS THIS CORRECT?  ANSWER Y/N/C.  ')
00057         PRINT 36
```

```
00058      36 FORMAT(28X,' '$)
00059         ACCEPT 4,IANSWR
00060       4 FORMAT (1A1)
00061         PRINT 1
00062         IF (IANSWR.EQ.'C') GO TO 10
00063         IF (IANSWR.EQ.'N') GO TO 5
00064   C
00065         PRINT *,'  DO YOU WISH TO DO MORE THAN ONE DAY OF WATER RIGHT '
00066         PRINT *,'  ACCOUNTING?  ANSWER Y/N/C. '
00067         PRINT 36
00068         ACCEPT 4,IANSWR
00069         PRINT 1
00070         IF (IANSWR.EQ.'C') GO TO 10
00071         IF (IANSWR.EQ.'N') GO TO 9
00072   C
00073       7 PRINT *,'  ENTER THE MONTH, DAY, AND YEAR FOR THE FINAL DAY '
00074         PRINT *,'  YOU WISH TO PERFORM THE ACCOUNTING. '
00075         PRINT *,'  FOR EXAMPLE : 07-15-1989. '
00076         PRINT 1
00077         PRINT *,'                ?????????? '
00078         PRINT 35
00079         ACCEPT 2, IM2,ID2,IY2
00080         PRINT 1
00081         PRINT 6, IM2,ID2,IY2
00082       6 FORMAT (1X,' THE LAST ACCOUNTING DAY WILL BE ',I2,'-',I2,'-',I4,
00083        !  /1X,' IS THIS CORRECT?  ANSWER Y/N/C. ')
00084         PRINT 36
00085         ACCEPT 4,IANSWR
00086         PRINT 1
00087         IF (IANSWR.EQ.'C') GO TO 10
00088         IF (IANSWR.EQ.'N') GO TO 7
00089         GO TO 8
00090       9 IM2=IM1
00091         ID2=ID1
00092         IY2=IY1
00093       8 CONTINUE
00094   C
00095   C***** CONVERT BEGINNING AND ENDING DATES TO SEVEN DIGIT DATE YYYYDDD,
00096   C***** WHERE YYYY IS THE YEAR AND DDD IS THE NUMERICAL DAY OF THE YEAR.
00097   C***** ALSO COMPUTE A SIMILAR DATE FOR THE DAY SIXTEEN DAYS PRIOR TO THE
00098   C***** FIRST ACCOUNTING DAY SO THAT UP TO FIFTEEN DAY AVERAGES CAN BE
00099   C***** COMPUTED FOR CERTAIN REACH GAINS. THEN PRINT RESULTS.
00100   C
00101         IM=IM1
00102         ID=ID1
00103         IY=IY1
00104         DO 13 K=2,3
00105         FEB=IY/4.0
00106         IFEB=FEB
00107         FEB=FEB-IFEB
00108         NDY=0
00109         DO 11 I=1,IM
00110         IF(I.EQ.IM) GO TO 12
00111         IF(I.EQ.2.AND.FEB.LT.0.001) JD(2)=29
00112      11 NDY=NDY+JD(I)
00113      12 JD(2)=28
00114         NDY=NDY+ID
```

```
00115         IYD(K)=(IY*1000)+NDY
00116         IF(K.EQ.3) GO TO 14
00117         IM=IM2
00118         ID=ID2
00119         IY=IY2
00120      13 CONTINUE
00121      14 IYD(1)=IYD(2)-16
00122         IY=IYD(1)/1000
00123         IF(IY1.EQ.IY) GO TO 15
00124         FEB=IY/4.0
00125         IFEB=FEB
00126         FEB=FEB-IFEB
00127         J=635
00128         IF(FEB.LT.0.001) J=634
00129         IYD(1)=IYD(1)-J
00130      15 CONTINUE
00131         PRINT 16, IYD(1),IYD(3),IYD(2),IYD(3)
00132      16 FORMAT(3X,'ALL DATA FROM DAY ',I8,' TO DAY ', I8,' IS BEING RETRIE
00133        !VED FROM'/3X,'THE HISTORY FILE TO RUN DAY ',I8,' THROUGH DAY ',I8)
%FORT-I-CHACONCONTD, Character, Hollerith or RAD50 constant continued across
        lines; may be non-portable
       [ TO DAY ', I8,'] in module SNKSEL$MAIN at line 132

00134         PRINT *,'  SO PLEASE HANG ON FOR A SECOND. '
00135         PRINT 1
00136      C
00137      C***** SELECT ALL DATA FROM HISTORY FILE FROM DATE IYD(1) THROUGH IYD(3)
00138      C***** AND WRITE DATA ON THREE OUTPUT FILES (R'S, F'S, AND {D,X}'S).
00139      C
00140         IERR=0
00141         IRECD=0
00142      19 READ(1,17,END=18,ERR=24) (A(I),I=1,3),IDATE,(D(I),I=1,4)
00143      17 FORMAT(2A4,A1,I7,4A7)
00144         IRECD=IRECD+1
00145         IF(IDATE.LT.IYD(1)) GO TO 19
00146         IF(IDATE.GT.IYD(3)) GO TO 19
00147         IF(A(3).EQ.'D') GO TO 21
00148         IF(A(3).EQ.'P') GO TO 21
00149         IF(A(3).EQ.'F') GO TO 22
00150         IF(A(3).EQ.'R') GO TO 23
00151         IF(A(3).EQ.'E') GO TO 37
00152         GO TO 19
00153      24 IERR=IERR+1
00154         PRINT *,'  AN ERROR HAS OCCURRED DURING HISTORY FILE READ.'
00155         PRINT 40, IRECD
00156      40 FORMAT(3X,'AT RECORD ',I8,' EXECUTION CONTINUING.')
00157         IF(IERR.LE.10) GO TO 19
00158         PRINT *,'  THE NUMBER OF ERRORS IN READING THE HISTORY FILE'
00159         PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
00160         GO TO 10
00161      23 WRITE(4,17) (A(I),I=1,3),IDATE,D(1),D(3),D(4)
00162         GO TO 19
00163      22 WRITE(3,20) (A(I),I=1,3),IDATE,D(1)
00164      20 FORMAT(2A4,A1,I7,A7)
00165         GO TO 19
00166      21 WRITE(2,20) (A(I),I=1,3),IDATE,D(1)
00167         GO TO 19
```

```
00168      37 WRITE(5,20) (A(I),I=1,3),IDATE,D(1)
00169         GO TO 19
00170      18 PRINT 1
00171         PRINT *,'  OUTPUT FILES OF DIVERSION, FLOW, AND RESERVOIR DATA '
00172         PRINT *,'  FROM HISTORY FILE FOR SPECIFIED ACCOUNTING PERIOD'
00173         PRINT *,'  HAVE NOW BEEN CREATED AND ARE READY TO BE '
00174         PRINT *,'  SORTED FOR INPUT TO ACCOUNTING PROGRAM. '
00175    C
00176    C***** COMPUTE DATE FOR WHICH DATA IS TO BE RETRIEVED FROM ALLOCATIONS
00177    C***** FILE (ONE DAY BEFORE FIRST ACCOUNTING DAY).
00178    C
00179         IYD(4)=IYD(2)-1
00180         IY=IYD(4)/1000
00181         IF(IY1.EQ.IY) GO TO 25
00182         FEB=IY/4.0
00183         IFEB=FEB
00184         FEB=FEB-IFEB
00185         J=635
00186         IF(FEB.LT.0.001) J=634
00187         IYD(4)=IYD(4)-J
00188      25 CONTINUE
00189         PRINT 1
00190         PRINT 26, IYD(4)
00191      26 FORMAT(3X,'ALL DATA FOR DAY ',I8,' IS BEING RETRIEVED FROM THE '/
00192        !3X,'ALLOCATIONS FILE.')
00193    C
00194    C***** SELECT ALL DATA FROM ALLOCATIONS FILE FOR DAY IYD(4) AND WRITE
00195    C***** THESE DATA ON THREE OUTPUT FILES (D'S, R'S, AND M'S).
00196    C
00197         IALK=0
00198         IERR=0
00199      29 READ(10,27,END=28,ERR=34) (A(I),I=1,3),IDATE,(A(I),I=4,16)
00200      27 FORMAT(2A4,A1,I7,12A4,A2)
00201         IF(IDATE.NE.IYD(4)) GO TO 29
00202         IALK=1
00203         IF(A(3).EQ.'D') GO TO 31
00204         IF(A(3).EQ.'R') GO TO 32
00205         IF(A(3).EQ.'M') GO TO 33
00206         GO TO 29
00207      34 IERR=IERR+1
00208         PRINT *,'  AN ERROR HAS OCCURRED DURING ALLOCATIONS FILE READ.'
00209         PRINT *,'  EXECUTION CONTINUING.'
00210         IF(IERR.LE.10) GO TO 29
00211         PRINT *,'  THE NUMBER OF ERRORS IN READING THE ALLOCATIONS FILE'
00212         PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
00213         GO TO 10
00214      33 WRITE(9,30) (A(I),I=1,3),IDATE,(A(I),I=4,16)
00215      30 FORMAT(2A4,A1,I7,12A4,A2)
00216         GO TO 29
00217      32 WRITE(8,30) (A(I),I=1,3),IDATE,(A(I),I=4,6)
00218         GO TO 29
00219      31 WRITE(7,30) (A(I),I=1,3),IDATE,(A(I),I=9,16)
00220         GO TO 29
00221      28 IF(IALK.EQ.0) GO TO 45
00222         PRINT 1
00223         PRINT *,'  OUTPUT FILES OF DIVERSION, RESERVOIR, AND'
00224         PRINT *,'  MISCELLANEOUS DATA FROM ALLOCATIONS FILE FOR '
```

```
00225          PRINT *,'  PREVIOUS DAY HAVE BEEN CREATED AND ARE NOW READY'
00226          PRINT *,'  TO BE SORTED FOR INPUT TO ACCOUNTING PROGRAM.'
00227       10 STOP
00228       45 PRINT 46, IYD(4)
00229       46 FORMAT(/3X,'  NO DATA FOR DAY ',I8,' WAS FOUND IN ALLOCATIONS')
00230          DO 39 I=1,20
00231       39 WRITE(9,38) IYD(4)
00232       38 FORMAT(9X,I7)
00233          STOP
00234          END
%FORT-I-VARUNUSED, Variable was declared but not used
        [OM] in module SNKSEL$MAIN
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	3315	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	2032	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	968	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	6315	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKSEL$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	FEB		105= 106 107(2)= 111 124= 125 126(2)= 128 182= 183 184(2)= 186
**	I*4	I		109= 110 111 112 142(4)= 161(2)= 163(2)= 166(2)= 168(2)= 199(4)= 214(4)= 217(4)= 219(4)= 230=
**	I*4	IALK		197= 202= 221
2-000000CC	I*4	IANSWR		59= 62 63 68= 70 71 85= 87 88
**	I*4	ID		102= 114 118=
2-000000C4	I*4	ID1		51= 54 91 102
2-000000D4	I*4	ID2		79= 81 91= 118
2-000000E0	I*4	IDATE		142= 145 146 161 163 166 168 199= 201 214 217 219
**	I*4	IERR		140= 153(2)= 157 198= 207(2)= 210
**	I*4	IFEB		106= 107 125= 126 183= 184
**	I*4	IM		101= 109 110 117=
2-000000C0	I*4	IM1		51= 54 90 101

2-000000D0	I*4	IM2	79=	81	90=	117			
**	I*4	IRECD	141=	144(2)=	155				
**	I*4	IY	103=	105	115	119=	122=	123	124	180=
			181	182					
2-000000C8	I*4	IY1	51=	54	92	103	123	181	
2-000000D8	I*4	IY2	79=	81	92=	119			
**	I*4	J	127=	128=	129	185=	186=	187	
**	I*4	K	104=	115	116				
**	I*4	NDY	108=	112(2)=	114(2)=	115			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000040	R*4	A		100	(25)	16	142=	147	148	149
						150	151	161	163	166
						168	199(2)=	203	204	205
						214(2)	217(2)	219(2)		
2-000000A4	CHAR	D		28	(4)	16	17	142=	161(3)	163
						166	168			
2-00000030	I*4	IYD		16	(4)	16	115=	121(2)=	122	129(2)=
						131(4)	145	146	179(2)=	180
						187(2)=	190	201	228	231
2-00000000	I*4	JD		48	(12)	16	20D	111=	112	113=
**	R*4	OM		48	(12)	16	18D			

LABELS

Address	Label	References						
1-00000000	1'	31	32#	42	47	53	61	69	76
		80	86	135	170	189	222		
1-00000019	2'	51	52#	79					
1-00000024	3'	54	55#						
1-0000007F	4'	59	60#	68	85				
0-00000184	5	44#	63						
1-00000082	6'	81	82#						
0-000003AC	7	73#	88						
0-00000546	8	89	93#						
0-00000538	9	71	90#						
0-00000CEC	10	62	70	87	160	213	227#		
**	11	109	112#						
0-0000059A	12	110	113#						
**	13	104	120#						
0-000005C2	14	116	121#						
0-00000605	15	123	130#						
1-000000DA	16'	131	132#						
1-0000014D	17'	142	143#	161					
0-00000918	18	142	170#						

```
  0-00000680  19                    142#     145      146      152      157      162      165      167
                                    169
  1-00000183  20'                   163      164#     166      168

  0-00000848  21                    147      148      166#
  0-000007F8  22                    149      163#
  0-00000780  23                    150      161#
  0-00000894  24                    142      153#
  0-000009E2  25                    181      188#

  1-0000018E  26'                   190      191#
  1-000001DB  27'                   199      200#
  0-00000C08  28                    199      221#
  0-00000A20  29                    199#     201      206      210      216      218      220
  1-000001EA  30'                   214      215#     217      219

  0-00000B40  31                    203      219#
  0-00000AF8  32                    204      217#
  0-00000AA8  33                    205      214#
  0-00000B88  34                    199      207#
  1-00000004  35'                    49       50#      78

  1-00000078  36'                    57       58#      67       84
  0-00000734  37                    151      168#
  1-0000022E  38'                   231      232#
      **      39                    230      231#
  1-0000015A  40'                   155      156#

  0-00000C98  45                    221      228#
  1-000001F9  46'                   228      229#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                                 References

        ASSIGN                                21       29
```

```
+-----------------------------------------------+
|            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 SNKSEL

  /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.SNAKE]SNKSEL.LIS;2
  /OBJECT=HYDRO:[BSUTTER.SNAKE]SNKSEL.OBJ;1


COMPILER:  DEC Fortran V6.3-141

COMPILATION STATISTICS

  Run Time:           0.76 seconds
  Elapsed Time:       1.65 seconds
  Page Faults:        1355
  Dynamic Memory:     1152 pages
```

SNKHSTSUM.LIS

```
0001    C  ************************************************
0002    C     THIS PROGRAM LISTS DATA FROM WD-01 HISTORY FILES
0003    C        IN USGS STYLE RECORDS BY IRRIGATION YEAR
0004    C        AND SUMS STATIONS BETWEEN DESIGNATED
0005    C                     POINTS.
0006    C
0007    C     THIS PROGRAM IS A MODIFICATION OF SNKHSTLST.FOR
0008    C                   BOB SUTTER - AUGUST 1991
0009    C ************************************************
0010    C
0011          LOGICAL*1 FILES(40)
0012          BYTE CNTRL
0013          INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
0014          REAL MEAN,MAX,MIN,NOVALUE
0015          CHARACTER NAME*70,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
0016          CHARACTER TTL*70,BLANK*1,SNAME*70,NNAME*70
0017          DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
0018          DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12),SDV(12,31)
0019          DIMENSION JDATE(12,31)
0020          DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
0021          CNTRL='00'X
0022          NOVALUE=9999999.
0023          BLANK=' '
0024          OPEN(UNIT=4,FILE='SNKHSTSUM.TTL',STATUS='OLD')
0025          OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
0026          OPEN(UNIT=8,FILE='SNKHSTSUM.OUT',STATUS='NEW',RECL=133)
0027    C
0028    C***** INQUIRE IF FILE IS TO BE USED TO SELECT SPECIFIC STATIONS TO BE
0029    C***** SUMMED FROM DATA FILE.  IF NOT, ALL DATA IN FILE WILL BE
0030    C***** SUMMED.
0031    C
0032      500 PRINT *,'************************************'
0033          PRINT *,'   YOU HAVE A CHOICE:                '
0034          PRINT *,'                                     '
0035          PRINT *,' EITHER SELECT STATIONS FROM INPUT FILE TO BE SUMMED,'
0036          PRINT *,' OR SUM ALL DATA OF SAME DAY IN INPUT FILE.          '
0037          PRINT *,'                                     '
0038          PRINT 904
0039      904 FORMAT('  TO USE A FILE OF SELECTED STATIONS ENTER Y/N: '$)
0040          ACCEPT 4, SEL
0041        4 FORMAT(A1)
0042          IF(SEL.NE.'Y') GO TO 11
0043    C
0044    C***** READ IN A LIST OF TITLES FOR EACH DATA SET TO BE SUMMED.  IF A
0045    C***** TITLE IS NOT PRESENT, THAT DATA WILL THEN NOT BE INCLUDED IN THE
0046    C***** SUMMED DATA SET.
0047    C
0048          PRINT *,'                                     '
0049          PRINT *,'************************************'
0050          PRINT *,'                                     '
0051          PRINT *,'   YOU HAVE ANOTHER CHOICE:          '
0052          PRINT *,'                                     '
0053          PRINT *,'  SUM ALL DIVERSIONS, FLOWS, PUMPS, OR RESERVOIRS '
0054          PRINT *,'  (STATION NUMBERS & TITLES READ FROM EXISTING FILE)'
0055          PRINT *,'                                     '
0056          PRINT *,'              OR                     '
0057          PRINT *,'                                     '
```

```
0058          PRINT *,'  SUM ONLY STATIONS WHICH YOU SELECT INDIVIDUALLY   '
0059          PRINT *,'  (YOU PROVIDE THE STATION NUMBER AND TITLE) '
0060          PRINT *,'                                              '
0061          PRINT *,'   TO SUM ALL DIVERSIONS, ANSWER  D          '
0062          PRINT *,'   TO SUM ALL RIVER FLOWS, ANSWER  F         '
0063          PRINT *,'   TO SUM ALL PUMPS, ANSWER  P              '
0064          PRINT *,'   TO SUM ALL RESERVOIRS, ANSWER  R         '
0065          PRINT *,'   TO SUM INDIVIDUAL STATIONS, ANSWER  I    '
0066          PRINT *,'   TO ENTER YOUR OWN GROUP OF STATIONS, ANSWER G '
0067          PRINT *,'   TO QUIT, ANSWER  Q  '
0068          PRINT *,'                                              '
0069          PRINT 905
0070      905 FORMAT('         ENTER CHOICE:'$)
0071          ACCEPT 4, IND
0072          IF(IND.EQ.'Q') GO TO 470
0073       11 PRINT *,' '
0074          PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
0075          PRINT *,'   IRRIGATION YEAR FOR WHICH DATA IS TO BE LISTED - '
0076          PRINT *,' '
0077          PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR IRRIGATION YEAR  '
0078          PRINT *,'   1990, ENTER 89-90    '
0079          PRINT *,'                __-__   '
0080          PRINT 901
0081      901 FORMAT('+               '$)
0082          KY=0
0083          ACCEPT 900, NP,NP1
0084      900 FORMAT(I2,1X,I2)
0085      C
0086      C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
0087      C
0088          NP=NP+1900
0089          NP1=NP1+1900
0090          IF(JMOD(NP,4).EQ.0) KY=1
0091          ID=(NP*1000)+305+KY
0092          KY=0
0093          IF(JMOD(NP1,4).EQ.0) KY=1
0094          ID1=(NP1*1000)+304+KY
0095      C
0096      C***** READ IN TITLES OF DATA TO BE SUMMED
0097      C
0098          IF(SEL.NE.'Y') GO TO 2
0099          IF(IND.EQ.'I') THEN
0100                       OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
0101        1              PRINT *,'   ENTER STATION NUMBER AND TITLE '
0102                       PRINT *,'   TO END   ENTER   99999999  '
0103                       PRINT *,'            ________  '
0104                       PRINT 903
0105        903            FORMAT('+           '$)
0106                       ACCEPT 301, ISTA,TTL
0107                       IF(ISTA .EQ. 99999999) GO TO 9
0108                       WRITE(3,301) ISTA,TTL
0109                       GO TO 1
0110          9            REWIND 3
0111                       GO TO 2
0112                       END IF
0113          IF(IND.NE.'G') GO TO 8
0114          PRINT 907
```

```
0115      907 FORMAT(' ENTER FILE NAME OF TITLES OF DATA TO BE SUMMED: '$)
0116          ACCEPT 908, IC,(FILES(I),I=1,IC)
0117      908 FORMAT(Q,40A1)
0118          CALL ASSIGN(3,FILES)
0119          GO TO 2
0120        8 IF(IND.EQ.'D') OPEN(UNIT=3,FILE='SNKDIVSUM.TTL',STATUS='OLD')
0121          IF(IND.EQ.'F') OPEN(UNIT=3,FILE='SNKFLO.TTL',STATUS='OLD')
0122          IF(IND.EQ.'P') OPEN(UNIT=3,FILE='SNKPMP.TTL',STATUS='OLD')
0123          IF(IND.EQ.'R') OPEN(UNIT=3,FILE='SNKRES.TTL',STATUS='OLD')
0124        2 PRINT 3
0125        3 FORMAT(' TO LIST STATIONS BEING SUMMED ENTER Y/N: '$)
0126          ACCEPT 4,PIS
0127          PRINT 7
0128        7 FORMAT(' TO MAKE A SEPARATE FILE OF SUMS ENTER Y/N: '$)
0129          ACCEPT 4,FSUM
0130          IF(FSUM.EQ.'Y') OPEN(UNIT=9,FILE='SNKHSTSUM.DAT',STATUS='NEW',
0131         1CARRIAGECONTROL='LIST')
0132          INREC=0
0133          INSTA=0
0134          IEND=0
0135          ISR=1
0136          STATION_NUM=0
0137        5 ISUM=0
0138          DO 40 I=1,12
0139          DO 40 J=1,31
0140       40 SDV(I,J)=NOVALUE
0141      C
0142      C***** IF NOT SELECTING SPECIFIC DATA FROM DATA SET TO BE SUMMED
0143      C***** SKIP SEARCH FOR STATION ROUTINE AND STATION READ STATEMENT
0144      C
0145         10 IF(SEL.NE.'Y') GO TO 15
0146            IF(INSTA.EQ.1) THEN
0147                           STA_ID=NSTA_ID
0148                           NAME=NNAME
0149                           INSTA=0
0150                           GO TO 15
0151                           ENDIF
0152            READ(3,301,END=450) STA_ID,NAME
0153      301 FORMAT(I8,A70)
0154      C
0155      C***** FILL ARRAY WITH MISSING VALUE INDICATORS
0156      C
0157         15 DO 20 I=1,12
0158            DO 20 J=1,31
0159       20 DV(I,J)=NOVALUE
0160            IF(SEL.NE.'Y') GO TO 13
0161      C
0162      C***** SKIP READ STATEMENT IF FIRST RECORD PREVIOUSLY READ
0163      C
0164            IF(STATION_NUM.EQ.STA_ID) GO TO 60
0165            IF(STATION_NUM.GT.STA_ID) GO TO 651
0166      C
0167      C***** READ A RECORD (STATION SEARCH ROUTINE)
0168      C
0169       26 IF(SEL.NE.'Y') GO TO 13
0170       25 READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0171      701 FORMAT(I8,A1,I4,I3,F7.0)
```

```
0172          IF(STATION_NUM.LT.STA_ID) GO TO 25
0173          IF(STATION_NUM.EQ.STA_ID) GO TO 60
0174        651 IF(TD.NE.STA_ID) THEN
0175              PRINT 601,STA_ID
0176        601   FORMAT(1X,'STATION ',I8,' NOT FOUND')
0177              GO TO 10
0178              END IF
0179          READ(3,301,END=99) NSTA_ID,NNAME
0180          INSTA=1
0181         35 IF(STATION_NUM.LT.NSTA_ID) THEN
0182              READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0183              GO TO 35
0184              END IF
0185          GO TO 100
0186      C
0187      C***** READ A RECORD (NO STATION SEARCH ROUTINE), WHEN THE STATION
0188      C***** NUMBER CHANGES, GO TO LISTING AND/OR SUMMING ROUTINE
0189      C
0190         13 IF(INSTA.EQ.1) GO TO 12
0191            READ(7,701,END=98) STATION_NUM,TYPE,YEAR,ND,VALUE
0192            IF(INREC.EQ.0) THEN
0193                           INREC=1
0194                           GO TO 12
0195                           ENDIF
0196            IF(STATION_NUM.LT.TD) GO TO 13
0197            IF(STATION_NUM.GT.TD) THEN
0198                                  INSTA=1
0199                                  GO TO 100
0200                                  ENDIF
0201         12 STA_ID=STATION_NUM
0202            INSTA=0
0203      C
0204      C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
0205      C
0206         60 IDATE=(YEAR*1000)+ND
0207            IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 26
0208            JD(2)=28
0209            IF(JMOD(YEAR,4) .EQ. 0) JD(2)=29
0210            LD=ND
0211            MON=0
0212         62 MON=MON+1
0213            IF(MON.GT.12) GO TO 68
0214            LD=LD-JD(MON)
0215            IF(LD.GT.0) GO TO 62
0216            LD=LD+JD(MON)
0217            DV(MON,LD)=VALUE
0218            JDATE(MON,LD)=IDATE
0219      C
0220      C***** SAVE VARIABLE VALUES IN PREPARATION FOR READING NEXT RECORD
0221      C***** SO THAT IF STATION CHANGES, WE HAVE PRESERVED DATA FOR
0222      C***** STATION TO BE LISTED.
0223      C
0224         68 CONTINUE
0225            TD=STATION_NUM
0226            P_YEAR=YEAR
0227            P_TYPE=TYPE
0228            IF(SEL.NE.'Y') GO TO 13
```

```
0229          GO TO 25
0230      C
0231      C***** ENTIRE YEAR FOR ONE STATION READ IN -
0232      C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
0233      C***** IF LAST STATION TITLE HAS BEEN READ, SET NEXT STATION
0234      C***** NUMBER = 99999999 SO THAT SUMS WILL BE PRINTED.
0235      C
0236         98 IEND=1
0237         99 STATION_NUM=99999999
0238        100 IF(ISUM.NE.1.AND.PIS.NE.'Y') GO TO 6
0239            LEAP=0
0240            IF(JMOD(NP1,4).EQ.0) LEAP=1
0241            IF(P_TYPE.NE.'R') GO TO 120
0242            DELTA_S(1)=DV(1,31)-DV(12,31)
0243            DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
0244            DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
0245            DELTA_S(4)=DV(4,30)-DV(3,31)
0246            DELTA_S(5)=DV(5,31)-DV(4,30)
0247            DELTA_S(6)=DV(6,30)-DV(5,31)
0248            DELTA_S(7)=DV(7,31)-DV(6,30)
0249            DELTA_S(8)=DV(8,31)-DV(7,31)
0250            DELTA_S(9)=DV(9,30)-DV(8,31)
0251            DELTA_S(10)=DV(10,30)-DV(9,31)
0252            DELTA_S(11)=NOVALUE
0253            DELTA_S(12)=DV(12,31)-DV(11,30)
0254            DO 119 I=2,12
0255        119 IF(ABS(DELTA_S(I)).GE.8000000.) DELTA_S(I)=NOVALUE
0256        120 DO 135 I=1,12
0257            D=31
0258            IF(I.EQ.2) D=28+LEAP
0259            IF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.9) D=30
0260            CFSD(I)=DV(I,1)
0261            D_ADJ=0
0262            IF(DV(I,1).EQ.NOVALUE) THEN
0263                                   CFSD(I)=0.0
0264                                   D_ADJ=1
0265                                   END IF
0266            MAX(I)=DV(I,1)
0267            IF(DV(I,1).EQ.NOVALUE) MAX(I)=-1.0
0268            MIN(I)=DV(I,1)
0269            DO 130 J=2,D
0270            DZ=DV(I,J)
0271            IF(DV(I,J).EQ.NOVALUE) THEN
0272                                   DZ=0.0
0273                                   D_ADJ=D_ADJ+1
0274                                   END IF
0275            CFSD(I)=CFSD(I)+DZ
0276            IF(DV(I,J).EQ.NOVALUE) THEN
0277                                   GO TO 129
0278                                   END IF
0279            IF(DV(I,J).GT.MAX(I)) MAX(I)=DV(I,J)
0280        129 IF(DV(I,J).LT.MIN(I)) MIN(I)=DV(I,J)
0281        130 CONTINUE
0282            IF(D_ADJ.GE.D) THEN
0283                           CFSD(I)=NOVALUE
0284                           DELTA_S(I)=NOVALUE
0285                           MEAN(I)=NOVALUE
```

```
0286                        GO TO 133
0287                        END IF
0288          MEAN(I)=CFSD(I)/(D-D_ADJ)
0289      133 ACFT(I)=CFSD(I)*1.9835
0290      135 CONTINUE
0291          CALL DATE(RUNDATE)
0292          WRITE(8,801) RUNDATE
0293      801 FORMAT('1',/,120X,A9)
0294          IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
0295      800 FORMAT(1H+,'PRELIMINARY DATA')
0296          WRITE(8,802) STA_ID,NAME
0297      802 FORMAT(//37X,I8,3X,A70)
0298          IF(P_TYPE .EQ.'R') THEN
0299                             WRITE(8,803) NP,NP1
0300      803 FORMAT(30X,'CONTENTS IN ACRE FEET, IRRIGATION YEAR NOVEMBER',
0301         *          I5,' TO OCTOBER',I5)
0302                             GO TO 160
0303                             ELSE
0304                             WRITE(8,804) NP,NP1
0305      804 FORMAT(30X,'DISCHARGE, CUBIC FEET PER SECOND, IRRIGATION YEAR',
0306         *          ' NOVEMBER',I5,' TO OCTOBER',I5)
0307                             WRITE(8,805)
0308      805 FORMAT(59X,'MEAN VALUES')
0309                             END IF
0310      160 WRITE(8,806)
0311      806 FORMAT(//,1X,'DAY',11X,'NOV',7X,'DEC',7X,'JAN',7X,'FEB',7X,
0312         *          'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',7X,'AUG',
0313         *          7X,'SEP',7X,'OCT')
0314          WRITE(8,807)
0315      807 FORMAT(' ')
0316          DO 180 J=1,31
0317          IF(J.EQ.6.OR.J.EQ.11.OR.J.EQ.16.OR.J.EQ.21.OR.J.EQ.26) THEN
0318                                    WRITE(8,807)
0319                                      END IF
0320          DO 170 I=1,12
0321          II=I-2
0322          IF(I.EQ.1) II=11
0323          IF(I.EQ.2) II=12
0324          IF(II.EQ.11) WRITE(8,807)
0325          IF(II.EQ.11) WRITE(8,811) CNTRL,J
0326      811 FORMAT(A1,I3,'    ')
0327          IF(DV(II,J).EQ.NOVALUE) THEN
0328                                    WRITE(8,809) CNTRL
0329                                    GO TO 170
0330                                    END IF
0331          IF(DV(II,J).GE.10.0) THEN
0332                                   IDV=JNINT(DV(II,J))
0333                                   WRITE(8,810) CNTRL,IDV
0334      810                          FORMAT(A1,I10)
0335                                   GO TO 170
0336                                   END IF
0337          IF(DV(II,J).LT.10.0) THEN
0338                                   WRITE(8,808) CNTRL,DV(II,J)
0339      808                          FORMAT(A1,F10.1)
0340                                   GO TO 170
0341                                   END IF
0342      809                          FORMAT(A1,'       ---')
```

```
0343      170 CONTINUE
0344      180 CONTINUE
0345          IF(P_TYPE.EQ.'R') THEN
0346                          WRITE(8,812)
0347      812                 FORMAT('0')
0348                          WRITE(8,813) CNTRL
0349      813                 FORMAT(A1,2X,'MAX  ')
0350                          DO 190 I=1,12
0351                          II=I-2
0352                          IF(I.EQ.1) II=11
0353                          IF(I.EQ.2) II=12
0354                          IF(MAX(II).EQ.-1.0) THEN
0355                                    WRITE(8,860) CNTRL
0356      860                           FORMAT(A1,'          ')
0357                                    GO TO 190
0358                                    END IF
0359                          IDV=JNINT(MAX(II))
0360                          WRITE(8,810) CNTRL,IDV
0361      190                 CONTINUE
0362                          WRITE(8,807)
0363                          WRITE(8,814) CNTRL
0364      814                 FORMAT(A1,2X,'MIN  ')
0365                          DO 200 I=1,12
0366                          II=I-2
0367                          IF(I.EQ.1) II=11
0368                          IF(I.EQ.2) II=12
0369                          IF(MIN(II).EQ.NOVALUE) THEN
0370                                                 WRITE(8,860) CNTRL
0371                                                 GO TO 200
0372                                                 END IF
0373                          IDV=JNINT(MIN(II))
0374                          WRITE(8,810) CNTRL,IDV
0375      200                 CONTINUE
0376                          WRITE(8,807)
0377                          WRITE(8,815) CNTRL
0378      815                 FORMAT(A1,1X,'CHNG            ')
0379                          DO 210 I=2,12
0380                          II=I-2
0381                          IF(I.EQ.2) II=12
0382                          IDV=JNINT(DELTA_S(II))
0383                          IF(DELTA_S(II).EQ.NOVALUE) THEN
0384                                                 WRITE(8,860) CNTRL
0385                                                 GO TO 210
0386                                                 END IF
0387                          WRITE(8,810) CNTRL,IDV
0388      210                 CONTINUE
0389                          GO TO 10
0390                          END IF
0391              WRITE(8,812)
0392              WRITE(8,816) CNTRL
0393      816 FORMAT(A1,'TOTAL  ')
0394              DO 220 I=1,12
0395              II=I-2
0396              IF(I.EQ.1) II=11
0397              IF(I.EQ.2) II=12
0398              IF(CFSD(II).GE.NOVALUE) THEN
0399                                      WRITE(8,860) CNTRL
```

```
0400                                GO TO 220
0401                                END IF
0402          IDV=JNINT(CFSD(II))
0403          WRITE(8,810) CNTRL,IDV
0404      220 CONTINUE
0405          WRITE(8,807)
0406          WRITE(8,818) CNTRL
0407      818 FORMAT(A1,' MEAN  ')
0408          DO 230 I=1,12
0409          II=I-2
0410          IF(I.EQ.1) II=11
0411          IF(I.EQ.2) II=12
0412          IF(MEAN(II).GE. NOVALUE) THEN
0413                                   WRITE(8,860) CNTRL
0414                                   GO TO 230
0415                                   END IF
0416          IF(MEAN(II).LT.10.0) THEN
0417                                WRITE(8,808) CNTRL,MEAN(II)
0418                                GO TO 230
0419                                END IF
0420          IDV=JNINT(MEAN(II))
0421          WRITE(8,810) CNTRL,IDV
0422      230 CONTINUE
0423          WRITE(8,807)
0424          WRITE(8,819) CNTRL
0425      819 FORMAT(A1,'  MAX  ')
0426          DO 240 I=1,12
0427          II=I-2
0428          IF(I.EQ.1) II=11
0429          IF(I.EQ.2) II=12
0430          IF(MAX(II).EQ.-1.0) THEN
0431                              WRITE(8,860) CNTRL
0432                              GO TO 240
0433                              END IF
0434          IF(MAX(II).LT.10.0) THEN
0435                              WRITE(8,808) CNTRL,MAX(II)
0436                              GO TO 240
0437                              END IF
0438          IDV=JNINT(MAX(II))
0439          WRITE(8,810) CNTRL,IDV
0440      240 CONTINUE
0441          WRITE(8,807)
0442          WRITE(8,820) CNTRL
0443      820 FORMAT(A1,'  MIN  ')
0444          DO 250 I=1,12
0445          II=I-2
0446          IF(I.EQ.1) II=11
0447          IF(I.EQ.2) II=12
0448          IF(MIN(II).GE.NOVALUE) THEN
0449                                 WRITE(8,860) CNTRL
0450                                 GO TO 250
0451                                 END IF
0452          IF(MIN(II).LT.10.0) THEN
0453                              WRITE(8,808) CNTRL,MIN(II)
0454                              GO TO 250
0455                              END IF
0456          IDV=JNINT(MIN(II))
```

```
0457          WRITE(8,810) CNTRL,IDV
0458        250 CONTINUE
0459    C         DO 260 I=1,12
0460    C         IFT(I)=JNINT(ACFT(I)/100.)
0461    C         IFT(I)=IFT(I)*100
0462    C260      CONTINUE
0463          WRITE(8,807)
0464          WRITE(8,821) CNTRL
0465        821 FORMAT(A1,'AC-FT  ')
0466          DO 270 I=1,12
0467          II=I-2
0468          IF(I.EQ.1) II=11
0469          IF(I.EQ.2) II=12
0470          IF(ACFT(II).GE.19834900.0) THEN
0471                                     WRITE(8,860) CNTRL
0472                                     GO TO 270
0473                                     END IF
0474    C         IF(ACFT(II).LE.999.) THEN
0475    C                                 IFT(II)=JNINT(ACFT(II))
0476    C                                  WRITE(8,810) CNTRL,IFT(II)
0477    C                                 GO TO 270
0478    C                                 END IF
0479          IFT(II)=JNINT(ACFT(II))
0480          WRITE(8,810) CNTRL,IFT(II)
0481        270 CONTINUE
0482          YT=0.0
0483          DO 280 I=1,12
0484          IF(CFSD(I) .EQ. NOVALUE) CFSD(I)=0.0
0485        280 YT=YT + CFSD(I)
0486          IYT=JNINT(YT)
0487          IYM=JNINT(YT/(365+LEAP))
0488    C         NT=JNINT(YT*1.9835/100.)
0489    C         IYA=NT*100
0490    C         IF(IYA.LT.9999) IYA=1.9835*YT
0491          IYA=1.9835*YT
0492          WRITE(8,807)
0493          WRITE(8,831) NP1,IYT,IYM,IYA
0494        831 FORMAT(11X,'IRRIGATION YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
0495           *          I6,4X,'AC-FT',I8)
0496    C
0497    C***** ENTER SUMMING ROUTINE UNLESS END OF DATA OR SUM HAS JUST
0498    C***** BEEN PRINTED.
0499    C
0500          IF(IEND.EQ.1) GO TO 450
0501          IF(ISUM.EQ.1) GO TO 5
0502          6 DO 460 I=1,12
0503          DO 460 J=1,31
0504          IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).EQ.NOVALUE) GO TO 460
0505          IF(DV(I,J).GE.0.0.AND.SDV(I,J).EQ.NOVALUE) SDV(I,J)=0.0
0506          IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).GT.0.0) DV(I,J)=0.0
0507          SDV(I,J)=SDV(I,J)+DV(I,J)
0508        460 CONTINUE
0509          IF(ISR.EQ.0) GO TO 480
0510          READ(4,301,END=10) IDSUM,SNAME
0511          ISR=0
0512        480 IF(STATION_NUM.LT.IDSUM) GO TO 10 ! GO TO NEXT STATION
0513        485 STA_ID=IDSUM
```

```
0514          NAME=SNAME
0515          DO 465 I=1,12
0516          DO 465 J=1,31
0517        465 DV(I,J)=SDV(I,J)
0518          ISUM=1
0519          ISR=1
0520      C
0521      C***** IF FSUM EQUALS 'Y' MAKE A FILE OF SUMMED DATA
0522      C
0523          IF(FSUM.NE.'Y') GO TO 100
0524          DO 885 K=1,12
0525          I=K+10
0526          IF(I.GT.12) I=I-12
0527          JMD=JD(I)
0528          DO 885 J=1,JMD
0529        885 IF(JDATE(I,J).GT.0.AND.SDV(I,J).NE.NOVALUE)
0530         1                    WRITE(9,880) IDSUM,JDATE(I,J),SDV(I,J)
0531        880 FORMAT(I8,'S',I7,F7.0)
0532          GO TO 100  ! ORGANIZE AND PRINT SUMS
0533        450 CONTINUE
0534          IF(ISR.EQ.0) GO TO 485
0535          PRINT *,'  '
0536          PRINT *,'  '
0537          PRINT *,'  REQUESTED STATION LISTINGS AND SUMS COMPLETE '
0538          PRINT *,'  '
0539          PRINT *,'  '
0540        470 PRINT *,'                                        '
0541          PRINT *,'  OUTPUT IS  THE LATEST VERSION OF SNKHSTSUM.OUT'
0542          PRINT *,'  '
0543          STOP
0544          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	6037	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	2251	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	5976	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	14264	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTSUM$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	CHAR	BLANK		16 23=
**	L*1	CNTRL		12 21= 325 328 333 338 348 355 360 363 370 377 384 387 392 399 406 413 417 421 424 435 439 442 449 453 464 471 480
**	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= 130 523
**	I*4	I		116(2)= 138= 140 157= 159 254= 255(2) 256= 258 259(3) 260(2) 263 266(2) 267(2) 268(2) 270 275(2) 276 279(4) 280(4) 283 285 288(2) 289(2) 320= 321 323 350= 351 352 353 366 367 368 379= 380 394= 395 396 397 408= 410 411 426= 427 428 444= 445 446 447 466= 468 469 483= 484(2) 485 504(2) 505(3) 506(3) 507(3) 515= 525= 526(3)= 527 529(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		510= 512 513 529
**	I*4	IDV		332= 333 359= 360 373= 374 382= 387 402= 403 420= 438= 439 456= 457
2-00001470	I*4	IEND		134= 236= 500

```
     **      I*4  II          321=     322=     323=     324      325      327
                                331      332      337      338      351=
                               353=     354      359      366=     367=
                               369      373      380=     381=     382
                               395=     396=     397=     398      402
                               410=     411=     412      416      417
                               427=     428=     429=     430      434
                               438      445=     446=     447=     448
                               453      456      467=     468=     469=
                               479(2)   480

  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=     509      511=     519=     534
  2-0000145C  I*4  ISTA        106=     107      108

  2-00001474  I*4  ISUM        137=     238      501      518=
     **       I*4  IYA         491=     493
     **       I*4  IYM         487=     493
     **       I*4  IYT         486=     493
     **       I*4  J           139=     140      158=     159      269=     270
                               271      276      279(2)   280(2)   316=
                               325      327      331      332      337
                               503=     504(2)   505(3)   506(3)   507(3)
                               517(2)   528=     529(4)

     **       I*4  JMD         527=     528
     **       I*4  K           524=     525
     **       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      487

     **       I*4  MON         211=     212(2)=  213      214      216      217
                               218
  2-00001318  CHAR NAME         15      148=     152=     296      514=
  2-0000147C  I*4  ND          170=     182=     191=     206      210
  2-000013F5  CHAR NNAME        16      148      179=
     **       R*4  NOVALUE      14       22=     140      159      252      255
                               262      267      271      276      283
                               285      327      369      383      398
                               448      484      504(2)   505      506

  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      493
  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      345

     **       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      510=     514
```

2-00001444	I*4	STATION_NUM	13 136= 164 165 170= 172 173 181 182= 191= 196 201 225 237= 512
2-00001440	I*4	STA_ID	13 147= 152= 164 165 172 173 174 175 201= 296
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	482= 485(2)= 486 487 491
**	CHAR	ZLCH	15

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-000006F0	R*4	ACFT		48	(12)	18 289= 470 479
2-00000690	R*4	CFSD		48	(12)	17 260= 263= 275(2)= 283= 288 289 398 402 485
2-000006C0	R*4	DELTA_S		48	(12)	18 242= 243= 244= 245= 246= 247= 248= 249= 251= 252= 253= 255(382 383
2-000000C0	R*4	DV		1488	(12, 31)	17 159= 217= 242(2) 243(2) 244(2) 245(2) 246(2) 247(249(2) 250(2) 251(2) 253(262 266 267 268 271 276 279(2) 280(331 332 337 338 505 506(2)= 507 517=
2-000012F0	L*1	FILES		40	(40)	11 116= 118A
2-00000720	I*4	IFT		48	(12)	18 479= 480
2-00000090	I*4	JD		48	(12)	17 20D 208= 209= 214 216 527
2-00000D20	I*4	JDATE		1488	(12, 31)	19 218= 529(2)
2-00000030	R*4	MAX		48	(12)	14 17 266= 267= 279(2)= 354 359 430 434 438
2-00000000	R*4	MEAN		48	(12)	14 18 285= 288= 412 416 417 420
2-00000060	R*4	MIN		48	(12)	14 17 268= 280(2)= 369 373 448 452 453
2-00000750	R*4	SDV		1488	(12, 31)	18 140= 504 505(2)= 506 507(2)= 517 529(2)

LABELS

Address	Label	References
0-00000534	1	101# 109
0-000006C8	2	98 111 119 124#
1-0000062A	3'	124 125#
1-000005A8	4'	40 41# 71 126 129
0-0000075C	5	137# 501
0-00001640	6	238 502#
1-00000658	7'	127 128#
0-0000067C	8	113 120#
0-00000604	9	107 110#
**	10	145# 177 389 510 512
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# 523 532.
**	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#
0-00000E64	170	320 329 335 340 343#
**	180	316 344#
0-00000F27	190	350 357 361#
0-00000FD7	200	365 371 375#
0-0000107D	210	379 385 388#
0-00001133	220	394 400 404#
0-0000121B	230	408 414 418 422#
0-00001307	240	426 432 436 440#
0-000013F3	250	444 450 454 458#
0-000014AD	270	466 472 481#
**	280	483 485#
1-00000688	301'	106 108 152 153# 179 510

```
  0-00001550  450                 152       500       533#

  0-0000168D  460                 502       503       504       508#
      **      465                 515       516       517#
  0-000015E3  470                  72       540#
  0-000016CE  480                 509       512#
  0-000016D8  485                 513#      534

      **      500                  32#
  1-00000699  601'                175       176#
  0-00000982  651                 165       174#
  1-0000068D  701'                170       171#      182       191
  1-000006BD  800'                294       295#

  1-000006B4  801'                292       293#
  1-000006D3  802'                296       297#
  1-000006DE  803'                299       300#
  1-00000723  804'                304       305#
  1-00000775  805'                307       308#

  1-00000785  806'                310       311#
  1-000007E3  807'                314       315#      318       324       362       376
                                                405       423       441       463       492
  1-000007F7  808'                338       339#      417       435       453
  1-000007FD  809'                328       342#
  1-000007F2  810'                333       334#      360       374       387       403
                                                421       439       457       480


  1-000007E7  811'                325       326#
  1-0000080C  812'                346       347#      391
  1-00000810  813'                348       349#
  1-0000082B  814'                363       364#
  1-00000837  815'                377       378#

  1-0000084E  816'                392       393#
  1-0000085A  818'                406       407#
  1-00000866  819'                424       425#
  1-00000872  820'                442       443#
  1-0000087E  821'                464       465#

  1-0000088A  831'                493       494#
  1-0000081C  860'                355       356#      370       384       399       413
                                                431       449       471
  1-000008C0  880'                529       531#
      **      885                 524       528       529#
  1-000005D8  900'                 83        84#

  1-000005C4  901'                 80        81#
  1-000005DF  903'                104       105#
  1-00000574  904'                 38        39#
  1-000005AB  905'                 69        70#
  1-000005EF  907'                114       115#

  1-00000624  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 SNKHSTSUM

/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.SNAKE]SNKHSTSUM.LIS;1
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTSUM.OBJ;1
```

COMPILATION STATISTICS

```
Run Time:           20.50 seconds
Elapsed Time:       22.02 seconds
Page Faults:        1076
Dynamic Memory:     1052 pages
```

SNKCHGS.LIS

```
00001   C*************************************************************************
00002   C
00003   C       PROGRAM TO CHANGE VALUES IN DATA SET SNKSYS.IND.
00004   C       THESE VALUES ARE THE INITIAL RESERVOIR RIGHT STORAGE ACCUMULATIONS,
00005   C       AND OTHER SYSTEM ACCUMULATIONS.   RJS - MAY 1988.
00006   C
00007   C*************************************************************************
00008
00009         DIMENSION CEV(9),GT(10),BRST(16),NAME(10),RIGHT(16),RESVR(9)
00010         CHARACTER*17 NAME,RIGHT,RESVR
00011         DATA NAME/'CHANGE IN STORAGE','STORAGE PAST MILNER',
00012        1'TTL ACNTD STORAGE','UNACCOUNTED STORE','TTL MINIDOKA CRDT',
00013        2'TOTAL RIRIE LOSS ','TOTAL WILLOW LOSS','TTL EAGLE RK CRDT',
00014        3'TTL AF GAIN DIFF ','TOTAL SWID'/
00015         DATA RIGHT/'JACKSON LAKE 1906','LAKE WALCOTT     ',
00016        1'JACKSON LAKE 1910','JACKSON LAKE 1913','HENRYS LAKE 1917 ',
00017        2'PALISADES 1921   ','ISLAND PARK 1921 ','AMERICAN FALLS #1',
00018        3'AMERICAN FALLS #2','ISLAND PARK 1935 ','GRASSY LAKE      ',
00019        4'PALISADES 1939   ','HENRYS LAKE 1965 ','RIRIE            ',
00020        4'AMERICAN FLS LTF ','PALISADES LTF    '/
00021         DATA RESVR/'JACKSON LAKE     ','PALISADES        ',
00022        1'HENRYS LAKE      ','ISLAND PARK      ','GRASSY LAKE      ',
00023        2'RIRIE            ','AMERICAN FALLS   ','LAKE WALCOTT     ',
00024        3'MILNER           '/
00025         CALL ASSIGN (1,'SNKSYS.IND')
00026         OPEN (UNIT=2,NAME='SNKSYS.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
00027         READ(1,2,END=6) (CEV(L),L=1,9)
00028       2 FORMAT(8F10.1)
00029         READ(1,2,END=6) (GT(L),L=1,10)
00030         READ(1,2,END=6) (BRST(L),L=1,16)
00031         PRINT *,'                                                      '
00032         PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR EVAPORATI
00033        1ON TOTALS '
00034         PRINT *,' THESE VALUES WILL THEN BECOME THE YEAR-TO-DATE TOTALS BE
00035        1GINNING   '
00036         PRINT *,' THE NEXT ACCOUNTING RUN '
00037         PRINT *,' '
00038         DO 14 I=1,9
00039         TYPE 11, RESVR(I),CEV(I)
00040         ACCEPT 102, IANS
00041         IF(IANS.NE.'Y') GO TO 14
00042         PRINT 103
00043         ACCEPT 4, CEV(I)
00044        14 CONTINUE
00045         WRITE(2,2) (CEV(L),L=1,9)
00046         PRINT *,' '
00047         PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE YEAR-TO-DATE TOTALS
00048        1  '
```

```
00049          PRINT *,' '
00050          DO 101 I=1,10
00051          TYPE 11, NAME(I),GT(I)
00052       11 FORMAT(' ',A17,' =',F10.1,' NEW VALUE? Y/N '$)
00053          ACCEPT 102, IANS
00054      102 FORMAT(A1)
00055          IF(IANS.NE.'Y') GO TO 101
00056          PRINT 103
00057      103 FORMAT(' ENTER NEW VALUE   ='$)
00058          ACCEPT 4, GT(I)
00059        4 FORMAT(F10.1)
00060      101 CONTINUE
00061          WRITE(2,2) (GT(L),L=1,10)
00062          PRINT *,'                                                    '
00063          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THE AMOUNT OF WATER ACCUM
00064         1ULATED '
00065          PRINT *,' IN EACH RESERVOIR STORAGE RIGHT '
00066          PRINT *,' '
00067          DO 104 I=1,16
00068          TYPE 11, RIGHT(I),BRST(I)
00069          ACCEPT 102, IANS
00070          IF(IANS.NE.'Y') GO TO 104
00071          PRINT 103
00072          ACCEPT 4, BRST(I)
00073      104 CONTINUE
00074          WRITE(2,2) (BRST(L),L=1,16)
00075          PRINT *,' '
00076        6 STOP
00077          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1176	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	469	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	1008	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	2653	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGS$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	I*4	I		38= 39(2) 43 50= 51(2) 58 67= 68(2) 72
2-000002E4	I*4	IANS		40= 41 53= 55 69= 70
2-000002E0	I*4	L		27(2)= 29(2)= 30(2)= 45(2)= 61(2)= 74(2)=

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-0000004C	R*4	BRST		64	(16)	9 30= 68 72= 74
2-00000000	R*4	CEV		36	(9)	9 27= 39 43= 45
2-00000024	R*4	GT		40	(10)	9 29= 51 58= 61
2-0000008C	CHAR	NAME		170	(10)	9 10 11D 51
2-00000246	CHAR	RESVR		153	(9)	9 10 21D 39
2-00000136	CHAR	RIGHT		272	(16)	9 10 15D 68

LABELS

Address	Label	References
1-00000000	2'	27 28# 29 30 45 61 74
1-00000041	4'	43 58 59# 72
0-00000491	6	27 29 30 76#
1-00000006	11'	39 51 52# 68
0-000001D8	14	38 41 44#
0-000002FF	101	50 55 60#
1-00000026	102'	40 53 54# 69
1-00000029	103'	42 56 57# 71
0-00000447	104	67 70 73#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	25

```
+-----------------------------------------------+
|           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 SNKCHGS

  /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.SNAKE]SNKCHGS.LIS;4
  /OBJECT=HYDRO:[BSUTTER.SNAKE]SNKCHGS.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

```
  Run Time:           0.33 seconds
  Elapsed Time:       1.33 seconds
  Page Faults:        770
  Dynamic Memory:     736 pages
```

SNKCHGS.LIS

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES IN DATA SET SNKCAN.IND.
0004    C       THESE VALUES ARE THE INITIAL CANAL STORAGE ALLOCATIONS.
0005    C
0006    C             RJS  -  MAY 1988, MODIFIED DECEMBER 1988
0007    C
0008    C*************************************************************************
0009
0010          DIMENSION STOR(500),SRM(500),NAME(500),ID(500),IST(500)
0011          CHARACTER*16 NAME
0012          CALL ASSIGN (1,'SNKCAN.IND')
0013          CALL ASSIGN (8,'SNKWRA.DPL')
0014          OPEN (UNIT=2,NAME='SNKCAN.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0015    C
0016    C***** READ ID NUMBERS & NAMES OF ALL DIVERSIONS FROM MASTER LIST.
0017    C***** IF IST = 1, DIVERSION NEVER HAS STORAGE AMOUNTS, SO WE CAN
0018    C***** SKIP IT.
0019    C
0020          DO 10 L=1,500
0021          READ(8,9,END=7) ID(L),IST(L),NAME(L)
0022        9 FORMAT(I10,14X,I1,A16)
0023       10 CONTINUE
0024        7 L=L-1
0025    C
0026    C***** CHECK IF WE ARE TO PRESERVE SOME VALUES FROM OLD DATA SET
0027    C
0028          PRINT *,' '
0029          PRINT *,' IF YOU ARE STARTING OVER WITH ALL NEW VALUES OF    '
0030          PRINT *,' STORAGE USED AND REMAINING STORAGE, THEN ANSWER    '
0031          PRINT *,' YES TO THE FOLLOWING QUESTION, OTHERWISE THE       '
0032          PRINT *,' THE CURRENT VALUES IN DATA SET SNKCAN.IND WILL BE  '
0033          PRINT *,' PRESERVED, ALLOWING YOU TO MODIFY ONLY THOSE YOU   '
0034          PRINT *,' WISH (THEN YOU DO NOT HAVE TO REENTER ALL VALUES).'
0035          PRINT *,' '
0036          TYPE 14
0037       14 FORMAT('   DO YOU WISH TO ENTER ALL NEW VALUES? Y/N '$)
0038          ACCEPT 102, IANS
0039          IF(IANS.EQ.'Y') GO TO 12
0040    C
0041    C***** NOW READ OLD VALUES OF STORAGE REMAINING IN ACCOUNTS, AND WHEN
0042    C***** ID NUMBERS MATCH UP, PLACE THAT VALUE OF STORAGE IN SRM ARRAY.
0043    C***** THIS ALLOWS DIVERSIONS TO BE ADDED OR REMOVED EASILY FROM
0044    C***** NAME FILE AND SRM ARRAY WILL BE CORRESPONDINGLY CORRECTED.
0045    C
0046          ISKIP=0
0047          DO 1 I=1,L
0048          STOR(I)=0.0
0049          SRM(I)=0.0
0050          IF(ISKIP.EQ.0) READ(1,2,END=1) ID2,STR,SREM
0051        2 FORMAT(I8,2F10.0)
0052          ISKIP=0
0053          IF(ID2.EQ.ID(I)) GO TO 5
0054          IF(ID2.GT.ID(I)) ISKIP=1
0055          GO TO 1
0056        5 STOR(I)=STR
0057          SRM(I)=SREM
```

```
0058        1 CONTINUE
0059     C
0060     C***** NOW ASK FOR NEW VALUES OF STORAGE.
0061     C
0062       12 PRINT *,' '
0063          PRINT *,' ENTER ANY CHANGES YOU WISH FOR DIVERSION STORAGE USED'
0064          PRINT *,' '
0065          PRINT *,' ANSWER   Y = YES    N OR HIT RETURN KEY = NO         '
0066          PRINT *,'          Q = QUIT   J = JUMP AROUND                   '
0067          PRINT *,' '
0068          JUMP=0.0
0069       13 DO 101 I=1,L
0070          IF(ID(I).LT.JUMP) GO TO 101
0071          IF(IST(I).EQ.1) GO TO 101
0072        6 TYPE 11, ID(I),NAME(I),STOR(I)
0073       11 FORMAT(' ',I8,1X,A16,' =',F10.1,' NEW VALUE? Y/N/J/Q '$)
0074          ACCEPT 102, IANS
0075      102 FORMAT(A1)
0076          IF(IANS.EQ.'Q') GO TO 15
0077          IF(IANS.EQ.'J') GO TO 3
0078          IF(IANS.NE.'Y') GO TO 101
0079          PRINT 103
0080      103 FORMAT('          ENTER NEW VALUE   ='$)
0081          READ(5,4,ERR=106) STOR(I)
0082        4 FORMAT(F10.0)
0083          GO TO 101
0084      106 PRINT 107
0085      107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0086          GO TO 6
0087      101 CONTINUE
0088          GO TO 15
0089        3 PRINT *,' '
0090          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0091          PRINT *,'                 46500 BEGINS FALL RIVER         '
0092          PRINT *,'                 49500 BEGINS LOWER HENRYS FORK  '
0093          PRINT *,'                 53000 BEGINS TETON RIVER        '
0094          PRINT *,'                 57000 BEGINS SNAKE BELOW HENRYS FORK '
0095          PRINT *,'                 57500 BEGINS WILLOW CREEK '
0096          PRINT *,'                 59000 BEGINS SNAKE BELOW WILLOW CREEK'
0097          PRINT *,'                 75000 BEGINS SNAKE BELOW BLACKFOOT '
0098          PRINT *,' '
0099          PRINT 104
0100      104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0101         15'/     ' **** USE ONLY LAST 5 DIGITS.                       '$)
0102          ACCEPT 105, JUMP
0103      105 FORMAT(I5)
0104          JUMP=JUMP+13000000
0105          GO TO 13
0106       15 PRINT *,' '
0107          PRINT *,' ENTER ANY CHANGES YOU WISH FOR REMAINING STORAGE'
0108          PRINT *,' '
0109          PRINT *,' ANSWER   Y = YES    N OR HIT RETURN KEY = NO         '
0110          PRINT *,'          Q = QUIT   J = JUMP AROUND                   '
0111          PRINT *,' '
0112          JUMP=0.0
0113       16 DO 18 I=1,L
0114          IF(ID(I).LT.JUMP) GO TO 18
```

```
0115          IF(IST(I).EQ.1) GO TO 18
0116       17 TYPE 11, ID(I),NAME(I),SRM(I)
0117          ACCEPT 102, IANS
0118          IF(IANS.EQ.'Q') GO TO 100
0119          IF(IANS.EQ.'J') GO TO 19
0120          IF(IANS.NE.'Y') GO TO 18
0121          PRINT 103
0122          READ(5,4,ERR=20) SRM(I)
0123          GO TO 18
0124       20 PRINT 107
0125          GO TO 17
0126       18 CONTINUE
0127          GO TO 100
0128       19 PRINT *,' '
0129          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0130          PRINT *,'                 46500 BEGINS FALL RIVER         '
0131          PRINT *,'                 49500 BEGINS LOWER HENRYS FORK  '
0132          PRINT *,'                 53000 BEGINS TETON RIVER        '
0133          PRINT *,'                 57000 BEGINS SNAKE BELOW HENRYS FORK '
0134          PRINT *,'                 57500 BEGINS WILLOW CREEK '
0135          PRINT *,'                 59000 BEGINS SNAKE BELOW WILLOW CREEK'
0136          PRINT *,'                 75000 BEGINS SNAKE BELOW BLACKFOOT '
0137          PRINT *,' '
0138          PRINT 104
0139          ACCEPT 105, JUMP
0140          JUMP=JUMP+13000000
0141          GO TO 16
0142    C
0143    C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0144    C***** ID NUMBERS AND STORAGE ACCOUNT VALUES TO NEW FILE.
0145    C
0146      100 PRINT 8
0147        8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE CANAL VALUES?
0148         1  Y/N '$)
0149          ACCEPT 102, IANS
0150          IF(IANS.EQ.'Y') GO TO 12
0151          WRITE(2,21) (ID(I),STOR(I),SRM(I),I=1,L)
0152       21 FORMAT(I8,2F10.1)
0153          PRINT *,' '
0154          PRINT *,' FILE SNKCAN.IND HAS BEEN UPDATED WITH NEW VALUES'
0155          PRINT *,' '
0156          PRINT *,' '
0157          PRINT *,' '
0158          STOP
0159          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	2397	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1386	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	16496	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	20279	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	I*4	I		47= 48 49 53 54 56 57 69= 70 71 72(3) 81 113= 114 115 116(122 151(4)=
2-00003E80	I*4	IANS		38= 39 74= 76 77 78 117= 118 119 120 149= 150
2-00003E88	I*4	ID2		50= 53 54
**	I*4	ISKIP		46= 50 52= 54=
2-00003E94	I*4	JUMP		68= 70 102= 104(2)= 112= 114 139= 140(2)=
**	I*4	L		20= 21(3) 24(2)= 47 69 113 151
2-00003E90	R*4	SREM		50= 57
2-00003E8C	R*4	STR		50= 56

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-00000FA0	I*4	ID		2000	(500)	10 21= 53 54 70 72 114 116 15
2-00001770	I*4	IST		2000	(500)	10 21= 71 115
2-00001F40	CHAR	NAME		8000	(500)	10 11 21= 72 116
2-000007D0	R*4	SRM		2000	(500)	10 49= 57= 116 122= 151
2-00000000	R*4	STOR		2000	(500)	10 48= 56= 72 81= 151

LABELS

Address	Label	References					
0-0000020B	1	47	50	55	58#		
1-00000429	2'	50	51#				
0-000003C8	3	77	89#				
1-0000047C	4'	81	82#	122			
0-00000200	5	53	56#				
**	6	72#	86				
0-00000078	7	21	24#				
1-00000523	8'	146	147#				
1-000003F0	9'	21	22#				
**	10	20	23#				
1-00000431	11'	72	73#	116			
0-0000020F	12	39	62#	150			
0-000002BC	13	69#	105				
1-000003F9	14'	36	37#				
0-0000052A	15	76	88	106#			
0-000005D4	16	113#	141				
**	17	116#	125				
0-0000083C	18	113	114	115	120	123	126#
0-000006E0	19	119	128#				
0-000006C4	20	122	124#				
1-00000562	21'	151	152#				
0-00000842	100	118	127	146#			
0-00000524	101	69	70	71	78	83	87#
1-00000459	102'	38	74	75#	117	149	
1-0000045C	103'	79	80#	121			
1-000004AD	104'	99	100#	138			
1-00000520	105'	102	103#	139			
0-000003AC	106	81	84#				
1-00000480	107'	84	85#	124			

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 SNKCHGC

  /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.SNAKE]SNKCHGC.LIS;5
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGC.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           5.98 seconds
  Elapsed Time:       9.04 seconds
  Page Faults:        819
  Dynamic Memory:     493 pages
```

SNKACHGC.LIS

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE CANAL REMAINING STORAGE FROM ALLOCATIONS
0004    C       FILE FOR INPUT TO ACCOUNTING PROGRAM.  RJS - JULY 1988.
0005    C
0006    C*************************************************************************
0007
0008          DIMENSION SRM(400),NAME(400),ID(400),CODE(400),SRU(400)
0009          DIMENSION ID1(400),JK(400)
0010          CHARACTER*8 CODE,CODEK
0011          CHARACTER*16 NAME
0012          CALL ASSIGN (1,'SNKWRA.ALD')
0013          CALL ASSIGN (8,'SNKWRA.DPL')
0014          OPEN (UNIT=2,NAME='SNKALD.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0015          READ(1,2,END=1) (ID1(K),CODE(K),SRU(K),SRM(K),K=1,400)
0016        2 FORMAT(I8,A8,2F10.1)
0017        1 KEND=K-1
0018          DO 10 L=1,400
0019          READ(8,9,END=7) ID(L),JK(L),NAME(L)
0020        9 FORMAT(I10,14X,I1,A16)
0021       10 CONTINUE
0022        7 LEND=L-1
0023          PRINT *,' '
0024          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE CANAL STORAGE ACCOU
0025         1NTS  '
0026          PRINT *,' '
0027          K=0
0028          KR=1
0029          DO 101 I=1,LEND
0030          IF(KR.GT.0) THEN
0031             K=K+1
0032             KR=0
0033             END IF
0034          IF(K.GT.KEND) GO TO 6
0035          CODEK=CODE(K)
0036          SRUI=SRU(K)
0037          SRMI=SRM(K)
0038          IF(ID1(K).EQ.ID(I)) GO TO 98
0039    C
0040    C***** IF JK(I) FROM THE TITLE LIST IS > 0, THE DIVERSION DOES NOT
0041    C***** HAVE RESERVOIR STORAGE, AND THEREFORE THERE IS NO NEED TO
0042    C***** PROMPT FOR A CHANGE
0043    C
0044        6 IF(JK(I).GT.0) GO TO 101
0045          SRUI=0.0
0046          SRMI=0.0
0047          GO TO 99
0048       98 KR=1
0049       99 TYPE 11, ID(I),NAME(I),SRMI
0050       11 FORMAT(' ',I8,1X,A16,' =',F10.1,' NEW VALUE? Y/N/Q '$)
0051          ACCEPT 102, IANS
0052      102 FORMAT(A1)
0053          IF(IANS.EQ.'Y') GO TO 100
0054          IF(IANS.EQ.'N') GO TO 3
0055          IF(IANS.EQ.'Q') GO TO 104
0056          GO TO 99
0057      100 PRINT 103
```

```
0058      103 FORMAT('         ENTER NEW VALUE   ='$)
0059          ACCEPT 4, SRMI
0060        4 FORMAT(F10.0)
0061        3 IF(SRMI.EQ.0.0.AND.SRUI.EQ.0.0) GO TO 101
0062          WRITE(2,2) ID(I),CODEK,SRUI,SRMI
0063      101 CONTINUE
0064          GO TO 5
0065      104 IF(K.LE.KEND) WRITE(2,2) (ID1(I),CODE(I),SRU(I),SRM(I),I=K,KEND)
0066        5 PRINT *,' '
0067          PRINT *,' FILE HAS BEEN UPDATED WITH NEW VALUES'
0068          PRINT *,' '
0069          PRINT *,' '
0070          PRINT *,' '
0071          STOP
0072          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	947	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	238	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	17808	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	18993	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKACHGC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
2-000044C0	CHAR	CODEK		10	35=	62					
**	I*4	I		29=	38	44	49(2)	62	65(5)=		
2-000044D4	I*4	IANS		51=	53	54	55				
2-000044C8	I*4	K		15(5)=	17	27=	31(2)=	34	35	36	37
				38	65(2)						
**	I*4	KEND		17=	34	65(2)					
**	I*4	KR		28=	30	32=	48=				
**	I*4	L		18=	19(3)	22					
**	I*4	LEND		22=	29						
2-000044D0	R*4	SRMI		37=	46=	49	59=	61	62		
**	R*4	SRUI		36=	45=	61	62				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00003840	CHAR	CODE		3200	(400)	8	10	15=	35	65
2-00000640	I*4	ID		1600	(400)	8	19=	38	49	62
2-000012C0	I*4	ID1		1600	(400)	9	15=	38	65	
2-00001900	I*4	JK		1600	(400)	9	19=	44		
2-00001F40	CHAR	NAME		6400	(400)	8	11	19=	49	
2-00000000	R*4	SRM		1600	(400)	8	15=	37	65	
2-00000C80	R*4	SRU		1600	(400)	8	15=	36	65	

LABELS

Address	Label	References			
0-0000008B	1	15	17#		
1-0000008E	2'	15	16#	62	65
0-000002D4	3	54	61#		
1-000000EA	4'	59	60#		
0-00000320	5	64	66#		
0-00000184	6	34	44#		
0-000000E4	7	19	22#		
1-00000098	9'	19	20#		
**	10	18	21#		
1-000000A1	11'	49	50#		
0-0000017E	98	38	48#		
0-00000192	99	47	49#	56	
0-00000298	100	53	57#		
0-0000031A	101	29	44	61	63#
1-000000C7	102'	51	52#		
1-000000CA	103'	57	58#		
0-00000224	104	55	65#		

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 SNKACHGC

  /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.SNAKE]SNKACHGC.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKACHGC.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:          3.36 seconds
  Elapsed Time:      5.12 seconds
  Page Faults:       315
  Dynamic Memory:    407 pages
```

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE RESERVOIR RIGHT STORAGE ACCUMULATIONS,
0004    C       AND OTHER SYSTEM ACCUMULATIONS FROM ALLOCATIONS FILE AS
0005    C       INPUT TO ACCOUNTING PROGRAM.   RJS - MAY 1988.
0006    C
0007    C*************************************************************************
0008
0009          DIMENSION CEV(9),GT(9),BRST(16),NAME(9),RIGHT(16),VAL(10)
0010          CHARACTER*16 CODE1,CODE2,CODE3,CODE4,CODE5,CODE6,CODE7
0011          CHARACTER*17 NAME,RIGHT
0012          DATA NAME/'CHANGE IN STORAGE','MILNER STORED    ',
0013         1'TOTAL NEW STORAGE','UNACCOUNTED STORE','TTL MINIDOKA CRDT',
0014         2'TOTAL RIRIE LOSS ','TOTAL WILLOW LOSS','TTL EAGLE RK CRDT',
0015         2'AM FALLS GAIN DIF'/
0016          DATA RIGHT/'JACKSON LAKE 1906','LAKE WALCOTT     ',
0017         1'JACKSON LAKE 1910','JACKSON LAKE 1913','HENRYS LAKE 1917 ',
0018         2'PALISADES 1921   ','ISLAND PARK 1921 ','AMERICAN FALLS #1',
0019         3'AMERICAN FALLS #2','ISLAND PARK 1935 ','GRASSY LAKE      ',
0020         4'PALISADES 1939   ','HENRYS LAKE 1965 ','RIRIE            ',
0021         5'AM FLS LAST FILL ','PAL LAST FILL    '/
0022          CALL ASSIGN (1,'SNKWRA.ALM')
0023          OPEN (UNIT=2,NAME='SNKALM.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0024          READ(1,3,END=6) CODE1,(BRST(L),L=1,5),CODE2,(BRST(L),L=6,10),
0025         1  CODE3,(BRST(L),L=11,15),CODE4,BRST(16)
0026        3 FORMAT(A16,5F10.0)
0027          READ(1,2,END=6) CODE5,(GT(L),L=1,5),CODE6,(GT(L),L=6,9)
0028        2 FORMAT(A16,5F10.0/A16,4F10.0)
0029          PRINT *,'                                                    '
0030          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR STORAGE
0031         1RIGHTS '
0032          PRINT *,' '
0033          DO 104 I=1,16
0034          TYPE 11, RIGHT(I),BRST(I)
0035          ACCEPT 102, IANS
0036          IF(IANS.NE.'Y') GO TO 104
0037          PRINT 103
0038          ACCEPT 4, BRST(I)
0039      104 CONTINUE
0040          WRITE(2,8) CODE1,(BRST(L),L=1,5),CODE2,(BRST(L),L=6,10),
0041         1   CODE3,(BRST(L),L=11,15),CODE4,BRST(16)
0042        8 FORMAT(A16,5F10.1)
0043          PRINT *,' '
0044          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE YEAR TO DATE TOTALS
0045         1  '
0046          PRINT *,' '
0047          DO 101 I=1,9
0048          TYPE 11, NAME(I),GT(I)
0049       11 FORMAT(' ',A17,' =',F10.1,' NEW VALUE? Y/N '$)
0050          ACCEPT 102, IANS
0051      102 FORMAT(A1)
0052          IF(IANS.NE.'Y') GO TO 101
0053          PRINT 103
0054      103 FORMAT(' ENTER NEW VALUE   ='$)
0055          ACCEPT 4, GT(I)
0056        4 FORMAT(F10.0)
0057      101 CONTINUE
```

```
0058          WRITE(2,9) CODE5,(GT(L),L=1,5),CODE6,(GT(L),L=6,9)
0059        9 FORMAT(A16,5F10.1/A16,4F10.1)
0060          PRINT *,' '
0061    C
0062    C****** TRANSFER REMAINING ALLOCATIONS DATA TO NEW FILE.  THESE DATA
0063    C****** ARE THE VOLUME LIMIT RIGHTS AND TOTAL PUMPING EXCHANGE DATA.
0064    C
0065          DO 7 I=1,9
0066          READ(1,1) CODE7,(VAL(L),L=1,5)
0067        1 FORMAT(A16,5F10.0)
0068          WRITE(2,5) CODE7,(VAL(L),L=1,5)
0069        5 FORMAT(A16,5F10.1)
0070        7 CONTINUE
0071        6 STOP
0072          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1163	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	331	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	908	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	2402	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKACHGS$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
2-00000259	CHAR	CODE1		10	24=	40				
2-00000269	CHAR	CODE2		10	24=	40				
2-00000279	CHAR	CODE3		10	24=	40				
2-00000289	CHAR	CODE4		10	24=	40				
2-00000299	CHAR	CODE5		10	27=	58				
2-000002A9	CHAR	CODE6		10	27=	58				
2-000002B9	CHAR	CODE7		10	66=	68				
**	I*4	I		33=	34(2)	38	47=	48(2)	55	65=
2-000002D0	I*4	IANS		35=	36	50=	52			
2-000002CC	I*4	L		24(6)=	27(4)=	40(6)=	58(4)=	66(2)=	68(2)=	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000048	R*4	BRST		64	(16)	9	24(4)=	34	38=	40(4)
2-00000000	R*4	CEV		36	(9)	9				
2-00000024	R*4	GT		36	(9)	9	27(2)=	48	55=	58(2)
2-000000B0	CHAR	NAME		153	(9)	9	11	12D	48	
2-00000149	CHAR	RIGHT		272	(16)	9	11	16D	34	
2-00000088	R*4	VAL		40	(10)	9	66=	68		

LABELS

Address	Label	References		
1-0000013B	1'	66	67#	
1-000000D4	2'	27	28#	
1-000000CC	3'	24	26#	
1-00000127	4'	38	55	56#
1-00000143	5'	68	69#	
0-00000484	6	24	27	71#
**	7	65	70#	
1-000000E4	8'	40	42#	
1-0000012B	9'	58	59#	
1-000000EC	11'	34	48	49#
0-00000393	101	47	52	57#
1-0000010C	102'	35	50	51#
1-0000010F	103'	37	53	54#
0-00000203	104	33	36	39#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	22
	FOR$OPEN	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 SNKACHGS

  /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.SNAKE]SNKACHGS.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKACHGS.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           3.60 seconds
  Elapsed Time:       12.67 seconds
  Page Faults:        700
  Dynamic Memory:     423 pages
```

SNKSTO.LIS

```
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      DIMENSION KREND(11),FILL(12),SPACE(12),YIELD(12),EVAP(12)
      DIMENSION PYLD(9),T(4),RNAME(12),BANK(450),GSTOT(9),JID(50)
      DIMENSION ISK(30),IOR(11),SPAC(450,12),RDIFF(450),GTOT(13)
      DIMENSION JDFL(450),TSPAC(12),ID2(450),PAVAIL(12)
      DIMENSION IK(450),NSTO(450,15),TOTR(450,14),CNSTO(12),CTOTC(12)
      DIMENSION FILLATE(12),TFILL(9)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      CHARACTER*14 RNAME
      CHARACTER*14 U,RNAM
      DATA KREND/13038150,13038500,13046450,13049500,13050590,13055351,
     1 13057260,13058550,13069500,13088000,99999999/
      DATA ISK/13010500,13032450,13039000,13042000,13042600,13046500,
     1 13057950,13059050,13076500,13076751,13081000,13081400,18*0/
C
C***** THE FOLLOWING DATA STATEMENT DICTATES THE ORDER FROM WHICH THE
C***** STORAGE IS REMOVED FROM RESERVOIRS.
C*****
C*****    1 - JACKSON LAKE      2 - PALISADES       3 - PAL WATER USERS
C*****    4 - PAL WINTER WATER  5 - HENRYS LAKE     6 - ISLAND PARK
C*****    7 - GRASSY LAKE       8 - RIRIE           9 - AMERICAN FALLS
C*****   10 - LAKE WALCOT
C*****   11 - FIRST 200,000 ACRE FEET IN JACKSON LAKE
C
      DATA IOR/6,7,11,10,9,3,4,2,1,5,8/
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      REAL*8 FILL,SPACE,YIELD,GTOT,PYLD,T,GSTOT,TSPAC,GRTOT,SGRTOT
      REAL*8 CTOTC,CNSTO,NSTO,BANK,SPAC,FILLATE,TFILL,TLATE,JDFL
      CALL DATE_AND_TIME(RUNDATE)
        OPEN(UNIT=1,NAME='INDIN',TYPE='OLD')
        OPEN(UNIT=5,NAME='CARRYIN',TYPE='OLD')
        OPEN(UNIT=6,NAME='REPORTOUT',TYPE='NEW')
        OPEN(UNIT=7,NAME='SPACEIN',TYPE='OLD')
        OPEN(UNIT=9,NAME='USEIN',TYPE='OLD')
C       THE OLD CALL DATE AND CALL ASSIGN STATEMENTS ARE THE FOLLOWING.
C      CALL DATE(RUNDATE)
C      CALL ASSIGN(1,'INDIN')
C      CALL ASSIGN(5,'CARRYIN')
C      CALL ASSIGN(6,'REPORTOUT')
C      CALL ASSIGN(7,'SPACEIN')
C      CALL ASSIGN(9,'USEIN')
C       WEIMIN LI 8/8/2002
      L=1
      I=1
   30 READ(7,10,END=20) ID(I),IK(I),UNAME(I),(STOR(I,K),K=1,12)
   10 FORMAT(I8,A1,A14,12F9.0)
      STOR(I,13)=0.0
      STOR(I,14)=0.0
      STOR(I,15)=0.0
      IF(ID(I).EQ.99999300) IPALU=I
      IF(ID(I).EQ.13087000) JNSTF=I
```

```
      IF(ID(I).EQ.13087500) JSSTF=I
      IF(ID(I).LT.KREND(L)) GO TO 40
      NDEND(L)=I-1
      L=L+1
   40 I=I+1
      GO TO 30
   20 CONTINUE
      NDEND(L)=I-1
      JEN=NDEND(10)
      JE=NDEND(11)
      DO 60 I=1,JE
      DO 60 K=1,12
      CRYO(I,K)=0.0
   60 SPAC(I,K)=STOR(I,K)
      DO 45 I=1,JE
      READ(5,12,END=5) ID2(I),U,(CRYO(I,J),J=1,12)
   12 FORMAT(I8,1X,A14,12F9.0)
      IF(ID2(I).EQ.ID(I)) GO TO 45
      TYPE 2, ID2(I),ID(I)
    2 FORMAT(' DIVERSION ',I8,' FROM SNKSTO.CRY IS DIFFERENT THAN DIVERS
     1ION ',I8,' FROM SNKSTO.SPA')
      STOP
   45 CONTINUE
    5 TYPE *, ' '
      TYPE *, '  ENTER FOUR DIGIT YEAR FOR WHICH YOU WANT   '
      TYPE *, '  TO COMPUTE STORAGE ALLOCATION AND USE.     '
      TYPE *, '  FOR EXAMPLE: 1995                          '
      PRINT 46
   46 FORMAT(15X,' '$)
      ACCEPT 49,IYR
   49 FORMAT(I4)
      READ(1,55) (RNAME(K),SPACE(K),FILL(K),EVAP(K),FILLATE(K),K=1,12)
   55 FORMAT(9X,A14,2X,4F10.0)
C
C***** COMPUTE AND PRINT TABLE OF EACH RESERVOIR'S SPACE, FILL,
C***** EVAPORATION AND YIELD.
C
      DO 90 K=1,12
      YIELD(K)=FILL(K)-EVAP(K)
      IF(FILL(K).GT.0.0) PAVAIL(K)=YIELD(K)/FILL(K)
      IF(FILL(K).LE.0.0) PAVAIL(K)=0.0
   90 CONTINUE
      WRITE(6,100) RUNDATE,IYR
  100 FORMAT(1H1//////115X,A9/////////////////,43X,I4,
     1  ' SNAKE 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=1,4
  105 T(K)=0.0
      DO 110 K=1,12
      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,A14,2X,4F12.1)
```

```
      IF(K.EQ.11) WRITE(6,125) (T(I),I=1,4)
  125 FORMAT(/32X,9HSUB TOTAL,7X,4F12.1/)
  110 CONTINUE
      WRITE(6,130) (T(K),K=1,4)
  130 FORMAT(/32X,5HTOTAL,11X,4F12.1)
      IT=1
  185 CONTINUE
      DO 155 I=1,10
      DO 155 K=1,15
  155 CTOT(I,K)=0.0
      CALL TABLE (0,1,1,1)
      CALL TABLE (1,2,0,1)
      CALL TABLE (6,7,1,2)
      CALL TABLE (8,9,0,2)
      CALL TABLE (9,10,0,3)
      CALL TABLE (2,3,1,4)
      CALL TABLE (4,5,0,4)
      CALL TABLE (3,4,0,5)
      CALL TABLE (5,6,0,6)
      CALL TABLE (7,8,0,7)
      CALL TABLE (10,11,0,8)
      WRITE(6,235) RUNDATE,IT
  235 FORMAT(1H1/////123X,A9///58X,4HPART,I2)
      IF(IT.EQ.1) WRITE(6,240) IYR
  240 FORMAT(1H //15X,I4,98H WATER DISTRICT 1 RESERVOIR SPACE BY REACH A
     1SSUMING FULL RESERVOIRS WITHOUT RENTAL POOL(ACRE-FEET)/)
      IF(IT.EQ.2) WRITE(6,250) IYR
  250 FORMAT(  //12X,I4,108H WATER DISTRICT 1 RESERVOIR STORAGE BY REACH
     1 AFTER PERCENT FILL AND EVAPORATION, WITH RENTAL POOL(ACRE-FEET)/)
      DO 195 K=1,15
  195 GTOT(K)=0.0
      WRITE(6,210)
  210 FORMAT(///// 8X,11HRIVER REACH,9X, 94H   JACKSON    PALISADES
     1   PAL USR       PAL WWS   HENRYS LAKE   ISLAND PARK   GRASSY LAKE
     2 /)
      DO 190 I=1,8
      DO 205 K=1,15
  205 GTOT(K)=GTOT(K)+CTOT(I,K)
  190 WRITE(6,220) (RCHNM(K,I),K=1,5),(CTOT(I,J),J=1,7)
  220 FORMAT(8X,5A4,7(F10.1,4X))
      WRITE(6,230) (GTOT(K),K=1,7)
  230 FORMAT(/8X,5HTOTAL,15X,7(F10.1,4X))
      WRITE(6,211)
  211 FORMAT(///// 8X,11HRIVER REACH,5X, 98H         RIRIE      AM FALLS
     1  LAKE WALCOTT       AMF LTF       PAL LTF   RENTAL POOL         T
     2OTAL/)
      DO 212 I=1,8
  212 WRITE(6,220) (RCHNM(K,I),K=1,5),(CTOT(I,J),J=8,12),
     1  (CTOT(I,J),J=14,15)
      WRITE(6,230) (GTOT(K),K=8,12),(GTOT(K),K=14,15)
      IF (IT.EQ.2) GO TO 200
      DO 232 K=1,12
  232 TSPAC(K)=GTOT(K)
C
C***** MOVE AS MUCH WWS CARRYOVER INTO MAIN PALISADES ACCOUNT AS
C***** POSSIBLE AS PER USBR METHOD OF ALLOCATING NEW STORAGE - NEW
C***** FILL IN WWS SPACE THEN DOES NOT INCLUDE CARRYOVER.
```

```
C
C***** THIS ROUTINE NOT USED BEGINNING IN 1992 BECAUSE IT PRODUCED
C***** NEGATIVE ALLOCATIONS WHEN THERE WAS LITTLE OR NO NEW FILL
C***** OF PALISADES REGULAR SPACE.  NOT SURE OF EXACT REASONING
C***** WHEN THIS WAS FIRST USED.
C
C       DO 25 I=1,JE
C       K=2
C       IF(I.EQ.IPALU) K=3
C       CRYO(I,K)=CRYO(I,K)+CRYO(I,4)
C       CRYO(I,4)=0.0
C       IF(CRYO(I,K).GT.SPAC(I,K)) THEN
C           CRYO(I,K)=SPAC(I,K)
C           CRYO(I,4)=CRYO(I,K)-SPAC(I,K)
C           END IF
C    25 CONTINUE
C
C***** INITIALIZE NEW STORAGE (NSTO) TO ZERO AND TOTAL CARRYOVER
C***** BY RESERVOIR ACCOUNT.
C
      DO 14 K=1,12
      CTOTC(K)=0.0
      DO 14 I=1,JE
      NSTO(I,K)=0.0
   14 CTOTC(K)=CTOTC(K)+CRYO(I,K)
      IJKSN=0
C
C***** BEGIN LOOP TO ALLOCATE NEW FILL TO THE TWELVE RESERVOIR
C***** ACCOUNTS.
C
      DO 1 K=1,12
      KRS=K
      IF(K.GT.3) KRS=K-1
      TSP=TSPAC(K)
C
C***** COMBINE PALISADES AND PALISADES WATER USERS.
C
      IF(K.EQ.2) TSP=TSPAC(2)+TSPAC(3)
      IF(TSP.LE.0.0) GO TO 1
      FIL=FILL(KRS)
      RNAM=RNAME(KRS)
      IF(K.EQ.3) THEN
           FIL=PUTSTO
           RNAM='PALISADES USRS'
           END IF
      IF(FIL.LE.TSP) GO TO 13
      TYPE 997, RNAM,FIL,TSP
  997 FORMAT(1X,A16,' HAS FILL =',F10.1,' > TOTAL SPACE =',F10.1)
      GO TO 400
   13 IF(FIL.EQ.TSP) GO TO 23
C
C***** ZERO CARRYOVER IF FAILURE TO FILL WAS RESULT OF FLOOD OPERATION
C***** OR OTHER RESERVOIR OPERATIONS.
C
      PRINT 22, RNAM,FIL,TSP
   22 FORMAT(1X,A16,' HAS FILL =',F10.1,' < TOTAL SPACE =',F10.1/
     1'   IS THE FAILURE TO FILL A RESULT OF FLOOD CONTROL? '/
```

```
      2'   OR DO SPACEHOLDERS NOT HAVE CARRYOVER PRIVILEGES? '/
      3'   IF YOU ANSWER Y, CARRYOVER WILL BE WIPED OUT. Y/N '$)
       ACCEPT 36, FLD
       IF(FLD.NE.'Y') GO TO 24
   23 KC=K
   27 DO 26 I=1,JE
   26 CRYO(I,KC)=0.0
      CTOTC(KC)=0.0
      IF(KC.EQ.2) THEN
                  KC=KC+1
                  GO TO 27
                  END IF
C
C***** COMPUTE NEW FILL TO EACH RESERVOIR BY SUBTRACTING CARRYOVER
C***** FROM TOTAL FILL. THAT AMOUNT (WAT) IS THEN DISTRIBUTED.
C
   24 TCR=CTOTC(K)
      IF(K.EQ.2) TCR=CTOTC(2)+CTOTC(3)
      CNSTO(K)=FIL-TCR
      WAT=CNSTO(K)
      IF(WAT.LE.-0.5) THEN
        PRINT 47, RNAM,TCR,FIL
   47 FORMAT(1X,A16,' HAS TOTAL CARRYOVER =',F10.1,' > TOTAL FILL=',
     1F10.1/'   DO YOU WISH TO STOP? ANSWER Y/N '$)
        ACCEPT 36, STP
        IF(STP.EQ.'Y') GO TO 400
        ENDIF
      J=0
C
C***** IF ALLOCATING JACKSON LAKE, REMOVE LAST TO FILL SPACE OF NORTH
C***** AND SOUTH SIDE TWIN FALLS CANALS (TJTFSP) AND ALSO CORRESPONDING
C***** CARRYOVER (TJTFCO) FROM TOTAL SPACE (TSP) AND TOTAL FIL (FIL),
C***** RESPECTIVELY.  ALSO CORRECT TOTAL CARRYOVER (TCR).  IF REVISED
C***** FILL (FIL) IS GREATER THAN REVISED TOTAL SPACE (TSP), THEN
C***** SPACE OTHER THAN TWIN FALLS JACKSON SPACE HAS FILLED COMPLETELY,
C***** SO SET TOTAL FILL EQUAL TO TOTAL SPACE.
C
      IF(K.GT.1) GO TO 6
      IF(IJKSN.GT.0.OR.FIL.EQ.TSP) GO TO 6
      IF(FLD.NE.'Y') THEN
           IJKSN=1
           TJNSTFSP=SPAC(JNSTF,1)
           TJSSTFSP=SPAC(JSSTF,1)
           SPAC(JNSTF,1)=0.0
           SPAC(JSSTF,1)=0.0
           TJTFSP=TJNSTFSP+TJSSTFSP
           TJNSTFCO=CRYO(JNSTF,1)
           TJSSTFCO=CRYO(JSSTF,1)
           TJTFCO=TJNSTFCO+TJSSTFCO
           CRYO(JNSTF,1)=0.0
           CRYO(JSSTF,1)=0.0
           TSP=TSP-TJTFSP
           FIL=FIL-TJTFCO
           TCR=TCR-TJTFCO
           IF(FIL.GE.TSP) THEN
               JKTFNSTO=FIL-TSP
               FIL=TSP
```

```
286                GO TO 23
287                ENDIF
288              JKTFNSTO=0.0
289              ENDIF
290 C
291 C***** SET TOTAL NEW STORAGE (TNSTOR) ASSIGNED TO SPACEHOLDERS EQUAL
292 C***** TO ZERO.  KEEP TRACK OF NUMBER OF PASSES MADE TO DISTRIBUTE
293 C***** ALL NEW STORAGE (CNSTO(K)) AND STOP IF IT GOES WILD.
294 C
295      6 TNSTOR=0.0
296        J=J+1
297        IF(J.GT.2000) TYPE 998, RNAM
298    998 FORMAT(' RUNAWAY LOOP ALLOCATING ',A16,' STORAGE')
299        TYPE 999, J,WAT
300    999 FORMAT(1X,I5,F12.4)
301 C
302 C***** USE TOTAL RESERVOIR SPACE FOR ANY OF THE AMERICAN FALLS OR
303 C***** PALISADES ACCOUNTS (EXCEPT LAST TO FILL), SINCE THOSE ARE ALL PROPORTIONAL TO
304 C***** TOTAL SPACE AS DESCRIBED BELOW.
305 C
306        IF(FLD.NE.'Y') THEN
307             IF(K.EQ.2.OR.K.EQ.4.) TSP=1200000.0
308             IF(K.EQ.9) TSP=1672590.0
309 C
310 C***** CHANGED 10/2002 --  THE ABOVE THREE STATEMENTS WERE ADDED AND THE FOLLOWING
311 C***** THREE STATEMENTS WERE DELETED SO THAT LAST TO FILL SPACE (11 AND 12) IN
312 C***** AMERICAN FALLS AND PALISADES FILLS PROPORTIONAL TO THE SPACE IN THOSE
313 C***** ACCOUNTS RATHER THAN PROPORTIONAL TO THE ENTIRE SPACE THAT EACH STORAGE
314 C***** HOLDER OWNS IN THE RESERVOPIR AS A WHOLE (THE PERCENTAGE IN THE
315 C***** CONTRACTS).
316 C
317 C       IF(FLD.NE.'Y') THEN
318 C            IF(K.EQ.2.OR.K.EQ.4.OR.K.EQ.12) TSP=1200000.0
319 C            IF(K.EQ.9.OR.K.EQ.11) TSP=1672590.0
320              ENDIF
321        KS=K
322        DO 4 I=1,JE
323        SPC=SPAC(I,KS)
324 C
325 C***** SUM ALL SPACE (SPC) ACCOUNTS FOR PALISADES AND AMERICAN FALLS
326 C***** TO DETERMINE PROPORTIONAL FILL BECAUSE USBR CONTRACTS READ
327 C***** AS A PERCENT DETERMINED FROM ALL STORAGE IN A RESERVOIR, NOT
328 C***** FROM EACH RESERVOIR RIGHT.  FOR PALISADES USERS (IPALU=1) BE
329 C***** SURE TO ADD ALL REGULAR PALISADES USERS SPACE (TSPAC(3)).
330 C
331        IF(FLD.NE.'Y') THEN
332           IF(KS.EQ.2.OR.KS.EQ.4) THEN
333                          SPC=SPAC(I,2)+SPAC(I,4)+SPAC(I,12)
334                          END IF
335           IF(KS.EQ.9) SPC=SPAC(I,9)+SPAC(I,11)
336 C
337 C
338 C***** CHANGED 10/2002 --  THE ABOVE FIVE STATEMENTS WERE ADDED AND THE FOLLOWING
339 C***** SIX STATEMENTS WERE DELETED SO THAT LAST TO FILL SPACE (11 AND 12) IN
340 C***** AMERICAN FALLS AND PALISADES FILLS PROPORTIONAL TO THE SPACE IN THOSE
341 C***** ACCOUNTS RATHER THAN PROPORTIONAL TO THE ENTIRE SPACE THAT EACH STORAGE
342 C***** HOLDER OWNS IN THE RESERVOPIR AS A WHOLE (THE PERCENTAGE IN THE
```

```
C***** CONTRACTS).
C
C         IF(KS.EQ.2.OR.KS.EQ.4.OR.KS.EQ.12) THEN
C                         SPC=SPAC(I,2)+SPAC(I,4)+SPAC(I,12)
C                         IF(KS.EQ.12.AND.I.EQ.IPALU)
C     1                   SPC=SPAC(I,4)+SPAC(I,12)+TSPAC(3)
C                         END IF
C         IF(KS.EQ.9.OR.KS.EQ.11) SPC=SPAC(I,9)+SPAC(I,11)
          ENDIF
C
C***** ALLOCATE NEW FILL (WAT) TO INDIVIDUAL ENTITIES AND USES WITHIN
C***** EACH RESERVOIR PROPORTIONAL TO SPACE OWNED.  LIMIT TOTAL FILL
C***** (NEW FILL PLUS CARRYOVER) OF EACH SPACEHOLDER (STOR) TO SPACE
C***** OWNED (SPAC).  KEEP TRACK OF TOTAL STORAGE ASSIGNED TO SPACE-
C***** HOLDERS (TNSTOR).
C
      IF(KS.EQ.11.OR.KS.EQ.12) THEN
           IF(FIL.EQ.TSPAC(KS)) GO TO 94
           ENDIF
      IF(CRYO(I,KS).GT.SPC) TYPE 996, UNAME(I)
  996 FORMAT(' CARRYOVER STORAGE FOR ',A14,' GREATER THAN SPACE')
      NSTO(I,KS)=((SPC/TSP)*WAT)+NSTO(I,KS)
      STOR(I,KS)=NSTO(I,KS)+CRYO(I,KS)
      IF(STOR(I,KS).LE.SPAC(I,KS)) GO TO 95
   94 STOR(I,KS)=SPAC(I,KS)
      NSTO(I,KS)=STOR(I,KS)-CRYO(I,KS)
   95 TNSTOR=TNSTOR+NSTO(I,KS)
    4 CONTINUE
C
C***** WHEN FINISHED WITH PALISADES REGULAR ACCOUNTS, THEN DETERMINE
C***** PALISADES WATER USERS FILL COLLECTIVELY AS ONE ACCOUNT.
C
      IF(KS.EQ.2) THEN
                  KS=KS+1
                  SPC=TSPAC(3)
      IF(FLD.NE.'Y') SPC=SPAC(IPALU,4)+SPAC(IPALU,12)+TSPAC(KS)
                  PUNSTO=((SPC/TSP)*WAT)+PUNSTO
                  PUTSTO=PUNSTO+CTOTC(KS)
                  IF(PUTSTO.GT.TSPAC(KS)) THEN
                                          PUTSTO=TSPAC(KS)
                                          PUNSTO=PUTSTO-CTOTC(KS)
                                          END IF
                  TNSTOR=TNSTOR+PUNSTO
                  END IF
C
C***** SUBTRACT TOTAL STORAGE ALLOCATED FROM TOTAL NEW FILL TO GET
C***** AMOUNT REMAINING TO BE ALLOCATED (WAT).
C
      WAT=CNSTO(K)-TNSTOR
      IF(WAT.GT.0.07) GO TO 6
C
C***** IF JACKSON LAKE SPACE DID NOT COMPLETELY FILL AND THERE WAS NO
C***** FLOOD CONTROL, THEN IJCKSN WILL EQUAL 1 AND THE FILL OF THE
C***** TWIN FALLS SPACE MUST NOW BE CALCULATED SINCE IT FILLS LAST
C***** WITHIN JACKSON LAKE.
C
      IF(K.GT.1) GO TO 1
```

```
      IF(IJKSN.EQ.1) THEN
          J=0
          WAT=JKTFNSTO
          SPAC(JNSTF,1)=TJNSTFSP
          SPAC(JSSTF,1)=TJSSTFSP
          CRYO(JNSTF,1)=TJNSTFCO
          CRYO(JSSTF,1)=TJSSTFCO
   98     TNSTOR=0.0
          J=J+1
          IF(J.GT.2000) TYPE 998, RNAM
          TYPE 999, J,WAT
          DO 97 I=JNSTF,JSSTF
          SPC=SPAC(I,1)
          NSTO(I,1)=((SPC/TJTFSP)*WAT)+NSTO(I,1)
          STOR(I,1)=NSTO(I,1)+CRYO(I,1)
          IF(STOR(I,1).LE.SPAC(I,1)) GO TO 96
          STOR(I,1)=SPAC(I,1)
          NSTO(I,1)=STOR(I,1)-CRYO(I,1)
   96     TNSTOR=TNSTOR+NSTO(I,1)
   97     CONTINUE
          WAT=JKTFNSTO-TNSTOR
          IF(WAT.GT.0.1) GO TO 98
          END IF
    1 CONTINUE
C
C***** TRANSFER RENTAL POOL TRANSFERS TO TABLE ARRAY,
C***** COMPUTE YIELD OF SPACE VALUES AFTER EVAPORATION.
C
      PRINT 3
    3 FORMAT(' DO YOU WANT TO READ STORAGE USE, ADJUSTMENTS, AND/OR'/
     1' RENTAL POOL TRANSACTIONS BY CANAL/USER? Y/N: '$)
      ACCEPT 36, USE
   36 FORMAT(A1)
      DO 165 I=1,JE
      BANK(I)=0.0
      IF(USE.NE.'Y') GO TO 320
      READ(9,260,END=35) JD,USED(I),ADJU(I),BANK(I),RFWB(I)
  260 FORMAT(I8,15X,4F10.0)
      IF(JD.EQ.ID(I)) GO TO 320
   35 WRITE(6,265) JD,ID(I)
  265 FORMAT(15H DIVERSION NO. ,I10,50H FROM SNKSTO'YR'.USE DOES NOT EQU
     1AL DIVERSION NO. ,I10,25H FROM SNKSTO'YR'.SPA LIST)
      GO TO 400
  320 STOR(I,14)=BANK(I)
      DO 175 K=1,12
      KS=K
      IF(K.GE.3) KS=K-1
  175 STOR(I,K)=STOR(I,K)*PAVAIL(KS)
  165 CONTINUE
      IT=2
      GO TO 185
  200 PRINT *,' '
      PRINT 201
  201 FORMAT(' DO YOU WISH TO CREATE A FILE (SNKCAN.IND) OF STORED WATER
     1 SUPPLY'/' BY DIVERSION FOR INPUT TO ACCOUNTING PROGRAM? Y/N: '$)
      ACCEPT 36, FILE
      IF(FILE.NE.'Y') GO TO 204
```

```
      OPEN(UNIT=2,NAME='SNKCAN.IND',TYPE='NEW',CARRIAGECONTROL='LIST')
      DUM=0.0
      DO 202 I=1,JEN
  202 WRITE(2,203) ID(I),DUM,STOR(I,15)
  203 FORMAT(I8,2F10.1)
  204 CONTINUE
C
C***** SET TABLE NUMBER TO BEGIN OUTPUT OF IRRIGATION SEASON
C***** STORED WATER TRANSACTIONS.
C
      IF(USE.NE.'Y') GO TO 400
      FTAB=21.0
   33 PRINT 28, FTAB
   28 FORMAT(' THE FIRST STORAGE TABLE FOR REPORT WILL BE NO. ',
     1 F5.1/' IS THIS OK? Y/N/Q '$)
      ACCEPT 36, TBL
      IF(TBL.EQ.'Q') GO TO 400
      IF(TBL.EQ.'Y') GO TO 310
      TYPE *, ' '
      PRINT 29
   29 FORMAT(' ENTER NEW BEGINNING TABLE NUMBER = '$)
      ACCEPT 32, FTAB
   32 FORMAT(F5.0)
      TYPE *, ' '
      GO TO 33
  310 IF(FTAB.LE.0.0) GO TO 400
      K=1
      DO 280 I=1,JE
C
C***** MOVE MILNER STORAGE TO MISCELLANEOUS TABLE SINCE STORAGE
C***** IS NOT ALLOCATED TO ANY USER
C
      IF(ID(I).EQ.99999950) THEN
                            STOR(I,13)=STOR(I,13)+YIELD(12)
                            JM=I
                            END IF
      ISKIP(I)=0.0
      IF(ISK(K).NE.ID(I)) GO TO 280
      ISKIP(I)=1
      K=K+1
  280 CONTINUE
      DO 315 I=1,JE
      EXCS(I)=0.0
      TCRYO(I)=0.0
      WBRV(I)=0.0
      BALN(I)=STOR(I,14)-USED(I)
      IF(BALN(I).GT.0.0) GO TO 275
  295 BALN(I)=BALN(I)+STOR(I,13)
      GO TO 290
  275 WBRV(I)=BALN(I)
      BALN(I)=STOR(I,13)
  290 BALN(I)=BALN(I)+RFWB(I)
C
C***** IF THERE IS A NEGATIVE ADJUSTMENT TO STORAGE USE (ADJU), THAT
C***** AMOUNT IS FIRST DEDUCTED FROM THE WATER REVERTING TO THE WATER
C***** BANK (WBRV), FOR ALL MISCELLANEOUS STORAGE ACCOUNTS.
C
```

```
      IF(ADJU(I).GE.0.0.OR.WBRV(I).LE.0.0.OR.ID(I).LT.90000000) GO TO
     1 285
      ADD=WBRV(I)
      WBRV(I)=WBRV(I)+ADJU(I)
      IF(WBRV(I).LT.0.0) WBRV(I)=0.0
      ADD=ADD-WBRV(I)
      BALN(I)=BALN(I)+ADD
  285 TCRYO(I)=BALN(I)+ADJU(I)
      IF(TCRYO(I).GE.0.0) GO TO 315
      EXCS(I)=TCRYO(I)*(-1.0)
      TCRYO(I)=0.0
  315 CONTINUE
      DO 455 I=1,8
      STOT(I,1)=CTOT(I,13)
      STOT(I,2)=CTOT(I,14)
      DO 455 K=3,9
  455 STOT(I,K)=0.0
      STOT(8,1)=STOT(8,1)+YIELD(12)
      ICONT=0
      CALL TABL2 (0,1,1,1)
      CALL TABL2 (1,2,0,1)
      CALL TABL2 (6,7,1,2)
      CALL TABL2 (8,9,0,2)
      CALL TABL2 (9,10,0,3)
      CALL TABL2 (2,3,1,4)
      CALL TABL2 (4,5,0,4)
      CALL TABL2 (3,4,0,5)
      CALL TABL2 (5,6,0,6)
      CALL TABL2 (7,8,0,7)
      CALL TABL2 (10,11,0,8)
      WRITE(6,540) RUNDATE,FTAB,IYR
  540 FORMAT(1H1/////123X,A9/////18X,5HTABLE,F4.0,2X,20HSUMMARY BY REACH
     1 OF ,I4,54H STORED WATER ACCOUNTS IN WATER DISTRICT 1 (ACRE-FEET))
      DO 595 K=1,9
  595 GSTOT(K)=0.0
      WRITE(6,510)
  510 FORMAT( /28X,76H            STORAGE OR                          RETUR
     1N TO                      /28X,67H           RENTAL POOL
     2 REVERTED TO SPACEHOLDER           /28X,99H   STORAGE   PURCHASE,
     3  STORAGE RENTAL POOL    FROM                 ADJUST-     EXCESS
     4    CARRY-/  4X,13HREACH         ,11X,99H  ALLOCATED SUPPLY (-)
     5 USED   FROM USER  RENTAL POOL   BALANCE       MENT       USED
     6   OVER/)
      DO 590 I=1,8
      DO 505 K=1,9
  505 GSTOT(K)=GSTOT(K)+STOT(I,K)
  590 WRITE(6,520) (RCHNM(K,I),K=1,5),(STOT(I,J),J=1,9)
  520 FORMAT( 4X,5A4,4X,9(2X,F9.1))
      WRITE(6,530) (GSTOT(K),K=1,9)
  530 FORMAT(/ 3X,6H TOTAL,19X,9(2X,F9.1))
C
C***** COMPUTE NET STORAGE USED (RDIFF).  MOVE LATE POWER STORAGE AT
C***** AMERICAN FALLS AND PALISADES BACK TO ORIGINAL RIGHT SINCE IT
C***** NOW LOSES ITS IDENTITY(NO CARRYOVER IN THIS SPACE.  IF THERE IS
C***** NO STORAGE IN PALISADES REGULAR ACCOUNT (STOR(I,2)), THEN ASSUME
C***** THIS POWER STORAGE CAME FROM PALISADES USERS (STOR(I,3)), SO ADD
C***** IT BACK THERE INSTEAD.  ALSO COMPUTE PROPORTIONAL SPACE BY USER
```

```
C***** IN FIRST 200,000 AF OF JACKSON LAKE FOR FIRST USE COMPUTATIONS
C***** BELOW.
C
      DO 608 I=1,JE
      DIFF(I)=STOR(I,13)-TCRYO(I)
      RDIFF(I)=DIFF(I)
      IF(STOR(I,2).LE.0.0) THEN
            STOR(I,3)=STOR(I,3)+STOR(I,12)
            GO TO 607
            ENDIF
      STOR(I,2)=STOR(I,2)+STOR(I,12)
  607 STOR(I,9)=STOR(I,9)+STOR(I,11)
      STOR(I,11)=0.0
      STOR(I,12)=0.0
  608 JDFL(I)=200000.*(SPAC(I,1)/TSPAC(1))
C
C***** COMPUTE CARRYOVER BY USER BY RESERVOIR.  WHEN IO=11, PROPORTION
C***** FIRST 200,000 ACRE FEET IN JACKSON LAKE AMONG SPACEHOLDERS
C***** ACCORDING TO SPACE OWNED.
C
      DO 615 LO=1,11
      IO=IOR(LO)
      IF(IO.NE.11) GO TO 613
      IO=1
      IF(GTOT(1).LE.200000.) GO TO 613
      TJL=0.0
      PTJL=0.0
  606 DO 612 I=1,JE
      DF=RDIFF(I)
      IF(DF.GT.JDFL(I)) DF=JDFL(I)
      IF(DF.LE.0.0) GO TO 612
      STOR(I,IO)=STOR(I,IO)-DF
      IF(STOR(I,IO).GE.0.0) RDIFF(I)=RDIFF(I)-DF
      IF(STOR(I,IO).LT.0.0) RDIFF(I)=RDIFF(I)-DF-STOR(I,IO)
      IF(STOR(I,IO).LT.0.0) STOR(I,IO)=0.0
      TJL=TJL+(DF-RDIFF(I))
  612 CONTINUE
      DTJL=TJL-PTJL
      IF(DTJL.LE.02) GO TO 615
      REMJ=200000.-TJL
      IF(REMJ.LE.0.2) GO TO 615
      DO 616 I=1,JE
  616 JDFL(I)=REMJ*(SPAC(I,1)/TSPAC(1))
      PTJL=TJL
      GO TO 606
  613 CONTINUE
      DO 614 I=1,JE
      DF=RDIFF(I)
      IF(DF.LE.0.0) GO TO 614
      STOR(I,IO)=STOR(I,IO)-DF
      IF(STOR(I,IO).GE.0.0) RDIFF(I)=RDIFF(I)-DF
      IF(STOR(I,IO).LT.0.0) RDIFF(I)=STOR(I,IO)*(-1.0)
      IF(STOR(I,IO).LT.0.0) STOR(I,IO)=0.0
  614 CONTINUE
  615 CONTINUE
C
C***** PLACE FREMONT MADISON CARRYOVER ENTIRELY IN ISLAND PARK
```

```
C***** SINCE THERE ARE NO INDIVIDUAL CARRYOVER PRIVILEGES
C
      K=NDEND(10)+1
      DO 618 I=K,JE
  618 IF(ID(I).EQ.99999600) STOR(I,6)=TCRYO(I)
      DO 655 I=1,10
      DO 655 K=1,12
  655 CTOT(I,K)=0.0
      CALL TABL3 (0,1,1,1)
      CALL TABL3 (1,2,0,1)
      CALL TABL3 (6,7,1,2)
      CALL TABL3 (8,9,0,2)
      CALL TABL3 (9,10,0,3)
      CALL TABL3 (2,3,1,4)
      CALL TABL3 (4,5,0,4)
      CALL TABL3 (3,4,0,5)
      CALL TABL3 (5,6,0,6)
      CALL TABL3 (7,8,0,7)
      CALL TABL3 (10,11,0,8)
      WRITE(6,652)
  652 FORMAT(1H1)
      WRITE(6,650) RUNDATE,IYR
  650 FORMAT(/123X,A9///35X,I4, 59H WATER DISTRICT 1 RESERVOIR CARRYOVER
     1 BY REACH  (ACRE-FEET)/)
      DO 695 K=1,12
  695 GTOT(K)=0.0
      WRITE(6,610)
  610 FORMAT(1H ,  14HRIVER REACH   ,6X, 99H  CARRYOVER JACKSON PALISADE
     1S  PAL USR   PAL WWS HNRYS L  ISL PRK GRSSY L   RIRIE  AM FALLS WA
     2LCOTT/)
      DO 690 I=1,8
      DO 605 K=1,10
  605 GTOT(K)=GTOT(K)+CTOT(I,K)
  690 WRITE(6,620) (RCHNM(K,I),K=1,5),STOT(I,9),(CTOT(I,J),J=1,10)
  620 FORMAT(1X,5A4,F10.1,F9.1,F10.1,F9.1,F10.1,F8.1,F9.1,2F8.1,
     1 F10.1,F8.1)
      WRITE(6,630) GSTOT(9),(GTOT(K),K=1,10)
  630 FORMAT(/6H TOTAL,15X,F10.1,F9.1,F10.1,F9.1,F10.1,F8.1,F9.1,
     12F8.1,F10.1,F8.1)
      GRTOT=0.0
      DO 640 I=1,10
  640 GRTOT=GRTOT+GTOT(I)
      SGRTOT=GRTOT+TCRYO(JE)+TCRYO(JM)
      WRITE(6,642)IYR
  642 FORMAT(///////     35X,I4, 64H WATER DISTRICT 1 RESERVOIR CARRYOVE
     1R - SYSTEM TOTAL (ACRE-FEET)/)
      WRITE(6,644) GRTOT,TCRYO(JE),TCRYO(JM),SGRTOT
  644 FORMAT(55X,'ALLOCATED        - ',F10.1/
     1       55X,'UNALLOCATED      - ',F10.1/
     2       55X,'MILNER RESERVOIR - ',F10.1//
     3       55X,'TOTAL SYSTEM     - ',F10.1)
      PRINT 16
   16 FORMAT(' DO YOU WISH TO ENTER LATE SEASON RESERVOIR FILL? Y/N '$)
      ACCEPT 7,ANSWER
    7 FORMAT(1A1)
      PRINT *,' '
      IF(ANSWER.NE.'Y') GO TO 8
```

```
685     15 DO 9 K=1,9
686        PRINT 11, RNAME(K),FILLATE(K)
687     11 FORMAT(1X,'LATE SEASON FILL FOR       ',A14,' = ',F10.1,' ENTER NE
688       1W FILL? Y/N/Q '$)
689        ACCEPT 7,ANSWER
690        IF(ANSWER.EQ.'Q') GO TO 18
691        IF(ANSWER.NE.'Y') GO TO 9
692        PRINT 19,RNAME(K)
693     19 FORMAT(1X,'ENTER LATE SEASON FILL FOR ',A14,' = '$)
694        ACCEPT 21,FILLATE(K)
695     21 FORMAT(F10.0)
696      9 CONTINUE
697     18 PRINT 17
698     17 FORMAT(' DO YOU WISH TO REENTER LATE SEASON RESERVOIR FILL? Y/N '
699       1$)
700        ACCEPT 7,ANSWER
701        IF(ANSWER.EQ.'Y') GO TO 15
702      8 WRITE(6,142)IYR
703    142 FORMAT(//////     35X,I4,' WATER DISTRICT 1 RESERVOIR TOTAL STORAG
704       1E - OCTOBER 31 (ACRE-FEET)'///43X,'RESERVOIR',12X,'CARRYOVER',
705       2'   LATE FILL',7X,'TOTAL'/)
706        GTOT(2)=GTOT(2)+GTOT(3)
707        GTOT(3)=TCRYO(JE)+TCRYO(JM)
708        TLATE=0.0
709        GTFILL=0.0
710        DO 146 K=1,10
711        K1=K
712        IF(K.GT.2) K1=K+1
713        IF(K1.EQ.11) K1=3
714        TFILL(K)=GTOT(K1)+FILLATE(K)
715        TLATE=TLATE+FILLATE(K)
716        GTFILL=GTFILL+TFILL(K)
717        IF(K.EQ.10) GO TO 148
718        WRITE(6,144) RNAME(K),GTOT(K1),FILLATE(K),TFILL(K)
719    144 FORMAT(43X,A14,4X,3F12.1)
720    146 CONTINUE
721    148 WRITE(6,147) GTOT(K1),FILLATE(K),TFILL(K)
722    147 FORMAT(43X,'OTHER',13X,3F12.1)
723        WRITE(6,145) SGRTOT,TLATE,GTFILL
724    145 FORMAT(/43X,'TOTAL',13X,3F12.1)
725     38 PRINT 31
726     31 FORMAT(' DO YOU WANT TO CREATE A FILE OF THE CARRYOVER'/
727       1' BY CANAL AND RESERVOIR? Y/N: '$)
728        ACCEPT 36, FILE
729        IF(FILE.EQ.'N') GO TO 400
730        IF(FILE.NE.'Y') GO TO 38
731        OPEN (UNIT=10,NAME='CARRYOUT',TYPE='NEW',CARRIAGECONTROL='LIST')
732        DO 37 I=1,JE
733     37 WRITE(10,34) ID(I),UNAME(I),(STOR(I,K),K=1,12)
734     34 FORMAT(I8,1X,A14,12F9.1)
735    400 CONTINUE
736        STOP
737        END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-00000000	SNKSTO$MAIN

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
1-00000040	R*4	ADD	1-000000BC	I*4	IPALU	1-0000005C	I*4	KS	1-00000070	R*4	TJNSTFCO
1-00000020	R*4	ANSWER	6-0001E950	I*4	IT	1-000000C8	I*4	L	1-0000007C	R*4	TJNSTFSP
1-0000002C	R*4	DF	6-0001E954	I*4	IYR	1-0000003C	I*4	LO	1-0000006C	R*4	TJSSTFCO
1-00000028	R*4	DTJL	1-000000A8	I*4	J	1-00000030	R*4	PTJL	1-00000078	R*4	TJSSTFSP
1-0000004C	R*4	DUM	1-00000050	I*4	JD	1-00000054	R*4	PUNSTO	1-00000068	R*4	TJTFCO
1-00000098	R*4	FIL	1-000000AC	I*4	JE	1-00000094	R*4	PUTSTO	1-00000074	R*4	TJTFSP
1-000000CC	R*4	FILE	1-000000B0	I*4	JEN	1-00000024	R*4	REMJ	1-00000000	R*8	TLATE
1-00000090	R*4	FLD	1-00000064	I*4	JKTFNSTO	1-000000D8	CHAR	RNAM	1-00000060	R*4	TNSTOR
6-0001E958	R*4	FTAB	1-00000044	I*4	JM	6-0001E960	CHAR	RUNDATE	1-0000009C	R*4	TSP
1-00000010	R*8	GRTOT	1-000000B8	I*4	JNSTF	1-00000008	R*8	SGRTOT	1-000000E8	CHAR	U
1-0000001C	R*4	GTFILL	1-000000B4	I*4	JSSTF	1-00000058	R*4	SPC	1-000000D0	R*4	USE
1-000000C4	I*4	I	1-000000C0	I*4	K	1-00000080	R*4	STP	1-00000084	R*4	WAT
6-0001E95C	I*4	ICONT	1-00000018	I*4	K1	1-00000048	R*4	TBL			
1-000000A4	I*4	IJKSN	1-0000008C	I*4	KC	1-00000088	R*4	TCR			
1-00000038	I*4	IO	1-000000A0	I*4	KRS	1-00000034	R*4	TJL			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
1-00018B60	R*8	BANK	3600	(450)
1-00021228	R*8	CNSTO	96	(12)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
1-000211C8	R*8	CTOTC	96	(12)
6-0001C278	R*4	DIFF	1800	(450)

1-00000148	R*4	EVAP	48	(12)
6-0000EF88	R*4	EXCS	1800	(450)
1-00021548	R*8	FILL	96	(12)
1-00021168	R*8	FILLATE	96	(12)
1-00021350	R*8	GSTOT	72	(9)
1-000212E8	R*8	GTOT	104	(13)
6-0001C9AC	I*4	ID	1800	(450)
1-00020300	I*4	ID2	1800	(450)
1-0001FBF0	I*4	IK	1800	(450)
2-00002050	I*4	IOR	44	(11)
2-000020A8	I*4	ISK	120	(30)
6-0001BB70	I*4	ISKIP	1800	(450)
1-0000D480	R*8	JDFL	3600	(450)
1-000215A8	I*4	JID	200	(50)
2-0000207C	I*4	KREND	44	(11)
6-0001C980	I*4	NDEND	44	(11)
1-00000180	R*8	NSTO	54000	(450, 15)
1-00000118	R*4	PAVAIL	48	(12)
1-00021440	R*8	PYLD	72	(9)
7-00000000	R*4	RCHNM	160	(5, 8)
1-00020A10	R*4	RDIFF	1800	(450)
6-0000FD98	R*4	RFWB	1800	(450)
1-00021398	CHAR	RNAME	168	(12)
1-0000E2A0	R*8	SPAC	43200	(450, 12)
1-000214E8	R*8	SPACE	96	(12)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
1-000000F8	R*8	T	32	(4)
6-0000DA70	R*8	TCRYO	3600	(450)
1-00021120	R*8	TFILL	72	(9)
1-00019980	R*4	TOTR	25200	(450, 14)
1-00021288	R*8	TSPAC	96	(12)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)
1-00021488	R*8	YIELD	96	(12)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
4-00002FFC	1	4-00001F78	24	4-000028A0	94	4-000011D8	185	4-00003ADC	290	4-000044F4	607
**	4	4-00001EE8	26	4-00002980	95	**	190	4-00003A88	295	**	608
**	5	4-00001ECC	27	4-00002F60	96	4-000014C8	195	**	310	4-00004B4C	612
4-000023A0	6	**	30	**	97	**	200	4-00003D78	315	4-00004658	613
**	8	**	33	4-00002CBC	98	**	202	4-00003290	320	4-00004860	614
4-000057BC	9	**	35	4-00000E28	105	4-00003604	204	4-000060A4	400	4-00004C80	615
**	13	**	37	4-00001114	110	4-00001560	205	4-00003E2C	455	4-00004BF4	616
**	14	**	38	**	146	**	212	4-00004144	505	4-00004CF4	618
4-000055E0	15	4-00000544	40	**	148	4-00001AC8	232	**	590	4-00005358	640
**	18	**	45	4-00001200	155	**	275	4-000040A8	595	4-00004D9C	655
4-00000560	20	**	60	**	165	4-0000391C	280	4-0000509C	605	**	690
4-00001EBC	23	4-00000D3C	90	4-00003308	175	4-00003CB0	285	4-000048A0	606	4-00005000	695

```
      SUBROUTINE TABLE (KB,KE,IRET,JT)
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      WRITE(6,160) IT,RUNDATE
  160 FORMAT(1H1/58X,4HPART,I2,59X,A9)
      IF(IT.EQ.1) WRITE(6,140) IYR
  140 FORMAT(//18X,I4,98H WATER DISTRICT 1 RESERVOIR SPACE BY USER ASSUM
     1ING FULL RESERVOIRS, WITHOUT RENTAL POOL(ACRE-FEET)/)
      IF(IT.EQ.2) WRITE(6,150) IYR
  150 FORMAT(//15X,I4,107H WATER DISTRICT 1 RESERVOIR STORAGE BY USER AF
     1TER PERCENT FILL AND EVAPORATION, WITH RENTAL POOL(ACRE-FEET)/)
      WRITE(6,145) (RCHNM(I,JT),I=1,5)
  145 FORMAT(54X,5A4/)
      WRITE(6,95)
   95 FORMAT(1H ,13HNUMBER   USER,11X,107H       JACKSN PALSDS PL USR PL
     1 WWS HNRY L ISL PK GRSY L  RIRIE AM FLS L WLCT AM LTF PL LTF RENTA
     2L     TOTAL/)
      IB=1
      IF(KB.EQ.0) GO TO 100
      IB=NDEND(KB)+1
  100 IE=NDEND(KE)
      DO 80 I=IB,IE
      STOR(I,13)=0.0
      DO 60 K=1,12
      CTOT(JT,K)=CTOT(JT,K)+STOR(I,K)
      STOR(I,13)=STOR(I,13)+STOR(I,K)
   60 CONTINUE
      STOR(I,15)=STOR(I,13)+STOR(I,14)
      CTOT(JT,13)=CTOT(JT,13)+STOR(I,13)
      CTOT(JT,14)=CTOT(JT,14)+STOR(I,14)
      CTOT(JT,15)=CTOT(JT,15)+STOR(I,15)
      WRITE(6,70) ID(I),UNAME(I),(JIDNNT(STOR(I,K)),K=1,12),
     1                           JIDNNT(STOR(I,14)),STOR(I,15)
   70 FORMAT(1X,I8,1X,A14,7X,13I7,F10.1)
   80 CONTINUE
      IF(IRET.GT.0) GO TO 10
      WRITE(6,20) (JIDNNT(CTOT(JT,K)),K=1,12),JIDNNT(CTOT(JT,14)),
     1                                        CTOT(JT,15)
   20 FORMAT(/10X,5HTOTAL,9X,7X,13I7,F10.1)
   10 CONTINUE
      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-000060C4	TABLE

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
6-0001E958	R*4	FTAB	1-00021674	I*4	IE	**	I*4	JT	6-0001E960	CHAR	RUNDATE
1-0002167C	I*4	I	**	I*4	IRET	1-00021670	I*4	K			
1-00021678	I*4	IB	6-0001E950	I*4	IT	**	I*4	KB			
6-0001E95C	I*4	ICONT	6-0001E954	I*4	IYR	**	I*4	KE			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
6-0001C278	R*4	DIFF	1800	(450)
6-0000EF88	R*4	EXCS	1800	(450)
6-0001C9AC	I*4	ID	1800	(450)
6-0001BB70	I*4	ISKIP	1800	(450)
6-0001C980	I*4	NDEND	44	(11)
7-00000000	R*4	RCHNM	160	(5, 8)
6-0000FD98	R*4	RFWB	1800	(450)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
6-0000DA70	R*8	TCRYO	3600	(450)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label
4-000068D8	10	**	60	**	80	4-00006330	100

```
      SUBROUTINE TABL2 (KB,KE,IRET,JT)
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      IF(ICONT.EQ.1) GO TO 120
      WRITE(6,140) RUNDATE,FTAB,IYR,(RCHNM(I,JT),I=1,5)
  140 FORMAT(1H1/////123X,A9//26X,5HTABLE,F4.0,4X,I4,25H STORED WATER AC
     1COUNTS - , 5A4,12H (ACRE-FEET))
      GO TO 130
  120 WRITE(6,150) FTAB
  150 FORMAT(1H1////////49X,5HTABLE,F4.0,4X,9HCONTINUED)
  130 WRITE(6,95)
   95 FORMAT( /28X,76H            STORAGE OR                          RETU
     1RN TO                     /28X,67H          RENTAL POOL
     2 REVERTED TO SPACEHOLDER          /28X,99H   STORAGE   PURCHASE,
     3  STORAGE RENTAL POOL    FROM                ADJUST-     EXCESS
     4    CARRY-/  4X,13HNUMBER   NAME,11X,99H  ALLOCATED SUPPLY (-)
     5 USED   FROM USER  RENTAL POOL   BALANCE       MENT       USED
     6   OVER/)
      IB=1
      IF(KB.EQ.0) GO TO 100
      IB=NDEND(KB)+1
  100 IE=NDEND(KE)
      DO 80 I=IB,IE
      STOT(JT,3)=STOT(JT,3)+USED(I)
      STOT(JT,4)=STOT(JT,4)+WBRV(I)
      STOT(JT,5)=STOT(JT,5)+RFWB(I)
      STOT(JT,6)=STOT(JT,6)+BALN(I)
      STOT(JT,7)=STOT(JT,7)+ADJU(I)
      STOT(JT,8)=STOT(JT,8)+EXCS(I)
      STOT(JT,9)=STOT(JT,9)+TCRYO(I)
      IF(ISKIP(I).GT.0) GO TO 80
      WRITE(6,70) ID(I),UNAME(I),STOR(I,13),STOR(I,14),USED(I),
     1 WBRV(I),RFWB(I),BALN(I),ADJU(I),EXCS(I),TCRYO(I)
   70 FORMAT( 4X,I8,1X,A14,9(2X,F9.1))
   80 CONTINUE
      IF(IRET.GT.0) GO TO 10
      FTAB=FTAB+1.0
      WRITE(6,20) (STOT(JT,K),K=1,9)
   20 FORMAT(/13X,5HTOTAL,9X,9(2X,F9.1))
      ICONT=0
      RETURN
   10 ICONT=1
      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-00006914	TABL2

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
6-0001E958	R*4	FTAB	1-00021684	I*4	IE	**	I*4	JT	6-0001E960	CHAR	RUNDATE
1-0002168C	I*4	I	**	I*4	IRET	1-00021680	I*4	K			
1-00021688	I*4	IB	6-0001E950	I*4	IT	**	I*4	KB			
6-0001E95C	I*4	ICONT	6-0001E954	I*4	IYR	**	I*4	KE			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
6-0001C278	R*4	DIFF	1800	(450)
6-0000EF88	R*4	EXCS	1800	(450)
6-0001C9AC	I*4	ID	1800	(450)
6-0001BB70	I*4	ISKIP	1800	(450)
6-0001C980	I*4	NDEND	44	(11)
7-00000000	R*4	RCHNM	160	(5, 8)
6-0000FD98	R*4	RFWB	1800	(450)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
6-0000DA70	R*8	TCRYO	3600	(450)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
**	10	4-00007034	80	4-00006B48	100	**	120	**	130

```
      SUBROUTINE TABL3 (KB,KE,IRET,JT)
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      WRITE(6,160)
  160 FORMAT(1H1)
      WRITE(6,150) RUNDATE,IYR
  150 FORMAT(/123X,A9/30X,I4, 66H WATER DISTRICT 1 RESERVOIR CARRYOVER S
     1TORAGE BY USER  (ACRE-FEET)/)
      WRITE(6,145) (RCHNM(I,JT),I=1,5)
  145 FORMAT(54X,5A4/)
      WRITE(6,95)
   95 FORMAT(1H ,13HNUMBER   USER,11X, 93HCARRYOVER JACKSON PALISADES PA
     1L USR  PAL WWS HNRYS L ISL PRK GRSSY L   RIRIE AM FALLS WALCOTT/)
      IB=1
      IF(KB.EQ.0) GO TO 100
      IB=NDEND(KB)+1
  100 IE=NDEND(KE)
      DO 80 I=IB,IE
      DO 60 K=1,10
      CTOT(JT,K)=CTOT(JT,K)+STOR(I,K)
   60 CONTINUE
      WRITE(6,70) ID(I),UNAME(I),TCRYO(I),(STOR(I,K),K=1,10)
   70 FORMAT( 1X,I8,1X,A14,F10.1,2F9.1,F8.1,F9.1,4F8.1,F9.1,F8.1)
   80 CONTINUE
      IF(IRET.GT.0) GO TO 10
      WRITE(6,20) STOT(JT,9),(CTOT(JT,K),K=1,10)
   20 FORMAT(/10X,5HTOTAL,9X,F10.1,2F9.1,F8.1,F9.1,4F8.1,F9.1,F8.1)
   10 CONTINUE
      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-00007164	TABL3

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
6-0001E958	R*4	FTAB	1-00021694	I*4	IE	**	I*4	JT	6-0001E960	CHAR	RUNDATE
1-0002169C	I*4	I	**	I*4	IRET	1-00021690	I*4	K			
1-00021698	I*4	IB	6-0001E950	I*4	IT	**	I*4	KB			
6-0001E95C	I*4	ICONT	6-0001E954	I*4	IYR	**	I*4	KE			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
6-0001C278	R*4	DIFF	1800	(450)
6-0000EF88	R*4	EXCS	1800	(450)
6-0001C9AC	I*4	ID	1800	(450)
6-0001BB70	I*4	ISKIP	1800	(450)
6-0001C980	I*4	NDEND	44	(11)
7-00000000	R*4	RCHNM	160	(5, 8)
6-0000FD98	R*4	RFWB	1800	(450)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
6-0000DA70	R*8	TCRYO	3600	(450)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label
4-000076B4	10	**	60	**	80	4-00007360	100

```
875        BLOCK DATA
876        COMMON /RN/RCHNM(5,8)
877        DATA RCHNM/
878       1 'IRWI','N TO',' LOR','ENZO','    ',
879       2 'LORE','NZO ','TO B','LACK','FOOT',
880       3 'BLAC','KFOO','T TO',' MIL','NER ',
881       4 'MAIN',' STE','M HE','NRYS',' FRK',
882       5 'FALL','S RI','VER ','    ','    ',
883       6 'TETO','N RI','VER ','    ','    ',
884       7 'WILL','OW C','REEK','    ','    ',
885       8 'MISC','ELLA','NEOU','S   ','    '/
886        END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ARRAYS

Address	Type	Name	Bytes	Dimensions
7-00000000	R*4	RCHNM	160	(5, 8)

```
COMMAND QUALIFIERS

  /ALIGNMENT=(COMMONS=(NONATURAL,PACKED,NOSTANDARD,NOMULTILANGUAGE),RECORDS=NATURAL,NOSEQUENCE)
  /ANNOTATIONS=(NOCODE,NODETAIL,NOFEEDBACK,NOINLINING,NOLOOP_TRANSFORMS,NOLOOP_UNROLLING,NOPREFETCHING,NOSHRINKWRAPPING,
          NOSOFTWARE_PIPELINING,NOTAIL_CALLS,NOTAIL_RECURSION)
  /ARCHITECTURE=GENERIC
  /ASSUME=(ACCURACY_SENSITIVE,ALTPARAM,NOBUFFERED_IO,NOBYTERECL,NODUMMY_ALIASES,NOF77RTL,NOFP_CONSTANT,NOINT_CONSTANT,
          NOMINUS0,PROTECT_CONSTANTS,NOSOURCE_INCLUDE,NOUNDERSCORE)
  /NOAUTOMATIC
  /NOBY_REF_CALL
  /CCDEFAULT=DEFAULT
  /CHECK=(NOARG_TEMP_CREATED,NOBOUNDS,FORMAT,NOFP_EXCEPTIONS,OUTPUT_CONVERSION,NOOVERFLOW,POWER,NOUNDERFLOW)
  /CONVERT=NATIVE
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /NODEFINE
  /DOUBLE_SIZE=64
  /NOD_LINES
  /ERROR_LIMIT=30
  /NOEXTEND_SOURCE
  /F77
  /NOFAST
  /FLOAT=G_FLOAT
  /GRANULARITY=QUADWORD
  /IEEE_MODE=FAST
  /INTEGER_SIZE=32
  /NOMACHINE_CODE
  /MATH_LIBRARY=ACCURATE
  /NOMODULE
  /NAMES=UPPERCASE
  /OPTIMIZE=(INLINE=NONE,LEVEL=,NOLOOPS,NOPIPELINE,TUNE=GENERIC,UNROLL=0)
  /NOPAD_SOURCE
  /REAL_SIZE=32
  /NORECURSIVE
  /REENTRANCY=NONE
  /ROUNDING_MODE=NEAREST
  /NOSEPARATE_COMPILATION
  /SEVERITY=(WARNING=WARNING)
  /SHOW=(NOINCLUDE,MAP)
  /SOURCE_FORM=FIXED
  /STANDARD=NONE
  /NOSYNCHRONOUS_EXCEPTIONS
  /NOSYNTAX_ONLY
  /NOTIE
  /VMS
  /WARNINGS=(ALIGNMENT,NOARGUMENT_CHECKING,NODECLARATIONS,GENERAL,GRANULARITY,NOIGNORE_LOC,NOTRUNCATED_SOURCE,UNCALLED,
          UNINITIALIZED,NOUNUSED,USAGE)

  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /INCLUDE=FORT$INCLUDE:
  /LIST=HYDRO:[PPACE.SNAKE]SNKSTO.LIS;22
  /OBJECT=HYDRO:[PPACE.SNAKE]SNKSTO.OBJ;3
  /NOLIBRARY

COMPILER: Compaq Fortran V7.5-1961-48BCD
```

COMPILATION STATISTICS

CPU time:	0.63	seconds
Elapsed time:	2.14	seconds
Pagefaults:	3409	
I/O Count:	69	

```
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      DIMENSION KREND(11),FILL(12),SPACE(12),YIELD(12),EVAP(12)
      DIMENSION PYLD(9),T(4),RNAME(12),BANK(450),GSTOT(9),JID(50)
      DIMENSION ISK(30),IOR(11),SPAC(450,12),RDIFF(450),GTOT(13)
      DIMENSION JDFL(450),TSPAC(12),ID2(450),PAVAIL(12)
      DIMENSION IK(450),NSTO(450,15),TOTR(450,14),CNSTO(12),CTOTC(12)
      DIMENSION FILLATE(12),TFILL(9)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      CHARACTER*14 RNAME
      CHARACTER*14 U,RNAM
      DATA KREND/13038150,13038500,13046450,13049500,13050590,13055351,
     1 13057260,13058550,13069500,13088000,99999999/
      DATA ISK/13010500,13032450,13039000,13042000,13042600,13046500,
     1 13057950,13059050,13076500,13076751,13081000,13081400,18*0/
C
C***** THE FOLLOWING DATA STATEMENT DICTATES THE ORDER FROM WHICH THE
C***** STORAGE IS REMOVED FROM RESERVOIRS.
C*****
C*****    1 - JACKSON LAKE      2 - PALISADES       3 - PAL WATER USERS
C*****    4 - PAL WINTER WATER  5 - HENRYS LAKE     6 - ISLAND PARK
C*****    7 - GRASSY LAKE       8 - RIRIE           9 - AMERICAN FALLS
C*****   10 - LAKE WALCOT
C*****   11 - FIRST 200,000 ACRE FEET IN JACKSON LAKE
C
      DATA IOR/6,7,11,10,9,3,4,2,1,5,8/
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      REAL*8 FILL,SPACE,YIELD,GTOT,PYLD,T,GSTOT,TSPAC,GRTOT,SGRTOT
      REAL*8 CTOTC,CNSTO,NSTO,BANK,SPAC,FILLATE,TFILL,TLATE,JDFL
      CALL DATE_AND_TIME(RUNDATE)
        OPEN(UNIT=1,NAME='INDIN',TYPE='OLD')
        OPEN(UNIT=5,NAME='CARRYIN',TYPE='OLD')
        OPEN(UNIT=6,NAME='REPORTOUT',TYPE='NEW')
        OPEN(UNIT=7,NAME='SPACEIN',TYPE='OLD')
        OPEN(UNIT=9,NAME='USEIN',TYPE='OLD')
C       THE OLD CALL DATE AND CALL ASSIGN STATEMENTS ARE THE FOLLOWING.
C      CALL DATE(RUNDATE)
C      CALL ASSIGN(1,'INDIN')
C      CALL ASSIGN(5,'CARRYIN')
C      CALL ASSIGN(6,'REPORTOUT')
C      CALL ASSIGN(7,'SPACEIN')
C      CALL ASSIGN(9,'USEIN')
C       WEIMIN LI 8/8/2002
      L=1
      I=1
   30 READ(7,10,END=20) ID(I),IK(I),UNAME(I),(STOR(I,K),K=1,12)
   10 FORMAT(I8,A1,A14,12F9.0)
      STOR(I,13)=0.0
      STOR(I,14)=0.0
      STOR(I,15)=0.0
      IF(ID(I).EQ.99999300) IPALU=I
      IF(ID(I).EQ.13087000) JNSTF=I
```

PROGRAM COMPILED ON ALPHA.

STATEMENTS 306-350 CHANGED TO MODIFY LAST-TO-FILL SPACE FILL SO THAT FILL IS PROPORTIONAL TO SPACE IN THAT ACCOUNT RATHER THAN IN TOTAL RESERVOIR

PJ 2-3-03

```
      IF(ID(I).EQ.13087500) JSSTF=I
      IF(ID(I).LT.KREND(L)) GO TO 40
      NDEND(L)=I-1
      L=L+1
   40 I=I+1
      GO TO 30
   20 CONTINUE
      NDEND(L)=I-1
      JEN=NDEND(10)
      JE=NDEND(11)
      DO 60 I=1,JE
      DO 60 K=1,12
      CRYO(I,K)=0.0
   60 SPAC(I,K)=STOR(I,K)
      DO 45 I=1,JE
      READ(5,12,END=5) ID2(I),U,(CRYO(I,J),J=1,12)
   12 FORMAT(I8,1X,A14,12F9.0)
      IF(ID2(I).EQ.ID(I)) GO TO 45
      TYPE 2, ID2(I),ID(I)
    2 FORMAT(' DIVERSION ',I8,' FROM SNKSTO.CRY IS DIFFERENT THAN DIVERS
     1ION ',I8,' FROM SNKSTO.SPA')
      STOP
   45 CONTINUE
    5 TYPE *, ' '
      TYPE *, '  ENTER FOUR DIGIT YEAR FOR WHICH YOU WANT   '
      TYPE *, '  TO COMPUTE STORAGE ALLOCATION AND USE.     '
      TYPE *, '  FOR EXAMPLE: 1995                          '
      PRINT 46
   46 FORMAT(15X,' '$)
      ACCEPT 49,IYR
   49 FORMAT(I4)
      READ(1,55) (RNAME(K),SPACE(K),FILL(K),EVAP(K),FILLATE(K),K=1,12)
   55 FORMAT(9X,A14,2X,4F10.0)
C
C***** COMPUTE AND PRINT TABLE OF EACH RESERVOIR'S SPACE, FILL,
C***** EVAPORATION AND YIELD.
C
      DO 90 K=1,12
      YIELD(K)=FILL(K)-EVAP(K)
      IF(FILL(K).GT.0.0) PAVAIL(K)=YIELD(K)/FILL(K)
      IF(FILL(K).LE.0.0) PAVAIL(K)=0.0
   90 CONTINUE
      WRITE(6,100) RUNDATE,IYR
  100 FORMAT(1H1//////115X,A9/////////////////,43X,I4,
     1  ' SNAKE 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=1,4
  105 T(K)=0.0
      DO 110 K=1,12
      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,A14,2X,4F12.1)
```

```
      IF(K.EQ.11) WRITE(6,125) (T(I),I=1,4)
  125 FORMAT(/32X,9HSUB TOTAL,7X,4F12.1/)
  110 CONTINUE
      WRITE(6,130) (T(K),K=1,4)
  130 FORMAT(/32X,5HTOTAL,11X,4F12.1)
      IT=1
  185 CONTINUE
      DO 155 I=1,10
      DO 155 K=1,15
  155 CTOT(I,K)=0.0
      CALL TABLE (0,1,1,1)
      CALL TABLE (1,2,0,1)
      CALL TABLE (6,7,1,2)
      CALL TABLE (8,9,0,2)
      CALL TABLE (9,10,0,3)
      CALL TABLE (2,3,1,4)
      CALL TABLE (4,5,0,4)
      CALL TABLE (3,4,0,5)
      CALL TABLE (5,6,0,6)
      CALL TABLE (7,8,0,7)
      CALL TABLE (10,11,0,8)
      WRITE(6,235) RUNDATE,IT
  235 FORMAT(1H1/////123X,A9///58X,4HPART,I2)
      IF(IT.EQ.1) WRITE(6,240) IYR
  240 FORMAT(1H //15X,I4,98H WATER DISTRICT 1 RESERVOIR SPACE BY REACH A
     1SSUMING FULL RESERVOIRS WITHOUT WATER BANK (ACRE-FEET)/)
      IF(IT.EQ.2) WRITE(6,250) IYR
  250 FORMAT(  //12X,I4,108H WATER DISTRICT 1 RESERVOIR STORAGE BY REACH
     1 AFTER PERCENT FILL AND EVAPORATION, WITH WATER BANK (ACRE-FEET)/)
      DO 195 K=1,15
  195 GTOT(K)=0.0
      WRITE(6,210)
  210 FORMAT(///// 8X,11HRIVER REACH,9X, 94H   JACKSON    PALISADES
     1   PAL USR       PAL WWS   HENRYS LAKE   ISLAND PARK   GRASSY LAKE
     2 /)
      DO 190 I=1,8
      DO 205 K=1,15
  205 GTOT(K)=GTOT(K)+CTOT(I,K)
  190 WRITE(6,220) (RCHNM(K,I),K=1,5),(CTOT(I,J),J=1,7)
  220 FORMAT(8X,5A4,7(F10.1,4X))
      WRITE(6,230) (GTOT(K),K=1,7)
  230 FORMAT(/8X,5HTOTAL,15X,7(F10.1,4X))
      WRITE(6,211)
  211 FORMAT(///// 8X,11HRIVER REACH,5X, 98H         RIRIE      AM FALLS
     1  LAKE WALCOTT       AMF LTF       PAL LTF    WATER BANK         T
     2OTAL/)
      DO 212 I=1,8
  212 WRITE(6,220) (RCHNM(K,I),K=1,5),(CTOT(I,J),J=8,12),
     1  (CTOT(I,J),J=14,15)
      WRITE(6,230) (GTOT(K),K=8,12),(GTOT(K),K=14,15)
      IF (IT.EQ.2) GO TO 200
      DO 232 K=1,12
  232 TSPAC(K)=GTOT(K)
C
C***** MOVE AS MUCH WWS CARRYOVER INTO MAIN PALISADES ACCOUNT AS
C***** POSSIBLE AS PER USBR METHOD OF ALLOCATING NEW STORAGE - NEW
C***** FILL IN WWS SPACE THEN DOES NOT INCLUDE CARRYOVER.
```

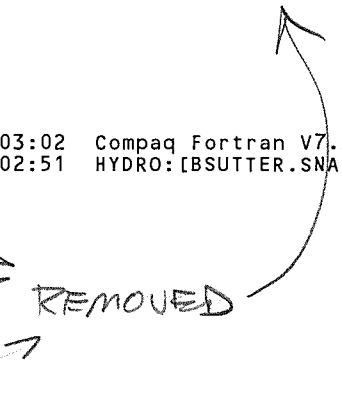
```
172 C
173 C***** THIS ROUTINE NOT USED BEGINNING IN 1992 BECAUSE IT PRODUCED
174 C***** NEGATIVE ALLOCATIONS WHEN THERE WAS LITTLE OR NO NEW FILL
175 C***** OF PALISADES REGULAR SPACE.  NOT SURE OF EXACT REASONING
176 C***** WHEN THIS WAS FIRST USED.
177 C
178 C       DO 25 I=1,JE
179 C       K=2
180 C       IF(I.EQ.IPALU) K=3
181 C       CRYO(I,K)=CRYO(I,K)+CRYO(I,4)
182 C       CRYO(I,4)=0.0
183 C       IF(CRYO(I,K).GT.SPAC(I,K)) THEN
184 C           CRYO(I,K)=SPAC(I,K)
185 C           CRYO(I,4)=CRYO(I,K)-SPAC(I,K)
186 C           END IF
187 C   25 CONTINUE
188 C
189 C***** INITIALIZE NEW STORAGE (NSTO) TO ZERO AND TOTAL CARRYOVER
190 C***** BY RESERVOIR ACCOUNT.
191 C
192       DO 14 K=1,12
193       CTOTC(K)=0.0
194       DO 14 I=1,JE
195       NSTO(I,K)=0.0
196    14 CTOTC(K)=CTOTC(K)+CRYO(I,K)
197       IJKSN=0
198 C
199 C***** BEGIN LOOP TO ALLOCATE NEW FILL TO THE TWELVE RESERVOIR
200 C***** ACCOUNTS.
201 C
202       DO 1 K=1,12
203       KRS=K
204       IF(K.GT.3) KRS=K-1
205       TSP=TSPAC(K)
206 C
207 C***** COMBINE PALISADES AND PALISADES WATER USERS.
208 C
209       IF(K.EQ.2) TSP=TSPAC(2)+TSPAC(3)
210       IF(TSP.LE.0.0) GO TO 1
211       FIL=FILL(KRS)
212       RNAM=RNAME(KRS)
213       IF(K.EQ.3) THEN
214            FIL=PUTSTO
215            RNAM='PALISADES USRS'
216            END IF
217       IF(FIL.LE.TSP) GO TO 13
218       TYPE 997, RNAM,FIL,TSP
219   997 FORMAT(1X,A16,' HAS FILL =',F10.1,' > TOTAL SPACE =',F10.1)
220       GO TO 400
221    13 IF(FIL.EQ.TSP) GO TO 23
222 C
223 C***** ZERO CARRYOVER IF FAILURE TO FILL WAS RESULT OF FLOOD OPERATION
224 C***** OR OTHER RESERVOIR OPERATIONS.
225 C
226       PRINT 22, RNAM,FIL,TSP
227    22 FORMAT(1X,A16,' HAS FILL =',F10.1,' < TOTAL SPACE =',F10.1/
228      1'   IS THE FAILURE TO FILL A RESULT OF FLOOD CONTROL? '/
```

```
      2'   OR DO SPACEHOLDERS NOT HAVE CARRYOVER PRIVILEGES? '/
      3'   IF YOU ANSWER Y, CARRYOVER WILL BE WIPED OUT. Y/N '$)
       ACCEPT 36, FLD
       IF(FLD.NE.'Y') GO TO 24
   23 KC=K
   27 DO 26 I=1,JE
   26 CRYO(I,KC)=0.0
      CTOTC(KC)=0.0
      IF(KC.EQ.2) THEN
                  KC=KC+1
                  GO TO 27
                  END IF
C
C***** COMPUTE NEW FILL TO EACH RESERVOIR BY SUBTRACTING CARRYOVER
C***** FROM TOTAL FILL. THAT AMOUNT (WAT) IS THEN DISTRIBUTED.
C
   24 TCR=CTOTC(K)
      IF(K.EQ.2) TCR=CTOTC(2)+CTOTC(3)
      CNSTO(K)=FIL-TCR
      WAT=CNSTO(K)
      IF(WAT.LE.-0.5) THEN
        PRINT 47, RNAM,TCR,FIL
   47 FORMAT(1X,A16,' HAS TOTAL CARRYOVER =',F10.1,' > TOTAL FILL=',
     1F10.1/'   DO YOU WISH TO STOP? ANSWER Y/N '$)
        ACCEPT 36, STP
        IF(STP.EQ.'Y') GO TO 400
        ENDIF
       J=0
C
C***** IF ALLOCATING JACKSON LAKE, REMOVE LAST TO FILL SPACE OF NORTH
C***** AND SOUTH SIDE TWIN FALLS CANALS (TJTFSP) AND ALSO CORRESPONDING
C***** CARRYOVER (TJTFCO) FROM TOTAL SPACE (TSP) AND TOTAL FIL (FIL),
C***** RESPECTIVELY.  ALSO CORRECT TOTAL CARRYOVER (TCR).  IF REVISED
C***** FILL (FIL) IS GREATER THAN REVISED TOTAL SPACE (TSP), THEN
C***** SPACE OTHER THAN TWIN FALLS JACKSON SPACE HAS FILLED COMPLETELY,
C***** SO SET TOTAL FILL EQUAL TO TOTAL SPACE.
C
       IF(K.GT.1) GO TO 6
       IF(IJKSN.GT.0.OR.FIL.EQ.TSP) GO TO 6
       IF(FLD.NE.'Y') THEN
            IJKSN=1
            TJNSTFSP=SPAC(JNSTF,1)
            TJSSTFSP=SPAC(JSSTF,1)
            SPAC(JNSTF,1)=0.0
            SPAC(JSSTF,1)=0.0
            TJTFSP=TJNSTFSP+TJSSTFSP
            TJNSTFCO=CRYO(JNSTF,1)
            TJSSTFCO=CRYO(JSSTF,1)
            TJTFCO=TJNSTFCO+TJSSTFCO
            CRYO(JNSTF,1)=0.0
            CRYO(JSSTF,1)=0.0
            TSP=TSP-TJTFSP
            FIL=FIL-TJTFCO
            TCR=TCR-TJTFCO
            IF(FIL.GE.TSP) THEN
                JKTFNSTO=FIL-TSP
                FIL=TSP
```

```
286                GO TO 23
287                ENDIF
288             JKTFNSTO=0.0
289             ENDIF
290 C
291 C***** SET TOTAL NEW STORAGE (TNSTOR) ASSIGNED TO SPACEHOLDERS EQUAL
292 C***** TO ZERO.  KEEP TRACK OF NUMBER OF PASSES MADE TO DISTRIBUTE
293 C***** ALL NEW STORAGE (CNSTO(K)) AND STOP IF IT GOES WILD.
294 C
295       6 TNSTOR=0.0
296         J=J+1
297         IF(J.GT.2000) TYPE 998, RNAM
298     998 FORMAT(' RUNAWAY LOOP ALLOCATING ',A16,' STORAGE')
299         TYPE 999, J,WAT
300     999 FORMAT(1X,I5,F12.4)
301 C
302 C***** USE TOTAL RESERVOIR SPACE FOR ANY OF THE AMERICAN FALLS OR
303 C***** PALISADES ACCOUNTS (EXCEPT LAST TO FILL), SINCE THOSE ARE ALL PROPORTIONAL TO
304 C***** TOTAL SPACE AS DESCRIBED BELOW.
305 C
306         IF(FLD.NE.'Y') THEN
307              IF(K.EQ.2.OR.K.EQ.4.) TSP=1200000.0
308              IF(K.EQ.9) TSP=1672590.0
309 C
310 C***** CHANGED 10/2002 --  THE ABOVE THREE STATEMENTS WERE ADDED AND THE FOLLOWING
311 C***** THREE STATEMENTS WERE DELETED SO THAT LAST TO FILL SPACE (11 AND 12) IN
312 C***** AMERICAN FALLS AND PALISADES FILLS PROPORTIONAL TO THE SPACE IN THOSE
313 C***** ACCOUNTS RATHER THAN PROPORTIONAL TO THE ENTIRE SPACE THAT EACH STORAGE
314 C***** HOLDER OWNS IN THE RESERVOPIR AS A WHOLE (THE PERCENTAGE IN THE
315 C***** CONTRACTS).
316 C
317 C        IF(FLD.NE.'Y') THEN
318 C             IF(K.EQ.2.OR.K.EQ.4.OR.K.EQ.12) TSP=1200000.0
319 C             IF(K.EQ.9.OR.K.EQ.11) TSP=1672590.0
320               ENDIF
321         KS=K
322         DO 4 I=1,JE
323         SPC=SPAC(I,KS)
324 C
325 C***** SUM ALL SPACE (SPC) ACCOUNTS FOR PALISADES AND AMERICAN FALLS
326 C***** TO DETERMINE PROPORTIONAL FILL BECAUSE USBR CONTRACTS READ
327 C***** AS A PERCENT DETERMINED FROM ALL STORAGE IN A RESERVOIR, NOT
328 C***** FROM EACH RESERVOIR RIGHT.  FOR PALISADES USERS (IPALU=1) BE
329 C***** SURE TO ADD ALL REGULAR PALISADES USERS SPACE (TSPAC(3)).
330 C
331         IF(FLD.NE.'Y') THEN
332            IF(KS.EQ.2.OR.KS.EQ.4) THEN
333                           SPC=SPAC(I,2)+SPAC(I,4)+SPAC(I,12)
334                           END IF
335            IF(KS.EQ.9) SPC=SPAC(I,9)+SPAC(I,11)
336 C
337 C
338 C***** CHANGED 10/2002 --  THE ABOVE FIVE STATEMENTS WERE ADDED AND THE FOLLOWING
339 C***** SIX STATEMENTS WERE DELETED SO THAT LAST TO FILL SPACE (11 AND 12) IN
340 C***** AMERICAN FALLS AND PALISADES FILLS PROPORTIONAL TO THE SPACE IN THOSE
341 C***** ACCOUNTS RATHER THAN PROPORTIONAL TO THE ENTIRE SPACE THAT EACH STORAGE
342 C***** HOLDER OWNS IN THE RESERVOPIR AS A WHOLE (THE PERCENTAGE IN THE
```

REMOVED

```
C***** CONTRACTS).
C
C         IF(KS.EQ.2.OR.KS.EQ.4.OR.KS.EQ.12) THEN
C                        SPC=SPAC(I,2)+SPAC(I,4)+SPAC(I,12)
C                        IF(KS.EQ.12.AND.I.EQ.IPALU)
C     1                  SPC=SPAC(I,4)+SPAC(I,12)+TSPAC(3)
C                        END IF
C         IF(KS.EQ.9.OR.KS.EQ.11) SPC=SPAC(I,9)+SPAC(I,11)
          ENDIF
C
C***** ALLOCATE NEW FILL (WAT) TO INDIVIDUAL ENTITIES AND USES WITHIN
C***** EACH RESERVOIR PROPORTIONAL TO SPACE OWNED.  LIMIT TOTAL FILL
C***** (NEW FILL PLUS CARRYOVER) OF EACH SPACEHOLDER (STOR) TO SPACE
C***** OWNED (SPAC).  KEEP TRACK OF TOTAL STORAGE ASSIGNED TO SPACE-
C***** HOLDERS (TNSTOR).
C
       IF(KS.EQ.11.OR.KS.EQ.12) THEN
            IF(FIL.EQ.TSPAC(KS)) GO TO 94
            ENDIF
       IF(CRYO(I,KS).GT.SPC) TYPE 996, UNAME(I)
  996  FORMAT(' CARRYOVER STORAGE FOR ',A14,' GREATER THAN SPACE')
       NSTO(I,KS)=((SPC/TSP)*WAT)+NSTO(I,KS)
       STOR(I,KS)=NSTO(I,KS)+CRYO(I,KS)
       IF(STOR(I,KS).LE.SPAC(I,KS)) GO TO 95
   94  STOR(I,KS)=SPAC(I,KS)
       NSTO(I,KS)=STOR(I,KS)-CRYO(I,KS)
   95  TNSTOR=TNSTOR+NSTO(I,KS)
    4  CONTINUE
C
C***** WHEN FINISHED WITH PALISADES REGULAR ACCOUNTS, THEN DETERMINE
C***** PALISADES WATER USERS FILL COLLECTIVELY AS ONE ACCOUNT.
C
       IF(KS.EQ.2) THEN
                   KS=KS+1
                   SPC=TSPAC(3)
       IF(FLD.NE.'Y') SPC=SPAC(IPALU,4)+SPAC(IPALU,12)+TSPAC(KS)
                   PUNSTO=((SPC/TSP)*WAT)+PUNSTO
                   PUTSTO=PUNSTO+CTOTC(KS)
                   IF(PUTSTO.GT.TSPAC(KS)) THEN
                                           PUTSTO=TSPAC(KS)
                                           PUNSTO=PUTSTO-CTOTC(KS)
                                           END IF
                   TNSTOR=TNSTOR+PUNSTO
                   END IF
C
C***** SUBTRACT TOTAL STORAGE ALLOCATED FROM TOTAL NEW FILL TO GET
C***** AMOUNT REMAINING TO BE ALLOCATED (WAT).
C
       WAT=CNSTO(K)-TNSTOR
       IF(WAT.GT.0.07) GO TO 6
C
C***** IF JACKSON LAKE SPACE DID NOT COMPLETELY FILL AND THERE WAS NO
C***** FLOOD CONTROL, THEN IJCKSN WILL EQUAL 1 AND THE FILL OF THE
C***** TWIN FALLS SPACE MUST NOW BE CALCULATED SINCE IT FILLS LAST
C***** WITHIN JACKSON LAKE.
C
       IF(K.GT.1) GO TO 1
```



```
      IF(IJKSN.EQ.1) THEN
          J=0
          WAT=JKTFNSTO
          SPAC(JNSTF,1)=TJNSTFSP
          SPAC(JSSTF,1)=TJSSTFSP
          CRYO(JNSTF,1)=TJNSTFCO
          CRYO(JSSTF,1)=TJSSTFCO
   98     TNSTOR=0.0
          J=J+1
          IF(J.GT.2000) TYPE 998, RNAM
          TYPE 999, J,WAT
          DO 97 I=JNSTF,JSSTF
          SPC=SPAC(I,1)
          NSTO(I,1)=((SPC/TJTFSP)*WAT)+NSTO(I,1)
          STOR(I,1)=NSTO(I,1)+CRYO(I,1)
          IF(STOR(I,1).LE.SPAC(I,1)) GO TO 96
          STOR(I,1)=SPAC(I,1)
          NSTO(I,1)=STOR(I,1)-CRYO(I,1)
   96     TNSTOR=TNSTOR+NSTO(I,1)
   97     CONTINUE
          WAT=JKTFNSTO-TNSTOR
          IF(WAT.GT.0.1) GO TO 98
          END IF
    1 CONTINUE
C
C***** TRANSFER WATER BANK TRANSFERS TO TABLE ARRAY,
C***** COMPUTE YIELD OF SPACE VALUES AFTER EVAPORATION.
C
      PRINT 3
    3 FORMAT(' DO YOU WANT TO READ STORAGE USE, ADJUSTMENTS, AND/OR'/
     1' WATER BANK TRANSACTIONS BY CANAL/USER? Y/N: '$)
      ACCEPT 36, USE
   36 FORMAT(A1)
      DO 165 I=1,JE
      BANK(I)=0.0
      IF(USE.NE.'Y') GO TO 320
      READ(9,260,END=35) JD,USED(I),ADJU(I),BANK(I),RFWB(I)
  260 FORMAT(I8,15X,4F10.0)
      IF(JD.EQ.ID(I)) GO TO 320
   35 WRITE(6,265) JD,ID(I)
  265 FORMAT(15H DIVERSION NO. ,I10,50H FROM SNKSTO'YR'.USE DOES NOT EQU
     1AL DIVERSION NO. ,I10,25H FROM SNKSTO'YR'.SPA LIST)
      GO TO 400
  320 STOR(I,14)=BANK(I)
      DO 175 K=1,12
      KS=K
      IF(K.GE.3) KS=K-1
  175 STOR(I,K)=STOR(I,K)*PAVAIL(KS)
  165 CONTINUE
      IT=2
      GO TO 185
  200 PRINT *,' '
      PRINT 201
  201 FORMAT(' DO YOU WISH TO CREATE A FILE (SNKCAN.IND) OF STORED WATER
     1 SUPPLY'/' BY DIVERSION FOR INPUT TO ACCOUNTING PROGRAM? Y/N: '$)
      ACCEPT 36, FILE
      IF(FILE.NE.'Y') GO TO 204
```

```
      OPEN(UNIT=2,NAME='SNKCAN.IND',TYPE='NEW',CARRIAGECONTROL='LIST')
      DUM=0.0
      DO 202 I=1,JEN
  202 WRITE(2,203) ID(I),DUM,STOR(I,15)
  203 FORMAT(I8,2F10.1)
  204 CONTINUE
C
C***** SET TABLE NUMBER TO BEGIN OUTPUT OF IRRIGATION SEASON
C***** STORED WATER TRANSACTIONS.
C
      IF(USE.NE.'Y') GO TO 400
      FTAB=21.0
   33 PRINT 28, FTAB
   28 FORMAT(' THE FIRST STORAGE TABLE FOR REPORT WILL BE NO. ',
     1 F5.1/' IS THIS OK? Y/N/Q '$)
      ACCEPT 36, TBL
      IF(TBL.EQ.'Q') GO TO 400
      IF(TBL.EQ.'Y') GO TO 310
      TYPE *, ' '
      PRINT 29
   29 FORMAT(' ENTER NEW BEGINNING TABLE NUMBER = '$)
      ACCEPT 32, FTAB
   32 FORMAT(F5.0)
      TYPE *, ' '
      GO TO 33
  310 IF(FTAB.LE.0.0) GO TO 400
      K=1
      DO 280 I=1,JE
C
C***** MOVE MILNER STORAGE TO MISCELLANEOUS TABLE SINCE STORAGE
C***** IS NOT ALLOCATED TO ANY USER
C
      IF(ID(I).EQ.99999950) THEN
                            STOR(I,13)=STOR(I,13)+YIELD(12)
                            JM=I
                            END IF
      ISKIP(I)=0.0
      IF(ISK(K).NE.ID(I)) GO TO 280
      ISKIP(I)=1
      K=K+1
  280 CONTINUE
      DO 315 I=1,JE
      EXCS(I)=0.0
      TCRYO(I)=0.0
      WBRV(I)=0.0
      BALN(I)=STOR(I,14)-USED(I)
      IF(BALN(I).GT.0.0) GO TO 275
  295 BALN(I)=BALN(I)+STOR(I,13)
      GO TO 290
  275 WBRV(I)=BALN(I)
      BALN(I)=STOR(I,13)
  290 BALN(I)=BALN(I)+RFWB(I)
C
C***** IF THERE IS A NEGATIVE ADJUSTMENT TO STORAGE USE (ADJU), THAT
C***** AMOUNT IS FIRST DEDUCTED FROM THE WATER REVERTING TO THE WATER
C***** BANK (WBRV), FOR ALL MISCELLANEOUS STORAGE ACCOUNTS.
C
```

```
      IF(ADJU(I).GE.0.0.OR.WBRV(I).LE.0.0.OR.ID(I).LT.90000000) GO TO
     1 285
      ADD=WBRV(I)
      WBRV(I)=WBRV(I)+ADJU(I)
      IF(WBRV(I).LT.0.0) WBRV(I)=0.0
      ADD=ADD-WBRV(I)
      BALN(I)=BALN(I)+ADD
  285 TCRYO(I)=BALN(I)+ADJU(I)
      IF(TCRYO(I).GE.0.0) GO TO 315
      EXCS(I)=TCRYO(I)*(-1.0)
      TCRYO(I)=0.0
  315 CONTINUE
      DO 455 I=1,8
      STOT(I,1)=CTOT(I,13)
      STOT(I,2)=CTOT(I,14)
      DO 455 K=3,9
  455 STOT(I,K)=0.0
      STOT(8,1)=STOT(8,1)+YIELD(12)
      ICONT=0
      CALL TABL2 (0,1,1,1)
      CALL TABL2 (1,2,0,1)
      CALL TABL2 (6,7,1,2)
      CALL TABL2 (8,9,0,2)
      CALL TABL2 (9,10,0,3)
      CALL TABL2 (2,3,1,4)
      CALL TABL2 (4,5,0,4)
      CALL TABL2 (3,4,0,5)
      CALL TABL2 (5,6,0,6)
      CALL TABL2 (7,8,0,7)
      CALL TABL2 (10,11,0,8)
      WRITE(6,540) RUNDATE,FTAB,IYR
  540 FORMAT(1H1/////123X,A9/////18X,5HTABLE,F4.0,2X,20HSUMMARY BY REACH
     1 OF ,I4,54H STORED WATER ACCOUNTS IN WATER DISTRICT 1 (ACRE-FEET))
      DO 595 K=1,9
  595 GSTOT(K)=0.0
      WRITE(6,510)
  510 FORMAT( /28X,76H            STORAGE OR                          RETUR
     1N TO                      /28X,67H            WATER BANK
     2 REVERTED TO SPACEHOLDER           /28X,99H   STORAGE   PURCHASE,
     3  STORAGE  WATER BANK    FROM                   ADJUST-     EXCESS
     4    CARRY-/  4X,13HREACH        ,11X,99H  ALLOCATED SUPPLY (-)
     5 USED   FROM USER   WATER BANK   BALANCE       MENT       USED
     6   OVER/)
      DO 590 I=1,8
      DO 505 K=1,9
  505 GSTOT(K)=GSTOT(K)+STOT(I,K)
  590 WRITE(6,520) (RCHNM(K,I),K=1,5),(STOT(I,J),J=1,9)
  520 FORMAT( 4X,5A4,4X,9(2X,F9.1))
      WRITE(6,530) (GSTOT(K),K=1,9)
  530 FORMAT(/ 3X,6H TOTAL,19X,9(2X,F9.1))
C
C***** COMPUTE NET STORAGE USED (RDIFF).  MOVE LATE POWER STORAGE AT
C***** AMERICAN FALLS AND PALISADES BACK TO ORIGINAL RIGHT SINCE IT
C***** NOW LOSES ITS IDENTITY(NO CARRYOVER IN THIS SPACE.  IF THERE IS
C***** NO STORAGE IN PALISADES REGULAR ACCOUNT (STOR(I,2)), THEN ASSUME
C***** THIS POWER STORAGE CAME FROM PALISADES USERS (STOR(I,3)), SO ADD
C***** IT BACK THERE INSTEAD.  ALSO COMPUTE PROPORTIONAL SPACE BY USER
```

```
C***** IN FIRST 200,000 AF OF JACKSON LAKE FOR FIRST USE COMPUTATIONS
C***** BELOW.
C
      DO 608 I=1,JE
      DIFF(I)=STOR(I,13)-TCRYO(I)
      RDIFF(I)=DIFF(I)
      IF(STOR(I,2).LE.0.0) THEN
            STOR(I,3)=STOR(I,3)+STOR(I,12)
            GO TO 607
            ENDIF
      STOR(I,2)=STOR(I,2)+STOR(I,12)
  607 STOR(I,9)=STOR(I,9)+STOR(I,11)
      STOR(I,11)=0.0
      STOR(I,12)=0.0
  608 JDFL(I)=200000.*(SPAC(I,1)/TSPAC(1))
C
C***** COMPUTE CARRYOVER BY USER BY RESERVOIR.  WHEN IO=11, PROPORTION
C***** FIRST 200,000 ACRE FEET IN JACKSON LAKE AMONG SPACEHOLDERS
C***** ACCORDING TO SPACE OWNED.
C
      DO 615 LO=1,11
      IO=IOR(LO)
      IF(IO.NE.11) GO TO 613
      IO=1
      IF(GTOT(1).LE.200000.) GO TO 613
      TJL=0.0
      PTJL=0.0
  606 DO 612 I=1,JE
      DF=RDIFF(I)
      IF(DF.GT.JDFL(I)) DF=JDFL(I)
      IF(DF.LE.0.0) GO TO 612
      STOR(I,IO)=STOR(I,IO)-DF
      IF(STOR(I,IO).GE.0.0) RDIFF(I)=RDIFF(I)-DF
      IF(STOR(I,IO).LT.0.0) RDIFF(I)=RDIFF(I)-DF-STOR(I,IO)
      IF(STOR(I,IO).LT.0.0) STOR(I,IO)=0.0
      TJL=TJL+(DF-RDIFF(I))
  612 CONTINUE
      DTJL=TJL-PTJL
      IF(DTJL.LE.02) GO TO 615
      REMJ=200000.-TJL
      IF(REMJ.LE.0.2) GO TO 615
      DO 616 I=1,JE
  616 JDFL(I)=REMJ*(SPAC(I,1)/TSPAC(1))
      PTJL=TJL
      GO TO 606
  613 CONTINUE
      DO 614 I=1,JE
      DF=RDIFF(I)
      IF(DF.LE.0.0) GO TO 614
      STOR(I,IO)=STOR(I,IO)-DF
      IF(STOR(I,IO).GE.0.0) RDIFF(I)=RDIFF(I)-DF
      IF(STOR(I,IO).LT.0.0) RDIFF(I)=STOR(I,IO)*(-1.0)
      IF(STOR(I,IO).LT.0.0) STOR(I,IO)=0.0
  614 CONTINUE
  615 CONTINUE
C
C***** PLACE FREMONT MADISON CARRYOVER ENTIRELY IN ISLAND PARK
```

```
C***** SINCE THERE ARE NO INDIVIDUAL CARRYOVER PRIVILEGES
C
      K=NDEND(10)+1
      DO 618 I=K,JE
  618 IF(ID(I).EQ.99999600) STOR(I,6)=TCRYO(I)
      DO 655 I=1,10
      DO 655 K=1,12
  655 CTOT(I,K)=0.0
      CALL TABL3 (0,1,1,1)
      CALL TABL3 (1,2,0,1)
      CALL TABL3 (6,7,1,2)
      CALL TABL3 (8,9,0,2)
      CALL TABL3 (9,10,0,3)
      CALL TABL3 (2,3,1,4)
      CALL TABL3 (4,5,0,4)
      CALL TABL3 (3,4,0,5)
      CALL TABL3 (5,6,0,6)
      CALL TABL3 (7,8,0,7)
      CALL TABL3 (10,11,0,8)
      WRITE(6,652)
  652 FORMAT(1H1)
      WRITE(6,650) RUNDATE,IYR
  650 FORMAT(/123X,A9///35X,I4, 59H WATER DISTRICT 1 RESERVOIR CARRYOVER
     1 BY REACH  (ACRE-FEET)/)
      DO 695 K=1,12
  695 GTOT(K)=0.0
      WRITE(6,610)
  610 FORMAT(1H ,  14HRIVER REACH   ,6X, 99H  CARRYOVER JACKSON PALISADE
     1S  PAL USR   PAL WWS HNRYS L  ISL PRK GRSSY L   RIRIE  AM FALLS WA
     2LCOTT/)
      DO 690 I=1,8
      DO 605 K=1,10
  605 GTOT(K)=GTOT(K)+CTOT(I,K)
  690 WRITE(6,620) (RCHNM(K,I),K=1,5),STOT(I,9),(CTOT(I,J),J=1,10)
  620 FORMAT(1X,5A4,F10.1,F9.1,F10.1,F9.1,F10.1,F8.1,F9.1,2F8.1,
     1 F10.1,F8.1)
      WRITE(6,630) GSTOT(9),(GTOT(K),K=1,10)
  630 FORMAT(/6H TOTAL,15X,F10.1,F9.1,F10.1,F9.1,F10.1,F8.1,F9.1,
     12F8.1,F10.1,F8.1)
      GRTOT=0.0
      DO 640 I=1,10
  640 GRTOT=GRTOT+GTOT(I)
      SGRTOT=GRTOT+TCRYO(JE)+TCRYO(JM)
      WRITE(6,642)IYR
  642 FORMAT(///////     35X,I4, 64H WATER DISTRICT 1 RESERVOIR CARRYOVE
     1R - SYSTEM TOTAL (ACRE-FEET)/)
      WRITE(6,644) GRTOT,TCRYO(JE),TCRYO(JM),SGRTOT
  644 FORMAT(55X,'ALLOCATED         - ',F10.1/
     1       55X,'UNALLOCATED       - ',F10.1/
     2       55X,'MILNER RESERVOIR - ',F10.1//
     3       55X,'TOTAL SYSTEM      - ',F10.1)
      PRINT 16
   16 FORMAT(' DO YOU WISH TO ENTER LATE SEASON RESERVOIR FILL? Y/N '$)
      ACCEPT 7,ANSWER
    7 FORMAT(1A1)
      PRINT *,' '
      IF(ANSWER.NE.'Y') GO TO 8
```

```
685     15 DO 9 K=1,9
686        PRINT 11, RNAME(K),FILLATE(K)
687     11 FORMAT(1X,'LATE SEASON FILL FOR       ',A14,' = ',F10.1,' ENTER NE
688       1W FILL? Y/N/Q '$)
689        ACCEPT 7,ANSWER
690        IF(ANSWER.EQ.'Q') GO TO 18
691        IF(ANSWER.NE.'Y') GO TO 9
692        PRINT 19,RNAME(K)
693     19 FORMAT(1X,'ENTER LATE SEASON FILL FOR ',A14,' = '$)
694        ACCEPT 21,FILLATE(K)
695     21 FORMAT(F10.0)
696      9 CONTINUE
697     18 PRINT 17
698     17 FORMAT(' DO YOU WISH TO REENTER LATE SEASON RESERVOIR FILL? Y/N '
699       1$)
700        ACCEPT 7,ANSWER
701        IF(ANSWER.EQ.'Y') GO TO 15
702      8 WRITE(6,142)IYR
703    142 FORMAT(//////     35X,I4,' WATER DISTRICT 1 RESERVOIR TOTAL STORAG
704       1E - OCTOBER 31 (ACRE-FEET)'///43X,'RESERVOIR',12X,'CARRYOVER',
705       2'   LATE FILL',7X,'TOTAL'/)
706        GTOT(2)=GTOT(2)+GTOT(3)
707        GTOT(3)=TCRYO(JE)+TCRYO(JM)
708        TLATE=0.0
709        GTFILL=0.0
710        DO 146 K=1,10
711        K1=K
712        IF(K.GT.2) K1=K+1
713        IF(K1.EQ.11) K1=3
714        TFILL(K)=GTOT(K1)+FILLATE(K)
715        TLATE=TLATE+FILLATE(K)
716        GTFILL=GTFILL+TFILL(K)
717        IF(K.EQ.10) GO TO 148
718        WRITE(6,144) RNAME(K),GTOT(K1),FILLATE(K),TFILL(K)
719    144 FORMAT(43X,A14,4X,3F12.1)
720    146 CONTINUE
721    148 WRITE(6,147) GTOT(K1),FILLATE(K),TFILL(K)
722    147 FORMAT(43X,'OTHER',13X,3F12.1)
723        WRITE(6,145) SGRTOT,TLATE,GTFILL
724    145 FORMAT(/43X,'TOTAL',13X,3F12.1)
725     38 PRINT 31
726     31 FORMAT(' DO YOU WANT TO CREATE A FILE OF THE CARRYOVER'/
727       1' BY CANAL AND RESERVOIR? Y/N: '$)
728        ACCEPT 36, FILE
729        IF(FILE.EQ.'N') GO TO 400
730        IF(FILE.NE.'Y') GO TO 38
731        OPEN (UNIT=10,NAME='CARRYOUT',TYPE='NEW',CARRIAGECONTROL='LIST')
732        DO 37 I=1,JE
733     37 WRITE(10,34) ID(I),UNAME(I),(STOR(I,K),K=1,12)
734     34 FORMAT(I8,1X,A14,12F9.1)
735    400 CONTINUE
736        STOP
737        END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-00000000	SNKSTO$MAIN

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
1-00000040	R*4	ADD	1-000000BC	I*4	IPALU	1-0000005C	I*4	KS	1-00000070	R*4	TJNSTFCO
1-00000020	R*4	ANSWER	6-0001E950	I*4	IT	1-000000C8	I*4	L	1-0000007C	R*4	TJNSTFSP
1-0000002C	R*4	DF	6-0001E954	I*4	IYR	1-0000003C	I*4	LO	1-0000006C	R*4	TJSSTFCO
1-00000028	R*4	DTJL	1-000000A8	I*4	J	1-00000030	R*4	PTJL	1-00000078	R*4	TJSSTFSP
1-0000004C	R*4	DUM	1-00000050	I*4	JD	1-00000054	R*4	PUNSTO	1-00000068	R*4	TJTFCO
1-00000098	R*4	FIL	1-000000AC	I*4	JE	1-00000094	R*4	PUTSTO	1-00000074	R*4	TJTFSP
1-000000CC	R*4	FILE	1-000000B0	I*4	JEN	1-00000024	R*4	REMJ	1-00000000	R*8	TLATE
1-00000090	R*4	FLD	1-00000064	I*4	JKTFNSTO	1-000000D8	CHAR	RNAM	1-00000060	R*4	TNSTOR
6-0001E958	R*4	FTAB	1-00000044	I*4	JM	6-0001E960	CHAR	RUNDATE	1-0000009C	R*4	TSP
1-00000010	R*8	GRTOT	1-000000B8	I*4	JNSTF	1-00000008	R*8	SGRTOT	1-000000E8	CHAR	U
1-0000001C	R*4	GTFILL	1-000000B4	I*4	JSSTF	1-00000058	R*4	SPC	1-000000D0	R*4	USE
1-000000C4	I*4	I	1-000000C0	I*4	K	1-00000080	R*4	STP	1-00000084	R*4	WAT
6-0001E95C	I*4	ICONT	1-00000018	I*4	K1	1-00000048	R*4	TBL			
1-000000A4	I*4	IJKSN	1-0000008C	I*4	KC	1-00000088	R*4	TCR			
1-00000038	I*4	IO	1-000000A0	I*4	KRS	1-00000034	R*4	TJL			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
1-00018B60	R*8	BANK	3600	(450)
1-00021228	R*8	CNSTO	96	(12)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
1-000211C8	R*8	CTOTC	96	(12)
6-0001C278	R*4	DIFF	1800	(450)

1-00000148	R*4	EVAP	48	(12)
6-0000EF88	R*4	EXCS	1800	(450)
1-00021548	R*8	FILL	96	(12)
1-00021168	R*8	FILLATE	96	(12)
1-00021350	R*8	GSTOT	72	(9)
1-000212E8	R*8	GTOT	104	(13)
6-0001C9AC	I*4	ID	1800	(450)
1-00020300	I*4	ID2	1800	(450)
1-0001FBF0	I*4	IK	1800	(450)
2-00002050	I*4	IOR	44	(11)
2-000020A8	I*4	ISK	120	(30)
6-0001BB70	I*4	ISKIP	1800	(450)
1-0000D480	R*8	JDFL	3600	(450)
1-000215A8	I*4	JID	200	(50)
2-0000207C	I*4	KREND	44	(11)
6-0001C980	I*4	NDEND	44	(11)
1-00000180	R*8	NSTO	54000	(450, 15)
1-00000118	R*4	PAVAIL	48	(12)
1-00021440	R*8	PYLD	72	(9)
7-00000000	R*4	RCHNM	160	(5, 8)
1-00020A10	R*4	RDIFF	1800	(450)
6-0000FD98	R*4	RFWB	1800	(450)
1-00021398	CHAR	RNAME	168	(12)
1-0000E2A0	R*8	SPAC	43200	(450, 12)
1-000214E8	R*8	SPACE	96	(12)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
1-000000F8	R*8	T	32	(4)
6-0000DA70	R*8	TCRYO	3600	(450)
1-00021120	R*8	TFILL	72	(9)
1-00019980	R*4	TOTR	25200	(450, 14)
1-00021288	R*8	TSPAC	96	(12)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)
1-00021488	R*8	YIELD	96	(12)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
4-00002FFC	1	4-00001F78	24	4-000028A0	94	4-000011D8	185	4-00003ADC	290	4-000044F4	607
**	4	4-00001EE8	26	4-00002980	95	**	190	4-00003A88	295	**	608
**	5	4-00001ECC	27	4-00002F60	96	4-000014C8	195	**	310	4-00004B4C	612
4-000023A0	6	**	30	**	97	**	200	4-00003D78	315	4-00004658	613
**	8	**	33	4-00002CBC	98	**	202	4-00003290	320	4-00004860	614
4-000057BC	9	**	35	4-00000E28	105	4-00003604	204	4-000060A4	400	4-00004C80	615
**	13	**	37	4-00001114	110	4-00001560	205	4-00003E2C	455	4-00004BF4	616
**	14	**	38	**	146	**	212	4-00004144	505	4-00004CF4	618
4-000055E0	15	4-00000544	40	**	148	4-00001AC8	232	**	590	4-00005358	640
**	18	**	45	4-00001200	155	**	275	4-000040A8	595	4-00004D9C	655
4-00000560	20	**	60	**	165	4-0000391C	280	4-0000509C	605	**	690
4-00001EBC	23	4-00000D3C	90	4-00003308	175	4-00003CB0	285	4-000048A0	606	4-00005000	695

```
      SUBROUTINE TABLE (KB,KE,IRET,JT)
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      WRITE(6,160) IT,RUNDATE
  160 FORMAT(1H1/58X,4HPART,I2,59X,A9)
      IF(IT.EQ.1) WRITE(6,140) IYR
  140 FORMAT(//18X,I4,98H WATER DISTRICT 1 RESERVOIR SPACE BY USER ASSUM
     1ING FULL RESERVOIRS, WITHOUT WATER BANK (ACRE-FEET)/)
      IF(IT.EQ.2) WRITE(6,150) IYR
  150 FORMAT(//15X,I4,107H WATER DISTRICT 1 RESERVOIR STORAGE BY USER AF
     1TER PERCENT FILL AND EVAPORATION, WITH WATER BANK (ACRE-FEET)/)
      WRITE(6,145) (RCHNM(I,JT),I=1,5)
  145 FORMAT(54X,5A4/)
      WRITE(6,95)
   95 FORMAT(1H ,13HNUMBER   USER,11X,107H       JACKSN PALSDS PL USR PL
     1 WWS HNRY L ISL PK GRSY L  RIRIE AM FLS L WLCT AM LTF PL LTF W BAN
     2K     TOTAL/)
      IB=1
      IF(KB.EQ.0) GO TO 100
      IB=NDEND(KB)+1
  100 IE=NDEND(KE)
      DO 80 I=IB,IE
      STOR(I,13)=0.0
      DO 60 K=1,12
      CTOT(JT,K)=CTOT(JT,K)+STOR(I,K)
      STOR(I,13)=STOR(I,13)+STOR(I,K)
   60 CONTINUE
      STOR(I,15)=STOR(I,13)+STOR(I,14)
      CTOT(JT,13)=CTOT(JT,13)+STOR(I,13)
      CTOT(JT,14)=CTOT(JT,14)+STOR(I,14)
      CTOT(JT,15)=CTOT(JT,15)+STOR(I,15)
      WRITE(6,70) ID(I),UNAME(I),(JIDNNT(STOR(I,K)),K=1,12),
     1                           JIDNNT(STOR(I,14)),STOR(I,15)
   70 FORMAT(1X,I8,1X,A14,7X,13I7,F10.1)
   80 CONTINUE
      IF(IRET.GT.0) GO TO 10
      WRITE(6,20) (JIDNNT(CTOT(JT,K)),K=1,12),JIDNNT(CTOT(JT,14)),
     1                                    CTOT(JT,15)
   20 FORMAT(/10X,5HTOTAL,9X,7X,13I7,F10.1)
   10 CONTINUE
      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-000060C4	TABLE

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
6-0001E958	R*4	FTAB	1-00021674	I*4	IE	**	I*4	JT	6-0001E960	CHAR	RUNDATE
1-0002167C	I*4	I	**	I*4	IRET	1-00021670	I*4	K			
1-00021678	I*4	IB	6-0001E950	I*4	IT	**	I*4	KB			
6-0001E95C	I*4	ICONT	6-0001E954	I*4	IYR	**	I*4	KE			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
6-0001C278	R*4	DIFF	1800	(450)
6-0000EF88	R*4	EXCS	1800	(450)
6-0001C9AC	I*4	ID	1800	(450)
6-0001BB70	I*4	ISKIP	1800	(450)
6-0001C980	I*4	NDEND	44	(11)
7-00000000	R*4	RCHNM	160	(5, 8)
6-0000FD98	R*4	RFWB	1800	(450)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
6-0000DA70	R*8	TCRYO	3600	(450)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label
4-000068D8	10	**	60	**	80	4-00006330	100

```
      SUBROUTINE TABL2 (KB,KE,IRET,JT)
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      IF(ICONT.EQ.1) GO TO 120
      WRITE(6,140) RUNDATE,FTAB,IYR,(RCHNM(I,JT),I=1,5)
  140 FORMAT(1H1/////123X,A9//26X,5HTABLE,F4.0,4X,I4,25H STORED WATER AC
     1COUNTS - , 5A4,12H (ACRE-FEET))
      GO TO 130
  120 WRITE(6,150) FTAB
  150 FORMAT(1H1////////49X,5HTABLE,F4.0,4X,9HCONTINUED)
  130 WRITE(6,95)
   95 FORMAT( /28X,76H            STORAGE OR                          RETU
     1RN TO                    /28X,67H            WATER BANK
     2 REVERTED TO SPACEHOLDER          /28X,99H   STORAGE   PURCHASE,
     3  STORAGE  WATER BANK    FROM              ADJUST-      EXCESS
     4    CARRY-/  4X,13HNUMBER   NAME,11X,99H  ALLOCATED SUPPLY (-)
     5 USED   FROM USER   WATER BANK   BALANCE       MENT       USED
     6   OVER/)
      IB=1
      IF(KB.EQ.0) GO TO 100
      IB=NDEND(KB)+1
  100 IE=NDEND(KE)
      DO 80 I=IB,IE
      STOT(JT,3)=STOT(JT,3)+USED(I)
      STOT(JT,4)=STOT(JT,4)+WBRV(I)
      STOT(JT,5)=STOT(JT,5)+RFWB(I)
      STOT(JT,6)=STOT(JT,6)+BALN(I)
      STOT(JT,7)=STOT(JT,7)+ADJU(I)
      STOT(JT,8)=STOT(JT,8)+EXCS(I)
      STOT(JT,9)=STOT(JT,9)+TCRYO(I)
      IF(ISKIP(I).GT.0) GO TO 80
      WRITE(6,70) ID(I),UNAME(I),STOR(I,13),STOR(I,14),USED(I),
     1 WBRV(I),RFWB(I),BALN(I),ADJU(I),EXCS(I),TCRYO(I)
   70 FORMAT( 4X,I8,1X,A14,9(2X,F9.1))
   80 CONTINUE
      IF(IRET.GT.0) GO TO 10
      FTAB=FTAB+1.0
      WRITE(6,20) (STOT(JT,K),K=1,9)
   20 FORMAT(/13X,5HTOTAL,9X,9(2X,F9.1))
      ICONT=0
      RETURN
   10 ICONT=1
      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-00006914	TABL2

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
6-0001E958	R*4	FTAB	1-00021684	I*4	IE	**	I*4	JT	6-0001E960	CHAR	RUNDATE
1-0002168C	I*4	I	**	I*4	IRET	1-00021680	I*4	K			
1-00021688	I*4	IB	6-0001E950	I*4	IT	**	I*4	KB			
6-0001E95C	I*4	ICONT	6-0001E954	I*4	IYR	**	I*4	KE			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
6-0001C278	R*4	DIFF	1800	(450)
6-0000EF88	R*4	EXCS	1800	(450)
6-0001C9AC	I*4	ID	1800	(450)
6-0001BB70	I*4	ISKIP	1800	(450)
6-0001C980	I*4	NDEND	44	(11)
7-00000000	R*4	RCHNM	160	(5, 8)
6-0000FD98	R*4	RFWB	1800	(450)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
6-0000DA70	R*8	TCRYO	3600	(450)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
**	10	4-00007034	80	4-00006B48	100	**	120	**	130

```
      SUBROUTINE TABL3 (KB,KE,IRET,JT)
      COMMON CTOT(10,15),STOR(450,15),STOT(10,9),TCRYO(450)
      COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
      COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
      COMMON ID(450),UNAME(450)
      COMMON IT,IYR,FTAB,ICONT,RUNDATE
      COMMON /RN/RCHNM(5,8)
      CHARACTER*9 RUNDATE
      CHARACTER*14 UNAME
      REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
      WRITE(6,160)
  160 FORMAT(1H1)
      WRITE(6,150) RUNDATE,IYR
  150 FORMAT(/123X,A9/30X,I4, 66H WATER DISTRICT 1 RESERVOIR CARRYOVER S
     1TORAGE BY USER  (ACRE-FEET)/)
      WRITE(6,145) (RCHNM(I,JT),I=1,5)
  145 FORMAT(54X,5A4/)
      WRITE(6,95)
   95 FORMAT(1H ,13HNUMBER   USER,11X, 93HCARRYOVER JACKSON PALISADES PA
     1L USR  PAL WWS HNRYS L ISL PRK GRSSY L   RIRIE AM FALLS WALCOTT/)
      IB=1
      IF(KB.EQ.0) GO TO 100
      IB=NDEND(KB)+1
  100 IE=NDEND(KE)
      DO 80 I=IB,IE
      DO 60 K=1,10
      CTOT(JT,K)=CTOT(JT,K)+STOR(I,K)
   60 CONTINUE
      WRITE(6,70) ID(I),UNAME(I),TCRYO(I),(STOR(I,K),K=1,10)
   70 FORMAT( 1X,I8,1X,A14,F10.1,2F9.1,F8.1,F9.1,4F8.1,F9.1,F8.1)
   80 CONTINUE
      IF(IRET.GT.0) GO TO 10
      WRITE(6,20) STOT(JT,9),(CTOT(JT,K),K=1,10)
   20 FORMAT(/10X,5HTOTAL,9X,F10.1,2F9.1,F8.1,F9.1,4F8.1,F9.1,F8.1)
   10 CONTINUE
      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ENTRY POINTS

Address	Name
4-00007164	TABL3

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
6-0001E958	R*4	FTAB	1-00021694	I*4	IE	**	I*4	JT	6-0001E960	CHAR	RUNDATE
1-DD02169C	I*4	I	**	I*4	IRET	1-00021690	I*4	K			
1-00021698	I*4	IB	6-0001E950	I*4	IT	**	I*4	KB			
6-0001E95C	I*4	ICONT	6-0001E954	I*4	IYR	**	I*4	KE			

ARRAYS

Address	Type	Name	Bytes	Dimensions
6-0001AD60	R*4	ADJU	1800	(450)
6-0001B468	R*4	BALN	1800	(450)
6-000104A0	R*8	CRYO	43200	(450, 12)
6-00000000	R*8	CTOT	1200	(10, 15)
6-0001C278	R*4	DIFF	1800	(450)
6-0000EF88	R*4	EXCS	1800	(450)
6-0001C9AC	I*4	ID	1800	(450)
6-0001BB70	I*4	ISKIP	1800	(450)
6-0001C980	I*4	NDEND	44	(11)
7-00000000	R*4	RCHNM	160	(5, 8)
6-0000FD98	R*4	RFWB	1800	(450)
6-000004B0	R*8	STOR	54000	(450, 15)
6-0000D7A0	R*8	STOT	720	(10, 9)
6-0000DA70	R*8	TCRYO	3600	(450)
6-0001D0B4	CHAR	UNAME	6300	(450)
6-0000E880	R*4	USED	1800	(450)
6-0000F690	R*4	WBRV	1800	(450)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label
4-000076B4	10	**	60	**	80	4-00007360	100

```
875        BLOCK DATA
876        COMMON /RN/RCHNM(5,8)
877        DATA RCHNM/
878       1 'IRWI','N TO',' LOR','ENZO','    ',
879       2 'LORE','NZO ','TO B','LACK','FOOT',
880       3 'BLAC','KFOO','T TO',' MIL','NER ',
881       4 'MAIN',' STE','M HE','NRYS',' FRK',
882       5 'FALL','S RI','VER ','    ','    ',
883       6 'TETO','N RI','VER ','    ','    ',
884       7 'WILL','OW C','REEK','    ','    ',
885       8 'MISC','ELLA','NEOU','S   ','    '/
886        END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
1	BSS	136864	NOPIC CON REL LCL NOSHR NOEXE RD WRT *64
2	$DATA$	8480	NOPIC CON REL LCL NOSHR NOEXE RD WRT OCTA
3	$LITERAL$	142	PIC CON REL LCL SHR NOEXE RD NOWRT OCTA
4	$CODE$	30444	PIC CON REL LCL SHR EXE NORD NOWRT OCTA
5	$LINK$	1000	NOPIC CON REL LCL NOSHR NOEXE RD NOWRT OCTA
6	$BLANK	125289	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
7	RN	160	NOPIC OVR REL GBL NOSHR NOEXE RD WRT OCTA
	Total Space Allocated	302379	

ARRAYS

Address	Type	Name	Bytes	Dimensions
7-00000000	R*4	RCHNM	160	(5, 8)

```
COMMAND QUALIFIERS

  /ALIGNMENT=(COMMONS=(NONATURAL,PACKED,NOSTANDARD,NOMULTILANGUAGE),RECORDS=NATURAL,NOSEQUENCE)
  /ANNOTATIONS=(NOCODE,NODETAIL,NOFEEDBACK,NOINLINING,NOLOOP_TRANSFORMS,NOLOOP_UNROLLING,NOPREFETCHING,NOSHRINKWRAPPING,
           NOSOFTWARE_PIPELINING,NOTAIL_CALLS,NOTAIL_RECURSION)
  /ARCHITECTURE=GENERIC
  /ASSUME=(ACCURACY_SENSITIVE,ALTPARAM,NOBUFFERED_IO,NOBYTERECL,NODUMMY_ALIASES,NOF77RTL,NOFP_CONSTANT,NOINT_CONSTANT,
           NOMINUS0,PROTECT_CONSTANTS,NOSOURCE_INCLUDE,NOUNDERSCORE)
  /NOAUTOMATIC
  /NOBY_REF_CALL
  /CCDEFAULT=DEFAULT
  /CHECK=(NOARG_TEMP_CREATED,NOBOUNDS,FORMAT,NOFP_EXCEPTIONS,OUTPUT_CONVERSION,NOOVERFLOW,POWER,NOUNDERFLOW)
  /CONVERT=NATIVE
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /NODEFINE
  /DOUBLE_SIZE=64
  /NOD_LINES
  /ERROR_LIMIT=30
  /NOEXTEND_SOURCE
  /F77
  /NOFAST
  /FLOAT=G_FLOAT
  /GRANULARITY=QUADWORD
  /IEEE_MODE=FAST
  /INTEGER_SIZE=32
  /NOMACHINE_CODE
  /MATH_LIBRARY=ACCURATE
  /NOMODULE
  /NAMES=UPPERCASE
  /OPTIMIZE=(INLINE=NONE,LEVEL=,NOLOOPS,NOPIPELINE,TUNE=GENERIC,UNROLL=0)
  /NOPAD_SOURCE
  /REAL_SIZE=32
  /NORECURSIVE
  /REENTRANCY=NONE
  /ROUNDING_MODE=NEAREST
  /NOSEPARATE_COMPILATION
  /SEVERITY=(WARNING=WARNING)
  /SHOW=(NOINCLUDE,MAP)
  /SOURCE_FORM=FIXED
  /STANDARD=NONE
  /NOSYNCHRONOUS_EXCEPTIONS
  /NOSYNTAX_ONLY
  /NOTIE
  /VMS
  /WARNINGS=(ALIGNMENT,NOARGUMENT_CHECKING,NODECLARATIONS,GENERAL,GRANULARITY,NOIGNORE_LOC,NOTRUNCATED_SOURCE,UNCALLED,
           UNINITIALIZED,NOUNUSED,USAGE)

  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /INCLUDE=FORT$INCLUDE:
  /LIST=HYDRO:[BSUTTER.SNAKE]SNKSTO.LIS;20
  /OBJECT=HYDRO:[BSUTTER.SNAKE]SNKSTO.OBJ;1
  /NOLIBRARY

COMPILER: Compaq Fortran V7.5-1961-48BCD
```

COMPILATION STATISTICS

CPU time:	0.61 seconds
Elapsed time:	2.17 seconds
Pagefaults:	2329
I/O Count:	66

```
00001         COMMON CTOT(10,15),STOR(450,15),ID(450),UNAME(450),STOT(10,9)
00002         COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
00003         COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
00004         COMMON TCRYO(450)
0D005         COMMON IT,IYR,FTAB,ICONT,RUNDATE
00006         COMMON /RN/RCHNM(5,8)
00007         DIMENSION KREND(11),FILL(12),SPACE(12),YIELD(12),EVAP(12)
D0008         DIMENSION PYLD(9),T(4),RNAME(12),BANK(450),GSTOT(9),JID(50)
00009         DIMENSION ISK(30),IOR(11),SPAC(450,12),RDIFF(450),GTOT(13)
00010         DIMENSION JDFL(450),TSPAC(12),ID2(450),PAVAIL(9)
00011         DIMENSION IK(450),NSTO(450,15),TOTR(450,14),CNSTO(12),CTOTC(12)
00012         DIMENSION FILLATE(12),TFILL(9)
00013         CHARACTER*9 RUNDATE
00014         CHARACTER*14 UNAME
00015         CHARACTER*14 RNAME
00016         CHARACTER*14 U,RNAM
00017         DATA KREND/13038150,13038500,13046450,13049500,13050590,13055351,
00018        1 13057260,13058550,13069500,13088000,99999999/
00019         DATA ISK/13010500,13032450,13039000,13042000,13042600,13046500,
00020        1 13057950,13059050,13076500,13076751,13081000,13081400,18*0/
00021   C
00022   C***** THE FOLLOWING DATA STATEMENT DICTATES THE ORDER FROM WHICH THE
00023   C***** STORAGE IS REMOVED FROM RESERVOIRS.
00024   C*****
00025   C*****    1 - JACKSON LAKE      2 - PALISADES       3 - PAL WATER USERS
00026   C*****    4 - PAL WINTER WATER  5 - HENRYS LAKE     6 - ISLAND PARK
00027   C*****    7 - GRASSY LAKE       8 - RIRIE           9 - AMERICAN FALLS
00028   C*****   10 - LAKE WALCOT
00029   C*****   11 - FIRST 2D0,DD0 ACRE FEET IN JACKSON LAKE
00030   C
00031         DATA IOR/6,7,11,10,9,3,4,2,1,5,8/
00032         REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
00033         REAL*8 FILL,SPACE,YIELD,GTOT,PYLD,T,GSTOT,TSPAC,GRTOT,SGRTOT
00034         REAL*8 CTOTC,CNSTO,NSTO,BANK,SPAC,FILLATE,TFILL,TLATE,JDFL
00035         CALL DATE(RUNDATE)
00036         CALL ASSIGN(1,'INDIN')
00037         CALL ASSIGN(5,'CARRYIN')
00038         CALL ASSIGN(6,'REPORTOUT')
00039         CALL ASSIGN(7,'SPACEIN')
00040         CALL ASSIGN(9,'USEIN')
00041         L=1
00042         I=1
00043      30 READ(7,10,END=20) ID(I),IK(I),UNAME(I),(STOR(I,K),K=1,12)
00044      10 FORMAT(I8,A1,A14,12F9.0)
00D45         STOR(I,13)=0.0
00046         STOR(I,14)=0.0
00047         STOR(I,15)=0.0
00048         IF(ID(I).EQ.99999300) IPALU=I
00049         IF(ID(I).EQ.13087000) JNSTF=I
00050         IF(ID(I).EQ.13087500) JSSTF=I
00051         IF(ID(I).LT.KREND(L)) GO TO 40
00052         NDEND(L)=I-1
00053         L=L+1
00054      40 I=I+1
00055         GO TO 30
00056      20 CONTINUE
00057         NDEND(L)=I-1
```

AS PER TONY'S COMMENT, CHANGE SO DON'T HAVE TO ENTER LATE FILL EACH TIME

```
00058         JEN=NDEND(10)
00059         JE=NDEND(11)
00060         DO 60 I=1,JE
00061         DO 60 K=1,12
00062         CRYO(I,K)=0.0
00063      6D SPAC(I,K)=STOR(I,K)
00064         DO 45 I=1,JE
00065         READ(5,12,END=5) ID2(I),U,(CRYO(I,J),J=1,12)
00066        12 FORMAT(I8,1X,A14,12F9.0)
00067         IF(ID2(I).EQ.ID(I)) GO TO 45
00068         TYPE 2, ID2(I),ID(I)
00069       2 FORMAT(' DIVERSION ',I8,' FROM SNKSTO.CRY IS DIFFERENT THAN DIVERS
00070        1ION ',I8,' FROM SNKSTO.SPA')
00071         STOP
00072        45 CONTINUE
00073       5 TYPE *, ' '
00074         TYPE *, '  ENTER FOUR DIGIT YEAR FOR WHICH YOU WANT    '
00075         TYPE *, '  TO COMPUTE STORAGE ALLOCATION AND USE.      '
00076         TYPE *, '  FOR EXAMPLE: 1995                           '
00077         PRINT 46
00D78        46 FORMAT(15X,' '$)
00079         ACCEPT 49,IYR
00080        49 FORMAT(I4)
00081         READ(1,55) (RNAME(K),SPACE(K),FILL(K),EVAP(K),FILLATE(K),K=1,12)
00082        55 FORMAT(9X,A14,2X,4F10.0)
00083    C
00084    C***** COMPUTE AND PRINT TABLE OF EACH RESERVOIR'S SPACE, FILL,
00085    C***** EVAPORATION AND YIELD.
00086    C
00087         DO 90 K=1,12
00088         YIELD(K)=FILL(K)-EVAP(K)
00089         IF(FILL(K).GT.0.0) PAVAIL(K)=YIELD(K)/FILL(K)
00090         IF(FILL(K).LE.0.0) PAVAIL(K)=0.0
00091        90 CONTINUE
00092         WRITE(6,100) RUNDATE,IYR
00093       100 FORMAT(1H1//////115X,A9////////////////,43X,I4,
00094          1  ' SNAKE RIVER STORED WATER BY RESERVOIR'//55X,' (ACRE-FEET)'/)
00095         WRITE(6,101)
00096       101 FORMAT(32X,9HRESERVOIR,14X,5HSPACE, 8X,4HFILL, 1X,11HEVAPORATION,
00097          1  7X,5HYIELD/)
00098         DO 105 K=1,4
00099       105 T(K)=0.0
00100         DO 110 K=1,12
00101         T(1)=T(1)+SPACE(K)
00102         T(2)=T(2)+FILL(K)
00103         T(3)=T(3)+EVAP(K)
00104         T(4)=T(4)+YIELD(K)
00105         WRITE(6,120) RNAME(K),SPACE(K),FILL(K),EVAP(K),YIELD(K)
00106       120 FORMAT(32X,A14,2X,4F12.1)
00107         IF(K.EQ.11) WRITE(6,125) (T(I),I=1,4)
00108       125 FORMAT(/32X,9HSUB TOTAL,7X,4F12.1/)
00109       110 CONTINUE
00110         WRITE(6,130) (T(K),K=1,4)
```

```
00111     130 FORMAT(/32X,5HTOTAL,11X,4F12.1)
00112         IT=1
00113     185 CONTINUE
00114         DO 155 I=1,10
00115         DO 155 K=1,15
00116     155 CTOT(I,K)=0.0
00117         CALL TABLE (0,1,1,1)
00118         CALL TABLE (1,2,0,1)
00119         CALL TABLE (6,7,1,2)
00120         CALL TABLE (8,9,0,2)
00121         CALL TABLE (9,10,0,3)
00122         CALL TABLE (2,3,1,4)
00123         CALL TABLE (4,5,0,4)
00124         CALL TABLE (3,4,0,5)
00125         CALL TABLE (5,6,0,6)
00126         CALL TABLE (7,8,0,7)
00127         CALL TABLE (10,11,0,8)
00128         WRITE(6,235) RUNDATE,IT
00129     235 FORMAT(1H1/////123X,A9///58X,4HPART,I2)
00130         IF(IT.EQ.1) WRITE(6,240) IYR
00131     240 FORMAT(1H //15X,I4,98H WATER DISTRICT 1 RESERVOIR SPACE BY REACH A
00132        1SSUMING FULL RESERVOIRS WITHOUT WATER BANK (ACRE-FEET)/)
00133         IF(IT.EQ.2) WRITE(6,250) IYR
00134     250 FORMAT(  //12X,I4,108H WATER DISTRICT 1 RESERVOIR STORAGE BY REACH
00135        1 AFTER PERCENT FILL AND EVAPORATION, WITH WATER BANK (ACRE-FEET)/)
00136         DO 195 K=1,15
00137     195 GTOT(K)=0.0
00138         WRITE(6,210)
00139     210 FORMAT(///// 8X,11HRIVER REACH,9X, 94H    JACKSON     PALISADES
0014D        1   PAL USR        PAL WWS    HENRYS LAKE    ISLAND PARK    GRASSY LAKE
00141        2 /)
00142         DO 190 I=1,8
00143         DO 205 K=1,15
00144     205 GTOT(K)=GTOT(K)+CTOT(I,K)
00145     190 WRITE(6,220) (RCHNM(K,I),K=1,5),(CTOT(I,J),J=1,7)
00146     220 FORMAT(8X,5A4,7(F10.1,4X))
00147         WRITE(6,230) (GTOT(K),K=1,7)
00148     230 FORMAT(/8X,5HTOTAL,15X,7(F10.1,4X))
00149         WRITE(6,211)
00150     211 FORMAT(///// 8X,11HRIVER REACH,5X, 98H          RIRIE      AM FALLS
00151        1  LAKE WALCOTT       AMF LTF       PAL LTF    WATER BANK          T
00152        2OTAL/)
```

```
00153         DO 212 I=1,8
00154       212 WRITE(6,220) (RCHNM(K,I),K=1,5),(CTOT(I,J),J=8,12),
00155          1  (CTOT(I,J),J=14,15)
00156           WRITE(6,230) (GTOT(K),K=8,12),(GTOT(K),K=14,15)
00157           IF (IT.EQ.2) GO TO 200
00158           DO 232 K=1,12
00159       232 TSPAC(K)=GTOT(K)
00160     C
00161     C***** MOVE AS MUCH WWS CARRYOVER INTO MAIN PALISADES ACCOUNT AS
00162     C***** POSSIBLE AS PER USBR METHOD OF ALLOCATING NEW STORAGE - NEW
00163     C***** FILL IN WWS SPACE THEN DOES NOT INCLUDE CARRYOVER.
00164     C
00165     C***** THIS ROUTINE NOT USED BEGINNING IN 1992 BECAUSE IT PRODUCED
00166     C***** NEGATIVE ALLOCATIONS WHEN THERE WAS LITTLE OR NO NEW FILL
00167     C***** OF PALISADES REGULAR SPACE.  NOT SURE OF EXACT REASONING
00168     C***** WHEN THIS WAS FIRST USED.
00169     C
00170     C        DO 25 I=1,JE
00171     C        K=2
00172     C        IF(I.EQ.IPALU) K=3
00173     C        CRYO(I,K)=CRYO(I,K)+CRYO(I,4)
00174     C        CRYO(I,4)=0.0
00175     C        IF(CRYO(I,K).GT.SPAC(I,K)) THEN
00176     C            CRYO(I,K)=SPAC(I,K)
00177     C            CRYO(I,4)=CRYO(I,K)-SPAC(I,K)
00178     C            END IF
00179     C    25 CONTINUE
00180     C
00181     C***** INITIALIZE NEW STORAGE (NSTO) TO ZERO AND TOTAL CARRYOVER
00182     C***** BY RESERVOIR ACCOUNT.
00183     C
00184           DO 14 K=1,12
00185           CTOTC(K)=0.0
00186           DO 14 I=1,JE
DD187           NSTO(I,K)=0.0
00188        14 CTOTC(K)=CTOTC(K)+CRYO(I,K)
00189           IJKSN=0
00190     C
00191     C***** BEGIN LOOP TO ALLOCATE NEW FILL TO THE TWELVE RESERVOIR
00192     C***** ACCOUNTS.
00193     C
00194           DO 1 K=1,12
00195           KRS=K
00196           IF(K.GT.3) KRS=K-1
00197           TSP=TSPAC(K)
00198     C
00199     C***** COMBINE PALISADES AND PALISADES WATER USERS.
00200     C
00201           IF(K.EQ.2) TSP=TSPAC(2)+TSPAC(3)
00202           IF(TSP.LE.0.0) GO TO 1
00203           FIL=FILL(KRS)
00204           RNAM=RNAME(KRS)
00205           IF(K.EQ.3) THEN
00206                FIL=PUTSTO
00207                RNAM='PALISADES USRS'
00208                END IF
```

```
00209         IF(FIL.LE.TSP) GO TO 13
00210         TYPE 997, RNAM,FIL,TSP
00211     997 FORMAT(1X,A16,' HAS FILL =',F10.1,' > TOTAL SPACE =',F10.1)
00212         GO TO 400
00213      13 IF(FIL.EQ.TSP) GO TO 23
00214    C
00215    C***** ZERO CARRYOVER IF FAILURE TO FILL WAS RESULT OF FLOOD OPERATION
00216    C***** OR OTHER RESERVOIR OPERATIONS.
00217    C
00218         PRINT 22, RNAM,FIL,TSP
00219      22 FORMAT(1X,A16,' HAS FILL =',F10.1,' < TOTAL SPACE =',F10.1/
00220        1'   IS THE FAILURE TO FILL A RESULT OF FLOOD CONTROL? '/
00221        2'   OR DO SPACEHOLDERS NOT HAVE CARRYOVER PRIVILEGES? '/
00222        3'   IF YOU ANSWER Y, CARRYOVER WILL BE WIPED OUT. Y/N '$)
00223         ACCEPT 36, FLD
00224         IF(FLD.NE.'Y') GO TO 24
00225      23 KC=K
00226      27 DO 26 I=1,JE
00227      26 CRYO(I,KC)=0.0
00228         CTOTC(KC)=0.0
00229         IF(KC.EQ.2) THEN
00230                     KC=KC+1
00231                     GO TO 27
00232                     END IF
00233    C
00234    C***** COMPUTE NEW FILL TO EACH RESERVOIR BY SUBTRACTING CARRYOVER
00235    C***** FROM TOTAL FILL. THAT AMOUNT (WAT) IS THEN DISTRIBUTED.
00236    C
00237      24 TCR=CTOTC(K)
00238         IF(K.EQ.2) TCR=CTOTC(2)+CTOTC(3)
00239         CNSTO(K)=FIL-TCR
00240         WAT=CNSTO(K)
00241         IF(WAT.LE.-0.5) THEN
00242           PRINT 47, RNAM,TCR,FIL
00243      47 FORMAT(1X,A16,' HAS TOTAL CARRYOVER =',F10.1,' > TOTAL FILL=',
00244        1F10.1/'   DO YOU WISH TO STOP? ANSWER Y/N '$)
00245           ACCEPT 36, STP
00246           IF(STP.EQ.'Y') GO TO 400
00247           ENDIF
00248         J=0
00249    C
00250    C***** IF ALLOCATING JACKSON LAKE, REMOVE LAST TO FILL SPACE OF NORTH
00251    C***** AND SOUTH SIDE TWIN FALLS CANALS (TJTFSP) AND ALSO CORRESPONDING
00252    C***** CARRYOVER (TJTFCO) FROM TOTAL SPACE (TSP) AND TOTAL FIL (FIL),
00253    C***** RESPECTIVELY.  ALSO CORRECT TOTAL CARRYOVER (TCR).  IF REVISED
00254    C***** FILL (FIL) IS GREATER THAN REVISED TOTAL SPACE (TSP), THEN
00255    C***** SPACE OTHER THAN TWIN FALLS JACKSON SPACE HAS FILLED COMPLETELY,
00256    C***** SO SET TOTAL FILL EQUAL TO TOTAL SPACE.
00257    C
00258         IF(K.GT.1) GO TO 6
00259         IF(IJKSN.GT.0.OR.FIL.EQ.TSP) GO TO 6
00260         IF(FLD.NE.'Y') THEN
00261              IJKSN=1
00262              TJNSTFSP=SPAC(JNSTF,1)
00263              TJSSTFSP=SPAC(JSSTF,1)
00264              SPAC(JNSTF,1)=0.0
00265              SPAC(JSSTF,1)=0.0
```

```
00266               TJTFSP=TJNSTFSP+TJSSTFSP
00267               TJNSTFCO=CRYO(JNSTF,1)
00268               TJSSTFCO=CRYO(JSSTF,1)
00269               TJTFCO=TJNSTFCO+TJSSTFCO
00270               CRYO(JNSTF,1)=0.0
00271               CRYO(JSSTF,1)=0.0
00272               TSP=TSP-TJTFSP
00273               FIL=FIL-TJTFCO
00274               TCR=TCR-TJTFCO
00275               IF(FIL.GE.TSP) THEN
00276                   JKTFNSTO=FIL-TSP
00277                   FIL=TSP
00278                   GO TO 23
00279                   ENDIF
00280               JKTFNSTO=0.0
00281               ENDIF
00282     C
00283     C***** SET TOTAL NEW STORAGE (TNSTOR) ASSIGNED TO SPACEHOLDERS EQUAL
00284     C***** TO ZERO.  KEEP TRACK OF NUMBER OF PASSES MADE TO DISTRIBUTE
00285     C***** ALL NEW STORAGE (CNSTO(K)) AND STOP IF IT GOES WILD.
00286     C
00287         6 TNSTOR=0.0
00288           J=J+1
00289           IF(J.GT.2000) TYPE 998, RNAM
00290       998 FORMAT(' RUNAWAY LOOP ALLOCATING ',A16,' STORAGE')
00291           TYPE 999, J,WAT
00292       999 FORMAT(1X,I5,F12.4)
00293     C
00294     C***** USE TOTAL RESERVOIR SPACE FOR ANY OF THE AMERICAN FALLS OR
00295     C***** PALISADES ACCOUNTS, SINCE THOSE ARE ALL PROPORTIONAL TO
00296     C***** TOTAL SPACE AS DESCRIBED BELOW.
00297     C
00298           IF(FLD.NE.'Y') THEN
00299                IF(K.EQ.2.OR.K.EQ.4.OR.K.EQ.12) TSP=1200000.0
00300                IF(K.EQ.9.OR.K.EQ.11) TSP=1672590.0
00301                ENDIF
00302           KS=K
00303           DO 4 I=1,JE
00304           SPC=SPAC(I,KS)
00305     C
00306     C***** SUM ALL SPACE (SPC) ACCOUNTS FOR PALISADES AND AMERICAN FALLS
00307     C***** TO DETERMINE PROPORTIONAL FILL BECAUSE USBR CONTRACTS READ
00308     C***** AS A PERCENT DETERMINED FROM ALL STORAGE IN A RESERVOIR, NOT
00309     C***** FROM EACH RESERVOIR RIGHT.  FOR PALISADES USERS (IPALU=1) BE
00310     C***** SURE TO ADD ALL REGULAR PALISADES USERS SPACE (TSPAC(3)).
00311     C
00312           IF(FLD.NE.'Y') THEN
00313              IF(KS.EQ.2.OR.KS.EQ.4.OR.KS.EQ.12) THEN
00314                             SPC=SPAC(I,2)+SPAC(I,4)+SPAC(I,12)
00315                             IF(KS.EQ.12.AND.I.EQ.IPALU)
00316           1                 SPC=SPAC(I,4)+SPAC(I,12)+TSPAC(3)
00317                             END IF
00318              IF(KS.EQ.9.OR.KS.EQ.11) SPC=SPAC(I,9)+SPAC(I,11)
00319              ENDIF
00320     C
00321     C***** ALLOCATE NEW FILL (WAT) TO INDIVIDUAL ENTITIES AND USES WITHIN
00322     C***** EACH RESERVOIR PROPORTIONAL TO SPACE OWNED.  LIMIT TOTAL FILL
```

```
00323   C***** (NEW FILL PLUS CARRYOVER) OF EACH SPACEHOLDER (STOR) TO SPACE
00324   C***** OWNED (SPAC).  KEEP TRACK OF TOTAL STORAGE ASSIGNED TO SPACE-
00325   C***** HOLDERS (TNSTOR).
00326   C
00327         IF(KS.EQ.11.OR.KS.EQ.12) THEN
00328              IF(FIL.EQ.TSPAC(KS)) GO TO 94
00329              ENDIF
00330         IF(CRYO(I,KS).GT.SPC) TYPE 996, UNAME(I)
00331     996 FORMAT(' CARRYOVER STORAGE FOR ',A14,' GREATER THAN SPACE')
00332         NSTO(I,KS)=((SPC/TSP)*WAT)+NSTO(I,KS)
00333         STOR(I,KS)=NSTO(I,KS)+CRYO(I,KS)
00334         IF(STOR(I,KS).LE.SPAC(I,KS)) GO TO 95
00335      94 STOR(I,KS)=SPAC(I,KS)
00336         NSTO(I,KS)=STOR(I,KS)-CRYO(I,KS)
00337      95 TNSTOR=TNSTOR+NSTO(I,KS)
00338       4 CONTINUE
00339   C
00340   C***** WHEN FINISHED WITH PALISADES REGULAR ACCOUNTS, THEN DETERMINE
00341   C***** PALISADES WATER USERS FILL COLLECTIVELY AS ONE ACCOUNT.
00342   C
00343         IF(KS.EQ.2) THEN
00344                     KS=KS+1
00345                     SPC=TSPAC(3)
00346         IF(FLD.NE.'Y') SPC=SPAC(IPALU,4)+SPAC(IPALU,12)+TSPAC(KS)
00347                     PUNSTO=((SPC/TSP)*WAT)+PUNSTO
00348                     PUTSTO=PUNSTO+CTOTC(KS)
00349                     IF(PUTSTO.GT.TSPAC(KS)) THEN
00350                                             PUTSTO=TSPAC(KS)
00351                                             PUNSTO=PUTSTO-CTOTC(KS)
00352                                             END IF
00353                     TNSTOR=TNSTOR+PUNSTO
00354                     END IF
00355   C
00356   C***** SUBTRACT TOTAL STORAGE ALLOCATED FROM TOTAL NEW FILL TO GET
00357   C***** AMOUNT REMAINING TO BE ALLOCATED (WAT).
00358   C
00359         WAT=CNSTO(K)-TNSTOR
00360         IF(WAT.GT.0.07) GO TO 6
00361   C
00362   C***** IF JACKSON LAKE SPACE DID NOT COMPLETELY FILL AND THERE WAS NO
00363   C***** FLOOD CONTROL, THEN IJCKSN WILL EQUAL 1 AND THE FILL OF THE
00364   C***** TWIN FALLS SPACE MUST NOW BE CALCULATED SINCE IT FILLS LAST
00365   C***** WITHIN JACKSON LAKE.
00366   C
00367         IF(K.GT.1) GO TO 1
00368         IF(IJKSN.EQ.1) THEN
00369             J=0
00370             WAT=JKTFNSTO
00371             SPAC(JNSTF,1)=TJNSTFSP
00372             SPAC(JSSTF,1)=TJSSTFSP
00373             CRYO(JNSTF,1)=TJNSTFCO
00374             CRYO(JSSTF,1)=TJSSTFCO
00375      98     TNSTOR=0.0
00376             J=J+1
00377             IF(J.GT.2000) TYPE 998, RNAM
00378             TYPE 999, J,WAT
00379             DO 97 I=JNSTF,JSSTF
```

```
0038D             SPC=SPAC(I,1)
00381             NSTO(I,1)=((SPC/TJTFSP)*WAT)+NSTO(I,1)
00382             STOR(I,1)=NSTO(I,1)+CRYO(I,1)
00383             IF(STOR(I,1).LE.SPAC(I,1)) GO TO 96
00384             STOR(I,1)=SPAC(I,1)
00385             NSTO(I,1)=STOR(I,1)-CRYO(I,1)
00386      96     TNSTOR=TNSTOR+NSTO(I,1)
00387      97     CONTINUE
00388             WAT=JKTFNSTO-TNSTOR
00389             IF(WAT.GT.0.1) GO TO 98
00390             END IF
00391       1 CONTINUE
00392   C
00393   C***** TRANSFER WATER BANK TRANSFERS TO TABLE ARRAY,
00394   C***** COMPUTE YIELD OF SPACE VALUES AFTER EVAPORATION.
00395   C
00396         PRINT 3
00397       3 FORMAT(' DO YOU WANT TO READ STORAGE USE, ADJUSTMENTS, AND/OR'/
00398        1' WATER BANK TRANSACTIONS BY CANAL/USER? Y/N: '$)
00399         ACCEPT 36, USE
00400      36 FORMAT(A1)
00401         DO 165 I=1,JE
00402         BANK(I)=0.0
00403         IF(USE.NE.'Y') GO TO 320
00404         READ(9,260,END=35) JD,USED(I),ADJU(I),BANK(I),RFWB(I)
00405     260 FORMAT(I8,15X,4F10.0)
00406         IF(JD.EQ.ID(I)) GO TO 320
00407      35 WRITE(6,265) JD,ID(I)
00408     265 FORMAT(15H DIVERSION NO. ,I10,50H FROM SNKSTO'YR'.USE DOES NOT EQU
00409        1AL DIVERSION NO. ,I10,25H FROM SNKSTO'YR'.SPA LIST)
00410         GO TO 400
00411     320 STOR(I,14)=BANK(I)
00412         DO 175 K=1,12
00413         KS=K
00414         IF(K.GE.3) KS=K-1
00415     175 STOR(I,K)=STOR(I,K)*PAVAIL(KS)
00416     165 CONTINUE
00417         IT=2
00418         GO TO 185
00419     200 PRINT *,' '
00420         PRINT 201
00421     201 FORMAT(' DO YOU WISH TO CREATE A FILE (SNKCAN.IND) OF STORED WATER
00422        1 SUPPLY'/' BY DIVERSION FOR INPUT TO ACCOUNTING PROGRAM? Y/N: '$)
00423         ACCEPT 36, FILE
00424         IF(FILE.NE.'Y') GO TO 204
00425         OPEN(UNIT=2,NAME='SNKCAN.IND',TYPE='NEW',CARRIAGECONTROL='LIST')
00426         DUM=0.0
00427         DO 202 I=1,JEN
00428     202 WRITE(2,203) ID(I),DUM,STOR(I,15)
```

```
00429     203 FORMAT(I8,2F10.1)
00430     204 CONTINUE
00431   C
00432   C***** SET TABLE NUMBER TO BEGIN OUTPUT OF IRRIGATION SEASON
00433   C***** STORED WATER TRANSACTIONS.
00434   C
00435         IF(USE.NE.'Y') GO TO 400
00436         FTAB=21.0
00437      33 PRINT 28, FTAB
00438      28 FORMAT(' THE FIRST STORAGE TABLE FOR REPORT WILL BE NO. ',
00439        1 F5.1/' IS THIS OK? Y/N/Q '$)
00440         ACCEPT 36, TBL
00441         IF(TBL.EQ.'Q') GO TO 400
00442         IF(TBL.EQ.'Y') GO TO 310
00443         TYPE *, ' '
00444         PRINT 29
00445      29 FORMAT(' ENTER NEW BEGINNING TABLE NUMBER = '$)
00446         ACCEPT 32, FTAB
00447      32 FORMAT(F5.0)
00448         TYPE *, ' '
00449         GO TO 33
00450     310 IF(FTAB.LE.0.0) GO TO 400
00451         K=1
00452         DO 280 I=1,JE
00453   C
00454   C***** MOVE MILNER STORAGE TO MISCELLANEOUS TABLE SINCE STORAGE
00455   C***** IS NOT ALLOCATED TO ANY USER
00456   C
00457         IF(ID(I).EQ.99999950) THEN
00458                               STOR(I,13)=STOR(I,13)+YIELD(12)
00459                               JM=I
00460                               END IF
00461         ISKIP(I)=0.0
00462         IF(ISK(K).NE.ID(I)) GO TO 280
00463         ISKIP(I)=1
00464         K=K+1
00465     280 CONTINUE
00466         DO 315 I=1,JE
00467         EXCS(I)=0.0
00468         TCRYO(I)=0.0
00469         WBRV(I)=0.0
00470         BALN(I)=STOR(I,14)-USED(I)
00471         IF(BALN(I).GT.0.0) GO TO 275
00472     295 BALN(I)=BALN(I)+STOR(I,13)
00473         GO TO 290
00474     275 WBRV(I)=BALN(I)
00475         BALN(I)=STOR(I,13)
00476     290 BALN(I)=BALN(I)+RFWB(I)
00477   C
00478   C***** IF THERE IS A NEGATIVE ADJUSTMENT TO STORAGE USE (ADJU), THAT
00479   C***** AMOUNT IS FIRST DEDUCTED FROM THE WATER REVERTING TO THE WATER
00480   C***** BANK (WBRV), FOR ALL MISCELLANEOUS STORAGE ACCOUNTS.
00481   C
00482         IF(ADJU(I).GE.0.0.OR.WBRV(I).LE.0.0.OR.ID(I).LT.90000000) GO TO
00483        1 285
00484         ADD=WBRV(I)
00485         WBRV(I)=WBRV(I)+ADJU(I)
```

```
00486          IF(WBRV(I).LT.0.0) WBRV(I)=0.0
00487          ADD=ADD-WBRV(I)
00488          BALN(I)=BALN(I)+ADD
00489      285 TCRYO(I)=BALN(I)+ADJU(I)
00490          IF(TCRYO(I).GE.0.0) GO TO 315
00491          EXCS(I)=TCRYO(I)*(-1.0)
00492          TCRYO(I)=0.0
00493      315 CONTINUE
00494          DO 455 I=1,8
00495          STOT(I,1)=CTOT(I,13)
00496          STOT(I,2)=CTOT(I,14)
00497          DO 455 K=3,9
00498      455 STOT(I,K)=0.0
00499          STOT(8,1)=STOT(8,1)+YIELD(12)
00500          ICONT=0
00501          CALL TABL2 (0,1,1,1)
00502          CALL TABL2 (1,2,0,1)
00503          CALL TABL2 (6,7,1,2)
00504          CALL TABL2 (8,9,0,2)
00505          CALL TABL2 (9,10,0,3)
00506          CALL TABL2 (2,3,1,4)
00507          CALL TABL2 (4,5,0,4)
00508          CALL TABL2 (3,4,0,5)
00509          CALL TABL2 (5,6,0,6)
00510          CALL TABL2 (7,8,0,7)
00511          CALL TABL2 (10,11,0,8)
00512          WRITE(6,540) RUNDATE,FTAB,IYR
00513      540 FORMAT(1H1/////123X,A9/////18X,5HTABLE,F4.0,2X,20HSUMMARY BY REACH
00514         1 OF ,I4,54H STORED WATER ACCOUNTS IN WATER DISTRICT 1 (ACRE-FEET))
00515          DO 595 K=1,9
00516      595 GSTOT(K)=0.0
00517          WRITE(6,510)
00518      510 FORMAT( /28X,76H            STORAGE OR                          RETUR
00519         1N TO                      /28X,67H            WATER BANK
00520         2 REVERTED TO SPACEHOLDER            /28X,99H   STORAGE   PURCHASE,
00521         3  STORAGE  WATER BANK    FROM                   ADJUST-     EXCESS
00522         4    CARRY-/  4X,13HREACH        ,11X,99H  ALLOCATED SUPPLY (-)
00523         5 USED   FROM USER   WATER BANK   BALANCE       MENT       USED
00524         6   OVER/)
        lines; may be non-portable
```

```
00525         DO 590 I=1,8
00526         DO 505 K=1,9
00527       505 GSTOT(K)=GSTOT(K)+STOT(I,K)
00528       590 WRITE(6,520) (RCHNM(K,I),K=1,5),(STOT(I,J),J=1,9)
00529       520 FORMAT( 4X,5A4,4X,9(2X,F9.1))
00530         WRITE(6,530) (GSTOT(K),K=1,9)
00531       530 FORMAT(/ 3X,6H TOTAL,19X,9(2X,F9.1))
00532     C
00533     C***** COMPUTE NET STORAGE USED (RDIFF).  MOVE LATE POWER STORAGE AT
00534     C***** AMERICAN FALLS AND PALISADES BACK TO ORIGINAL RIGHT SINCE IT
00535     C***** NOW LOSES ITS IDENTITY(NO CARRYOVER IN THIS SPACE.  IF THERE IS
00536     C***** NO STORAGE IN PALISADES REGULAR ACCOUNT (STOR(I,2)), THEN ASSUME
00537     C***** THIS POWER STORAGE CAME FROM PALISADES USERS (STOR(I,3)), SO ADD
00538     C***** IT BACK THERE INSTEAD.  ALSO COMPUTE PROPORTIONAL SPACE BY USER
00539     C***** IN FIRST 200,000 AF OF JACKSON LAKE FOR FIRST USE COMPUTATIONS
00540     C***** BELOW.
00541     C
00542           DO 608 I=1,JE
00543           DIFF(I)=STOR(I,13)-TCRYO(I)
00544           RDIFF(I)=DIFF(I)
00545           IF(STOR(I,2).LE.0.0) THEN
00546                 STOR(I,3)=STOR(I,3)+STOR(I,12)
00547                 GO TO 607
00548                 ENDIF
00549           STOR(I,2)=STOR(I,2)+STOR(I,12)
00550       607 STOR(I,9)=STOR(I,9)+STOR(I,11)
00551           STOR(I,11)=0.0
00552           STOR(I,12)=0.0
00553       6D8 JDFL(I)=200000.*(SPAC(I,1)/TSPAC(1))
00554     C
00555     C***** COMPUTE CARRYOVER BY USER BY RESERVOIR.  WHEN IO=11, PROPORTION
00556     C***** FIRST 200,000 ACRE FEET IN JACKSON LAKE AMONG SPACEHOLDERS
00557     C***** ACCORDING TO SPACE OWNED.
00558     C
00559           DO 615 LO=1,11
00560           IO=IOR(LO)
00561           IF(IO.NE.11) GO TO 613
00562           IO=1
00563           IF(GTOT(1).LE.200000.) GO TO 613
00564           TJL=0.0
00565           PTJL=0.0
00566       606 DO 612 I=1,JE
00567           DF=RDIFF(I)
00568           IF(DF.GT.JDFL(I)) DF=JDFL(I)
00569           IF(DF.LE.0.0) GO TO 612
00570           STOR(I,IO)=STOR(I,IO)-DF
00571           IF(STOR(I,IO).GE.0.0) RDIFF(I)=RDIFF(I)-DF
00572           IF(STOR(I,IO).LT.0.0) RDIFF(I)=RDIFF(I)-DF-STOR(I,IO)
00573           IF(STOR(I,IO).LT.0.0) STOR(I,IO)=0.0
00574           TJL=TJL+(DF-RDIFF(I))
00575       612 CONTINUE
00576           DTJL=TJL-PTJL
00577           IF(DTJL.LE.02) GO TO 615
00578           REMJ=200000.-TJL
00579           IF(REMJ.LE.0.2) GO TO 615
```

```
00580          DO 616 I=1,JE
00581       616 JDFL(I)=REMJ*(SPAC(I,1)/TSPAC(1))
00582          PTJL=TJL
00583          GO TO 606
00584       613 CONTINUE
00585          DO 614 I=1,JE
00586          DF=RDIFF(I)
00587          IF(DF.LE.0.0) GO TO 614
00588          STOR(I,IO)=STOR(I,IO)-DF
00589          IF(STOR(I,IO).GE.0.0) RDIFF(I)=RDIFF(I)-DF
00590          IF(STOR(I,IO).LT.0.0) RDIFF(I)=STOR(I,IO)*(-1.0)
00591          IF(STOR(I,IO).LT.0.0) STOR(I,IO)=0.0
00592       614 CONTINUE
00593       615 CONTINUE
00594     C
00595     C***** PLACE FREMONT MADISON CARRYOVER ENTIRELY IN ISLAND PARK
00596     C***** SINCE THERE ARE NO INDIVIDUAL CARRYOVER PRIVILEGES
00597     C
00598          K=NDEND(10)+1
00599          DO 618 I=K,JE
00600       618 IF(ID(I).EQ.99999600) STOR(I,6)=TCRYO(I)
00601          DO 655 I=1,10
00602          DO 655 K=1,12
00603       655 CTOT(I,K)=0.0
00604          CALL TABL3 (0,1,1,1)
00605          CALL TABL3 (1,2,0,1)
00606          CALL TABL3 (6,7,1,2)
00607          CALL TABL3 (8,9,0,2)
00608          CALL TABL3 (9,10,0,3)
00609          CALL TABL3 (2,3,1,4)
00610          CALL TABL3 (4,5,0,4)
00611          CALL TABL3 (3,4,0,5)
00612          CALL TABL3 (5,6,0,6)
00613          CALL TABL3 (7,8,0,7)
00614          CALL TABL3 (10,11,0,8)
00615          WRITE(6,652)
00616       652 FORMAT(1H1)
00617          WRITE(6,650) RUNDATE,IYR
00618       650 FORMAT(/123X,A9///35X,I4, 59H WATER DISTRICT 1 RESERVOIR CARRYOVER
00619         1 BY REACH  (ACRE-FEET)/)
00620          DO 695 K=1,12
00621       695 GTOT(K)=0.0
00622          WRITE(6,610)
00623       610 FORMAT(1H ,  14HRIVER REACH   ,6X, 99H  CARRYOVER JACKSON PALISADE
00624         1S  PAL USR   PAL WWS HNRYS L  ISL PRK GRSSY L   RIRIE  AM FALLS WA
00625        2LCOTT/)
00626          DO 690 I=1,8
00627          DO 605 K=1,10
00628       605 GTOT(K)=GTOT(K)+CTOT(I,K)
```

```
00629     690 WRITE(6,620) (RCHNM(K,I),K=1,5),STOT(I,9),(CTOT(I,J),J=1,10)
00630     620 FORMAT(1X,5A4,F10.1,F9.1,F10.1,F9.1,F10.1,F8.1,F9.1,2F8.1,
00631        1 F10.1,F8.1)
00632         WRITE(6,630) GSTOT(9),(GTOT(K),K=1,10)
00633     630 FORMAT(/6H TOTAL,15X,F10.1,F9.1,F10.1,F9.1,F10.1,F8.1,F9.1,
00634        12F8.1,F10.1,F8.1)
00635         GRTOT=0.0
00636         DO 640 I=1,10
00637     640 GRTOT=GRTOT+GTOT(I)
00638         SGRTOT=GRTOT+TCRYO(JE)+TCRYO(JM)
00639         WRITE(6,642)IYR
00640     642 FORMAT(///////     35X,I4, 64H WATER DISTRICT 1 RESERVOIR CARRYOVE
00641        1R - SYSTEM TOTAL (ACRE-FEET)/)
0D642         WRITE(6,644) GRTOT,TCRYO(JE),TCRYO(JM),SGRTOT
00643     644 FORMAT(55X,'ALLOCATED         - ',F10.1/
00644        1       55X,'UNALLOCATED       - ',F10.1/
00645        2       55X,'MILNER RESERVOIR - ',F10.1//
00646        3       55X,'TOTAL SYSTEM      - ',F10.1)
00647         PRINT 16
00648      16 FORMAT(' DO YOU WISH TO ENTER LATE SEASON RESERVOIR FILL? Y/N '$)
00649         ACCEPT 7,ANSWER
00650       7 FORMAT(1A1)
00651         PRINT *,' '
00652         IF(ANSWER.NE.'Y') GO TO 8
00653      15 DO 9 K=1,9
00654         PRINT 11, RNAME(K),FILLATE(K)
00655      11 FORMAT(1X,'LATE SEASON FILL FOR       ',A14,' = ',F10.1,' ENTER NE
00656        1W FILL? Y/N/Q '$)
00657         ACCEPT 7,ANSWER
00658         IF(ANSWER.EQ.'Q') GO TO 18
00659         IF(ANSWER.NE.'Y') GO TO 9
00660         PRINT 19,RNAME(K)
00661      19 FORMAT(1X,'ENTER LATE SEASON FILL FOR ',A14,' = '$)
00662         ACCEPT 21,FILLATE(K)
00663      21 FORMAT(F10.0)
00664       9 CONTINUE
00665      18 PRINT 17
00666      17 FORMAT(' DO YOU WISH TO REENTER LATE SEASON RESERVOIR FILL? Y/N '
00667        1$)
00668         ACCEPT 7,ANSWER
00669         IF(ANSWER.EQ.'Y') GO TO 15
00670       8 WRITE(6,142)IYR
00671     142 FORMAT(//////     35X,I4,' WATER DISTRICT 1 RESERVOIR TOTAL STORAG
00672        1E - OCTOBER 31 (ACRE-FEET)'///43X,'RESERVOIR',12X,'CARRYOVER',
00673        2'   LATE FILL',7X,'TOTAL'/)
```

```
00674          GTOT(2)=GTOT(2)+GTOT(3)
00675          GTOT(3)=TCRYO(JE)+TCRYO(JM)
00676          TLATE=0.0
00677          GTFILL=0.0
00678          DO 146 K=1,10
00679          K1=K
00680          IF(K.GT.2) K1=K+1
00681          IF(K1.EQ.11) K1=3
00682          TFILL(K)=GTOT(K1)+FILLATE(K)
00683          TLATE=TLATE+FILLATE(K)
00684          GTFILL=GTFILL+TFILL(K)
00685          IF(K.EQ.10) GO TO 148
00686          WRITE(6,144) RNAME(K),GTOT(K1),FILLATE(K),TFILL(K)
00687      144 FORMAT(43X,A14,4X,3F12.1)
00688      146 CONTINUE
00689      148 WRITE(6,147) GTOT(K1),FILLATE(K),TFILL(K)
00690      147 FORMAT(43X,'OTHER',13X,3F12.1)
00691          WRITE(6,145) SGRTOT,TLATE,GTFILL
00692      145 FORMAT(/43X,'TOTAL',13X,3F12.1)
00693       38 PRINT 31
00694       31 FORMAT(' DO YOU WANT TO CREATE A FILE OF THE CARRYOVER'/
00695         1' BY CANAL AND RESERVOIR? Y/N: '$)
00696          ACCEPT 36, FILE
00697          IF(FILE.EQ.'N') GO TO 400
00698          IF(FILE.NE.'Y') GO TO 38
00699          OPEN (UNIT=10,NAME='CARRYOUT',TYPE='NEW',CARRIAGECONTROL='LIST')
00700          DO 37 I=1,JE
00701       37 WRITE(10,34) ID(I),UNAME(I),(STOR(I,K),K=1,12)
00702       34 FORMAT(I8,1X,A14,12F9.1)
00703      400 CONTINUE
00704          STOP
00705          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	7182	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	3556	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	112460	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3	$BLANK	125289	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
4	RN	160	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
	Total Space Allocated	248647	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKSTO$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	ADD		484= 487(2)= 488
2-0001B314	R*4	ANSWER		649= 652 657= 658 659 668= 669
**	R*4	DF		567= 568(2)= 569 570 571 572 574 586= 587 588 589
**	R*4	DTJL		576= 577
**	R*4	DUM		426= 428
**	R*4	FIL		203= 206= 209 210 213 218 239 242 259 273(2)= 275 276 277= 328
2-0001B2E8	R*4	FILE		423= 424 696= 697 698
2-0001B2AC	R*4	FLD		223= 224 260 298 312 346
3-0001E958	R*4	FTAB	COMM	5 436= 437 446= 450 512
**	R*8	GRTOT		33 635= 637(2)= 638 642
**	R*4	GTFILL		677= 684(2)= 691
**	I*4	I		42= 43(4) 45 46 47 48(2) 49(2) 50(2) 51 52 54(2)= 57 60= 62 63(2) 64= 65(2) 67(2) 68(2) 107(2)= 114= 116 142= 144 145(2) 153= 154(3) 186= 187 188 226= 227 303= 304 314(3) 315(3) 318(2) 330(2) 332(2) 333(3) 334(2) 335(2) 336(3) 337 379= 380 381(2) 382(3) 383(2) 384(2) 385(3) 386 401= 402 404(4) 406 407 411(2) 415(2) 427= 428(2) 452= 457 458(2) 459 461 462 463 466= 467 468 469 470(3) 471 472(3) 474(2) 475(2) 476(3) 482(3) 484 485(3) 486(2) 487 488(2) 489(3) 490 491(2) 492 494= 495(2) 496(2) 498 525= 527 528(2) 542= 543(3) 544(2) 545 546(3) 549(3) 550(3) 551 552 553(2) 566= 567 568(2) 570(2) 571(3) 572(4) 573(2) 574 580= 581(2) 585= 586 588(2) 589(3) 590(3) 591(2) 599= 600(3) 601= 603 626= 628 629(3) 636= 637 700= 701(3)
3-0001E95C	I*4	ICONT	COMM	5 500=

```
     **       I*4   IJKSN                    189=       259        261=       368
     **       I*4   IO                       560=       561        562=       570(2)     571        572(2)     573(2)     588(2)
                                             589        590(2)     591(2)

  2-0001B28C  I*4   IPALU                     48=        315        346(2)
  3-0001E950  I*4   IT              COMM       5         112=       128        130        133        157        417=
  3-0001E954  I*4   IYR             COMM       5          79=        92        130        133        512        617        639
                                             670
  2-0001B2A0  I*4   J                        65(2)=     145(2)=    154(4)=    248=       288(2)=    289        291        369=
                                             376(2)=    377        378        528(2)=    629(2)=
  2-0001B2E4  I*4   JD                       404=       406        407

     **       I*4   JE                        59=        60         64        186        226        303        401        452
                                             466        542        566        580        585        599        638        642
                                             675        700
     **       I*4   JEN                      58=        427
  2-0001B2CC  I*4   JKTFNSTO                276=       280=       370        388
  2-0001B2F8  I*4   JM                       459=       638        642        675
  2-0001B290  I*4   JNSTF                    49=        262        264        267        270        371        373        379

  2-0001B294  I*4   JSSTF                    50=        263        265        268        271        372        374        379
  2-0001B288  I*4   K                        43(2)=      61=        62         63(2)      81(6)=     87=        88(3)      89(4)
                                              90(2)      98=        99        100=       101        102        103        104
                                             105(5)     107        110(2)=    115=       116        136=       137        143=
                                             144(3)     145(2)=    147(2)=    154(2)=    156(4)=    158=       159(2)     184=
                                             185        187        188(3)     194=       195        196(2)     197        201
                                             205        225        237        238        239        240        258        299(3)
                                             300(2)     302        359        367        412=       413        414(2)     415(2)
                                             451=       462        464(2)=    497=       498        515=       516        526=
                                             527(3)     528(2)=    530(2)=    598=       599        602=       603        620=
                                             621        627=       628(3)     629(2)=    632(2)=    653=       654(2)     660
                                             662        678=       679        680(2)     682(2)     683        684        685
                                             686(3)     689(2)     701(2)=
     **       I*4   K1                       679=       680=       681(2)=    682        686        689
     **       I*4   KC                       225=       227        228        229        230(2)=
     **       I*4   KRS                      195=       196=       203        204

     **       I*4   KS                       302=       304        313(3)     315        318(2)     327(2)     328        330
                                             332(2)     333(3)     334(2)     335(2)     336(3)     337        343        344(2)=
                                             346        348        349        350        351        413=       414=       415
     **       I*4   L                        41=        51         52          53(2)=     57
     **       I*4   LO                       559=       560
     **       R*4   PTJL                     565=       576        582=
  2-0001B2D4  R*4   PUNSTO                   347(2)=    348        351=       353

  2-0001B2A8  R*4   PUTSTO                   206        348=       349        350=       351
     **       R*4   REMJ                     578=       579        581
  2-0001B27A  CHAR  RNAM                      16        204=       207=       210        218        242        289        377
  3-0001E960  CHAR  RUNDATE        COMM       5         13          35A        92        128        512        617
     **       R*8   SGRTOT                   33         638=       642        691

     **       R*4   SPC                      304=       314=       315=       318=       330        332        345=       346=
                                             347        380=       381
  2-0001B2B4  R*4   STP                      245=       246
  2-0001B2F0  R*4   TBL                      440=       441        442
     **       R*4   TCR                      237=       238=       239        242        274(2)=
     **       R*4   TJL                      564=       574(2)=    576        578        582
```

2-0001B2C4	R*4	TJNSTFCO	267=	269	373					
2-0001B2B8	R*4	TJNSTFSP	262=	266	371					
2-0001B2C8	R*4	TJSSTFCO	268=	269	374					
2-0001B2BC	R*4	TJSSTFSP	263=	266	372					
**	R*4	TJTFCO	269=	273	274					
2-0001B2C0	R*4	TJTFSP	266=	272	381					
**	R*8	TLATE	34	676=	683(2)=	691				
**	R*4	TNSTOR	287=	337(2)=	353(2)=	359	375=	386(2)=	388	
**	R*4	TSP	197=	201=	202	209	210	213	218	259
			272(2)=	275	276	277	299=	300=	332	347
2-0001B26C	CHAR	U	16	65=						
2-0001B2DC	R*4	USE	399=	403	435					
**	R*4	WAT	240=	241	291	332	347	359=	360	370=
			378	381	388=	389				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
3-0001BEF4	R*4	ADJU	COMM	1800	(450)	3	404=	482	485	489
3-0001C5FC	R*4	BALN	COMM	1800	(450)	3	470=	471	472(2)=	474
						475=	476(2)=	488(2)=	489	
2-00000140	R*8	BANK		3600	(450)	8	34	402=	404=	411
2-00019A20	R*8	CNSTO		96	(12)	11	34	239=	240	359
3-00011634	R*8	CRYO	COMM	43200	(450, 12)	2	32	62=	65=	188
						227=	267	268	270=	271=
						330	333	336	373=	374=
						382	385			
3-00000000	R*8	CTOT	COMM	1200	(10, 15)	1	32	116=	144	145
						154(2)	495	496	603=	628
						629				
2-00019A80	R*8	CTOTC		96	(12)	11	34	185=	188(2)=	228=
						237	238(2)	348	351	
3-0001D40C	R*4	DIFF	COMM	1800	(450)	3	543=	544		
2-00019BB4	R*4	EVAP		48	(12)	7	81=	88	103	105
3-0001011C	R*4	EXCS	COMM	1800	(450)	2	467=	491=		
2-00000000	R*8	FILL		96	(12)	7	33	81=	88	89(2)
						90	102	105	203	
2-00019AE0	R*8	FILLATE		96	(12)	12	34	81=	654	662=
						682	683	686	689	
2-00000F50	R*8	GSTOT		72	(9)	8	33	516=	527(2)=	530
						632				
2-0000B858	R*8	GTOT		104	(13)	9	33	137=	144(2)=	147
						156(2)	159	563	621=	628(2)=
						632	637	674(3)=	675=	682
						686	689			
3-0000D7A0	I*4	ID	COMM	1800	(450)	1	43=	48	49	50
						51	67	68	406	407
						428	457	462	482	600
						701				

2-0001A390	I*4	ID2		1800	(450)	10	65=	67	68	
2-0001AABC	I*4	IK		1800	(450)	11	43=			
2-00019C5C	I*4	IOR		44	(11)	9	31D	560		
2-00019BE4	I*4	ISK		120	(30)	9	19D	462		
3-0001CD04	I*4	ISKIP	COMM	1800	(450)	3	461=	463=		
2-0000B8C0	R*8	JDFL		3600	(450)	10	34	553=	568(2)	581=
**	I*4	JID		200	(50)	8				
2-00019B88	I*4	KREND		44	(11)	7	17D	51		
3-0001DB14	I*4	NDEND	COMM	44	(11)	3	52=	57=	58	59
						598				
2-0000C730	R*8	NSTO		54000	(450, 15)	11	34	187=	332(2)=	333
						336=	337	381(2)=	382	385=
						386				
2-0001AA98	R*4	PAVAIL		36	(9)	10	89=	90=	415	
**	R*8	PYLD		72	(9)	8	33			
4-00000000	R*4	RCHNM	COMM	160	(5, 8)	6	145	154	528	629
2-00019C88	R*4	RDIFF		1800	(450)	9	544=	567	571(2)=	572(2)=
						574	586	589(2)=	590=	
3-00010F2C	R*4	RFWB	COMM	1800	(450)	2	404=	476		
2-0001B1C4	CHAR	RNAME		168	(12)	8	15	81=	105	204
						654	660	686		
2-00000F98	R*8	SPAC		43200	(450, 12)	9	34	63=	262	263
						264=	265=	304	314(3)	315(2)
						318(2)	334	335	346(2)	371=
						372=	380	383	384	553
						581				
2-00000060	R*8	SPACE		96	(12)	7	33	81=	101	105
3-000004B0	R*8	STOR	COMM	54000	(450, 15)	1	32	43=	45=	46=
						47=	63	333=	334	335=
						336	382=	383	384=	385
						411=	415(2)=	428	458(2)=	470
						472	475	543	545	546(3)=
						549(3)=	550(3)=	551=	552=	570(2)=
						571	572(2)	573(2)=	588(2)=	589
						590(2)	591(2)=	600=	701	
3-0000F744	R*8	STOT	COMM	720	(10, 9)	1	32	495=	496=	498=
						499(2)=	527	528	629	
2-00000120	R*8	T		32	(4)	8	33	99=	101(2)=	102(2)=
						103(2)=	104(2)=	107	110	
3-0001DB40	R*8	TCRYO	COMM	3600	(450)	4	32	468=	489=	490
						491	492=	543	600	638(2)
						642(2)	675(2)			
2-00019B40	R*8	TFILL		72	(9)	12	34	682=	684	686
						689				
**	R*4	TOTR		25200	(450, 14)	11				
2-0000C6D0	R*8	TSPAC		96	(12)	10	33	159=	197	201(2)
						315	328	345	346	349
						350	553	581		
3-0000DEA8	CHAR	UNAME	COMM	6300	(450)	1	14	43=	330	701
3-0000FA14	R*4	USED	COMM	1800	(450)	2	404=	470		
3-00010824	R*4	WBRV	COMM	1800	(450)	2	469=	474=	482	484
						485(2)=	486(2)=	487		

```
  2-000000C0   R*8  YIELD                        96  (12)               7        33         88=       89       104
                                                                      105       458        499

LABELS

   Address     Label                      References

  0-00000E5F   1                          194       202       367       391#
  1-00000018   2'                          68        69#
  1-0000051E   3'                         396       397#
      **       4                          303       338#
  0-0000027E   5                           65        73#

  0-00000ABF   6                          258       259       287#      360
  1-00000B21   7'                         649       650#      657       668
  0-000019ED   8                          652       670#
  0-000019A0   9                          653       659       664#
  1-00000000   10'                         43        44#

  1-00000B24   11'                        654       655#
  1-0000000C   12'                         65        66#
  0-000008B8   13                         209       213#
      **       14                         184       186       188#
  0-000018D4   15                         653#      669

  1-00000AE7   16'                        647       648#
  1-00000B94   17'                        665       666#
  0-000019A6   18                         658       665#
  1-00000B68   19'                        660       661#
  0-00000178   20                          43        56#

  1-00000B90   21'                        662       663#
  1-00000390   22'                        218       219#
  0-00000924   23                         213       225#      278
  0-00000968   24                         224       237#
      **       26                         226       227#

  0-00000928   27                         226#      231
  1-0000067D   28'                        437       438#
  1-000006CA   29'                        444       445#
  0-00000094   30                          43#       55
  1-00000C83   31'                        693       694#

  1-000006F2   32'                        446       447#
  0-0000106C   33                         437#      449
  1-00000CD6   34'                        701       702#
  0-00000F27   35                         404       407#
  1-00000587   36'                        223       245       399       400#      423       440       696

      **       37                         700       701#
  0-00001B3C   38                         693#      698
  0-00000172   40                          51        54#
  0-00000278   45                          64        67        72#
  1-0000006B   46'                         77        78#

  1-00000463   47'                        242       243#
```

```
  1-00000072  49'          79       80#
  1-00000075  55'          81       82#
      **      60           60       61        63#
      **      90           87       91#

  0-00000C70  94          328      335#
  0-00000C8A  95          334      337#
  0-00000E3E  96          383      386#
      **      97          379      387#
  0-00000D94  98         375#      389

  1-00000081  100'         92       93#
  1-000000DE  101'         95       96#
      **      105          98       99#
      **      110         100      109#
  1-00000116  120'        105      106#

  1-00000122  125'        107      108#
  1-00000139  130'        110      111#
  1-00000BD0  142'        670      671#
  1-00000C54  144'        686      687#
  1-00000C71  145'        691      692#

      **      146         678      688#
  1-00000C60  147'        689      690#
  0-00001ACD  148         685      689#
      **      155         114      115       116#
      **      165         401      416#

      **      175         412      415#
  0-00000524  185        113#      418
      **      190         142      145#
      **      195         136      137#
  0-00000F9C  200         157      419#

  1-000005F9  201'        420      421#
      **      202         427      428#
  1-00000675  203'        428      429#
  0-00001051  204         424      430#
      **      205         143      144#

  1-0000024A  210'        138      139#
  1-000002EA  211'        149      150#
      **      212         153      154#
  1-000002C2  220'        145      146#      154
  1-000002D3  230'        147      148#      156

      **      232         158      159#
  1-0000014B  235'        128      129#
  1-00000165  240'        130      131#
  1-000001D4  250'        133      134#
  1-0000058A  260'        404      405#

  1-00000594  265'        407      408#
  0-000011EC  275         471      474#
  0-00001190  280         452      462       465#
  0-0000123E  285         482      489#
```

```
  0-000011FA  290                    473       476#

      **      295                    472#
  0-00001140  310                    442       450#
  0-00001259  315                    466       490       493#
  0-00000F58  320                    403       406       411#
  0-00001C07  400                    212       246       410       435       441       450       697       703#

      **      455                    494       497       498#
      **      505                    526       527#
  1-00000766  510'                   517       518#
  1-000008E2  520'                   528       529#
  1-000008F5  530'                   530       531#

  1-000006F6  540'                   512       513#
      **      590                    525       528#
      **      595                    515       516#
      **      605                    627       628#
  0-000014C0  606                    566#      583

  0-00001467  607                    547       550#
      **      608                    542       553#
  1-0000095C  610'                   622       623#
  0-00001538  612                    566       569       575#
  0-00001590  613                    561       563       584#

  0-000015E6  614                    585       587       592#
  0-000015EA  615                    559       577       579       593#
      **      616                    580       581#
      **      618                    599       600#
  1-000009D8  620'                   629       630#

  1-000009FF  630'                   632       633#
      **      640                    636       637#
  1-00000A2B  642'                   639       640#
  1-00000A7A  644'                   642       643#
  1-00000911  650'                   617       618#

  1-0000090D  652'                   615       616#
      **      655                    601       602       603#
      **      690                    626       629#
      **      695                    620       621#
  1-000004ED  996'                   330       331#

  1-00000366  997'                   210       211#
  1-000004BD  998'                   289       290#      377
  1-000004E5  999'                   291       292#      378
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References							
	ASSIGN	36	37	38	39	40			
	DATE	35							
	TABL2	501	502	503	504	505	506	507	508
		509	510	511					

TABL3	604	605	606	607	608	609	610	611
	612	613	614					
TABLE	117	118	119	120	121	122	123	124
	125	126	127					

```
00001          SUBROUTINE TABLE (KB,KE,IRET,JT)
00002          COMMON CTOT(10,15),STOR(450,15),ID(450),UNAME(450),STOT(10,9)
00003          COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
00004          COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
00005          COMMON TCRYO(450)
00006          COMMON IT,IYR,FTAB,ICONT,RUNDATE
00007          COMMON /RN/RCHNM(5,8)
00008          CHARACTER*9 RUNDATE
00009          CHARACTER*14 UNAME
00010          REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
00011          WRITE(6,160) IT,RUNDATE
00012      160 FORMAT(1H1/58X,4HPART,I2,59X,A9)
00013          IF(IT.EQ.1) WRITE(6,140) IYR
00014      140 FORMAT(//18X,I4,98H WATER DISTRICT 1 RESERVOIR SPACE BY USER ASSUM
00015         1ING FULL RESERVOIRS, WITHOUT WATER BANK (ACRE-FEET)/)
00016          IF(IT.EQ.2) WRITE(6,150) IYR
00017      150 FORMAT(//15X,I4,107H WATER DISTRICT 1 RESERVOIR STORAGE BY USER AF
00018         1TER PERCENT FILL AND EVAPORATION, WITH WATER BANK (ACRE-FEET)/)
00019          WRITE(6,145) (RCHNM(I,JT),I=1,5)
00020      145 FORMAT(54X,5A4/)
00021          WRITE(6,95)
00022       95 FORMAT(1H ,13HNUMBER   USER,11X,107H       JACKSN PALSDS PL USR PL
00023         1 WWS HNRY L ISL PK GRSY L  RIRIE AM FLS L WLCT AM LTF PL LTF W BAN
00024         2K     TOTAL/)
00025          IB=1
00026          IF(KB.EQ.0) GO TO 100
00027          IB=NDEND(KB)+1
00028      100 IE=NDEND(KE)
00029          DO 80 I=IB,IE
00030          STOR(I,13)=0.0
00031          DO 60 K=1,12
00032          CTOT(JT,K)=CTOT(JT,K)+STOR(I,K)
00033          STOR(I,13)=STOR(I,13)+STOR(I,K)
00034       60 CONTINUE
00035          STOR(I,15)=STOR(I,13)+STOR(I,14)
00036          CTOT(JT,13)=CTOT(JT,13)+STOR(I,13)
00037          CTOT(JT,14)=CTOT(JT,14)+STOR(I,14)
00038          CTOT(JT,15)=CTOT(JT,15)+STOR(I,15)
00039          WRITE(6,70) ID(I),UNAME(I),(JIDNNT(STOR(I,K)),K=1,12),
00040         1                           JIDNNT(STOR(I,14)),STOR(I,15)
00041       70 FORMAT(1X,I8,1X,A14,7X,13I7,F10.1)
00042       80 CONTINUE
00043          IF(IRET.GT.0) GO TO 10
00044          WRITE(6,20) (JIDNNT(CTOT(JT,K)),K=1,12),JIDNNT(CTOT(JT,14)),
00045         1                                    CTOT(JT,15)
```

```
00046      20 FORMAT(/10X,5HTOTAL,9X,7X,13I7,F10.1)
00047      10 CONTINUE
00048         RETURN
00049         END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	562	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	423	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	20	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3	$BLANK	125289	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
4	RN	160	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
	Total Space Allocated	126454	

ENTRY POINTS

Address	Type	Name	References
0-00000000		TABLE	1#

VARIABLES

Address	Type	Name	Attributes	References						
3-0001E958	R*4	FTAB	COMM	6						
**	I*4	I		19(2)=	29=	30	32	33(3)	35(3)	36	37
				38	39(5)					
**	I*4	IB		25=	27=	29				
3-0001E95C	I*4	ICONT	COMM	6						
**	I*4	IE		28=	29					
AP-0000000C@	I*4	IRET		1	43					
3-0001E950	I*4	IT	COMM	6	11	13	16			
3-0001E954	I*4	IYR	COMM	6	13	16				
AP-00000010@	I*4	JT		1	19	32(2)	36(2)	37(2)	38(2)	44(3)
**	I*4	K		31=	32(3)	33	39(2)=	44(2)=		
AP-00000004@	I*4	KB		1	26	27				
AP-00000008@	I*4	KE		1	28					
3-0001E960	CHAR	RUNDATE	COMM	6	8	11				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References	
3-0001BEF4	R*4	ADJU	COMM	1800	(450)	4	
3-0001C5FC	R*4	BALN	COMM	1800	(450)	4	
3-00011634	R*8	CRYO	COMM	43200	(450, 12)	3	10

3-00000000	R*8	CTOT	COMM	1200	(10, 15)	2	10	32(2)=	36(2)=	37(2)=
						38(2)=	44(3)			
3-0001D40C	R*4	DIFF	COMM	1800	(450)	4				
3-0001011C	R*4	EXCS	COMM	1800	(450)	3				
3-0000D7A0	I*4	ID	COMM	1800	(450)	2	39			
3-0001CD04	I*4	ISKIP	COMM	1800	(450)	4				
3-0001DB14	I*4	NDEND	COMM	44	(11)	4	27	28		
4-00000000	R*4	RCHNM	COMM	160	(5, 8)	7	19			
3-00010F2C	R*4	RFWB	COMM	1800	(450)	3				
3-000004B0	R*8	STOR	COMM	54000	(450, 15)	2	10	30=	32	33(3)=
						35(3)=	36	37	38	39(3)
3-0000F744	R*8	STOT	COMM	720	(10, 9)	2	10			
3-0001DB40	R*8	TCRYO	COMM	3600	(450)	5	10			
3-0000DEA8	CHAR	UNAME	COMM	6300	(450)	2	9	39		
3-0000FA14	R*4	USED	COMM	1800	(450)	3				
3-00010824	R*4	WBRV	COMM	1800	(450)	3				

LABELS

Address	Label	References	
0-00000231	10	43	47#
1-00000191	20'	44	46#
**	60	31	34#
1-0000017F	70'	39	41#
**	80	29	42#
1-000000FC	95'	21	22#
0-000000F6	100	26	28#
1-00000013	140'	13	14#
1-000000F4	145'	19	20#
1-0000007F	150'	16	17#
1-00000000	160'	11	12#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
I*4	JIDNNT	39(2)	44(2)

```
00001          SUBROUTINE TABL2 (KB,KE,IRET,JT)
00002          COMMON CTOT(10,15),STOR(450,15),ID(450),UNAME(450),STOT(10,9)
00003          COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
00004          COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
00005          COMMON TCRYO(450)
00006          COMMON IT,IYR,FTAB,ICONT,RUNDATE
00007          COMMON /RN/RCHNM(5,8)
00008          CHARACTER*9 RUNDATE
00009          CHARACTER*14 UNAME
00010          REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
00011          IF(ICONT.EQ.1) GO TO 120
00012          WRITE(6,140) RUNDATE,FTAB,IYR,(RCHNM(I,JT),I=1,5)
00013      140 FORMAT(1H1/////123X,A9//26X,5HTABLE,F4.0,4X,I4,25H STORED WATER AC
00014         1COUNTS - , 5A4,12H (ACRE-FEET))
00015          GO TO 130
00016      120 WRITE(6,150) FTAB
00017      150 FORMAT(1H1////////49X,5HTABLE,F4.0,4X,9HCONTINUED)
00018      130 WRITE(6,95)
00019       95 FORMAT( /28X,76H            STORAGE OR                          RETU
00020         1RN TO                    /28X,67H          WATER BANK
00021         2 REVERTED TO SPACEHOLDER          /28X,99H   STORAGE   PURCHASE,
00022         3  STORAGE  WATER BANK    FROM                ADJUST-     EXCESS
00023         4    CARRY-/  4X,13HNUMBER   NAME,11X,99H  ALLOCATED SUPPLY (-)
00024         5 USED   FROM USER   WATER BANK   BALANCE       MENT       USED
00025         6   OVER/)
00026          IB=1
00027          IF(KB.EQ.0) GO TO 100
00028          IB=NDEND(KB)+1
00029      100 IE=NDEND(KE)
00030          DO 80 I=IB,IE
00031          STOT(JT,3)=STOT(JT,3)+USED(I)
00032          STOT(JT,4)=STOT(JT,4)+WBRV(I)
00033          STOT(JT,5)=STOT(JT,5)+RFWB(I)
00034          STOT(JT,6)=STOT(JT,6)+BALN(I)
00035          STOT(JT,7)=STOT(JT,7)+ADJU(I)
00036          STOT(JT,8)=STOT(JT,8)+EXCS(I)
00037          STOT(JT,9)=STOT(JT,9)+TCRYO(I)
```

```
00038          IF(ISKIP(I).GT.0) GO TO 80
00039          WRITE(6,70) ID(I),UNAME(I),STOR(I,13),STOR(I,14),USED(I),
00040         1 WBRV(I),RFWB(I),BALN(I),ADJU(I),EXCS(I),TCRYO(I)
00041       70 FORMAT( 4X,I8,1X,A14,9(2X,F9.1))
00042       80 CONTINUE
00043          IF(IRET.GT.0) GO TO 10
00044          FTAB=FTAB+1.0
00045          WRITE(6,20) (STOT(JT,K),K=1,9)
00046       20 FORMAT(/13X,5HTOTAL,9X,9(2X,F9.1))
00047          ICONT=0
00048          RETURN
00049       10 ICONT=1
00050          RETURN
00051          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	608	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	535	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	20	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3	$BLANK	125289	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
4	RN	160	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
	Total Space Allocated	126612	

ENTRY POINTS

Address	Type	Name	References
0-00000000		TABL2	1#

VARIABLES

Address	Type	Name	Attributes	References						
3-0001E958	R*4	FTAB	COMM	6	12	16	44(2)=			
**	I*4	I		12(2)=	30=	31	32	33	34	35	36
				37	38	39(11)					
**	I*4	IB		26=	28=	30					
3-0001E95C	I*4	ICONT	COMM	6	11	47=	49=				
**	I*4	IE		29=	30						
AP-0000000C@	I*4	IRET		1	43						
3-0001E950	I*4	IT	COMM	6							
3-0001E954	I*4	IYR	COMM	6	12						
AP-00000010@	I*4	JT		1	12	31(2)	32(2)	33(2)	34(2)	35(2)	36(2)
				37(2)	45						
**	I*4	K		45(2)=							
AP-00000004@	I*4	KB		1	27	28					

AP-00000008@	I*4	KE		1	29	
3-0001E960	CHAR	RUNDATE	COMM	6	8	12

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
3-0001BEF4	R*4	ADJU	COMM	1800	(450)	4	35	39		
3-0001C5FC	R*4	BALN	COMM	1800	(450)	4	34	39		
3-00011634	R*8	CRYO	COMM	43200	(450, 12)	3	10			
3-00000000	R*8	CTOT	COMM	1200	(10, 15)	2	10			
3-0001D40C	R*4	DIFF	COMM	1800	(450)	4				
3-0001011C	R*4	EXCS	COMM	1800	(450)	3	36	39		
3-0000D7A0	I*4	ID	COMM	1800	(450)	2	39			
3-0001CD04	I*4	ISKIP	COMM	1800	(450)	4	38			
3-0001DB14	I*4	NDEND	COMM	44	(11)	4	28	29		
4-00000000	R*4	RCHNM	COMM	160	(5, 8)	7	12			
3-00010F2C	R*4	RFWB	COMM	1800	(450)	3	33	39		
3-000004B0	R*8	STOR	COMM	54000	(450, 15)	2	10	39(2)		
3-0000F744	R*8	STOT	COMM	720	(10, 9)	2	10	31(2)=	32(2)=	33(2)=
						34(2)=	35(2)=	36(2)=	37(2)=	45
3-0001DB40	R*8	TCRYO	COMM	3600	(450)	5	10	37	39	
3-0000DEA8	CHAR	UNAME	COMM	6300	(450)	2	9	39		
3-0000FA14	R*4	USED	COMM	1800	(450)	3	31	39		
3-00010824	R*4	WBRV	COMM	1800	(450)	3	32	39		

LABELS

Address	Label	References		
0-00000258	10	43	49#	
1-00000200	20'	45	46#	
1-000001ED	70'	39	41#	
0-00000201	80	30	38	42#
1-00000071	95'	18	19#	
0-000000C5	100	27	29#	
0-00000070	120	11	16#	
0-00000093	130	15	18#	
1-00000000	140'	12	13#	
1-0000004C	150'	16	17#	

```
00001          SUBROUTINE TABL3 (KB,KE,IRET,JT)
00002          COMMON CTOT(10,15),STOR(450,15),ID(450),UNAME(450),STOT(10,9)
00003          COMMON USED(450),EXCS(450),WBRV(450),RFWB(450),CRYO(450,12)
00004          COMMON ADJU(450),BALN(450),ISKIP(450),DIFF(450),NDEND(11)
00005          COMMON TCRYO(450)
00006          COMMON IT,IYR,FTAB,ICONT,RUNDATE
00007          COMMON /RN/RCHNM(5,8)
00008          CHARACTER*9 RUNDATE
00009          CHARACTER*14 UNAME
00010          REAL*8 CTOT,STOR,STOT,CRYO,TCRYO
00011          WRITE(6,160)
00012      160 FORMAT(1H1)
00013          WRITE(6,150) RUNDATE,IYR
00014      150 FORMAT(/123X,A9/30X,I4, 66H WATER DISTRICT 1 RESERVOIR CARRYOVER S
00015         1TORAGE BY USER  (ACRE-FEET)/)
00016          WRITE(6,145) (RCHNM(I,JT),I=1,5)
00017      145 FORMAT(54X,5A4/)
00018          WRITE(6,95)
00019       95 FORMAT(1H ,13HNUMBER   USER,11X, 93HCARRYOVER JACKSON PALISADES PA
00020         1L USR  PAL WWS HNRYS L ISL PRK GRSSY L   RIRIE AM FALLS WALCOTT/)
00021          IB=1
00022          IF(KB.EQ.0) GO TO 100
00023          IB=NDEND(KB)+1
00024      100 IE=NDEND(KE)
00025          DO 80 I=IB,IE
00026          DO 60 K=1,10
00027          CTOT(JT,K)=CTOT(JT,K)+STOR(I,K)
00028       60 CONTINUE
00029          WRITE(6,70) ID(I),UNAME(I),TCRYO(I),(STOR(I,K),K=1,10)
00030       70 FORMAT( 1X,I8,1X,A14,F10.1,2F9.1,F8.1,F9.1,4F8.1,F9.1,F8.1)
00031       80 CONTINUE
00032          IF(IRET.GT.0) GO TO 10
00033          WRITE(6,20) STOT(JT,9),(CTOT(JT,K),K=1,10)
00034       20 FORMAT(/10X,5HTOTAL,9X,F10.1,2F9.1,F8.1,F9.1,4F8.1,F9.1,F8.1)
00035       10 CONTINUE
00036          RETURN
00037          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	436	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	281	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	20	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3	$BLANK	125289	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
4	RN	160	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
	Total Space Allocated	126186	

ENTRY POINTS

Address	Type	Name	References
0-00000000		TABL3	1#

VARIABLES

Address	Type	Name	Attributes	References			
3-0001E958	R*4	FTAB	COMM	6			
**	I*4	I		16(2)=	25=	27	29(4)
**	I*4	IB		21=	23=	25	
3-0001E95C	I*4	ICONT	COMM	6			
**	I*4	IE		24=	25		
AP-0000000C@	I*4	IRET		1	32		
3-0001E950	I*4	IT	COMM	6			
3-0001E954	I*4	IYR	COMM	6	13		
AP-00000010@	I*4	JT		1	16	27(2)	33(2)
**	I*4	K		26=	27(3)	29(2)=	33(2)=
AP-00000004@	I*4	KB		1	22	23	
AP-00000008@	I*4	KE		1	24		
3-0001E960	CHAR	RUNDATE	COMM	6	8	13	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References			
3-0001BEF4	R*4	ADJU	COMM	1800	(450)	4			
3-0001C5FC	R*4	BALN	COMM	1800	(450)	4			
3-00011634	R*8	CRYO	COMM	43200	(450, 12)	3	10		
3-00000000	R*8	CTOT	COMM	1200	(10, 15)	2	10	27(2)=	33
3-0001D40C	R*4	DIFF	COMM	1800	(450)	4			
3-0001011C	R*4	EXCS	COMM	1800	(450)	3			
3-0000D7A0	I*4	ID	COMM	1800	(450)	2	29		
3-0001CD04	I*4	ISKIP	COMM	1800	(450)	4			
3-0001DB14	I*4	NDEND	COMM	44	(11)	4	23	24	
4-00000000	R*4	RCHNM	COMM	160	(5, 8)	7	16		

3-00010F2C	R*4	RFWB	COMM	1800	(450)	3			
3-000004B0	R*8	STOR	COMM	54000	(450, 15)	2	10	27	29
3-0000F744	R*8	STOT	COMM	720	(10, 9)	2	10	33	
3-0001DB40	R*8	TCRYO	COMM	3600	(450)	5	10	29	
3-0000DEA8	CHAR	UNAME	COMM	6300	(450)	2	9	29	
3-0000FA14	R*4	USED	COMM	1800	(450)	3			
3-00010824	R*4	WBRV	COMM	1800	(450)	3			

LABELS

Address	Label	References	
0-000001B3	10	32	35#
1-000000F3	20'	33	34#
**	60	26	28#
1-000000D1	70'	29	30#
**	80	25	31#
1-0000005C	95'	18	19#
0-000000B6	100	22	24#
1-00000054	145'	16	17#
1-00000004	150'	13	14#
1-00000000	160'	11	12#

```
00001          BLOCK DATA
00002          COMMON /RN/RCHNM(5,8)
00003          DATA RCHNM/
00004         1 'IRWI','N TO',' LOR','ENZO','    ',
00005         2 'LORE','NZO ','TO B','LACK','FOOT',
00006         3 'BLAC','KFOO','T TO',' MIL','NER ',
00007         4 'MAIN',' STE','M HE','NRYS',' FRK',
00008         5 'FALL','S RI','VER ','    ','    ',
00009         6 'TETO','N RI','VER ','    ','    ',
00010         7 'WILL','OW C','REEK','    ','    ',
00011         8 'MISC','ELLA','NEOU','S   ','    '/
00012          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	RN	160	PIC OVR REL GBL SHR NOEXE RD WRT QUAD
	Total Space Allocated	160	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
0-00000000	R*4	RCHNM	COMM	160	(5, 8)	2 3D

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 SNKSTO

  /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.SNAKE]SNKSTO.LIS;6
  /OBJECT=HYDRO:[BSUTTER.SNAKE]SNKSTO.OBJ;1


COMPILER:  DEC Fortran V6.3-141

COMPILATION STATISTICS

  Run Time:           2.76 seconds
  Elapsed Time:       4.61 seconds
  Page Faults:        6759
  Dynamic Memory:     2592 pages
```

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES IN DATA SET SNKSTO'YR'.SPA
0004    C       THESE VALUES ARE THE STORAGE SPACE ENTITLEMENTS IN UPPER
0005    C       SNAKE RIVER RESERVOIRS.  RJS - OCTOBER 1988.
0006    C
0007    C*************************************************************************
0008
0009          DIMENSION SPACE(400,12),RNAME(12)
0010          DIMENSION ID(400),IK(400),DNAME(400)
0011          CHARACTER*14 DNAME
0012          CHARACTER*10 RNAME
0013          DATA RNAME/'JACKSON L ','PALISADES ','PAL W USRS','PAL WWS   ',
0014         1'HENRYS LK ','ISLAND PRK','GRASSY LK ','RIRIE     ','AMERICAN F',
0015         2'LK WALCOTT','AM FLS PWR','PALSDS PWR'/
0016          DATA BLANK/' '/
0017          CALL ASSIGN (1,'SPACEIN')
0018          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0019          CALL ASSIGN (3,'OLDSPACE')
0020          IANS=BLANK
0021    C
0022    C***** READ COMPLETE FILE OF EXISTING STORAGE VALUES
0023    C
0024          DO 9 I=1,400
0025        9 READ(1,2,ERR=5,END=5) ID(I),IK(I),DNAME(I),(SPACE(I,L),L=1,12)
0026        2 FORMAT(I8,A1,A14,12F9.0)
0027        5 IF(I.EQ.1) GO TO 10
0028          NSPACE=I-1
0029          GO TO 12
0030    C
0031    C***** CREATE NEW FILE FROM LAST YEAR'S FILE IF THIS YEAR'S FILE
0032    C***** DOES NOT YET EXIST.
0033    C
0034       10 PRINT *,' '
0035          PRINT *,' NO FILE EXISTS FOR CURRENT YEAR, CREATING NEW  '
0036          PRINT *,' FROM LAST YEAR''S DATA '
0037          PRINT *,' '
0038          DO 14 I=1,400
0039       14 READ(3,2,ERR=12,END=15) ID(I),IK(I),DNAME(I),(SPACE(I,L),L=1,12)
0040       15 NSPACE=I-1
0041    C
0042    C***** INQUIRE FOR CHANGES TO SPACE TABLE
0043    C
0044    C
0045       12 PRINT *,' '
0046          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE SPACE ENTITLEMENTS'
0047          PRINT *,' '
0048          PRINT *,' ANSWER   Y = YES    N OR HIT RETURN KEY = NO          '
0049          PRINT *,'          Q = QUIT   J = JUMP AROUND                    '
0050          PRINT *,' '
0051          JUMP=0.0
0052       13 DO 101 I=1,NSPACE
0053          IF(ID(I).LT.JUMP) GO TO 101
0054          IF(IK(I).EQ.'S') GO TO 101
0055          DO 108 L=1,12
0056        6 TYPE 11, ID(I),DNAME(I),RNAME(L),SPACE(I,L)
0057       11 FORMAT(' ',I8,1X,A14,' SPACE IN ',A10,' =',F10.1,
```

```
0058          1          ' NEW VALUE? Y/N/J/Q '$)
0059           ACCEPT 102, IANS
0060       102 FORMAT(A1)
0061           IF(IANS.EQ.'Q') GO TO 100
0062           IF(IANS.EQ.'J') GO TO 3
0063           IF(IANS.NE.'Y') GO TO 108
0064           PRINT 103
0065       103 FORMAT('                              ENTER NEW VALUE   ='$)
0066           READ(5,4,ERR=106) SPACE(I,L)
0067         4 FORMAT(F10.0)
0068           GO TO 108
0069       106 PRINT 107
0070       107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0071           GO TO 6
0072       108 CONTINUE
0073       101 CONTINUE
0074           GO TO 100
0075         3 PRINT *,' '
0076           PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0077           PRINT *,'                 46500 BEGINS FALL RIVER         '
0078           PRINT *,'                 49500 BEGINS LOWER HENRYS FORK  '
0079           PRINT *,'                 53000 BEGINS TETON RIVER        '
0080           PRINT *,'                 57000 BEGINS SNAKE BELOW HENRYS FORK '
0081           PRINT *,'                 57500 BEGINS WILLOW CREEK '
0082           PRINT *,'                 59000 BEGINS SNAKE BELOW WILLOW CREEK'
0083           PRINT *,'                 75000 BEGINS SNAKE BELOW BLACKFOOT '
0084           PRINT *,' '
0085           PRINT 104
0086       104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0087          15'/     ' **** USE ONLY LAST 5 DIGITS.                    '$)
0088           ACCEPT 105, JUMP
0089       105 FORMAT(I5)
0090           JUMP=JUMP+13000000
0091           IF(JUMP.GT.13090000) JUMP=JUMP+86900000
0092           GO TO 13
0093       C
0094       C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0095       C***** ID NUMBERS AND STORAGE ACCOUNT VALUES TO NEW FILE.
0096       C
0097         100 PRINT 8
0098           8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE SPACE VALUES?
0099           1  Y/N '$)
0100           ACCEPT 102, IANS
0101           IF(IANS.EQ.'Y') GO TO 12
0102           DO 1 I=1,NSPACE
0103         1 WRITE(2,18) ID(I),IK(I),DNAME(I),(SPACE(I,L),L=1,12)
0104        18 FORMAT(I8,A1,A14,12F9.1)
0105           PRINT *,' '
0106           PRINT *,' FILE SNKSTO''YR''.SPA HAS BEEN UPDATED WITH NEW VALUES'
0107           PRINT *,' '
0108           PRINT *,' '
0109           PRINT *,' '
0110           STOP
0111           END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1598	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1086	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	28472	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	31156	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGSS$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
**	R*4	BLANK		16D	20					
**	I*4	I		24=	25(4)	27	28	38=	39(4)	40	52=
					53	54	56(3)	66	102=	103(4)	
2-00006DD8	I*4	IANS		20=	59=	61	62	63	100=	101	
2-00006DE4	I*4	JUMP		51=	53	88=	90(2)=	91(3)=			
2-00006DDC	I*4	L		25(2)=	39(2)=	55=	56(2)	66	103(2)=		
2-00006DE0	I*4	NSPACE		28=	40=	52	102				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-000057F8	CHAR	DNAME		5600	(400)	10	11	25=	39=	56
								103		
2-00004B00	I*4	ID		1600	(400)	10	25=	39=	53	56
								103		
2-00005140	I*4	IK		1600	(400)	10	25=	39=	54	103
2-00005780	CHAR	RNAME		120	(12)	9	12	13D	56	
2-00000000	R*4	SPACE		19200	(400, 12)	9	25=	39=	56	66=
								103		

LABELS

Address	Label	References		
**	1	102	103#	
1-000002D5	2'	25	26#	39
0-00000380	3	62	75#	
1-0000034C	4'	66	67#	
0-0000009F	5	25(2)	27#	

```
  0-0000027C  6                          56#        71
  1-000003F3  8'                         97         98#
      **      9                          24         25#
  0-000000AC  10                         27         34#
  1-000002E1  11'                        56         57#

  0-0000018B  12                         29         39         45#       101
  0-00000238  13                         52#        92
      **      14                         38         39#
  0-00000187  15                         39         40#
  1-00000432  18'                       103        104#

  0-000004F2  100                        61         74         97#
  0-000004EC  101                        52         53         54         73#
  1-00000317  102'                       59         60#       100
  1-0000031A  103'                       64         65#
  1-0000037D  104'                       85         86#

  1-000003F0  105'                       88         89#
  0-00000358  106                        66         69#
  1-00000350  107'                       69         70#
  0-00000374  108                        55         63         68         72#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                                 References

        ASSIGN                                 17        19
        FOR$OPEN                               18
```

```
  +-----------------------------------------------+
  |           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 SNKCHGSS

  /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.SNAKE]SNKCHGSS.LIS;5
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGSS.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           4.37 seconds
  Elapsed Time:       5.42 seconds
  Page Faults:        759
  Dynamic Memory:     447 pages
```

SNKCHGSC.LIS

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES IN DATA SET SNKSTO'YR'.CRY
0004    C       THESE VALUES ARE THE CARRYOVER STORAGE BY USER AND RESERVOIR
0005    C       FROM THE PREVIOUS YEAR.    RJS - OCTOBER 1988.
0006    C
0007    C*************************************************************************
0008
0009          DIMENSION CRYO(400,12),RNAME(10)
0010          DIMENSION ID(400),IK(400),DNAME(400)
0011          CHARACTER*14 DNAME
0012          CHARACTER*10 RNAME
0013          DATA RNAME/'JACKSON L ','PALISADES ','PAL W USRS','PAL WWS   ',
0014         1'HENRYS LK ','ISLAND PRK','GRASSY LK ','RIRIE     ','AMERICAN F',
0015         2'LK WALCOTT'/
0016          DATA BLANK/' '/
0017          CALL ASSIGN (1,'CARRYIN')
0018          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0019          IANS=BLANK
0020    C
0021    C***** READ COMPLETE FILE OF EXISTING STORAGE VALUES
0022    C
0023          DO 9 I=1,400
0024        9 READ(1,2,END=5) ID(I),IK(I),DNAME(I),(CRYO(I,L),L=1,12)
0025        2 FORMAT(I8,A1,A14,12F9.0)
0026        5 NCRYO=I-1
0027    C
0028    C***** INQUIRE FOR CHANGES TO CARRYOVER TABLE
0029    C
0030    C
0031       12 PRINT *,' '
0032          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE CARRYOVER VALUES'
0033          PRINT *,' '
0034          PRINT *,' ANSWER   Y = YES    N OR HIT RETURN KEY = NO          '
0035          PRINT *,'          Q = QUIT   J = JUMP AROUND                    '
0036          PRINT *,' '
0037          JUMP=0.0
0038       13 DO 101 I=1,NCRYO
0039          IF(ID(I).LT.JUMP) GO TO 101
0040          IF(IK(I).EQ.'S') GO TO 101
0041          DO 108 L=1,10
0042        6 TYPE 11, ID(I),DNAME(I),RNAME(L),CRYO(I,L)
0043       11 FORMAT(' ',I8,1X,A14,' CRYO IN ',A10,' =',F10.1,
0044         1          ' NEW VALUE? Y/N/J/Q '$)
0045          ACCEPT 102, IANS
0046      102 FORMAT(A1)
0047          IF(IANS.EQ.'Q') GO TO 100
0048          IF(IANS.EQ.'J') GO TO 3
0049          IF(IANS.NE.'Y') GO TO 108
0050          PRINT 103
0051      103 FORMAT('                                ENTER NEW VALUE   ='$)
0052          READ(5,4,ERR=106) CRYO(I,L)
0053        4 FORMAT(F10.0)
0054          GO TO 108
0055      106 PRINT 107
0056      107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0057          GO TO 6
```

```
0058      108 CONTINUE
0059      101 CONTINUE
0060          GO TO 100
0061        3 PRINT *,' '
0062          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0063          PRINT *,'                 46500 BEGINS FALL RIVER         '
0064          PRINT *,'                 49500 BEGINS LOWER HENRYS FORK  '
0065          PRINT *,'                 53000 BEGINS TETON RIVER        '
0066          PRINT *,'                 57000 BEGINS SNAKE BELOW HENRYS FORK '
0067          PRINT *,'                 57500 BEGINS WILLOW CREEK '
0068          PRINT *,'                 59000 BEGINS SNAKE BELOW WILLOW CREEK'
0069          PRINT *,'                 75000 BEGINS SNAKE BELOW BLACKFOOT '
0070          PRINT *,' '
0071          PRINT 104
0072      104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0073         15'/     ' **** USE ONLY LAST 5 DIGITS.                    '$)
0074          ACCEPT 105, JUMP
0075      105 FORMAT(I5)
0076          JUMP=JUMP+13000000
0077          IF(JUMP.GT.13090000) JUMP=JUMP+86900000
0078          GO TO 13
0079      C
0080      C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0081      C***** ID NUMBERS AND CARRYOVER VALUES TO NEW FILE.
0082      C
0083        100 PRINT 8
0084          8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE CARRYOVER VALUES?
0085           1  Y/N '$)
0086          ACCEPT 102, IANS
0087          IF(IANS.EQ.'Y') GO TO 12
0088          DO 1 I=1,NCRYO
0089        1 WRITE(2,18) ID(I),IK(I),DNAME(I),(CRYO(I,L),L=1,12)
0090         18 FORMAT(I8,A1,A14,12F9.1)
0091          PRINT *,' '
0092          PRINT *,' FILE SNKSTO''YR''.CRY HAS BEEN UPDATED WITH NEW VALUES'
0093          PRINT *,' '
0094          PRINT *,' '
0095          PRINT *,' '
0096          STOP
0097          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1346	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1003	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	28376	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	30725	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGSC$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
**	R*4	BLANK		16D	19					
**	I*4	I		23=	24(4)	26	38=	39	40	42(3) 52
					88=	89(4)				
2-00006DC4	I*4	IANS		19=	45=	47	48	49	86=	87
2-00006DCC	I*4	JUMP		37=	39	74=	76(2)=	77(3)=		
2-00006DC8	I*4	L		24(2)=	41=	42(2)	52	89(2)=		
**	I*4	NCRYO		26=	38	88				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000000	R*4	CRYO		19200	(400, 12)	9	24=	42	52=	89
2-000057E4	CHAR	DNAME		5600	(400)	10	11	24=	42	89
2-00004B00	I*4	ID		1600	(400)	10	24=	39	42	89
2-00005140	I*4	IK		1600	(400)	10	24=	40	89	
2-00005780	CHAR	RNAME		100	(10)	9	12	13D	42	

LABELS

Address	Label	References	
**	1	88	89#
1-0000027F	2'	24	25#
0-00000284	3	48	61#
1-000002F5	4'	52	53#
0-00000093	5	24	26#
0-00000180	6	42#	57
1-0000039C	8'	83	84#
**	9	23	24#

```
  1-0000028B  11'                          42        43#
  0-00000098  12                           31#       87

  0-0000013C  13                           38#       78
  1-000003DF  18'                          89        90#
  0-000003F6  100                          47        60        83#
  0-000003F0  101                          38        39        40        59#
  1-000002C0  102'                         45        46#       86

  1-000002C3  103'                         50        51#
  1-00000326  104'                         71        72#
  1-00000399  105'                         74        75#
  0-0000025C  106                          52        55#
  1-000002F9  107'                         55        56#

  0-00000278  108                          41        49        54        58#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	17
	FOR$OPEN	18

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 SNKCHGSC

  /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.SNAKE]SNKCHGSC.LIS;3
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGSC.OBJ;1
```

COMPILATION STATISTICS

Run Time:	5.65 seconds
Elapsed Time:	6.31 seconds
Page Faults:	733
Dynamic Memory:	414 pages

"SELECTED DRAINAGE AREAS AND AVERAGE ANNUAL FLOWS IN IDAHO PANHANDLE"

USGS IDEN. NUMBER	RECORD IN YEARS	AREA IN SQ. MILES	ANN. FLOWS ACRE-FOOT	LOGARITHM OF AREA	LOGARITHM OF ANNUAL FLOWS	DRAINAGE BASIN
12305000	59	11740.	10060000.	4.0696680969116	7.00259798071991	KOOTENAI R NR LEONIA
12306500	58	570.	505700.	2.75587485567249	5.70389295363254	MOYIE R
12318500	58	13400.	11172000.	4.12710479836481	7.04813092702897	KOOTENAI R
12321500	57	97.	143500.	1.98677173426624	5.15685190107001	BOUNDARY CK
12322000	59	13700.	11480000.	4.13672056715641	7.05994188806195	KOOT. NR PORTHILL
12394000	39	611.	941000.	2.78604121024255	5.97358962342726	PRIEST R NR COOLIN
12395500	73	24200.	18700000.	4.38381536598043	7.2718416065365	PEND OREILLE R
12411000	36	335.	514400.	2.52504480703685	5.71130095991617	COEUR D'ALENE
12413140	19	14.9	26730.	1.17318626841227	4.42699895875654	PLACER CK
12413500	54	1223.0	1842000.	3.08742645703629	6.26528962586083	COEUR D'ALENE NR CTLD
12414500	68	1030.0	1708000.	3.01283722470517	6.23248786635299	ST. JOE RIVER
12414900	22	275.0	256500.	2.43933269383026	5.40908736944784	ST. MARIE R.
12416000	28	22.0	20650.0	1.34242268082221	4.31492005599242	HAYDEN CREEK
12419000	75	3840.0	4525000.0	3.58433122436753	6.65561858354122	SPOKANE RIVER
12311000	43	133.0	105800.0	2.12385164096709	5.02448566769917	DEEP CREEK
12305500	45	53.0	87660.0	1.72427586960079	4.94280146631794	BOULDER CK
12320500	29	29.0	46190.0	1.46239799789896	4.66454796224655	LONG CANYON
12321000	30	70.0	138300.0	1.84509804001426	5.14082218010931	SMITH'S CK
12316800	17	23.0	28180.0	1.36172783601759	4.44994098877334	MISSION CK
12392300	17	124.0	247100.0	2.09342168516224	5.39287274540208	PACK RIVER
	SUM	71489.9	62548710.0	52.02135105	113.84802131	
	AVERAGE	3574.5	3127435.5	2.60106755	5.69240107	
	STAND. DEV.	6701.994	5332431.646	1.029656	0.9573700	
		CONSTANT	293469.777	CONSTANT	3.293243	
		SLOPE	792.829	SLOPE	0.92237	
		FIT	0.992927	FIT	0.9841	

SIMPLE REGRESSION ANALYSIS, SELECTED PANHANDLE BASINS:

LOG Q = LOG(K*A**B)

= LOG (K) + B*LOG (A)

= 3.293243 +0.92237* LOG(A)

WHERE: Q = AVERAGE ANNUAL FLOWS

A = DRAINAGE AREA IN SQUARE MILES

SNKCHGSU. LIS

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES STORAGE USE VALUES IN DATA SET
0004    C       SNKSTO'YR'.USE.          RJS  -  APRIL, 1989.
0005    C
0006    C*************************************************************************
0007    C
0008          DIMENSION USED(400),ADJU(400),BANK(400),RFWB(400)
0009          DIMENSION ID(400),IK(400),DNAME(400)
0010          CHARACTER*14 DNAME
0011          DATA BLANK/' '/
0012          CALL ASSIGN (1,'USEIN')
0013          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0014          PRINT *,' '
0015          IANS=BLANK
0016    C
0017    C***** READ COMPLETE FILE OF EXISTING STORAGE USE AND OTHER VALUES
0018    C
0019          DO 9 I=1,400
0020        9 READ(1,2,END=5) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0021         1 RFWB(I)
0022        2 FORMAT(I8,A1,A14,4F10.0)
0023        5 NDIVNS=I-1
0024    C
0025    C***** INQUIRE FOR CHANGES TO SPACE TABLE
0026    C
0027    C
0028       12 PRINT *,' '
0029          PRINT *,' ENTER CHANGES YOU WISH FOR THESE STORAGE WATER USED. '
0030          PRINT *,' '
0031          PRINT *,' ANSWER    Y = YES     N OR HIT RETURN KEY = NO          '
0032          PRINT *,'           Q = QUIT    J = JUMP AROUND                    '
0033          PRINT *,' '
0034          JUMP=0.0
0035       13 DO 101 I=1,NDIVNS
0036          IF(ID(I).LT.JUMP) GO TO 101
0037          IF(IK(I).EQ.'S') GO TO 101
0038        6 TYPE 11, ID(I),DNAME(I),USED(I)
0039       11 FORMAT(' ',I8,1X,A14,' STORAGE WATER USED  =',F10.1,' NEW VALUE? Y
0040         1/N/J/Q '$)
0041          ACCEPT 102, IANS
0042      102 FORMAT(A1)
0043          IF(IANS.EQ.'Q') GO TO 100
0044          IF(IANS.EQ.'J') GO TO 3
0045          IF(IANS.NE.'Y') GO TO 101
0046          PRINT 103
0047      103 FORMAT('                              ENTER NEW VALUE   ='$)
0048          READ(5,4,ERR=106) USED(I)
0049          FORMAT(F10.0)
0050          GO TO 101
0051      106 PRINT 107
0052      107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0053          GO TO 6
0054      101 CONTINUE
0055          GO TO 100
0056        3 PRINT *,' '
0057          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
```

```
0058          PRINT *,'                 46500 BEGINS FALL RIVER         '
0059          PRINT *,'                 49500 BEGINS LOWER HENRYS FORK  '
0060          PRINT *,'                 53000 BEGINS TETON RIVER        '
0061          PRINT *,'                 57000 BEGINS SNAKE BELOW HENRYS FORK '
0062          PRINT *,'                 57500 BEGINS WILLOW CREEK '
0063          PRINT *,'                 59000 BEGINS SNAKE BELOW WILLOW CREEK'
0064          PRINT *,'                 75000 BEGINS SNAKE BELOW BLACKFOOT '
0065          PRINT *,'                 99000 BEGINS MISCELLANEOUS USERS '
0066          PRINT *,' '
0067          PRINT 104
0068        104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0069           15'/     ' **** USE ONLY LAST 5 DIGITS.                    '$)
0070          ACCEPT 105, JUMP
0071        105 FORMAT(I5)
0072          JUMP=JUMP+13000000
0073          IF(JUMP.GT.13090000) JUMP=JUMP+86900000
0074          GO TO 13
0075      C
0076      C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0077      C***** ID NUMBERS AND STORAGE ACCOUNT VALUES TO NEW FILE.
0078      C
0079        100 PRINT 8
0080          8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE USE VALUES?
0081           1  Y/N '$)
0082          ACCEPT 102, IANS
0083          IF(IANS.EQ.'Y') GO TO 12
0084          DO 1 I=1,NDIVNS
0085         1 WRITE(2,18) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0086          1 RFWB(I)
0087         18 FORMAT(I8,A1,A14,4F10.1)
0088          PRINT *,' '
0089          PRINT *,' FILE SNKSTO''YR''.USE HAS BEEN UPDATED WITH NEW VALUES'
0090          PRINT *,' '
0091          PRINT *,' '
0092          PRINT *,' '
0093          STOP
0094          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1381	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1052	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	15480	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	17913	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGSU$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
**	R*4	BLANK		11D	15					
**	I*4	I		19=	20(7)	23	35=	36	37	38(3) 48
					84=	85(7)				
2-00003B60	I*4	IANS		15=	41=	43	44	45	82=	83
2-00003B64	I*4	JUMP		34=	36	70=	72(2)=	73(3)=		
**	I*4	NDIVNS		23=	35	84				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000640	R*4	ADJU		1600	(400)	8	20=	85		
2-00000C80	R*4	BANK		1600	(400)	8	20=	85		
2-00002580	CHAR	DNAME		5600	(400)	9	10	20=	38	85
2-00001900	I*4	ID		1600	(400)	9	20=	36	38	85
2-00001F40	I*4	IK		1600	(400)	9	20=	37	85	
2-000012C0	R*4	RFWB		1600	(400)	8	20=	85		
2-00000000	R*4	USED		1600	(400)	8	20=	38	48=	85

LABELS

Address	Label	References	
**	1	84	85#
1-000002AF	2'	20	22#
0-0000027C	3	44	56#
1-0000032C	4'	48	49#
0-000000BF	5	20	23#
**	6	38#	53
1-000003D3	8'	79	80#

```
     **      9                              19        20#
  1-000002BB 11'                            38        39#
  0-000000C4 12                             28#       83

  0-0000016C 13                             35#       74
  1-00000410 18'                            85        87#
  0-0000040A 100                            43        55        79#
  0-00000404 101                            35        36        37        45        50        54#
  1-000002F7 102'                           41        42#       82

  1-000002FA 103'                           46        47#
  1-0000035D 104'                           67        68#
  1-000003D0 105'                           70        71#
  0-00000260 106                            48        51#
  1-00000330 107'                           51        52#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	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 SNKCHGSU

/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.SNAKE]SNKCHGSU.LIS;1
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGSU.OBJ;1
```

COMPILATION STATISTICS

Run Time:	5.72 seconds
Elapsed Time:	8.24 seconds
Page Faults:	328
Dynamic Memory:	430 pages

SNPCHGST.LIS

SNKCHGST.LIS

```
0001    C*************************************************************************
0002    C
0003    C        PROGRAM TO CHANGE VALUES IN DATA SET SNKSTO.IND.
0004    C       THESE VALUES ARE THE INITIAL RESERVOIR FILL AND
0005    C       EITHER ESTIMATED OR ACTUAL EVAPORATION. RJS - OCTOBER 1988.
0006    C
0007    C*************************************************************************
0008
0009          DIMENSION ID(12),RNAME(12),SPACE(12),FILL(12),EVAP(12),FLATE(12)
0010          CHARACTER*16 RNAME
0011          CALL ASSIGN (1,'RESIN')
0012          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0013          CALL ASSIGN (3,'OLDRES')
0014          READ(1,2,END=8,ERR=8) (ID(K),RNAME(K),SPACE(K),FILL(K),
0015         1 EVAP(K),FLATE(K),K=1,12)
0016        2 FORMAT(I8,1X,A16,4F10.0)
0017          GO TO 1
0018        8 PRINT *,' '
0019          PRINT *,' END OF FILE ON RESERVOIR FILE OR NO FILE EXISTS - '
0020          PRINT *,' CREATING NEW FILE FROM PREVIOUS YEAR''S FILE '
0021          PRINT *,' '
0022          READ(3,2,END=9,ERR=9) (ID(K),RNAME(K),SPACE(K),FILL(K),
0023         1 EVAP(K),FLATE(K),K=1,12)
0024          GO TO 1
0025        9 PRINT *,' '
0026          PRINT *,' END OF FILE ON LAST YEAR''S RESERVOIR FILE - ABORT '
0027          PRINT *,' '
0028          GO TO 6
0029        1 PRINT *,' '
0030          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR SPACE VAL
0031         1UES  '
0032          PRINT *,' '
0033          DO 105 I=1,12
0034          TYPE 11, RNAME(I),SPACE(I)
0035       11 FORMAT(' ',A16,' =',F10.1,' NEW VALUE? Y/N '$)
0036          ACCEPT 102, IANS
0037      102 FORMAT(A1)
0038          IF(IANS.NE.'Y') GO TO 105
0039          PRINT 103
0040      103 FORMAT(' ENTER NEW VALUE  ='$)
0041          ACCEPT 4, SPACE(I)
0042        4 FORMAT(F10.0)
0043      105 CONTINUE
0044          PRINT *,' '
0045          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR FILL VALU
0046         1ES  '
0047          PRINT *,' '
0048          DO 101 I=1,12
0049          TYPE 11, RNAME(I),FILL(I)
0050          ACCEPT 102, IANS
0051          IF(IANS.NE.'Y') GO TO 101
0052          PRINT 103
0053          ACCEPT 4, FILL(I)
0054      101 CONTINUE
0055          PRINT *,'                                                        '
0056          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR EVAPORATI
0057         1ON VALUES '
```

```
0058          PRINT *,' '
0059          DO 104 I=1,12
0060          TYPE 11, RNAME(I),EVAP(I)
0061          ACCEPT 102, IANS
0062          IF(IANS.NE.'Y') GO TO 104
0063          PRINT 103
0064          ACCEPT 4, EVAP(I)
0065      104 CONTINUE
0066          PRINT *,' '
0067          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR LATE FILL
0068         1 VALUES  '
0069          PRINT *,' '
0070          DO 106 I=1,12
0071          TYPE 11, RNAME(I),FLATE(I)
0072          ACCEPT 102, IANS
0073          IF(IANS.NE.'Y') GO TO 106
0074          PRINT 103
0075          ACCEPT 4, FLATE(I)
0076      106 CONTINUE
0077          PRINT *,'                                                    '
0078          PRINT 5
0079        5 FORMAT(' DO YOU WISH TO CHECK OR REENTER THESE VALUES? Y/N '$)
0080          ACCEPT 102, IANS
0081          IF(IANS.EQ.'Y') GO TO 1
0082          WRITE(2,7) (ID(K),RNAME(K),SPACE(K),FILL(K),EVAP(K),FLATE(K),
0083         1 K=1,12)
0084        7 FORMAT(I8,1X,A16,4F10.1)
0085          PRINT *,' '
0086        6 STOP
0087          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1759	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	632	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	740	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3131	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGSI$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
**	I*4	I		33=	34(2)	41	48=	49(2)	53	59=	60(2)
					64	70=	71(2)	75			
2-000001B4	I*4	IANS		36=	38	50=	51	61=	62	72=	73
					80=	81					
2-000001B0	I*4	K		14(7)=	22(7)=	82(7)=					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000090	R*4	EVAP		48	(12)	9	14=	22=	60	64=
								82		
2-00000060	R*4	FILL		48	(12)	9	14=	22=	49	53=
								82		
2-000000C0	R*4	FLATE		48	(12)	9	14=	22=	71	75=
								82		
2-00000000	I*4	ID		48	(12)	9	14=	22=	82	
2-000000F0	CHAR	RNAME		192	(12)	9	10	14=	22=	34
								49	60	71 8
2-00000030	R*4	SPACE		48	(12)	9	14=	22=	34	41=
								82		

LABELS

Address	Label	References				
0-00000184	1	17	24	29#	81	
1-000001EB	2'	14	16#	22		
1-00000231	4'	41	42#	53	64	75
1-00000235	5'	78	79#			
0-000006D8	6	28	86#			

```
  1-0000026C  7'                82        84#
  0-000000A0  8                 14(2)     18#
  0-00000684  9                 22(2)     25#
  1-000001F7  11'               34        35#       49        60        71
  0-00000383  101               48        51        54#

  1-00000217  102'              36        37#       50        61        72        80
  1-0000021A  103'              39        40#       52        63        74
  0-00000487  104               59        62        65#
  0-0000027F  105               33        38        43#
  0-0000058B  106               70        73        76#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                              References

        ASSIGN                             11        13
        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 SNKCHGSI

  /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.SNAKE]SNKCHGSI.LIS;3
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGSI.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           6.13 seconds
  Elapsed Time:       9.90 seconds
  Page Faults:        720
  Dynamic Memory:     437 pages
```

```
0001    C     ****************************************************************
0002    C
0003    C             PROGRAM SNKWRAPUR - WATER RIGHT ACCOUNTING PURGE
0004    C
0005    C       THIS PROGRAM SELECTS FROM THE HISTORY FILE (SNKWRA.HST) THE
0006    C       DATA NEEDED FOR THE WATER RIGHT ACCOUNTING FOR THE NEW YEAR.
0007    C       THE DATA ARE BROKEN INTO TWO SEPARATE FILES, ONE FROM OCTOBER 1
0008    C       OF THE PREVIOUS YEAR TO OCTOBER 31 OF THE CURRENT YEAR; AND ONE
0009    C       FROM OCTOBER 1 OF THE CURRENT YEAR.  THE PREVIOUS YEAR FILE IS
0010    C       NAMED SNKAYR.HST (WHICH SHOULD LATER BE RENAMED WITH YR EQUAL TO
0011    C       THE PROPER YEAR) AND THE NEW FILE IS SNKWRA.PUR (WHICH WILL
0012    C       LATER BE RENAMED TO SNKWRA.HST).
0013    C
0014    C                       BOB SUTTER  JULY 31,1984
0015    C
0016    C                       REVISED FOR SNAKE RIVER MAY 12,1989
0017    C
0018    C     ****************************************************************
0019    C
0020    C
0021          DIMENSION JD(12),IYD(3)
0022          CHARACTER*28 VALU
0023          CHARACTER*7 VAL(4)
0024          INTEGER*4 ISTA,IDATE
0025          EQUIVALENCE (VALU,VAL)
0026          DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
0027          CALL ASSIGN(1,'HISTORY')
0028          OPEN (UNIT=2,NAME='OLDHIST',TYPE='NEW',CARRIAGECONTROL='LIST')
0029          OPEN (UNIT=3,NAME='NEWHIST',TYPE='NEW',CARRIAGECONTROL='LIST')
0030    C
0031          PRINT 1
0032        1 FORMAT (' ')
0033          PRINT *,' ******************************************************'
0034          PRINT *,' '
0035          PRINT *,'  THIS PROCEDURE CREATES TWO FILES FROM THE HISTORY '
0036          PRINT *,'  FILE, ONE WITH DATA FROM OCTOBER 1 OF THE PREVIOUS '
0037          PRINT *,'  YEAR TO OCTOBER 31 OF THE CURRENT YEAR, AND ONE FROM '
0038          PRINT *,'  OCTOBER 1 OF THE CURRENT YEAR.  THE LATTER FILE '
0039          PRINT *,'  WILL BECOME THE NEW HISTORY FILE.'
0040          PRINT *,' '
0041          PRINT *,' ******************************************************'
0042          PRINT 1
0043    C
0044        5 PRINT *,'  ONE MORE TIME - PLEASE ENTER THE'
0045          PRINT *,'  IRRIGATION YEAR FOR WHICH YOU ARE CREATING'
0046          PRINT *,'  AN ARCHIVE FILE.  THIS IS USUALLY LAST YEAR.   '
0047          PRINT *,'  FOR EXAMPLE : 1989  '
0048          PRINT 1
0049          PRINT *,'                ????  '
0050          PRINT 35
0051       35 FORMAT('+                '$)
0052          ACCEPT 2, IY2
0053        2 FORMAT(I4)
0054          PRINT 1
0055          PRINT 3, IY2
0056        3 FORMAT (1X,' THE ARCHIVE ACCOUNTING YEAR IS ',I4/
0057         !   2X,'IS THIS CORRECT?  ANSWER Y/N/C.  ')
```

```
0058          PRINT 36
0059       36 FORMAT(28X,' '$)
0060          ACCEPT 4,IANSWR
0061        4 FORMAT (1A1)
0062          PRINT 1
0063          IF (IANSWR.EQ.'C') GO TO 10
0064          IF (IANSWR.EQ.'N') GO TO 5
0065     C
0066          IY1=IY2-1
0067          ID1=1
0068          ID2=31
0069          IM1=10
0070          IM2=10
0071        8 CONTINUE
0072     C
0073     C***** CONVERT BEGINNING AND ENDING DATES TO SEVEN DIGIT DATE YYYYDDD,
0074     C***** WHERE YYYY IS THE YEAR AND DDD IS THE NUMERICAL DAY OF THE YEAR.
0075     C
0076          IM=IM1
0077          ID=ID1
0078          IY=IY1
0079          DO 13 K=1,2
0080          FEB=IY/4.0
0081          IFEB=FEB
0082          FEB=FEB-IFEB
0083          NDY=0
0084          DO 11 I=1,IM
0085          IF(I.EQ.IM) GO TO 12
0086          IF(I.EQ.2.AND.FEB.LT.0.001) JD(2)=29
0087       11 NDY=NDY+JD(I)
0088       12 JD(2)=28
0089          NDY=NDY+ID
0090          IYD(K)=(IY*1000)+NDY
0091          IF(K.EQ.2) GO TO 13
0092          IM=IM2
0093          ID=ID2
0094          IY=IY2
0095       13 CONTINUE
0096          IY=IYD(1)/1000
0097          IF(IY1.EQ.IY) GO TO 15
0098          FEB=IY/4.0
0099          IFEB=FEB
0100          FEB=FEB-IFEB
0101          J=635
0102          IF(FEB.LT.0.001) J=634
0103          IYD(1)=IYD(1)-J
0104       15 CONTINUE
0105          IYD(3)=IYD(2)-30
0106          PRINT 16, IYD(1),IYD(2),IYD(3)
0107       16 FORMAT(3X,'ALL DATA FROM DAY ',I8,' TO DAY ', I8,' IS BEING RETRIE
0108         !VED FROM'/3X,'THE HISTORY FILE TO CREATE A HISTORY ARCHIVE FILE,'/
0109         !3X,'AND ALL DATA FROM DAY ',I8,' IS BEING SELECTED FOR THE NEW  '/
0110         !3X,'(PURGED) HISTORY FILE')
0111          PRINT *,'  SO PLEASE HANG ON FOR A WHILE. '
0112          PRINT 1
0113     C
0114     C***** SELECT ALL DATA FROM HISTORY FILE FROM DATE IYD(1) THROUGH IYD(2)
```

```
0115    C***** AND WRITE DATA TO THE ARCHIVE FILE.  ALSO SELECT ALL DATA FROM
0116    C***** HISTORY FILE FROM DATE IYD(3) FOR NEW (PURGED) HISTORY FILE.
0117    C
0118          IARCH=0
0119          IPHST=0
0120          IERR=0
0121       19 READ(1,17,END=18,ERR=24) ISTA,S,IDATE,VALU
0122       17 FORMAT(I8,A1,I7,A28)
0123          IF(IDATE.LT.IYD(1)) GO TO 19
0124          IF(IDATE.GT.IYD(2)) GO TO 29
0125          IF(S.EQ.'R') WRITE(2,17) ISTA,S,IDATE,VALU
0126          IF(S.EQ.'F') WRITE(2,27) ISTA,S,IDATE,(VAL(I),I=1,2)
0127       27 FORMAT(I8,A1,I7,2A7)
0128          IF(S.EQ.'D') WRITE(2,17) ISTA,S,IDATE,VALU
0129          IF(S.EQ.'P') WRITE(2,17) ISTA,S,IDATE,VALU
0130          IF(S.EQ.'E') WRITE(2,28) ISTA,S,IDATE,VAL(1)
0131       28 FORMAT(I8,A1,I7,A7)
0132          IARCH=IARCH+1
0133          GO TO 23
0134       24 IERR=IERR+1
0135          PRINT *,'  AN ERROR HAS OCCURRED DURING HISTORY FILE READ.'
0136          PRINT *,'  EXECUTION CONTINUING.'
0137          IF(IERR.LE.10) GO TO 19
0138          PRINT *,'  THE NUMBER OF ERRORS IN READING THE HISTORY FILE'
0139          PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
0140          GO TO 10
0141       23 IF(IDATE.LT.IYD(3)) GO TO 19
0142       29 IF(S.EQ.'R') WRITE(3,17) ISTA,S,IDATE,VALU
0143          IF(S.EQ.'F') WRITE(3,27) ISTA,S,IDATE,(VAL(I),I=1,2)
0144          IF(S.EQ.'D') WRITE(3,17) ISTA,S,IDATE,VALU
0145          IF(S.EQ.'P') WRITE(3,17) ISTA,S,IDATE,VALU
0146          IF(S.EQ.'E') WRITE(3,28) ISTA,S,IDATE,VAL(1)
0147          IPHST=IPHST+1
0148          GO TO 19
0149       18 PRINT 1
0150          PRINT *,'  OUTPUT FILES OF ARCHIVE HISTORY RECORDS AND PURGED '
0151          PRINT *,'  HISTORY RECORDS (BEGINNING OCTOBER 1, HAVE NOW'
0152          PRINT *,'  BEEN CREATED.'
0153          PRINT *,'  '
0154          TYPE 25, IARCH,IPHST
0155       25 FORMAT(/2X,'ARCHIVE HISTORY FILE CONTAINS    ',I10,' RECORDS'/
0156         !         2X,'PURGED(NEW) HISTORY FILE CONTAINS',I10,' RECORDS')
0157          PRINT *,' '
0158       10 STOP
0159          END
```

MODIFIED 7/23/90 TO MOVE ALL 4 FIELDS OF RESERVOIR RECORDS — TO CAPTURE AMERICAN FALLS PPT.

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	2223	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1275	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	444	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3942	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTPUR$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	FEB		80= 81 82(2)= 86 98= 99 100(2)= 102
**	I*4	I		84= 85 86 87 126(2)= 143(2)=
2-00000064	I*4	IANSWR		60= 63 64
**	I*4	IARCH		118= 132(2)= 154
**	I*4	ID		77= 89 93=
**	I*4	ID1		67= 77
**	I*4	ID2		68= 93
2-0000005C	I*4	IDATE		24 121= 123 124 125 126 128 129 130 141 142 143 144 145 146
**	I*4	IERR		120= 134(2)= 137
**	I*4	IFEB		81= 82 99= 100
**	I*4	IM		76= 84 85 92=
**	I*4	IM1		69= 76
**	I*4	IM2		70= 92
**	I*4	IPHST		119= 147(2)= 154
2-00000058	I*4	ISTA		24 121= 125 126 128 129 130 142 143 144 145 146
**	I*4	IY		78= 80 90 94= 96= 97 98
**	I*4	IY1		66= 78 97
2-00000060	I*4	IY2		52= 55 66 94
**	I*4	J		101= 102= 103
**	I*4	K		79= 90 91
**	I*4	NDY		83= 87(2)= 89(2)= 90
2-0000006C	R*4	S		121= 125(2) 126(2) 128(2) 129(2) 130(2) 142(2) 143(2) 144(2) 145(2) 146(2)
2-00000000	CHAR	VALU	EQUIV	22 25 121= 125 128 129 142 144 145

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-0000004C	I*4	IYD		12	(3)	21 90= 96 103(2)= 105(2)= 106(3) 123 124 141
2-0000001C	I*4	JD		48	(12)	21 26D 86= 87 88=
2-00000000	CHAR	VAL	EQUIV	28	(4)	23 25 126 130 143 146

LABELS

Address	Label	References
1-00000337	1'	31 32# 42 48 54 62 112 149
1-00000350	2'	52 53#
1-00000353	3'	55 56#
1-000003A7	4'	60 61#
0-0000014C	5	44# 64
**	8	71#
0-000008A8	10	63 140 158#
**	11	84 87#
0-00000320	12	85 88#
0-00000342	13	79 91 95#
0-0000037F	15	97 104#
1-000003AA	16'	106 107#
1-00000479	17'	121 122# 125 128 129 142 144 145
0-000007DC	18	121 149#
0-000003F4	19	121# 123 137 141 148
**	23	133 141#
0-0000075C	24	121 134#
1-00000496	25'	154 155#
1-00000482	27'	126 127# 143
1-0000048D	28'	130 131# 146
0-000005DF	29	124 142#
1-0000033B	35'	50 51#
1-000003A0	36'	58 59#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	ASSIGN	27
	FOR$OPEN	28 29

```
+-----------------------------------------------+
|           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 SNKHSTPUR

  /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.SNAKE]SNKHSTPUR.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTPUR.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           5.94 seconds
  Elapsed Time:       10.02 seconds
  Page Faults:        822
  Dynamic Memory:     500 pages
```

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES WATER BANK TRANSACTIONS IN DATA SET
0004    C       SNKSTO'YR'.USE.  ENTER POSITIVE VALUES FOR THOSE RECEIVING WATER
0005    C       THRU THE WATER BANK , AND NEGATIVE VALUES FOR THOSE SUPPLYING
0006    C       WATER TO THE BANK.   RJS - OCTOBER, 1988.
0007    C
0008    C*************************************************************************
0009    C
0010          DIMENSION USED(400),ADJU(400),BANK(400),RFWB(400)
0011          DIMENSION ID(400),IK(400),DNAME(400)
0012          CHARACTER*14 DNAME
0013          DATA BLANK/' '/
0014          CALL ASSIGN (1,'USEIN')
0015          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0016          PRINT *,' '
0017          IANS=BLANK
0018    C
0019    C***** READ COMPLETE FILE OF EXISTING STORAGE USE AND OTHER VALUES
0020    C
0021          DO 9 I=1,400
0022        9 READ(1,2,END=5) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0023         1 RFWB(I)
0024        2 FORMAT(I8,A1,A14,4F10.0)
0025        5 NDIVNS=I-1
0026    C
0027    C***** INQUIRE FOR CHANGES TO SPACE TABLE
0028    C
0029    C
0030       12 PRINT *,' '
0031          PRINT *,' ENTER CHANGES YOU WISH FOR THESE WATER BANK VALUES: '
0032          PRINT *,' ENTER POSITIVE VALUES FOR THOSE REQUESTING STORAGE,  '
0033          PRINT *,' ENTER NEGATIVE VALUES FOR THOSE SUPPLYING STORAGE.  '
0034          PRINT *,' '
0035          PRINT *,' ANSWER   Y = YES    N OR HIT RETURN KEY = NO         '
0036          PRINT *,'          Q = QUIT   J = JUMP AROUND                   '
0037          PRINT *,' '
0038          JUMP=0.0
0039       13 DO 101 I=1,NDIVNS
0040          IF(ID(I).LT.JUMP) GO TO 101
0041          IF(IK(I).EQ.'S') GO TO 101
0042        6 TYPE 11, ID(I),DNAME(I),BANK(I)
0043       11 FORMAT(' ',I8,1X,A14,' WATER BANK TRANSFER =',F10.1,' NEW VALUE? Y
0044         1/N/J/Q '$)
0045          ACCEPT 102, IANS
0046      102 FORMAT(A1)
0047          IF(IANS.EQ.'Q') GO TO 100
0048          IF(IANS.EQ.'J') GO TO 3
0049          IF(IANS.NE.'Y') GO TO 101
0050          PRINT 103
0051      103 FORMAT('                                ENTER NEW VALUE   ='$)
0052          READ(5,4,ERR=106) BANK(I)
0053        4 FORMAT(F10.0)
0054          GO TO 101
0055      106 PRINT 107
0056      107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0057          GO TO 6
```

```
0058      101 CONTINUE
0059          GO TO 100
0060        3 PRINT *,' '
0061          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0062          PRINT *,'                 46500 BEGINS FALL RIVER         '
0063          PRINT *,'                 49500 BEGINS LOWER HENRYS FORK  '
0064          PRINT *,'                 53000 BEGINS TETON RIVER        '
0065          PRINT *,'                 57000 BEGINS SNAKE BELOW HENRYS FORK '
0066          PRINT *,'                 57500 BEGINS WILLOW CREEK '
0067          PRINT *,'                 59000 BEGINS SNAKE BELOW WILLOW CREEK'
0068          PRINT *,'                 75000 BEGINS SNAKE BELOW BLACKFOOT '
0069          PRINT *,' '
0070          PRINT 104
0071      104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0072         15'/     ' **** USE ONLY LAST 5 DIGITS.                    '$)
0073          ACCEPT 105, JUMP
0074      105 FORMAT(I5)
0075          JUMP=JUMP+13000000
0076          IF(JUMP.GT.13090000) JUMP=JUMP+86900000
0077          GO TO 13
0078      C
0079      C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0080      C***** ID NUMBERS AND STORAGE ACCOUNT VALUES TO NEW FILE.
0081      C
0082        100 PRINT 8
0083          8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE SPACE VALUES?
0084           1  Y/N '$)
0085           ACCEPT 102, IANS
0086           IF(IANS.EQ.'Y') GO TO 12
0087           DO 1 I=1,NDIVNS
0088         1 WRITE(2,18) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0089          1 RFWB(I)
0090        18 FORMAT(I8,A1,A14,4F10.1)
0091           PRINT *,' '
0092           PRINT *,' FILE SNKSTO''YR''.USE HAS BEEN UPDATED WITH NEW VALUES'
0093           PRINT *,' '
0094           PRINT *,' '
0095           PRINT *,' '
0096           STOP
0097           END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1409	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1110	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	15488	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	18007	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGWB$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
**	R*4	BLANK		13D	17					
**	I*4	I		21=	22(7)	25	39=	40	41	42(3) 52
					87=	88(7)				
2-00003B60	I*4	IANS		17=	45=	47	48	49	85=	86
2-00003B64	I*4	JUMP		38=	40	73=	75(2)=	76(3)=		
**	I*4	NDIVNS		25=	39	87				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000640	R*4	ADJU		1600	(400)	10	22=	88		
2-00000C80	R*4	BANK		1600	(400)	10	22=	42	52=	88
2-00002580	CHAR	DNAME		5600	(400)	11	12	22=	42	88
2-00001900	I*4	ID		1600	(400)	11	22=	40	42	88
2-00001F40	I*4	IK		1600	(400)	11	22=	41	88	
2-000012C0	R*4	RFWB		1600	(400)	10	22=	88		
2-00000000	R*4	USED		1600	(400)	10	22=	88		

LABELS

Address	Label	References	
**	1	87	88#
1-000002E7	2'	22	24#
0-000002B4	3	48	60#
1-00000364	4'	52	53#
0-000000BF	5	22	25#
**	6	42#	57
1-0000040B	8'	82	83#

```
     **      9                          21        22#
  1-000002F3 11'                        42        43#
  0-000000C4 12                         30#       86

  0-000001A4 13                         39#       77
  1-0000044A 18'                        88        90#
  0-00000426 100                        47        59        82#
  0-00000420 101                        39        40        41        49        54        58#
  1-0000032F 102'                       45        46#       85

  1-00000332 103'                       50        51#
  1-00000395 104'                       70        71#
  1-00000408 105'                       73        74#
  0-00000298 106                        52        55#
  1-00000368 107'                       55        56#
```

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 SNKCHGWB

  /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.SNAKE]SNKCHGWB.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGWB.OBJ;1
```

COMPILATION STATISTICS

Run Time:	5.62 seconds
Elapsed Time:	7.97 seconds
Page Faults:	357
Dynamic Memory:	438 pages

From: DWR70::GINGERICH 27-APR-1990 11:19:29.98
To: DWR01::SUTTER
CC:
Subj: SNKSTO. AND SNKGETSU. COMPATIBILITY

B,

WHILE WORKING WITH THE STORAGE ALLOCATIONS PROGRAM WE'VE DISCOVERED A PROBLEM CONCERNING THE SNKSTO'IY'.USE FILE AND HOW ITS ALTERED WITHIN SNKGETSU., THE PROGRAM THAT RETRIEVES STORAGE USE VALUES FROM THE ALLOCATIONS FILE.

SNKGETSU. CREATES SNKSTO'IY'.USE WITH FOUR COLUMNS OF VALUES -- IDENTIFICATION #, DIVERSION NAME, STORAGE USED (OR ZERO), AND ZERO. SNKSTO. LOOKS FOR SNKSTO'IY'.USE TO HAVE SIX VALUES (IT ONLY READS FIVE OF THEM, PASSING OVER DIVERSION NAME) -- IDENTIFICATION #, DIVERSION NAME, STORAGE USED, ADJUSTMENTS, WATER BANK SUPPLY OR PURCHASE, AND WATER BANK WATER REVERTED TO THE SPACEHOLDER FROM THE WATER BANK. THEREFORE, IF WE USE SNKGETSU. AFTER WE'VE MADE WATER BANK ACTIVITY AND OTHER ADJUSTMENTS SNKSTO'IY'.USE IS ALTERED SUCH THAT THOSE CHANGES ARE LOST; THEY BECOME ZERO SINCE THE LAST COLUMN OF SNKSTO'IY'.USE CREATED BY SNKGETSU. IS ZERO AND THE NEXT TWO COLUMNS SNKSTO. LOOKS FOR DO NOT EXIST.

THIS WOULDN'T BE A FATAL PROBLEM IF STORAGE USED VALUES ARE FINALIZED BEFORE STORAGE CALCULATIONS ARE PERFORMED, BUT I IMAGINE THIS AS A RARE CASE. IN ORDER TO HAVE SNKGETSU. CREATE THE PROPERLY FORMATTED .USE FILE IT WILL HAVE TO READ WATER BANK AND OTHER ADJUSTMENTS AND THEN WRITE THEM OUT WITH THE NEW STORAGE USED VALUES.

```
0001    C
0002    C             PROGRAM SNKGETSU - STORAGE USE SELECT
0003    C
0004    C       THIS PROGRAM SELECTS FROM THE ALLOCATIONS FILE THE STOARGE
0005    C       USED BY EACH USER ENTITY ON THE SELECTED DATE.  THESE DATA
0006    C       ARE PLACED IN A FILE CALLED SNKSTO'YR'.USE FOR INPUT TO
0007    C       STORAGE ACCOUNTING PROGRAM.
0008    C
0009    C                       BOB SUTTER  OCTOBER 1988
0010    C
0011    C     ****************************************************************
0012    C
0013    C
0014          DIMENSION OM(12),JD(12)
0015          CHARACTER*14 DNAME
0016          DATA OM/'OCT','NOV','DEC','JAN','FEB','MAR','APR','MAY','JUN',
0017         !  'JUL','AUG','SEP'/
0018          DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
0019          DATA BLANK/' '/
0020          CALL ASSIGN(1,'SNKWRA.ALC')
0021          OPEN (UNIT=2,NAME='USEIN',TYPE='NEW',CARRIAGECONTROL='LIST')
0022          CALL ASSIGN(3,'SPACEIN')
0023          PRINT 1
0024        1 FORMAT (' ')
0025          PRINT *,' ************************************************************'
0026          PRINT *,' '
0027          PRINT *,'  THIS PROCEDURE SELECTS THE STORAGE USE DATA FROM THE  '
0028          PRINT *,'  LATEST ALLOCATION FILE FOR INPUT TO THE STORAGE '
0029          PRINT *,'  ACCOUNTING PROGRAM.  ENTER THE DATE AND ANSWER THE    '
0030          PRINT *,'  FOLLOWING QUESTIONS USING Y FOR YES, N FOR NO, AND C  '
0031          PRINT *,'  IF YOU WISH TO CANCEL ALL.  '
0032          PRINT *,' '
0033          PRINT *,' ************************************************************'
0034          PRINT 1
0035    C
0036        5 PRINT *,'  ENTER THE MONTH, DAY, AND YEAR FOR THE DAY YOU WANT TO'
0037          PRINT *,'  RETRIEVE THE STORED WATER USES.  '
0038          PRINT *,'  FOR EXAMPLE : 06-12-1989  '
0039          PRINT 1
0040          PRINT *,'                ??????????  '
0041          PRINT 35
0042       35 FORMAT('+                '$)
0043          ACCEPT 2, IM,ID,IY
0044        2 FORMAT(I2,1X,I2,1X,I4)
0045          PRINT 1
0046          PRINT 3, IM,ID,IY
0047        3 FORMAT (1X,' THE STORAGE USE DATE IS ',I2,'-',I2,'-',I4/
0048         !   2X,'IS THIS CORRECT?  ANSWER Y/N/C.  ')
0049          PRINT 36
0050       36 FORMAT(28X,' '$)
0051          ACCEPT 4,IANSWR
0052        4 FORMAT (1A1)
0053          PRINT 1
0054          IF (IANSWR.EQ.'C') GO TO 100
0055          IF (IANSWR.EQ.'N') GO TO 5
0056          ZERO=0.0
0057    C
```

```
0058     C***** CONVERT DATE TO SEVEN DIGIT DATE YYYYDDD,
0059     C***** WHERE YYYY IS THE YEAR AND DDD IS THE NUMERICAL DAY OF THE YEAR.
0060     C
0061           FEB=IY/4.0
0062           IFEB=FEB
0063           FEB=FEB-IFEB
0064           NDY=0
0065           DO 11 I=1,IM
0066           IF(I.EQ.IM) GO TO 12
0067           IF(I.EQ.2.AND.FEB.LT.0.001) JD(2)=29
0068        11 NDY=NDY+JD(I)
0069        12 JD(2)=28
0070           NDY=NDY+ID
0071           IYD=(IY*1000)+NDY
0072           PRINT 16, IYD
0073        16 FORMAT(3X,'ALL STORAGE USE DATA FROM DAY ',I8,' IS BEING RETRIEVED
0074          1 FROM'/3X,'THE LATEST ALLOCATIONS FILE, SO PLEASE HANG ON FOR A WH
0075          2ILE. ')
0076           PRINT 1
0077     C
0078     C***** SELECT ALL STORAGE USE DATA FROM ALLOCATIONS FILE FOR DAY IYD AND
0079     C***** WRITE THESE DATA TO FILE SNKSTO.USE.
0080     C
0081           IALK=0
0082           IERR=0
0083        29 READ(1,27,END=28,ERR=34) ID,A,IDATE,USED
0084        27 FORMAT(I8,A1,I7,20X,F10.0)
0085           IF(IDATE.NE.IYD) GO TO 29
0086           IALK=1
0087           IF(A.EQ.'D') GO TO 61
0088           GO TO 29
0089        34 IERR=IERR+1
0090           PRINT *,'  AN ERROR HAS OCCURRED DURING ALLOCATIONS FILE READ.'
0091           PRINT *,'  EXECUTION CONTINUING.'
0092           IF(IERR.LE.10) GO TO 29
0093           PRINT *,'  THE NUMBER OF ERRORS IN READING THE ALLOCATIONS FILE'
0094           PRINT *,'  RECORDS HAS EXCEEDED TEN, EXECUTION TERMINATED.'
0095           GO TO 100
0096        61 READ(3,38) ID2,ISK,DNAME
0097        38 FORMAT(I8,A1,A14)
0098           IF(ID.EQ.ID2) GO TO 31
0099     C
0100     C***** WRITE OUT ZERO'S FOR DIVERSIONS HAVING NO STORAGE OR HAVE
0101     C***** HAVE NOT USED STORAGE.
0102     C
0103           IF(ISK.EQ.'S'.OR.ID2.LT.ID) THEN
0104                                       WRITE(2,30) ID2,DNAME,ZERO,ZERO
0105                                       GO TO 61
0106                                       ENDIF
0107           PRINT 50, ID,ID2
0108           WRITE(2,50) ID,ID2
0109        50 FORMAT(' DIVERSION NO. ',I8,' FROM LAST DAY ACCOUNTING FILE DOES N
0110          1OT EQUAL'/' DIVERSION.NO. ',I8,' FROM SNKSTO.SPA FILE'/)
0111           GO TO 100
0112        31 WRITE(2,30) ID2,DNAME,USED,ZERO
0113        30 FORMAT(I8,' ',A14,2F10.1)
0114           GO TO 29
```

```
0115     C
0116     C***** CHECK IF THERE ARE MORE DIVERSION NAMES ON NAME FILE.
0117     C
0118        28 READ(3,38,END=62) ID2,ISK,DNAME
0119           WRITE(2,30) ID2,DNAME,ZERO,ZERO
0120           GO TO 28
0121     C
0122     C***** IF NO VALUES FOR REQUESTED DAY WERE FOUND, PRINT MESSAGE.
0123     C
0124        62 IF(IALK.EQ.0) GO TO 45
0125           PRINT *,'  '
0126           PRINT *,'  STORAGE USE DATA FROM ALLOCATIONS FILE HAVE BEEN '
0127           PRINT *,'  RETRIEVED AND PLACED IN FILE SNKSTO''YR''.USE.      '
0128           PRINT *,'  '
0129           STOP
0130     C
0131     C***** IF NO DATA WAS FOUND IN ALLOCATIONS FILE, CREATE ZERO FILE.
0132     C
0133        45 PRINT 46, IYD
0134        46 FORMAT(/'  NO DATA FOR DAY ',I8,' WAS FOUND IN ALLOCATIONS.'//)
0135           PRINT *,'  '
0136           PRINT *,'  FILE SNKSTO''YR''.USE CREATED WITH ALL ZERO''S FOR'
0137           PRINT *,'  STORAGE USE VALUES.'
0138           PRINT *,'  '
0139       100 STOP
0140           END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1779	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1290	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	436	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3505	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKGETSU$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
2-00000084	R*4	A		83= 87
**	R*4	BLANK		19D
2-00000060	CHAR	DNAME		15 96= 104 112 118= 119
**	R*4	FEB		61= 62 63(2)= 67
**	I*4	I		65= 66 67 68

**	I*4	IALK	81=	86=	124					
2-0000007C	I*4	IANSWR	51=	54	55					
2-00000074	I*4	ID	43=	46	70	83=	98	103	107	108
2-00000090	I*4	ID2	96=	98	103	104	107	108	112	118=
				119						
2-00000088	I*4	IDATE	83=	85						
**	I*4	IERR	82=	89(2)=	92					
**	I*4	IFEB	62=	63						
2-00000070	I*4	IM	43=	46	65	66				
2-00000094	I*4	ISK	96=	103	118=					
2-00000078	I*4	IY	43=	46	61	71				
**	I*4	IYD	71=	72	85	133				
**	I*4	NDY	64=	68(2)=	70(2)=	71				
2-0000008C	R*4	USED	83=	112						
**	R*4	ZERO	56=	104(2)	112	119(2)				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000030	I*4	JD		48	(12)	14	18D	67=	68	69=
2-00000000	R*4	OM		48	(12)	14	16D			

LABELS

Address	Label	References						
1-00000347	1'	23	24#	34	39	45	53	76
1-00000360	2'	43	44#					
1-0000036B	3'	46	47#					
1-000003C2	4'	51	52#					
0-0000014C	5	36#	55					
**	11	65	68#					
0-00000314	12	66	69#					
1-000003C5	16'	72	73#					
1-00000445	27'	83	84#					
0-00000568	28	83	118#	120				
0-00000364	29	83#	85	88	92	114		
1-000004C7	30'	104	112	113#	119			
0-000004A8	31	98	112#					
0-000004E8	34	83	89#					
1-0000034B	35'	41	42#					
1-000003BB	36'	49	50#					
1-00000451	38'	96	97#	118				
0-0000065C	45	124	133#					
1-000004D4	46'	133	134#					
1-00000458	50'	107	108	109#				
0-000003C0	61	87	96#	105				

```
  0-000005E0  62                          118       124#
  0-000006EC  100                          54        95       111       139#


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                                References

        ASSIGN                                 20        22
        FOR$OPEN                               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 SNKGETSU

  /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.SNAKE]SNKGETSU.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKGETSU.OBJ;1


COMPILATION STATISTICS

  Run Time:           4.72 seconds
  Elapsed Time:       6.87 seconds
  Page Faults:        724
  Dynamic Memory:     442 pages
```

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES STORAGE USE ADJUSTMENTS IN DATA SET
0004    C       SNKSTO'YR'.USE.  ENTER POSITIVE VALUES FOR ADDITIONAL STORAGE,
0005    C       AND NEGATIVE VALUES FOR STORAGE REMOVED.   RJS  -  APRIL, 1989.
0006    C
0007    C*************************************************************************
0008    C
0009          DIMENSION USED(400),ADJU(400),BANK(400),RFWB(400)
0010          DIMENSION ID(400),IK(400),DNAME(400)
0011          CHARACTER*14 DNAME
0012          DATA BLANK/' '/
0013          CALL ASSIGN (1,'USEIN')
0014          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0015          PRINT *,' '
0016          IANS=BLANK
0017    C
0018    C***** READ COMPLETE FILE OF EXISTING STORAGE USE AND OTHER VALUES
0019    C
0020          DO 9 I=1,400
0021        9 READ(1,2,END=5) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0022         1 RFWB(I)
0023        2 FORMAT(I8,A1,A14,4F10.0)
0024        5 NDIVNS=I-1
0025    C
0026    C***** INQUIRE FOR CHANGES TO SPACE TABLE
0027    C
0028    C
0029       12 PRINT *,' '
0030          PRINT *,' ENTER CHANGES YOU WISH FOR THESE STORAGE WATER USED. '
0031          PRINT *,' '
0032          PRINT *,' ANSWER   Y = YES    N OR HIT RETURN KEY = NO         '
0033          PRINT *,'          Q = QUIT   J = JUMP AROUND                   '
0034          PRINT *,' '
0035          JUMP=0.0
0036       13 DO 101 I=1,NDIVNS
0037          IF(ID(I).LT.JUMP) GO TO 101
0038          IF(IK(I).EQ.'S') GO TO 101
0039        6 TYPE 11, ID(I),DNAME(I),ADJU(I)
0040       11 FORMAT(' ',I8,1X,A14,' STORAGE ADJUSTMENT  =',F10.1,' NEW VALUE? Y
0041         1/N/J/Q '$)
0042          ACCEPT 102, IANS
0043      102 FORMAT(A1)
0044          IF(IANS.EQ.'Q') GO TO 100
0045          IF(IANS.EQ.'J') GO TO 3
0046          IF(IANS.NE.'Y') GO TO 101
0047          PRINT 103
0048      103 FORMAT('                              ENTER NEW VALUE   ='$)
0049          READ(5,4,ERR=106) ADJU(I)
0050        4 FORMAT(F10.0)
0051          GO TO 101
0052      106 PRINT 107
0053      107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0054          GO TO 6
0055      101 CONTINUE
0056          GO TO 100
0057        3 PRINT *,' '
```

```
0058          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0059          PRINT *,'                  46500 BEGINS FALL RIVER         '
0060          PRINT *,'                  49500 BEGINS LOWER HENRYS FORK  '
0061          PRINT *,'                  53000 BEGINS TETON RIVER        '
0062          PRINT *,'                  57000 BEGINS SNAKE BELOW HENRYS FORK '
0063          PRINT *,'                  57500 BEGINS WILLOW CREEK '
0064          PRINT *,'                  59000 BEGINS SNAKE BELOW WILLOW CREEK'
0065          PRINT *,'                  75000 BEGINS SNAKE BELOW BLACKFOOT '
0066          PRINT *,'                  99000 BEGINS MISCELLANEOUS USERS '
0067          PRINT *,' '
0068          PRINT 104
0069        104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0070           15'/     ' **** USE ONLY LAST 5 DIGITS.                    '$)
0071          ACCEPT 105, JUMP
0072        105 FORMAT(I5)
0073          JUMP=JUMP+13000000
0074          IF(JUMP.GT.13090000) JUMP=JUMP+86900000
0075          GO TO 13
0076      C
0077      C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0078      C***** ID NUMBERS AND STORAGE ACCOUNT VALUES TO NEW FILE.
0079      C
0080        100 PRINT 8
0081          8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE SPACE VALUES?
0082           1  Y/N '$)
0083          ACCEPT 102, IANS
0084          IF(IANS.EQ.'Y') GO TO 12
0085          DO 1 I=1,NDIVNS
0086         1 WRITE(2,18) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0087           1 RFWB(I)
0088         18 FORMAT(I8,A1,A14,4F10.1)
0089          PRINT *,' '
0090          PRINT *,' FILE SNKSTO''YR''.USE HAS BEEN UPDATED WITH NEW VALUES'
0091          PRINT *,' '
0092          PRINT *,' '
0093          PRINT *,' '
0094          STOP
0095          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1381	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1054	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	15480	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	17915	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGSA$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	BLANK		12D	16						
**	I*4	I		20=	21(7)	24	36=	37	38	39(3)	49
					85=	86(7)					
2-00003B60	I*4	IANS		16=	42=	44	45	46	83=	84	
2-00003B64	I*4	JUMP		35=	37	71=	73(2)=	74(3)=			
**	I*4	NDIVNS		24=	36	85					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000640	R*4	ADJU		1600	(400)	9	21=	39	49=	86
2-00000C80	R*4	BANK		1600	(400)	9	21=	86		
2-00002580	CHAR	DNAME		5600	(400)	10	11	21=	39	86
2-00001900	I*4	ID		1600	(400)	10	21=	37	39	86
2-00001F40	I*4	IK		1600	(400)	10	21=	38	86	
2-000012C0	R*4	RFWB		1600	(400)	9	21=	86		
2-00000000	R*4	USED		1600	(400)	9	21=	86		

LABELS

Address	Label	References	
**	1	85	86#
1-000002AF	2'	21	23#
0-0000027C	3	45	57#
1-0000032C	4'	49	50#
0-000000BF	5	21	24#
**	6	39#	54
1-000003D3	8'	80	81#

```
     **      9                          20        21#
  1-000002BB 11'                        39        40#
  0-000000C4 12                         29#       84

  0-0000016C 13                         36#       75
  1-00000412 18'                        86        88#
  0-0000040A 100                        44        56        80#
  0-00000404 101                        36        37        38        46        51        55#
  1-000002F7 102'                       42        43#       83

  1-000002FA 103'                       47        48#
  1-0000035D 104'                       68        69#
  1-000003D0 105'                       71        72#
  0-00000260 106                        49        52#
  1-00000330 107'                       52        53#
```

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 SNKCHGSA

/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.SNAKE]SNKCHGSA.LIS;6
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGSA.OBJ;1
```

COMPILATION STATISTICS

Run Time:	5.82 seconds
Elapsed Time:	10.69 seconds
Page Faults:	756
Dynamic Memory:	430 pages

SNKCHGRR.LIS

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES IN DATA SET SNKSTO'YR'.USE OF WATER
0004    C       RETURNED TO USERS FROM WATER BANK BECAUSE IT WENT UNUSED.
0005    C                         RJS  -  APRIL, 1989.
0006    C
0007    C*************************************************************************
0008    C
0009          DIMENSION USED(400),ADJU(400),BANK(400),RFWB(400)
0010          DIMENSION ID(400),IK(400),DNAME(400)
0011          CHARACTER*14 DNAME
0012          DATA BLANK/' '/
0013          CALL ASSIGN (1,'USEIN')
0014          OPEN (UNIT=2,NAME='SNKSTO.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0015          PRINT *,' '
0016          IANS=BLANK
0017    C
0018    C***** READ COMPLETE FILE OF EXISTING STORAGE USE AND OTHER VALUES
0019    C
0020          DO 9 I=1,400
0021        9 READ(1,2,END=5) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0022         1 RFWB(I)
0023        2 FORMAT(I8,A1,A14,4F10.0)
0024        5 NDIVNS=I-1
0025    C
0026    C***** INQUIRE FOR CHANGES TO SPACE TABLE
0027    C
0028    C
0029       12 PRINT *,' '
0030          PRINT *,' ENTER CHANGES YOU WISH FOR THESE WATER BANK RETURNS. '
0031          PRINT *,' '
0032          PRINT *,' ANSWER    Y = YES     N OR HIT RETURN KEY = NO          '
0033          PRINT *,'           Q = QUIT    J = JUMP AROUND                    '
0034          PRINT *,' '
0035          JUMP=0.0
0036       13 DO 101 I=1,NDIVNS
0037          IF(ID(I).LT.JUMP) GO TO 101
0038          IF(IK(I).EQ.'S') GO TO 101
0039        6 TYPE 11, ID(I),DNAME(I),RFWB(I)
0040       11 FORMAT(' ',I8,1X,A14,' WATER BANK RETURN   =',F10.1,' NEW VALUE? Y
0041         1/N/J/Q '$)
0042          ACCEPT 102, IANS
0043      102 FORMAT(A1)
0044          IF(IANS.EQ.'Q') GO TO 100
0045          IF(IANS.EQ.'J') GO TO 3
0046          IF(IANS.NE.'Y') GO TO 101
0047          PRINT 103
0048      103 FORMAT('                              ENTER NEW VALUE   ='$)
0049          READ(5,4,ERR=106) RFWB(I)
0050        4 FORMAT(F10.0)
0051          GO TO 101
0052      106 PRINT 107
0053      107 FORMAT(//' ERROR READING INPUT DATA - TRY AGAIN '//)
0054          GO TO 6
0055      101 CONTINUE
0056          GO TO 100
0057        3 PRINT *,' '
```

```
0058          PRINT *,'     DIVERSION # 40000 BEGINS HENRYS FORK        '
0059          PRINT *,'                  46500 BEGINS FALL RIVER        '
0060          PRINT *,'                  49500 BEGINS LOWER HENRYS FORK  '
0061          PRINT *,'                  53000 BEGINS TETON RIVER        '
0062          PRINT *,'                  57000 BEGINS SNAKE BELOW HENRYS FORK '
0063          PRINT *,'                  57500 BEGINS WILLOW CREEK '
0064          PRINT *,'                  59000 BEGINS SNAKE BELOW WILLOW CREEK'
0065          PRINT *,'                  75000 BEGINS SNAKE BELOW BLACKFOOT '
0066          PRINT *,'                  99000 BEGINS MISCELLANEOUS USERS '
0067          PRINT *,' '
0068          PRINT 104
0069        104 FORMAT(/' SELECT DIVERSION # YOU WANT TO JUMP TO, EXAMPLE = 3802
0070           15'/     ' **** USE ONLY LAST 5 DIGITS.                    '$)
0071          ACCEPT 105, JUMP
0072        105 FORMAT(I5)
0073          JUMP=JUMP+13000000
0074          IF(JUMP.GT.13090000) JUMP=JUMP+86900000
0075          GO TO 13
0076      C
0077      C***** CHECK ONE MORE TIME IF ALL CHANGES HAVE BEEN MADE THEN WRITE
0078      C***** ID NUMBERS AND STORAGE ACCOUNT VALUES TO NEW FILE.
0079      C
0080        100 PRINT 8
0081          8 FORMAT(/' DO YOU WISH TO CHANGE/REVIEW ANY MORE SPACE VALUES?
0082           1  Y/N '$)
0083          ACCEPT 102, IANS
0084          IF(IANS.EQ.'Y') GO TO 12
0085          DO 1 I=1,NDIVNS
0086         1 WRITE(2,18) ID(I),IK(I),DNAME(I),USED(I),ADJU(I),BANK(I),
0087           1 RFWB(I)
0088        18 FORMAT(I8,A1,A14,4F10.1)
0089          PRINT *,' '
0090          PRINT *,' FILE SNKSTO''YR''.USE HAS BEEN UPDATED WITH NEW VALUES'
0091          PRINT *,' '
0092          PRINT *,' '
0093          PRINT *,' '
0094          STOP
0095          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1381	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	1054	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	15480	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	17915	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGBR$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
**	R*4	BLANK		12D	16					
**	I*4	I		20=	21(7)	24	36=	37	38	39(3) 49
					85=	86(7)				
2-00003B60	I*4	IANS		16=	42=	44	45	46	83=	84
2-00003B64	I*4	JUMP		35=	37	71=	73(2)=	74(3)=		
**	I*4	NDIVNS		24=	36	85				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000640	R*4	ADJU		1600	(400)	9	21=	86		
2-00000C80	R*4	BANK		1600	(400)	9	21=	86		
2-00002580	CHAR	DNAME		5600	(400)	10	11	21=	39	86
2-00001900	I*4	ID		1600	(400)	10	21=	37	39	86
2-00001F40	I*4	IK		1600	(400)	10	21=	38	86	
2-000012C0	R*4	RFWB		1600	(400)	9	21=	39	49=	86
2-00000000	R*4	USED		1600	(400)	9	21=	86		

LABELS

Address	Label	References	
**	1	85	86#
1-000002AF	2'	21	23#
0-0000027C	3	45	57#
1-0000032C	4'	49	50#
0-000000BF	5	21	24#
**	6	39#	54
1-000003D3	8'	80	81#

```
     **      9                          20        21#
  1-000002BB 11'                        39        40#
  0-000000C4 12                         29#       84

  0-0000016C 13                         36#       75
  1-00000412 18'                        86        88#
  0-0000040A 100                        44        56        80#
  0-00000404 101                        36        37        38        46        51        55#
  1-000002F7 102'                       42        43#       83

  1-000002FA 103'                       47        48#
  1-0000035D 104'                       68        69#
  1-000003D0 105'                       71        72#
  0-00000260 106                        49        52#
  1-00000330 107'                       52        53#
```

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 SNKCHGBR

/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.SNAKE]SNKCHGBR.LIS;1
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGBR.OBJ;1
```

COMPILATION STATISTICS

Run Time:	5.44 seconds
Elapsed Time:	6.73 seconds
Page Faults:	713
Dynamic Memory:	430 pages

SNKDENT, LLS

```
00001   C
00002   C       **** PROGRAM TO ENTER RIVER DATA FROM PARTIAL RECORDS AND
00003   C       PERFORM STRAIGHT LINE INTERPOLATION BETWEEN KNOWN POINTS.
00004   C       OUTPUTS DATA IN FORMAT ACCEPTABLE TO WATER RIGHT ACCOUNTING
00005   C       PROGRAMS.   RHL - JUNE 1981.
00006   C
00007   C       **** MODIFIED TO ENTER SNAKE RIVER WATER RIGHT ACCOUNTING
00008   C       DATA.  RJS - NOVEMBER 1988.
00009   C
00010   C
00011         LOGICAL*1 FILES(40)
00012         REAL*8 XFLOWA,XFLOWB,DIFF,XD,DAYFL
00013         INTEGER*4 NDAYA,NDAYB,NNDAY,JDAY,KDAY
00014         DIMENSION JDAY(12),TITLE(11),KDAY(12)
00015         DATA JDAY/0,31,59,90,120,151,181,212,243,273,304,334/
00016         DATA KDAY/31,28,31,30,31,30,31,31,30,31,30,31/
00017         DATA BLANK/' '/
00018   C
00019   C***** ASSIGN INPUT-OUTPUT UNITS.
00020   C
00021         OPEN(UNIT=1,NAME='SNKDENT.OUT',TYPE='NEW',
00022        1CARRIAGECONTROL='LIST')
00023         CALL ASSIGN(3,'SNKEXC.TTL')
00024         CALL ASSIGN(4,'SNKFLO.TTL')
00025         CALL ASSIGN(5,'SNKRES.TTL')
00026         CALL ASSIGN(6,'SNKSTR.TTL')
00027   C
00028   C***** READ DATA USING TITLE PROMPTS OR HAND ENTER STATION NUMBER.
00029   C
00030         IN=1
00031         NEWYR=0
00032         MTHB=0
00033         TYPE *, ' ************************************************'
00034         TYPE *, ' '
00035         TYPE *, ' SNAKE RIVER DATA ENTRY AND INTERPOLATION PROGRAM. '
00036         TYPE *, ' STRAIGHT LINE INTERPOLATION BETWEEN KNOWN POINTS.'
00037         TYPE *, ' '
00038         TYPE *, ' ************************************************'
00039         TYPE *, ' '
00040       8 PRINT *,'   ENTER FOUR DIGIT CALENDAR YEAR OF THE BEGINNING   '
00041         PRINT *,'   PERIOD  FOR WHICH DATA IS TO BE ENTERED -'
00042         PRINT *,' '
00043         PRINT *,'   FOR EXAMPLE - TO ENTER DATA BEGINNING JAN - DEC  '
00044         PRINT *,'   1990, ENTER 1990      '
00045         PRINT *,'                ----      '
00046         PRINT 901
00047     901 FORMAT('+               '$)
00048         ACCEPT 7, IYR
00049       7 FORMAT(I4)
00050         JYR=IYR
00051      16 IYR=JYR
00052         PRINT *,'************************************************'
00053         PRINT *,'                                                '
00054         PRINT *,'   YOU HAVE A CHOICE:                           '
00055         PRINT *,'                                                '
00056         PRINT *,'   ENTER GROUPS OF DIVERSIONS, FLOWS, EXCHANGE PUMPS,'
00057         PRINT *,'   DRAINS, OR RESERVOIRS  (STATION NUMBERS AND TITLES'
```

```
00058         PRINT *,'   READ IN FROM FILES)                                '
00059         PRINT *,'                                                       '
00060         PRINT *,'                              OR                       '
00061         PRINT *,'                                                       '
00062         PRINT *,'   ENTER SINGLE STATIONS                               '
00063         PRINT *,'   (YOU PROVIDE THE STATION NUMBER AND TITLE)          '
00064         PRINT *,'                                                       '
00065         PRINT *,'   TO ENTER GROUPS , ENTER  G                          '
00066         PRINT *,'   TO ENTER SINGLE STATIONS, ENTER  S                  '
00067         PRINT *,'   TO QUIT, ENTER  Q                                   '
00068         PRINT *,'                                                       '
00069         PRINT *,'*******************************************************'
00070         PRINT *,'                                                       '
00071         PRINT 904
00072     904 FORMAT('        ENTER CHOICE:'$)
00073         ACCEPT 102, IND
00074         IF( IND .EQ. 'Q' ) GO TO 500
00075         IF( IND .EQ. 'S' ) GO TO 5
00076         PRINT *,'                                                       '
00077         PRINT *,'*******************************************************'
00078         PRINT *,'                                                       '
00079         PRINT *,'   YOU HAVE ANOTHER CHOICE:                            '
00080         PRINT *,'                                                       '
00081         PRINT *,'   TO ENTER DIVERSIONS, ANSWER  D                      '
00082         PRINT *,'   TO ENTER ALL RIVER FLOWS, ANSWER  F                 '
00083         PRINT *,'   TO ENTER ALL EXCHANGE PUMPS, ANSWER  E              '
00084         PRINT *,'   TO ENTER ALL RESERVOIRS, ANSWER  R                  '
00085         PRINT *,'   TO QUIT, ANSWER  Q  '
00086         PRINT *,'                                                 '
00087         PRINT *,'*******************************************************'
00088         PRINT *,'                                                 '
00089         PRINT 904
00090         ACCEPT 102, IND
00091         IF( IND .EQ. 'Q' ) GO TO 500
00092         IF( IND .NE. 'D' ) GO TO 911
00093         PRINT *,'                                                       '
00094         PRINT *,'*******************************************************'
00095         PRINT *,'                                                       '
00096         PRINT *,'   YOU HAVE STILL ANOTHER CHOICE:                      '
00097         PRINT *,'                                                       '
00098         PRINT *,'   TO ENTER ALL 300 DIVERSIONS, ANSWER  D              '
00099         PRINT *,'   TO ENTER ALL CANAL DIVERSIONS, ANSWER  C            '
00100         PRINT *,'   TO ENTER ALL PUMP DIVERSIONS, ANSWER  P             '
00101         PRINT *,'   TO ENTER DIVERSIONS TO LORENZO, ANSWER  L           '
00102         PRINT *,'   TO ENTER HENRYS FORK DIVERSIONS, ANSWER  H          '
00103         PRINT *,'   TO ENTER DIVERSIONS LORENZO TO BLACKFOOT, ANSWER B'
00104         PRINT *,'   TO ENTER DIVERSIONS BLACKFOOT TO MILNER, ANSWER M '
00105         PRINT *,'   TO ENTER FILE NAME OF DIVERSION TITLES ANSWER T    '
00106         PRINT *,'   TO QUIT, ANSWER  Q  '
00107         PRINT *,'                                                 '
00108         PRINT *,'*******************************************************'
00109         PRINT *,'                                                 '
00110         PRINT 904
00111         ACCEPT 102, INK
00112         IF( INK .EQ. 'Q' ) GO TO 500
00113         IF(INK.EQ.'D') OPEN(UNIT=2,FILE='SNKDIV.TTL',STATUS='OLD')
00114         IF(INK.EQ.'C') OPEN(UNIT=2,FILE='SNKCAN.TTL',STATUS='OLD')
```

```
00115         IF(INK.EQ.'P') OPEN(UNIT=2,FILE='SNKPMP.TTL',STATUS='OLD')
00116         IF(INK.EQ.'L') OPEN(UNIT=2,FILE='SNKTOL.TTL',STATUS='OLD')
00117         IF(INK.EQ.'H') OPEN(UNIT=2,FILE='SNKHEN.TTL',STATUS='OLD')
00118         IF(INK.EQ.'B') OPEN(UNIT=2,FILE='SNKLTB.TTL',STATUS='OLD')
00119         IF(INK.EQ.'M') OPEN(UNIT=2,FILE='SNKBTM.TTL',STATUS='OLD')
00120         IF(INK.NE.'T') GO TO 12
00121         PRINT 907
00122     907 FORMAT('     ENTER FILE NAME:'$)
00123         ACCEPT 908, IC,(FILES(I),I=1,IC)
00124     908 FORMAT(Q,40A1)
00125         CALL ASSIGN(2,FILES)
00126         GO TO 12
00127    911  IF(IND.EQ.'F') GO TO 13
00128         IF(IND.EQ.'E') GO TO 2
00129         IF(IND.EQ.'R') GO TO 14
00130         IF(IND.EQ.'S') GO TO 5
00131         IF(IND.EQ.'I') GO TO 4
00132         IF(IND.EQ.'Q') GO TO 500
00133      12 S='D'
00134         IN=2
00135         GO TO 15
00136       2 S='F'
00137         IN=3
00138         GO TO 15
00139      13 S='F'
00140         IN=4
00141         GO TO 15
00142      14 S='R'
00143         IN=5
00144         GO TO 15
00145       4 S='D'
00146         IN=6
00147      15 REWIND IN
00148         S3=S
00149      29 S=S3
00150         READ(IN,1,END=16) ID,S2,(TITLE(I),I=1,11)
00151       1 FORMAT(I8,A1,11A4)
00152         IF(S2.NE.BLANK) S=S2
00153         TYPE 3, (TITLE(I),I=1,11)
00154       3 FORMAT(' HIT "RETURN" TO ENTER DATA FOR '//5X,11A4//
00155        1 ' OTHERWISE ENTER "N" TO GO TO NEXT STATION, OR   '/
00156        2 '           ENTER "E" TO EXIT THIS GROUP ',$)
00157         ACCEPT 102, ILOGIC
00158         IF(ILOGIC.EQ.'N') GO TO 29
00159         IF(ILOGIC.EQ.'E') GO TO 16
00160         GO TO 17
00161    C
00162    C***** IF MODE IS 'S' , ENTER STATION ID NUMBERS INDIVIDUALLY
00163    C
00164      22 IYR=JYR
00165         IF(IND.NE.'S') GO TO 29
00166       5 TYPE *, ' ENTER 9 DIGIT STATION NO. - EX. 13059800F'
00167         TYPE *, ' SUFFIX: D=DIV, F=FLOW OR DRAIN, R=RESERVR.'
00168         TYPE *, 'NNNNNNNNA    '
00169         ACCEPT 6, ID,S
00170       6 FORMAT(I8,A1)
00171      17 CONTINUE
```

```
00172      10 TYPE *, ' '
00173         TYPE *, ' ENTER FIRST MONTH, DAY, AND FLOW.'
00174         TYPE *, ' EXAMPLE: MAY 9, 222.2CFS = 0409222.2'
00175         TYPE *, ' TO BEGIN NEXT STATION, ENTER 80 FOR MONTH.'
00176         TYPE *, ' TO EXIT ENTER 90.'
00177         TYPE *, ' '
00178         TYPE *, 'MMDDFLOW.00    '
00179         ACCEPT 11, MTHA,IDAYA,XFLOWA
00180      11 FORMAT(I2,I2,F7.0)
00181         IF(MTHA.EQ.80) GO TO 22
00182         IF(MTHA.EQ.90) GO TO 16
00183         TYPE 101, MTHA,IDAYA,XFLOWA
00184     101 FORMAT(3X,'DATA ACCEPTED: MONTH=',I2,/,20X,'DAY=',I2,/,19X,'DATA
00185        1=',F10.1,/,'   **** IF DATA IS INCORRECT ENTER "N".',$)
00186         ACCEPT 102, ILOGIC
00187     102 FORMAT(1A1)
00188         IF(ILOGIC.EQ.'N') GO TO 10
00189         MTHC=MTHA
00190         IDAYC=IDAYA
00191    C
00192    C***** CHECK FOR LEAP YEAR CORRECTION, THEN COMPUTE DAY OF *****
00193    C***** THE YEAR FOR FIRST DATA POINT                      *****
00194    C
00195       9 ILEAP=0
00196         ADIFF=MOD(IYR,4)
00197         IF(ADIFF.EQ.0) ILEAP=1
00198         IF(NEWYR.EQ.1) GO TO 110
00199         NDAYA=JDAY(MTHA)+IDAYA
00200         IF(MTHA.GT.2) NDAYA=NDAYA+ILEAP
00201         IF(S.NE.'R') GO TO 23
00202         WRITE(1,25) ID,S,IYR,NDAYA,JIDNNT(XFLOWA)
00203      25 FORMAT(I8,A1,I4,I3,I7)
00204         GO TO 24
00205      23 WRITE(1,21) ID,S,IYR,NDAYA,XFLOWA
00206      21 FORMAT(I8,A1,I4,I3,F7.1)
00207    C
00208    C***** IF DATA BEING ENTERED IS 13203600 - BELOW DIVERSION DAM,
00209    C***** COMPUTE RELEASES FOR FLOW MAINTENANCE FROM FISH AND GAME
00210    C***** STORAGE AND USBR STORAGE IN LUCKY PEAK.
00211    C
00212      24 IF(ID.NE.13203600) GO TO 31
00213         FGF=70.
00214         IF(FGF.GT.XFLOWA) FGF=XFLOWA
00215         USBRF=XFLOWA-FGF
00216         IF(USBRF.GT.150.) USBRF=150.
00217         IF(XFLOWA.GT.325.) FGF=0.0
00218         IF(XFLOWA.GT.325.) USBRF=0.0
00219         IF(FGF.GT.0.0.OR.USBRF.GT.0.0) GO TO 34
00220         NA=NDAYA-ILEAP
00221         IF(NA.LT.91.OR.NA.GT.304) GO TO 31
00222      34 WRITE(1,32) IYR,NDAYA,FGF
00223      32 FORMAT('13201990D',I4,I3,F7.1)
00224         WRITE(1,33) IYR,NDAYA,USBRF
```

```
00225      33 FORMAT('13201991D',I4,I3,F7.1)
00226      31 NCTN=1
00227         TYPE 150, NCTN
00228     110 NEWYR=0
00229    C
00230    C***** ACCEPT SECOND DAY AND FLOW. *****
00231    C
00232         GO TO 27
00233     112 TYPE *, ' **** SECOND DATE IS SAME AS PREVIOUS - NN=0'
00234      27 TYPE *, ' '
00235         TYPE *, ' ENTER SECOND OR NEXT DAY AND FLOW.'
00236         TYPE *, ' TO SELECT NEW FIRST DATE, ENTER 70 FOR MONTH.'
00237         TYPE *, ' TO BEGIN NEXT STATION, ENTER 80 FOR MONTH.'
00238         TYPE *, ' TO EXIT ENTER 90.'
00239         TYPE *, ' '
00240         TYPE *, 'MMDDFLOW.00 '
00241         ACCEPT 11, MTHB,IDAYB,XFLOWB
00242         IF(MTHB.LE.0) GO TO 30
00243         IF(MTHB.LE.12) GO TO 19
00244         IF(MTHB.EQ.80) GO TO 22
00245         IF(MTHB.EQ.90) GO TO 16
00246         IF(MTHB.EQ.70) GO TO 17
00247    C
00248    C***** GENERATE DATE AS ONE DAY LATER IF NO DATE ENTERED. *****
00249    C
00250      30 MTHB=MTHC
00251         IDAYB=IDAYC+1
00252         KD=KDAY(MTHB)
00253         IF(ILEAP.GT.0.AND.MTHB.EQ.2) KD=29
00254         IF(IDAYB.LE.KD) GO TO 19
00255         MTHB=MTHB+1
00256         IDAYB=1
00257         IF(MTHB.GT.12) MTHB=1
00258    C
00259    C***** WRITE DATA TO SCREEN FOR VERIFICATION. *****
00260    C
00261      19 TYPE 101, MTHB,IDAYB,XFLOWB
00262         ACCEPT 102, ILOGIC
00263         IF(ILOGIC.EQ.'N') GO TO 27
00264    C
00265    C***** DETERMINE DAY OF YEAR AND CORRECT FOR LEAP YEAR. *****
00266    C
00267         NDAYB=JDAY(MTHB)+IDAYB
00268         IF(MTHB.GT.2) NDAYB=NDAYB+ILEAP
00269         MTHC=MTHB
00270         IDAYC=IDAYB
00271    C
00272    C***** PERFORM INTERPOLATION. *****
00273    C
00274         NN=NDAYB-NDAYA
00275         IF(NDAYA.GT.NDAYB) NN=365-NDAYA+ILEAP+NDAYB
00276         IF(NN.LT.1) GO TO 112
00277         DIFF=XFLOWB-XFLOWA
00278         XD=DIFF/NN
00279      18 DO 28 J=1,NN
00280         DAYFL=XFLOWA+FLOAT(J)*XD
00281         NNDAY=NDAYA+J
```

```
00282   C
00283   C***** ERROR CHECK FOR RUN AWAY DATA - MORE THAN 35 DAYS. ******
00284   C
00285         IF(J.EQ.36) GO TO 35
00286         GO TO 39
00287      35 TYPE *, ' ==> CAUTION I HAVE CREATED >35 DAYS OF DATA. <=='
00288         TYPE *, ' IS THIS OK????  ANSWER Y/N.'
00289         ACCEPT 36, JLOGIC
00290      36 FORMAT(1A1)
00291         IF(JLOGIC.EQ.'N') GO TO 500
00292   C
00293   C***** CORRECT FOR END OF YEAR. [ DEC==>JAN ] *****
00294   C
00295      39 IF(NNDAY.GT.(365+ILEAP)) GO TO 40
00296         GO TO 20
00297      40 NNDAY=NNDAY-365-ILEAP
00298         IF(NEWYR.EQ.1) GO TO 20
00299         IYR=IYR+1
00300         NEWYR=1
00301         TYPE *, ' ++++ NOTICE:  ENTERING NEW YEAR. ++++'
00302      20 IF(S.NE.'R') GO TO 26
00303         WRITE(1,25) ID,S,IYR,NNDAY,JIDNNT(DAYFL)
00304         GO TO 37
00305      26 WRITE(1,21) ID,S,IYR,NNDAY,DAYFL
00306   C
00307   C***** IF DATA BEING ENTERED IS 13203600 - BELOW DIVERSION DAM,
00308   C***** COMPUTE RELEASES FOR FLOW MAINTENANCE FROM FISH AND GAME
00309   C***** STORAGE AND USBR STORAGE IN LUCKY PEAK.
00310   C
00311      37 IF(ID.NE.13203600) GO TO 28
00312         FGF=70.
00313         IF(FGF.GT.DAYFL) FGF=DAYFL
00314         USBRF=DAYFL-FGF
00315         IF(USBRF.GT.150.) USBRF=150.
00316         IF(DAYFL.GT.325.) FGF=0.0
00317         IF(DAYFL.GT.325.) USBRF=0.0
00318         IF(FGF.GT.0.0.OR.USBRF.GT.0.0) GO TO 38
00319         NA=NNDAY-ILEAP
00320         IF(NA.LT.91.OR.NA.GT.304) GO TO 28
00321      38 WRITE(1,32) IYR,NNDAY,FGF
00322         WRITE(1,33) IYR,NNDAY,USBRF
00323      28 CONTINUE
00324   C
00325   C***** WRITE OUTPUT INFORMATION TO SCREEN. *****
00326   C
00327         NCTN=NN
00328         TYPE 150, NCTN
00329     150 FORMAT(1X,I4,' LINES ENTERED IN OUTPUT FILE.')
00330   C
00331   C***** RENAME SECOND DAY AND FLOW TO FIRST. *****
00332   C
00333         NDAYA=NDAYB
00334         XFLOWA=XFLOWB
00335   C
00336   C***** RETURN TO ACCEPT NEXT DAY AND FLOW. *****
00337   C
00338         IF(NEWYR.EQ.1) GO TO 9
```

```
00339          GO TO 27
00340      500 STOP 'OUTPUT IN "SNKDENT.OUT"'
00341          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	4991	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	2908	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	1300	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	9199	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKDENT$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	ADIFF		196=	197						
2-000000CC	R*4	BLANK		17D	152						
**	R*8	DAYFL		12	280=	303	305	313(2)	314	316	317
**	R*8	DIFF		12	277=	278					
**	R*4	FGF		213=	214(2)=	215	217=	219	222	312=	313(2)=
				314	316=	318	321				
2-000000E4	I*4	I		123(2)=	150(2)=	153(2)=					
2-000000E0	I*4	IC		123(2)=							
2-000000F4	I*4	ID		150=	169=	202	205	212	303	305	311
2-00000104	I*4	IDAYA		179=	183	190	199				
2-00000108	I*4	IDAYB		241=	251=	254	256=	261	267	270	
**	I*4	IDAYC		190=	251	270=					
**	I*4	ILEAP		195=	197=	200	220	253	268	275	295
				297	319						
2-000000FC	I*4	ILOGIC		157=	158	159	186=	188	262=	263	
**	I*4	IN		30=	134=	137=	140=	143=	146=	147	150
2-000000D8	I*4	IND		73=	74	75	90=	91	92	127	128
				129	130	131	132	165			
2-000000DC	I*4	INK		111=	112	113	114	115	116	117	118
				119	120						
2-000000D4	I*4	IYR		48=	50	51=	164=	196	202	205	222
				224	299(2)=	303	305	321	322		
**	I*4	J		279=	280	281	285				
2-00000110	I*4	JLOGIC		289=	291						
**	I*4	JYR		50=	51	164					

**	I*4	KD	252=	253=	254				
2-00000100	I*4	MTHA	179=	181	182	183	189	199	200
2-000000D0	I*4	MTHB	32=	241=	242	243	244	245	246	250=
			252	253	255(2)=	257(2)=	261	267	268	269
**	I*4	MTHC	189=	250	269=					
**	I*4	NA	220=	221(2)	319=	320(2)				
**	I*4	NCTN	226=	227	327=	328				
2-000000C8	I*4	NDAYA	13	199=	200(2)=	202	205	220	222	224
			274	275(2)	281	333=				
**	I*4	NDAYB	13	267=	268(2)=	274	275(2)	333		
**	I*4	NEWYR	31=	198	228=	298	300=	338		
**	I*4	NN	274=	275=	276	278	279	327		
**	I*4	NNDAY	13	281=	295	297(2)=	303	305	319	321
			322							
2-000000EC	R*4	S	133=	136=	139=	142=	145=	148	149=	152=
			169=	201	202	205	302	303	305	
2-000000F8	R*4	S2	150=	152(2)						
2-000000F0	R*4	S3	148=	149						
**	R*4	USBRF	215=	216(2)=	218=	219	224	314=	315(2)=	317=
			318	322						
**	R*8	XD	12	278=	280					
2-000000B8	R*8	XFLOWA	12	179=	183	202	205	214(2)	215	217
			218	277	280	334=				
2-000000C0	R*8	XFLOWB	12	241=	261	277	334			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-0000008C	L*1	FILES		40	(40)	11	123=	125A		
2-00000000	I*4	JDAY		48	(12)	13	14	15D	199	267
2-00000030	I*4	KDAY		48	(12)	13	14	16D	252	
2-00000060	R*4	TITLE		44	(11)	14	150=	153		

LABELS

Address	Label	References		
1-0000004F	1'	150	151#	
0-000009AC	2	128	136#	
1-00000058	3'	153	154#	
0-0000098C	4	131	145#	
0-000009D8	5	75	130	166#
1-000000E4	6'	169	170#	
1-00000014	7'	48	49#	
**	8	40#		
0-00000BE8	9	195#	338	
0-00000A5C	10	172#	188	
1-000000E9	11'	179	180#	241

0-0000097F	12	120	126	133#			
0-000009BC	13	127	139#				
0-0000099C	14	129	142#				
0-000009C7	15	135	138	141	144	147#	
0-00000200	16	51#	150	159	182	245	
0-00000A5C	17	160	171#	246			
**	18	279#					
0-0000100E	19	243	254	261#			
0-000011BA	20	296	298	302#			
1-0000015E	21'	205	206#	305			
0-00000EDD	22	164#	181	244			
0-00000C70	23	201	205#				
0-00000CB7	24	204	212#				
1-00000153	25'	202	203#	303			
0-0000120C	26	302	305#				
0-00000DC0	27	232	234#	263	339		
0-00001337	28	279	311	320	323#		
0-00000EEE	29	149#	158	165			
0-00000FD0	30	242	250#				
0-00000D9B	31	212	221	226#			
1-0000016A	32'	222	223#	321			
1-0000017D	33'	224	225#	322			
0-00000D39	34	219	222#				
0-00001110	35	285	287#				
1-00000190	36'	289	290#				
0-00001252	37	304	311#				
0-000012D4	38	318	321#				
0-00001176	39	286	295#				
0-00001188	40	295	297#				
1-000000F1	101'	183	184#	261			
1-00000150	102'	73	90	111	157	186	187# 262
0-00000DBB	110	198	228#				
0-000010AB	112	233#	276				
1-00000193	150'	227	328	329#			
0-00001374	500	74	91	112	132	291	340#
1-00000000	901'	46	47#				
1-00000017	904'	71	72#	89	110		
1-00000030	907'	121	122#				
1-00000049	908'	123	124#				
0-00000940	911	92	127#				

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References				
	ASSIGN	23	24	25	26	125
R*4	FLOAT	280				
I*4	JIDNNT	202	303			

```
   I*4  MOD                                    196

  +-----------------------------------------------+
  |            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 SNKDENT

  /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.SNAKE]SNKDENT.LIS;1
  /OBJECT=HYDRO:[BSUTTER.SNAKE]SNKDENT.OBJ;1


COMPILER:  DEC Fortran V6.3-141

COMPILATION STATISTICS

  Run Time:           1.01 seconds
  Elapsed Time:       1.85 seconds
  Page Faults:        2467
  Dynamic Memory:     1408 pages
```

```
0001    C
0002    C         **** PROGRAM TO ENTER RIVER DATA FROM DATA ENTRY FILES TO
0003    C         PERMANENT HISTORY FILE.  IF DATA ALREADY EXISTS FOR STATION
0004    C         AND DAY BEING ENTERED, OLD DATA WILL BE REPLACED WITH NEW
0005    C         ENTRY. BOB SUTTER - OCTOBER 1986.
0006    C
0007    C       MODIFIED FOR SNAKE RIVER - RJS - DECEMBER 1988
0008    C
0009          CHARACTER*7 VAOLD(4),VANEW(4),VANEW2(4)
0010          CHARACTER*28 VOLD,VNEW,VNEW2
0011          INTEGER*4 ISTA,ISTA2,JSTA,IDATE,IDATE2,JDATE
0012          EQUIVALENCE (VOLD,VAOLD),(VNEW,VANEW),(VNEW2,VANEW2)
0013    C
0014          CALL ASSIGN(1,'ENTRY')
0015          CALL ASSIGN(2,'OLDHIST')
0016          OPEN(UNIT=3,NAME='NEWHIST',TYPE='NEW',
0017         1CARRIAGECONTROL='LIST')
0018    C
0019    C***** INITIALIZE INDICATORS
0020    C
0021          JSKIP=0
0022          IREP=0
0023          INEW=0
0024          ISAM=0
0025          KEND=0
0026    C
0027    C***** READ TWO  RECORDS FROM FROM ENTRY FILE AND ONE RECORD
0028    C***** FROM OLD HISTORY FILE.  IF THERE ARE DUPLICATE RECORDS
0029    C***** ON ENTRY FILE, USE ONLY FINAL ENTRY.  SKIP READING DATA
0030    C***** FROM OLD HISTORY FILE IF LAST ENTRY DATA HAS BEEN MERGED
0031    C***** WITH DATA FROM OLD HISTORY FILE.
0032    C
0033          READ(1,1,END=10)ISTA,S,IDATE,VNEW
0034        1 FORMAT(I8,A1,I7,A28)
0035          GO TO 19
0036        2 IF(KEND.EQ.1) GO TO 10
0037          ISTA=ISTA2
0038          S=S2
0039          IDATE=IDATE2
0040          VNEW=VNEW2
0041       19 READ(1,1,END=17) ISTA2,S2,IDATE2,VNEW2
0042          IF(ISTA2.EQ.ISTA.AND.IDATE2.EQ.IDATE) GO TO 2
0043          GO TO 3
0044       17 KEND=1
0045        3 IF(JSKIP.EQ.0) READ(2,1,END=20)JSTA,T,JDATE,VOLD
0046          JSKIP=0
0047    C
0048    C***** CHECK RELATIVE VALUE OF STATION NUMBERS OF OLD
0049    C***** AND NEW ENTRY DATA.
0050    C
0051          IF(JSTA-ISTA) 4,5,6
0052    C
0053    C***** CHECK RELATIVE VALUE OF DATES OF OLD AND NEW
0054    C***** ENTRY DATA.
0055    C
0056        5 IF(JDATE-IDATE) 4,7,6
0057    C
```

```
0058    C***** WRITE DATA FROM ENTRY FILE TO NEW HISTORY FILE TO
0059    C***** EITHER REPLACE OR BE MERGED WITH OLD DATA.
0060    C
0061         6 JSKIP=1
0062           INEW=INEW+1
0063           GO TO 9
0064         7 IREP=IREP+1
0065           DO 18 I=1,4
0066        18 IF(VANEW(I).EQ.'        ') VANEW(I)=VAOLD(I)
0067         9 IF(S.EQ.'R') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0068           IF(S.EQ.'F') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0069           IF(S.EQ.'D') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0070           IF(S.EQ.'P') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0071        26 FORMAT(I8,A1,I7,4A7)
0072           IF(S.EQ.'E') WRITE(3,27) ISTA,S,IDATE,(VANEW(I),I=1,1)
0073        27 FORMAT(I8,A1,I7,A7)
0074           GO TO 2
0075    C
0076    C***** WRITE DATA RECORDS FROM OLD HISTORY FILE TO NEW
0077    C***** HISTORY FILE UNTIL DATA FROM ENTRY FILE IS TO
0078    C***** REPLACE OLD DATA OR BE MERGED WITH OLD DATA.
0079    C
0080         4 ISAM=ISAM+1
0081           IF(T.EQ.'R') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0082           IF(T.EQ.'F') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0083           IF(T.EQ.'D') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0084           IF(T.EQ.'P') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0085           IF(T.EQ.'E') WRITE(3,27) JSTA,T,JDATE,(VAOLD(I),I=1,1)
0086           GO TO 3
0087    C
0088    C***** WRITE REMAINDER OF OLD HISTORY FILE TO NEW FILE
0089    C***** AFTER DATA ENTRY FILE HAS BEEN EXHAUSTED.
0090    C
0091        10 IF(JSKIP.EQ.1) GO TO 14
0092        15 READ(2,1,END=100)JSTA,T,JDATE,VOLD
0093        14 ISAM=ISAM+1
0094           IF(T.EQ.'R') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0095           IF(T.EQ.'F') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0096           IF(T.EQ.'D') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0097           IF(T.EQ.'P') WRITE(3,26) JSTA,T,JDATE,(VAOLD(I),I=1,4)
0098           IF(T.EQ.'E') WRITE(3,27) JSTA,T,JDATE,(VAOLD(I),I=1,1)
0099           GO TO 15
0100    C
0101    C***** WRITE REMAINDER OF DATA ENTRY FILE TO NEW HISTORY
0102    C***** FILE AFTER OLD HISTORY FILE HAS BEEN EXHAUSTED.
0103    C
0104        22 IF(KEND.EQ.1) GO TO 100
0105        24 ISTA=ISTA2
0106           S=S2
0107           IDATE=IDATE2
0108           VNEW=VNEW2
0109           READ(1,1,END=37) ISTA2,S2,IDATE2,VNEW2
0110           IF(ISTA2.EQ.ISTA.AND.IDATE2.EQ.IDATE) GO TO 24
0111           GO TO 20
0112        37 KEND=1
0113        20 INEW=INEW+1
0114           IF(S.EQ.'R') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
```

```
0115          IF(S.EQ.'F') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0116          IF(S.EQ.'D') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0117          IF(S.EQ.'P') WRITE(3,26) ISTA,S,IDATE,(VANEW(I),I=1,4)
0118          IF(S.EQ.'E') WRITE(3,27) ISTA,S,IDATE,(VANEW(I),I=1,1)
0119          GO TO 22
0120      C
0121      C***** END OF ALL DATA SETS
0122      C
0123        100 TYPE *, ' HISTORY FILE HAS NOW BEEN UPDATED '
0124            ITOT=IREP+ISAM+INEW
0125            TYPE *, ' '
0126            TYPE 11, IREP,INEW,ISAM,ITOT
0127         11 FORMAT(/2X,'UPDATED FILE HAS',I10,3X,'RECORDS REPLACED'/
0128           1 18X,I10,3X,'NEW RECORDS'/18X,I10,3X,'RECORDS TRANSFERRED'/
0129           2 4X,'FOR A TOTAL OF',I10,3X,'HISTORY FILE RECORDS')
0130            TYPE *, ' '
0131        500 STOP
0132            END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	2413	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	239	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	316	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	2968	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTUP$MAIN	

VARIABLES

Address	Type	Name	Attributes	References							
**	I*4	I		65=	66(3)	67(2)=	68(2)=	69(2)=	70(2)=	72(2)=	81(2)=	
					82(2)=	83(2)=	84(2)=	85(2)=	94(2)=	95(2)=	96(2)=	97(
					98(2)=	114(2)=	115(2)=	116(2)=	117(2)=	118(2)=		
2-00000060	I*4	IDATE		11	33=	39=	42	56	67	68	69	
					70	72	107=	110	114	115	116	117
					118							
2-00000064	I*4	IDATE2		11	39	41=	42	107	109=	110		
**	I*4	INEW		23=	62(2)=	113(2)=	124	126				
**	I*4	IREP		22=	64(2)=	124	126					
**	I*4	ISAM		24=	80(2)=	93(2)=	124	126				
2-00000054	I*4	ISTA		11	33=	37=	42	51	67	68	69	
					70	72	105=	110	114	115	116	117

```
                                                    118
  2-00000058  I*4   ISTA2                  11       37        41=       42        105       109=      110
        **    I*4   ITOT                  124=      126
  2-00000068  I*4   JDATE                  11       45=       56        81        82        83        84        85
                                                     92=       94        95        96        97        98

        **    I*4   JSKIP                  21=      45        46=       61=       91
  2-0000005C  I*4   JSTA                   11       45=       51        81        82        83        84        85
                                                     92=       94        95        96        97        98
        **    I*4   KEND                   25=      36        44=       104       112=
  2-0000006C  R*4   S                      33=      38=       67(2)     68(2)     69(2)     70(2)     72(2)     106=
                                                    114(2)    115(2)    116(2)    117(2)    118(2)
  2-00000070  R*4   S2                     38       41=       106       109=

  2-00000074  R*4   T                      45=      81(2)     82(2)     83(2)     84(2)     85(2)     92=       94(2)
                                                     95(2)     96(2)     97(2)     98(2)
  2-0000001C  CHAR VNEW           EQUIV    10       12        33=       40=       108=
  2-00000000  CHAR VNEW2          EQUIV    10       12        40        41=       108       109=
  2-00000038  CHAR VOLD           EQUIV    10       12        45=       92=
```

ARRAYS

```
    Address   Type  Name          Attributes      Bytes  Dimensions      References

  2-0000001C  CHAR VANEW            EQUIV           28   (4)                9         12        66(2)=    67        68
                                                                                                   69        70        72        11
                                                                                                   116       117       118
  2-00000000  CHAR VANEW2           EQUIV           28   (4)                9         12
  2-00000038  CHAR VAOLD            EQUIV           28   (4)                9         12        66        81        82
                                                                                                   83        84        85        9
                                                                                                   96        97        98
```

LABELS

```
    Address   Label                      References

  1-00000042  1'                        33        34#       41        45        92        109
  0-00000542  2                         36#       42        74
  0-000000CB  3                         43        45#       86
  0-00000564  4                         51        56        80#
        **    5                         51        56#

  0-0000035C  6                         51        56        61#
  0-00000337  7                         56        64#
  0-00000361  9                         63        67#
  0-000006F4  10                        33        36        91#
  1-0000005F  11'                      126        127#

  0-000006FC  14                        91        93#
  0-00000889  15                        92#       99
  0-000000C8  17                        41        44#
        **    18                        65        66#
  0-00000070  19                        35        41#

  0-0000011C  20                        45        111       113#
```

```
      **      22                          104#      119
      **      24                          105#      110
  1-0000004B  26'                          67        68        69        70        71#       81        82        83
                                                     84        94        95        96        97       114       115       116
                                                    117
  1-00000056  27'                          72        73#       85        98       118

  0-00000314  37                          109       112#
  0-000008CF  100                          92       104       123#
      **      500                         131#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                              References

        ASSIGN                              14        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 SNKHSTUP

  /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.SNAKE]SNKHSTUP.LIS;1
  /OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTUP.OBJ;1
```

COMPILATION STATISTICS

```
  Run Time:           6.43 seconds
  Elapsed Time:      15.48 seconds
  Page Faults:       832
  Dynamic Memory:    539 pages
```

CANDIDATE 115

```
0001    C
0002    C*****CHANGE DATE FROM YEAR-MONTH-DAY (YYYYMMDD) TO YEAR-DAY OF THE
0003    C*****YEAR (YYYYDDD). HYDROMET FORMAT TO HISTORY FILE FORMAT.
0004    C
0005          REAL*8 ADIFF
0006          DIMENSION JDAY(12)
0007          CHARACTER*28 DATA
0008          DATA JDAY/0,31,59,90,120,151,181,212,243,273,304,334/
0009          OPEN(UNIT=1,FILE='INFILE',TYPE='OLD')
0010          OPEN(UNIT=2,FILE='OUTFILE',TYPE='NEW',CARRIAGECONTROL='LIST')
0011    C
0012    C***** READ A RECORD FROM OLD FILE
0013    C
0014        1 READ(1,10,END=500)  ID,S,IC,IYR,MTHA,IDAYA,DATA
0015       10 FORMAT(I8,A1,4I2,A28)
0016    C
0017    C***** CHECK FOR LEAP YEAR CORRECTION, THEN COMPUTE DAY OF YEAR
0018    C***** FROM MONTH AND DAY OF MONTH.
0019    C
0020          ILEAP=0
0021          ADIFF=MOD(IYR,4)
0022          IF(ADIFF.EQ.0) ILEAP=1
0023          NDAYA=JDAY(MTHA)+IDAYA
0024          IF(MTHA.GT.2) NDAYA=NDAYA+ILEAP
0025    C
0026    C***** WRITE DATA TO NEW FILE IN PROPER FORMAT FOR ACCOUNTING
0027    C***** PROGRAM.  THEN RETURN TO GET ANOTHER RECORD.
0028    C
0029          WRITE(2,12) ID,S,IC,IYR,NDAYA,DATA
0030       12 FORMAT(I8,A1,2I2,I3,A28)
0031          GO TO 1
0032      500 STOP
0033          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	259	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	39	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	176	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	474	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGDATE$MAIN	

VARIABLES

Address	Type	Name	Attributes	References		
**	R*8	ADIFF		5	21=	22
2-00000030	CHAR	DATA		7	14=	29
2-00000054	I*4	IC		14=	29	
2-0000004C	I*4	ID		14=	29	
2-00000060	I*4	IDAYA		14=	23	
**	I*4	ILEAP		20=	22=	24
2-00000058	I*4	IYR		14=	21	29
2-0000005C	I*4	MTHA		14=	23	24
**	I*4	NDAYA		23=	24(2)=	29
2-00000050	R*4	S		14=	29	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
2-00000000	I*4	JDAY		48	(12)	6	8D	23

LABELS

Address	Label	References	
0-0000001C	1	14#	31
1-0000000F	10'	14	15#
1-0000001A	12'	29	30#
0-000000FC	500	14	32#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	FOR$OPEN	9	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 SNKCHGDATE

/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.SNAKE]SNKCHGDATE.LIS;1
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKCHGDATE.OBJ;1
```

COMPILATION STATISTICS

```
Run Time:           2.19 seconds
Elapsed Time:       3.07 seconds
Page Faults:        648
Dynamic Memory:     341 pages
```

SNR FORMAT 115

```
 1 C
 2 C***** REFORMAT THE HYDROMET DATA FILES ACCORDING TO
 3 C***** THE FORMAT THAT WAS READ FROM FORMAT FILE (SNKFORMAT.DAT).
 4 C
 5 C***** IF COLUMN 9 OF THE OUTPUT RECORD IS AN I (INST DIVERSION) THEN MAKE
 6 C***** IT A D BEFORE OUTPUTING RECORD.
 7 C
 8 C*****MODIFIED JANUARY 1989 TO SET UP FILES TO BE TRANSFERRED TO IDWR
 9 C*****VAX - BOB SUTTER.
10 C
11       IMPLICIT INTEGER*4(A-Z)
12       CHARACTER STN_NAME_D*9,Y_M_D*8,PCODE*3,VALUE*10
13       CHARACTER STN_NAME_F*9,F_STRING*56
14       CHARACTER STN_NAME_O*9,Y_M_D_O*8,VALUE_O(9)*7
15       CHARACTER BLANK_28*28,ALL_VALUE*28,OUT_REC*80,Y_M_D_T*9
16       CHARACTER MONTHS*60
17       INTEGER F_PCODE_L(4)
18       CHARACTER F_PCODE(4)*2
19       EQUIVALENCE (OUT_REC,STN_NAME_O),(OUT_REC(10:),Y_M_D_O)
20       EQUIVALENCE (OUT_REC(18:),VALUE_O(1))
21       EQUIVALENCE (ALL_VALUE,VALUE_O(1))
22       DATA BLANK_28(1:28)/'                            '/
23       DATA MONTHS(1:30)/'JAN01FEB02MAR03APR04MAY05JUN06'/
24       DATA MONTHS(31:60)/'JUL07AUG08SEP09OCT10NOV11DEC12'/
25 C
26 C*****GET TODAYS DATE TO PLUG IN FOR STATIONS ON FORMAT FILE
27 C*****THAT HAVE NO DATA
28 C
29       CALL DATE_AND_TIME(Y_M_D_T) ! GET TODAYS DATE
30 C       THE OLD CALL DATE STATEMENT IS THE FOLLOWING.
31 C      CALL DATE(Y_M_D_T) ! GET TODAYS DATE
32 C       WEIMIN LI 8/8/2002
33 C
34 C*****REFORMAT DATE INTO YYYYMMDD
35 C
36       J=INDEX(MONTHS,Y_M_D_T(4:6))
37       Y_M_D_T(5:6)=MONTHS(J+3:J+4)
38       Y_M_D_T(3:4)=Y_M_D_T(8:9)
39       Y_M_D_T(7:8)=Y_M_D_T(1:2)
40       IF(Y_M_D_T(3:4).LT.'80')THEN  ! SHOULD YEAR 19XX OR 20XX ?
41          Y_M_D_T(1:2)='20'
42          ELSE
43          Y_M_D_T(1:2)='19'
44       ENDIF
45       OPEN(UNIT=2,FILE='INFILE',STATUS='OLD') ! GET HYDROMET DATA FILE
46       OPEN(UNIT=1,STATUS='OLD',FILE='SNKFORMAT.DAT') ! GET FORMAT FILE
47       OPEN(UNIT=7,STATUS='NEW',FILE='OUTFILE',CARRIAGECONTROL='LIST') ! NEW FILE
48 C
49 C*****READ FROM HYDROMET DATA FILE
50 C
51 C      READ(2,"(/)")
52        do
53                READ(2,10,END=500) STN_NAME_D,Y_M_D,PCODE,VALUE
54                if(stn_name_d /= '') exit
55        end do
56    10 FORMAT(A9,A8,A3,A10)
57 C
```

```
C*****READ FROM STATE FORMAT FILE
C
C
   20 READ(1,21,END=1000) STN_NAME_F,F_STRING
   21 FORMAT(A9,A36)
C
C*****COMPARE STATE AND DATA RECORD FOR MATCHUP
C
   22 IF(STN_NAME_F.LT.STN_NAME_D)THEN ! GENERATE A BLANK OUTPUT RECORD
C         STN_NAME_O=STN_NAME_F ! DO NOT GENERATE BLANK RECORD -
C         Y_M_D_O=Y_M_D_T  ! USE TODAYS DATE            10 JAN 89 - RJS
C         ALL_VALUE=BLANK_28
C
C*****THE STATION HAS NO DATA SO FIX UP AN OUTPUT RECORD
C*****AND WRITE IT - CHANGE "I" RECORDS TO "D" RECORDS.
C
C         IF(OUT_REC(9:9).EQ.'I') OUT_REC(9:9)='D'
C         WRITE(7,23)OUT_REC
          GO TO 20 ! GO READ NEXT RECORD FROM FORMAT FILE
       ENDIF
C
C*****DID WE GET DATA FOR A STATION THAT IS NOT IN THE
C*****FORMAT FILE ?  IF SO READ ANOTHER RECORD AND GO BACK
C*****TO CHECK FOR MATCH.
C
       IF(STN_NAME_F.GT.STN_NAME_D)THEN ! THROW DATA AWAY
          READ(2,10,END=500)STN_NAME,Y_M_D,PCODE,VALUE
          GO TO 22
       ENDIF
C
C*****WE HAVE A MATCH, SO BEGIN TO BUILD AN OUTPUT RECORD.
C*****BLANK OUT THE OUTPUT STRING - BUILD FORMAT TABLE
C*****FROM THE RECORD READ FROM FORMAT.STE.  FIRST FIND THE
C*****LENGTH (F_PCODE_L) OF EACH OF THE POSSIBLE 4 PARAMETER
C*****CODES AND THEN REWRITE THEM IN ARRAY F_PCODE.
C
       POS1=2
       POS2=0
       DO 30 I=1,4
       IF(POS1.EQ.0)THEN  ! THERE ARE NO MORE PARAMETERS
          F_PCODE_L(I)=0
       ELSE
          POS2=INDEX(F_STRING(POS1:),',') ! FIND COMMA AFTER PARAMETER CODE
          IF(POS2.EQ.0)THEN ! WE HAVE FOUND LAST CODE IN STRING
            IF(F_STRING(POS1+1:POS1+1).EQ.' ') THEN
               F_PCODE_L(I)=1
            ELSE
               F_PCODE_L(I)=2
            ENDIF
            F_PCODE(I)(1:F_PCODE_L(I))=F_STRING(POS1:POS1+F_PCODE_L(I)-1)
            POS1=0
          ELSE
            F_PCODE_L(I)=POS2-1 ! GET NEXT PARAMETER CODE LENGTH
            F_PCODE(I)(1:F_PCODE_L(I))=F_STRING(POS1:POS1+POS2-1)
             POS1=POS1+POS2
          ENDIF
       ENDIF
```

```
115     30 CONTINUE
116        STN_NAME_O=STN_NAME_F ! INSERT STATION NAME
117     35 Y_M_D_O=Y_M_D
118        ALL_VALUE=BLANK_28
119  C
120  C*****WHEN PCODE IS AN '*' THEN MAKE CORRESPONDING OUTPUT FIELD BLANK.
121  C
122        DO 37 I=1,4
123        IF(F_PCODE(I)(1:F_PCODE_L(I)).EQ.'*') VALUE_O(I)(1:7)='       '
124     37 CONTINUE
125  C
126  C*****FIGURE OUT WHERE IN THE OUTPUT RECORD THAT THE DATA WILL
127  C*****BE PLACED.
128  C
129     36 IF(PCODE(2:2).EQ.' ') THEN ! DETERMINE PARAMETER CODE LENGTH
130           PCODE_L=1
131        ELSE
132           PCODE_L=2
133        ENDIF
134        I=1
135        DO WHILE (I.LE.4)
136        IF(PCODE_L.NE.F_PCODE_L(I)) GO TO 55 ! WRONG PARAMETER CODE LENGTH
137        IF(PCODE(1:PCODE_L).NE.F_PCODE(I)(1:F_PCODE_L(I))) GO TO 55
138  C
139  C*****WE HAVE A PARAMETER CODE MATCH, NOW SETUP TO MOVE OVER 7 SIGNIFANT
140  C*****DIGITS TO OUTPUT RECORD.  IF DATA IS MISSING LEAVE ALL BLANKS.
141  C
142        IF(PCODE(3:3).EQ.'M'.OR.VALUE(1:7).EQ.' 998877') THEN
143           VALUE_O(I)(1:7)='       '
144           GO TO 90
145        ENDIF
146        IF(VALUE(1:3).EQ.'000'.OR.VALUE(1:3).EQ.'   ') THEN
147           VALUE_O(I)(1:7)=VALUE(4:10)
148        ELSE
149           VALUE_O(I)(1:7)=VALUE(1:7)
150        ENDIF
151        GO TO 90
152     55 I=I+1
153        ENDDO
154  C
155  C*****READ NEXT HYDROMET DATA RECORD
156  C
157     90 READ(2,10,END=110) STN_NAME_D,Y_M_D,PCODE,VALUE
158  C
159  C*****HAS STATION CHANGED OR DATE CHANGED
160  C
161        IF(STN_NAME_D.NE.STN_NAME_O.OR.Y_M_D.NE.Y_M_D_O) THEN
162           IF(OUT_REC(9:9).EQ.'I') OUT_REC(9:9)='D'
163           WRITE(7,23)OUT_REC
164     23    FORMAT(A)
165           IF(STN_NAME_D.NE.STN_NAME_O) GO TO 20 ! GO GET NEXT FORMAT RECORD
166           GO TO 35 ! SAME STATION NEW DAY
167        ELSE
168           GO TO 36 ! SAME STATION, SAME DAY, GO FIND WHICH PARAMETER
169        ENDIF
170  C
171  C*****WE HAVE RUN OUT OF DATA
```

```
            172 C*****SET UP AN OUTPUT RECORD FOR CURRENT RECORD ON FORMAT FILE
            173 C*****AND WRITE IT OUT. THEN GENERATE RECORDS FOR ALL OTHER STATIONS
            174 C*****ON FORMAT FILE. (DO NOT GENERATE RECORDS - 10 JAN 89 - RJS)
            175 C
            176 C  100 STN_NAME_O=STN_NAME_F
            177 C      Y_M_D_O=Y_M_D_T
            178 C      ALL_VALUE=BLANK_28
            179 C
            180 C*****WHEN PCODE ON FORMAT RECORD IS AN '*' THEN MAKE THE
            181 C*****CORRESPONDING OUTPUT FIELD ZEROS
            182 C
            183   110 CONTINUE
            184       DO 120 I=1,4
            185       IF(F_PCODE(I)(1:F_PCODE_L(I)).EQ.'*')
            186      1    VALUE_O(I)(1:7)='       '
            187   120 CONTINUE
            188       IF(OUT_REC(9:9).EQ.'I') OUT_REC(9:9)='D'
            189       WRITE(7,23) OUT_REC
            190 C
            191 C*****READ THE REST OF RECORDS FROM THE FORMAT FILE AND GENERATE OUTPUT
            192 C*****RECORDS. (DO NOT READ REST OF FORMAT RECORDS - 10 JAN 89 - RJS)
            193 C
            194 C  500 READ(1,21,END=1000) STN_NAME
            195 C      GO TO 100
            196   500 CONTINUE
            197  1000 STOP
            198       END


PROGRAM SECTIONS

    Name                              Bytes   Attributes

  1 $BSS$                               264 NOPIC CON REL LCL NOSHR NOEXE   RD   WRT OCTA
  2 $DATA$                              200 NOPIC CON REL LCL NOSHR NOEXE   RD   WRT OCTA
  3 $CODE$                             4900   PIC CON REL LCL   SHR   EXE NORD NOWRT OCTA
  4 $LINK$                              469 NOPIC CON REL LCL NOSHR NOEXE   RD NOWRT OCTA

    Total Space Allocated              5833


ENTRY POINTS

 Address      Name

  3-00000000  SNKFORMAT$MAIN
```

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name	Address	Type	Name
1-000000C9	CHAR	ALL_VALUE	2-00000068	CHAR	MONTHS	1-00000020	I*4	POS2	1-00000088	CHAR	VALUE
2-000000A8	CHAR	BLANK_28	1-000000B8	CHAR	OUT_REC	1-00000028	I*4	STN_NAME	1-00000010	CHAR	Y_M_D
1-00000040	CHAR	F_STRING	1-00000008	CHAR	PCODE	1-00000098	CHAR	STN_NAME_D	1-000000C1	CHAR	Y_M_D_O
1-0000001C	I*4	I	1-00000018	I*4	PCODE_L	1-00000078	CHAR	STN_NAME_F	1-00000030	CHAR	Y_M_D_T
1-0000002C	I*4	J	1-00000024	I*4	POS1	1-000000B8	CHAR	STN_NAME_O			

ARRAYS

Address	Type	Name	Bytes	Dimensions
1-00000000	CHAR	F_PCODE	8	(4)
1-000000A8	I*4	F_PCODE_L	16	(4)
1-000000C9	CHAR	VALUE_O	80	(9)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
**	20	3-00000A54	35	3-00000D40	55	3-0000114C	110	3-0000130C	1000
3-00000494	22	3-00000BEC	36	**	90	3-00001274	120		
3-00000A00	30	3-00000BCC	37	**	LA	3-00001304	500		

```
COMMAND QUALIFIERS

  /ALIGNMENT=(COMMONS=(NONATURAL,PACKED,NOSTANDARD,NOMULTILANGUAGE),RECORDS=NATURAL,NOSEQUENCE)
  /ANNOTATIONS=(NOCODE,NODETAIL,NOFEEDBACK,NOINLINING,NOLOOP_TRANSFORMS,NOLOOP_UNROLLING,NOPREFETCHING,NOSHRINKWRAPPING,
          NOSOFTWARE_PIPELINING,NOTAIL_CALLS,NOTAIL_RECURSION)
  /ARCHITECTURE=GENERIC
  /ASSUME=(ACCURACY_SENSITIVE,ALTPARAM,NOBUFFERED_IO,NOBYTERECL,NODUMMY_ALIASES,NOF77RTL,NOFP_CONSTANT,NOINT_CONSTANT,
          NOMINUS0,PROTECT_CONSTANTS,NOSOURCE_INCLUDE,NOUNDERSCORE)
  /NOAUTOMATIC
  /NOBY_REF_CALL
  /CCDEFAULT=DEFAULT
  /CHECK=(NOARG_TEMP_CREATED,BOUNDS,FORMAT,NOFP_EXCEPTIONS,OUTPUT_CONVERSION,NOOVERFLOW,POWER,NOUNDERFLOW)
  /CONVERT=NATIVE
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /NODEFINE
  /DOUBLE_SIZE=64
  /NOD_LINES
  /ERROR_LIMIT=30
  /NOEXTEND_SOURCE
  /F77
  /NOFAST
  /FLOAT=G_FLOAT
  /GRANULARITY=QUADWORD
  /IEEE_MODE=FAST
  /INTEGER_SIZE=32
  /NOMACHINE_CODE
  /MATH_LIBRARY=ACCURATE
  /NOMODULE
  /NAMES=UPPERCASE
  /OPTIMIZE=(INLINE=NONE,LEVEL=,NOLOOPS,NOPIPELINE,TUNE=GENERIC,UNROLL=0)
  /NOPAD_SOURCE
  /REAL_SIZE=32
  /NORECURSIVE
  /REENTRANCY=NONE
  /ROUNDING_MODE=NEAREST
  /NOSEPARATE_COMPILATION
  /SEVERITY=(WARNING=WARNING)
  /SHOW=(NOINCLUDE,MAP)
  /SOURCE_FORM=FIXED
  /STANDARD=NONE
  /NOSYNCHRONOUS_EXCEPTIONS
  /NOSYNTAX_ONLY
  /NOTIE
  /VMS
  /WARNINGS=(ALIGNMENT,NOARGUMENT_CHECKING,NODECLARATIONS,GENERAL,GRANULARITY,NOIGNORE_LOC,NOTRUNCATED_SOURCE,UNCALLED,
          UNINITIALIZED,NOUNUSED,USAGE)

  /NOANALYSIS_DATA
  /NODIAGNOSTICS
  /INCLUDE=FORT$INCLUDE:
  /LIST=HYDRO:[PPACE.SNAKE]SNKFORMAT.LIS;5
  /OBJECT=HYDRO:[PPACE.SNAKE]SNKFORMAT.OBJ;1
  /NOLIBRARY

COMPILER: Compaq Fortran V7.5-1961-48BCD
```

COMPILATION STATISTICS

CPU time:	0.12 seconds
Elapsed time:	1.31 seconds
Pagefaults:	833
I/O Count:	14

```
0001    C
0002    C***** REFORMAT THE HYDROMET DATA FILES ACCORDING TO
0003    C***** THE FORMAT THAT WAS READ FROM FORMAT FILE (SNKFORMAT.DAT).
0004    C
0005    C***** IF COLUMN 9 OF THE OUTPUT RECORD IS AN I (INST DIVERSION) THEN MAKE
0006    C***** IT A D BEFORE OUTPUTING RECORD.
0007    C
0008    C*****MODIFIED JANUARY 1989 TO SET UP FILES TO BE TRANSFERRED TO IDWR
0009    C*****VAX - BOB SUTTER.
0010    C
0011          IMPLICIT INTEGER*4(A-Z)
0012          CHARACTER STN_NAME_D*9,Y_M_D*8,PCODE*3,VALUE*10
0013          CHARACTER STN_NAME_F*9,F_STRING*56
0014          CHARACTER STN_NAME_O*9,Y_M_D_O*8,VALUE_O(9)*7
0015          CHARACTER BLANK_28*28,ALL_VALUE*28,OUT_REC*80,Y_M_D_T*9
0016          CHARACTER MONTHS*60
0017          INTEGER F_PCODE_L(4)
0018          CHARACTER F_PCODE(4)*2
0019          EQUIVALENCE (OUT_REC,STN_NAME_O),(OUT_REC(10:),Y_M_D_O)
0020          EQUIVALENCE (OUT_REC(18:),VALUE_O(1))
0021          EQUIVALENCE (ALL_VALUE,VALUE_O(1))
0022          DATA BLANK_28(1:28)/'                            '/
0023          DATA MONTHS(1:30)/'JAN01FEB02MAR03APR04MAY05JUN06'/
0024          DATA MONTHS(31:60)/'JUL07AUG08SEP09OCT10NOV11DEC12'/
0025    C
0026    C*****GET TODAYS DATE TO PLUG IN FOR STATIONS ON FORMAT FILE
0027    C*****THAT HAVE NO DATA
0028    C
0029          CALL DATE(Y_M_D_T) ! GET TODAYS DATE
0030    C
0031    C*****REFORMAT DATE INTO YYYYMMDD
0032    C
0033          J=INDEX(MONTHS,Y_M_D_T(4:6))
0034          Y_M_D_T(5:6)=MONTHS(J+3:J+4)
0035          Y_M_D_T(3:4)=Y_M_D_T(8:9)
0036          Y_M_D_T(7:8)=Y_M_D_T(1:2)
0037          IF(Y_M_D_T(3:4).LT.'80')THEN  ! SHOULD YEAR 19XX OR 20XX ?
0038             Y_M_D_T(1:2)='20'
0039             ELSE
0040             Y_M_D_T(1:2)='19'
0041          ENDIF
0042          OPEN(UNIT=2,FILE='INFILE',STATUS='OLD') ! GET HYDROMET DATA FILE
0043          OPEN(UNIT=1,STATUS='OLD',FILE='SNKFORMAT.DAT') ! GET FORMAT FILE
0044          OPEN(UNIT=7,STATUS='NEW',FILE='OUTFILE',CARRIAGECONTROL='LIST') ! NEW FILE
0045    C
0046    C*****READ FROM HYDROMET DATA FILE
0047    C
0048          READ(2,10,END=500) STN_NAME_D,Y_M_D,PCODE,VALUE
0049       10 FORMAT(A9,A8,A3,A10)
0050    C
0051    C*****READ FROM STATE FORMAT FILE
0052    C
0053       20 READ(1,21,END=1000) STN_NAME_F,F_STRING
0054       21 FORMAT(A9,A36)
0055    C
0056    C*****COMPARE STATE AND DATA RECORD FOR MATCHUP
0057    C
```

```
0058        22 IF(STN_NAME_F.LT.STN_NAME_D)THEN ! GENERATE A BLANK OUTPUT RECORD
0059     C          STN_NAME_O=STN_NAME_F ! DO NOT GENERATE BLANK RECORD -
0060     C          Y_M_D_O=Y_M_D_T  ! USE TODAYS DATE            10 JAN 89 - RJS
0061     C          ALL_VALUE=BLANK_28
0062     C
0063     C*****THE STATION HAS NO DATA SO FIX UP AN OUTPUT RECORD
0064     C*****AND WRITE IT - CHANGE "I" RECORDS TO "D" RECORDS.
0065     C
0066     C          IF(OUT_REC(9:9).EQ.'I') OUT_REC(9:9)='D'
0067     C          WRITE(7,23)OUT_REC
0068              GO TO 20 ! GO READ NEXT RECORD FROM FORMAT FILE
0069           ENDIF
0070     C
0071     C*****DID WE GET DATA FOR A STATION THAT IS NOT IN THE
0072     C*****FORMAT FILE ?  IF SO READ ANOTHER RECORD AND GO BACK
0073     C*****TO CHECK FOR MATCH.
0074     C
0075           IF(STN_NAME_F.GT.STN_NAME_D)THEN ! THROW DATA AWAY
0076              READ(2,10,END=500)STN_NAME,Y_M_D,PCODE,VALUE
0077              GO TO 22
0078           ENDIF
0079     C
0080     C*****WE HAVE A MATCH, SO BEGIN TO BUILD AN OUTPUT RECORD.
0081     C*****BLANK OUT THE OUTPUT STRING - BUILD FORMAT TABLE
0082     C*****FROM THE RECORD READ FROM FORMAT.STE.  FIRST FIND THE
0083     C*****LENGTH (F_PCODE_L) OF EACH OF THE POSSIBLE 4 PARAMETER
0084     C*****CODES AND THEN REWRITE THEM IN ARRAY F_PCODE.
0085     C
0086           POS1=2
0087           POS2=0
0088           DO 30 I=1,4
0089           IF(POS1.EQ.0)THEN  ! THERE ARE NO MORE PARAMETERS
0090              F_PCODE_L(I)=0
0091           ELSE
0092              POS2=INDEX(F_STRING(POS1:),',') ! FIND COMMA AFTER PARAMETER CODE
0093              IF(POS2.EQ.0)THEN ! WE HAVE FOUND LAST CODE IN STRING
0094                IF(F_STRING(POS1+1:POS1+1).EQ.' ') THEN
0095                   F_PCODE_L(I)=1
0096                ELSE
0097                   F_PCODE_L(I)=2
0098                ENDIF
0099                F_PCODE(I)(1:F_PCODE_L(I))=F_STRING(POS1:POS1+F_PCODE_L(I)-1)
0100                POS1=0
0101             ELSE
0102                F_PCODE_L(I)=POS2-1 ! GET NEXT PARAMETER CODE LENGTH
0103                F_PCODE(I)(1:F_PCODE_L(I))=F_STRING(POS1:POS1+POS2-1)
0104                  POS1=POS1+POS2
0105             ENDIF
0106           ENDIF
0107        30 CONTINUE
0108           STN_NAME_O=STN_NAME_F ! INSERT STATION NAME
0109        35 Y_M_D_O=Y_M_D
0110           ALL_VALUE=BLANK_28
0111     C
0112     C*****WHEN PCODE IS AN '*' THEN MAKE CORRESPONDING OUTPUT FIELD BLANK.
0113     C
0114           DO 37 I=1,4
```

```
0115          IF(F_PCODE(I)(1:F_PCODE_L(I)).EQ.'*') VALUE_O(I)(1:7)='       '
0116       37 CONTINUE
0117    C
0118    C*****FIGURE OUT WHERE IN THE OUTPUT RECORD THAT THE DATA WILL
0119    C*****BE PLACED.
0120    C
0121       36 IF(PCODE(2:2).EQ.' ') THEN ! DETERMINE PARAMETER CODE LENGTH
0122             PCODE_L=1
0123          ELSE
0124             PCODE_L=2
0125          ENDIF
0126          I=1
0127          DO WHILE (I.LE.4)
0128          IF(PCODE_L.NE.F_PCODE_L(I)) GO TO 55 ! WRONG PARAMETER CODE LENGTH
0129          IF(PCODE(1:PCODE_L).NE.F_PCODE(I)(1:F_PCODE_L(I))) GO TO 55
0130    C
0131    C*****WE HAVE A PARAMETER CODE MATCH, NOW SETUP TO MOVE OVER 7 SIGNIFANT
0132    C*****DIGITS TO OUTPUT RECORD.  IF DATA IS MISSING LEAVE ALL BLANKS.
0133    C
0134          IF(PCODE(3:3).EQ.'M'.OR.VALUE(1:7).EQ.' 998877') THEN
0135             VALUE_O(I)(1:7)='       '
0136             GO TO 90
0137          ENDIF
0138          IF(VALUE(1:3).EQ.'000'.OR.VALUE(1:3).EQ.'   ') THEN
0139             VALUE_O(I)(1:7)=VALUE(4:10)
0140          ELSE
0141             VALUE_O(I)(1:7)=VALUE(1:7)
0142          ENDIF
0143          GO TO 90
0144       55 I=I+1
0145          ENDDO
0146    C
0147    C*****READ NEXT HYDROMET DATA RECORD
0148    C
0149       90 READ(2,10,END=110) STN_NAME_D,Y_M_D,PCODE,VALUE
0150    C
0151    C*****HAS STATION CHANGED OR DATE CHANGED
0152    C
0153          IF(STN_NAME_D.NE.STN_NAME_O.OR.Y_M_D.NE.Y_M_D_O) THEN
0154             IF(OUT_REC(9:9).EQ.'I') OUT_REC(9:9)='D'
0155             WRITE(7,23)OUT_REC
0156       23    FORMAT(A)
0157             IF(STN_NAME_D.NE.STN_NAME_O) GO TO 20 ! GO GET NEXT FORMAT RECORD
0158             GO TO 35 ! SAME STATION NEW DAY
0159          ELSE
0160             GO TO 36 ! SAME STATION, SAME DAY, GO FIND WHICH PARAMETER
0161          ENDIF
0162    C
0163    C*****WE HAVE RUN OUT OF DATA
0164    C*****SET UP AN OUTPUT RECORD FOR CURRENT RECORD ON FORMAT FILE
0165    C*****AND WRITE IT OUT. THEN GENERATE RECORDS FOR ALL OTHER STATIONS
0166    C*****ON FORMAT FILE. (DO NOT GENERATE RECORDS - 10 JAN 89 - RJS)
0167    C
0168    C  100 STN_NAME_O=STN_NAME_F
0169    C      Y_M_D_O=Y_M_D_T
0170    C      ALL_VALUE=BLANK_28
0171    C
```

```
0172     C*****WHEN PCODE ON FORMAT RECORD IS AN '*' THEN MAKE THE
0173     C*****CORRESPONDING OUTPUT FIELD ZEROS
0174     C
0175       110 CONTINUE
0176           DO 120 I=1,4
0177           IF(F_PCODE(I)(1:F_PCODE_L(I)).EQ.'*')
0178          1    VALUE_O(I)(1:7)='       '
0179       120 CONTINUE
0180           IF(OUT_REC(9:9).EQ.'I') OUT_REC(9:9)='D'
0181           WRITE(7,23) OUT_REC
0182     C
0183     C*****READ THE REST OF RECORDS FROM THE FORMAT FILE AND GENERATE OUTPUT
0184     C*****RECORDS. (DO NOT READ REST OF FORMAT RECORDS - 10 JAN 89 - RJS)
0185     C
0186     C  500 READ(1,21,END=1000) STN_NAME
0187     C      GO TO 100
0188       500 CONTINUE
0189      1000 STOP
0190           END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	967	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	73	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	528	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1568	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKFORMAT$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
2-00000011	CHAR	ALL_VALUE	EQUIV	15 21 110=
2-000000C7	CHAR	BLANK_28		15 22D 110
2-0000008F	CHAR	F_STRING		13 53= 92 94 99 103
**	I*4	I		88= 90 95 97 99(3) 102 103(2) 114= 115(3) 126= 127 129(2) 135 139 141 144(2)= 177(3)
**	I*4	J		33= 34(2)
2-000000EC	CHAR	MONTHS		16 23D 24D 33 34
2-00000000	CHAR	OUT_REC	EQUIV	15 19(2) 20 154(2)= 155 180(2)= 181

```
  2-00000079  CHAR PCODE                                    12      48=     76=     121     129     134
                                                                    149=
      **      I*4  PCODE_L                                 122=    124=    128     129
      **      I*4  POS1                                     86=     89      92      94(2)   99(2)   100=
                                                                    103(2)  104(2)=

      **      I*4  POS2                                     87=     92=     93      102     103     104
  2-00000128  I*4  STN_NAME                                 76=
  2-00000068  CHAR STN_NAME_D                               12      48=     58      75      149=    153
                                                                    157
  2-00000086  CHAR STN_NAME_F                               13      53=     58      75      108
  2-00000000  CHAR STN_NAME_O                     EQUIV     14      19      108=    153     157

  2-0000007C  CHAR VALUE                                    12      48=     76=     134     138(2)  139
                                                                    141     149=
  2-00000071  CHAR Y_M_D                                    12      48=     76=     109     149=    153
  2-00000009  CHAR Y_M_D_O                        EQUIV     14      19      109=    153
  2-000000E3  CHAR Y_M_D_T                                  15      29A     33      34=     35(2)=  36(2)=
                                                                    37      38=     40=


ARRAYS

    Address   Type  Name          Attributes       Bytes  Dimensions     References

  2-00000060  CHAR F_PCODE                             8  (4)            18      99=     103=    115     129
                                                                                         177
  2-00000050  I*4  F_PCODE_L                          16  (4)            17      90=     95=     97=     99(2)
                                                                                         102=    103     115     128
                                                                                         177
  2-00000011  CHAR VALUE_O             EQUIV          63  (9)            14      20      21      115=    135=
                                                                                         139=    141=    177=


LABELS

    Address   Label                                  References

  1-0000001F  10'                                       48      49#     76      149
      **      20                                        53#     68      157
  1-00000028  21'                                       53      54#
      **      22                                        58#     77
  1-0000002D  23'                                      155     156#    181

      **      30                                        88     107#
  0-000001E8  35                                       109#    158
      **      36                                       121#    160
      **      37                                       114     116#
  0-000002C0  55                                       128     129     144#

  0-000002C8  90                                       136     143     149#
  0-00000364  110                                      149     175#
      **      120                                      176     179#
  0-000003C0  500                                       48      76     188#
  0-000003C0  1000                                      53     189#
```

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References		
	FOR$DATE_T_DS	29		
	FOR$OPEN	42	43	44
I*4	LIB$INDEX	33	92	

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 SNKFORMAT

/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.SNAKE]SNKFORMAT.LIS;1
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKFORMAT.OBJ;1

COMPILATION STATISTICS

Run Time:	7.28 seconds
Elapsed Time:	10.70 seconds
Page Faults:	429
Dynamic Memory:	467 pages

SNK HST.SUMW).LIS

```
0001    C  ************************************************
0002    C     THIS PROGRAM LISTS DATA FROM WD-01 HISTORY FILES
0003    C      IN USGS STYLE RECORDS BY WATER YEAR
0004    C        AND SUMS STATIONS BETWEEN DESIGNATED
0005    C                     POINTS.
0006    C
0007    C     THIS PROGRAM IS A MODIFICATION OF SNKHSTLST.FOR
0008    C                   BOB SUTTER - FEBRUARY 1989
0009    C ************************************************
0010    C
0011          LOGICAL*1 FILES(40)
0012          BYTE CNTRL
0013          INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
0014          REAL MEAN,MAX,MIN,NOVALUE
0015          CHARACTER NAME*70,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
0016          CHARACTER TTL*70,BLANK*1,SNAME*70,NNAME*70
0017          DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
0018          DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12),SDV(12,31)
0019          DIMENSION JDATE(12,31)
0020          DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
0021          CNTRL='00'X
0022          NOVALUE=9999999.
0023          BLANK=' '
0024          OPEN(UNIT=4,FILE='SNKHSTSUM.TTL',STATUS='OLD')
0025          OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
0026          OPEN(UNIT=8,FILE='SNKHSTSUM.OUT',STATUS='NEW',RECL=133)
0027    C
0028    C***** INQUIRE IF FILE IS TO BE USED TO SELECT SPECIFIC STATIONS TO BE
0029    C***** SUMMED FROM DATA FILE.  IF NOT, ALL DATA IN FILE WILL BE
0030    C***** SUMMED.
0031    C
0032      500 PRINT *,'************************************'
0033          PRINT *,'   YOU HAVE A CHOICE:               '
0034          PRINT *,'                                    '
0035          PRINT *,' EITHER SELECT STATIONS FROM INPUT FILE TO BE SUMMED,'
0036          PRINT *,' OR SUM ALL DATA OF SAME DAY IN INPUT FILE.          '
0037          PRINT *,'                                    '
0038          PRINT 904
0039      904 FORMAT('  TO USE A FILE OF SELECTED STATIONS ENTER Y/N: '$)
0040          ACCEPT 4, SEL
0041        4 FORMAT(A1)
0042          IF(SEL.NE.'Y') GO TO 11
0043    C
0044    C***** READ IN A LIST OF TITLES FOR EACH DATA SET TO BE SUMMED.  IF A
0045    C***** TITLE IS NOT PRESENT, THAT DATA WILL THEN NOT BE INCLUDED IN THE
0046    C***** SUMMED DATA SET.
0047    C
0048          PRINT *,'                                    '
0049          PRINT *,'************************************'
0050          PRINT *,'                                    '
0051          PRINT *,'   YOU HAVE ANOTHER CHOICE:         '
0052          PRINT *,'                                    '
0053          PRINT *,'  SUM ALL DIVERSIONS, FLOWS, PUMPS, OR RESERVOIRS '
0054          PRINT *,'  (STATION NUMBERS & TITLES READ FROM EXISTING FILE)'
0055          PRINT *,'                                    '
0056          PRINT *,'              OR                    '
0057          PRINT *,'                                    '
```

```
0058          PRINT *,'  SUM ONLY STATIONS WHICH YOU SELECT INDIVIDUALLY   '
0059          PRINT *,'  (YOU PROVIDE THE STATION NUMBER AND TITLE) '
0060          PRINT *,'                                              '
0061          PRINT *,'   TO SUM ALL DIVERSIONS, ANSWER  D          '
0062          PRINT *,'   TO SUM ALL RIVER FLOWS, ANSWER  F          '
0063          PRINT *,'   TO SUM ALL PUMPS, ANSWER  P          '
0064          PRINT *,'   TO SUM ALL RESERVOIRS, ANSWER  R          '
0065          PRINT *,'   TO SUM INDIVIDUAL STATIONS, ANSWER  I  '
0066          PRINT *,'   TO ENTER YOUR OWN GROUP OF STATIONS, ANSWER G '
0067          PRINT *,'   TO QUIT, ANSWER  Q  '
0068          PRINT *,'                                              '
0069          PRINT 905
0070        905 FORMAT('         ENTER CHOICE:'$)
0071          ACCEPT 4, IND
0072          IF(IND.EQ.'Q') GO TO 470
0073         11 PRINT *,' '
0074          PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
0075          PRINT *,'   WATER YEAR FOR WHICH DATA IS TO BE LISTED - '
0076          PRINT *,' '
0077          PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR WATER YEAR  '
0078          PRINT *,'   1990, ENTER 89-90   '
0079          PRINT *,'                __-__    '
0080          PRINT 901
0081        901 FORMAT('+               '$)
0082          KY=0
0083          ACCEPT 900, NP,NP1
0084        900 FORMAT(I2,1X,I2)
0085        C
0086        C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD.
0087        C
0088          NP=NP+1900
0089          NP1=NP1+1900
0090          IF(JMOD(NP,4).EQ.0) KY=1
0091          ID=(NP*1000)+274+KY
0092          KY=0
0093          IF(JMOD(NP1,4).EQ.0) KY=1
0094          ID1=(NP1*1000)+273+KY
0095        C
0096        C***** READ IN TITLES OF DATA TO BE SUMMED
0097        C
0098          IF(SEL.NE.'Y') GO TO 2
0099          IF(IND.EQ.'I') THEN
0100                       OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
0101           1           PRINT *,'   ENTER STATION NUMBER AND TITLE '
0102                       PRINT *,'   TO END   ENTER   99999999  '
0103                       PRINT *,'            ________  '
0104                       PRINT 903
0105          903          FORMAT('+           '$)
0106                       ACCEPT 301, ISTA,TTL
0107                       IF(ISTA .EQ. 99999999) GO TO 9
0108                       WRITE(3,301) ISTA,TTL
0109                       GO TO 1
0110           9           REWIND 3
0111                       GO TO 2
0112                       END IF
0113          IF(IND.NE.'G') GO TO 8
0114          PRINT 907
```

```
0115      907 FORMAT(' ENTER FILE NAME OF TITLES OF DATA TO BE SUMMED: '$)
0116          ACCEPT 908, IC,(FILES(I),I=1,IC)
0117      908 FORMAT(Q,40A1)
0118          CALL ASSIGN(3,FILES)
0119          GO TO 2
0120        8 IF(IND.EQ.'D') OPEN(UNIT=3,FILE='SNKDIVSUM.TTL',STATUS='OLD')
0121          IF(IND.EQ.'F') OPEN(UNIT=3,FILE='SNKFLO.TTL',STATUS='OLD')
0122          IF(IND.EQ.'P') OPEN(UNIT=3,FILE='SNKPMP.TTL',STATUS='OLD')
0123          IF(IND.EQ.'R') OPEN(UNIT=3,FILE='SNKRES.TTL',STATUS='OLD')
0124        2 PRINT 3
0125        3 FORMAT(' TO LIST STATIONS BEING SUMMED ENTER Y/N: '$)
0126          ACCEPT 4,PIS
0127          PRINT 7
0128        7 FORMAT(' TO MAKE A SEPARATE FILE OF SUMS ENTER Y/N: '$)
0129          ACCEPT 4,FSUM
0130          IF(FSUM.EQ.'Y') OPEN(UNIT=9,FILE='SNKHSTSUM.DAT',STATUS='NEW',
0131         1CARRIAGECONTROL='LIST')
0132          INREC=0
0133          INSTA=0
0134          IEND=0
0135          ISR=1
0136          STATION_NUM=0
0137        5 ISUM=0
0138          DO 40 I=1,12
0139          DO 40 J=1,31
0140       40 SDV(I,J)=NOVALUE
0141      C
0142      C***** IF NOT SELECTING SPECIFIC DATA FROM DATA SET TO BE SUMMED
0143      C***** SKIP SEARCH FOR STATION ROUTINE AND STATION READ STATEMENT
0144      C
0145       10 IF(SEL.NE.'Y') GO TO 15
0146          IF(INSTA.EQ.1) THEN
0147                         STA_ID=NSTA_ID
0148                         NAME=NNAME
0149                         INSTA=0
0150                         GO TO 15
0151                         ENDIF
0152          READ(3,301,END=450) STA_ID,NAME
0153      301 FORMAT(I8,A70)
0154      C
0155      C***** FILL ARRAY WITH MISSING VALUE INDICATORS
0156      C
0157       15 DO 20 I=1,12
0158          DO 20 J=1,31
0159       20 DV(I,J)=NOVALUE
0160          IF(SEL.NE.'Y') GO TO 13
0161      C
0162      C***** SKIP READ STATEMENT IF FIRST RECORD PREVIOUSLY READ
0163      C
0164          IF(STATION_NUM.EQ.STA_ID) GO TO 60
0165          IF(STATION_NUM.GT.STA_ID) GO TO 651
0166      C
0167      C***** READ A RECORD (STATION SEARCH ROUTINE)
0168      C
0169       26 IF(SEL.NE.'Y') GO TO 13
0170       25 READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0171      701 FORMAT(I8,A1,I4,I3,F7.0)
```

```
0172          IF(STATION_NUM.LT.STA_ID) GO TO 25
0173          IF(STATION_NUM.EQ.STA_ID) GO TO 60
0174       651 IF(TD.NE.STA_ID) THEN
0175            PRINT 601,STA_ID
0176       601  FORMAT(1X,'STATION ',I8,' NOT FOUND')
0177            GO TO 10
0178            END IF
0179          READ(3,301,END=99) NSTA_ID,NNAME
0180          INSTA=1
0181        35 IF(STATION_NUM.LT.NSTA_ID) THEN
0182            READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
0183            GO TO 35
0184            END IF
0185          GO TO 100
0186     C
0187     C***** READ A RECORD (NO STATION SEARCH ROUTINE), WHEN THE STATION
0188     C***** NUMBER CHANGES, GO TO LISTING AND/OR SUMMING ROUTINE
0189     C
0190        13 IF(INSTA.EQ.1) GO TO 12
0191          READ(7,701,END=98) STATION_NUM,TYPE,YEAR,ND,VALUE
0192          IF(INREC.EQ.0) THEN
0193                         INREC=1
0194                         GO TO 12
0195                         ENDIF
0196          IF(STATION_NUM.LT.TD) GO TO 13
0197          IF(STATION_NUM.GT.TD) THEN
0198                               INSTA=1
0199                               GO TO 100
0200                               ENDIF
0201        12 STA_ID=STATION_NUM
0202          INSTA=0
0203     C
0204     C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
0205     C
0206        60 IDATE=(YEAR*1000)+ND
0207          IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 26
0208          JD(2)=28
0209          IF(JMOD(YEAR,4) .EQ. 0) JD(2)=29
0210          LD=ND
0211          MON=0
0212        62 MON=MON+1
0213          IF(MON.GT.12) GO TO 68
0214          LD=LD-JD(MON)
0215          IF(LD.GT.0) GO TO 62
0216          LD=LD+JD(MON)
0217          DV(MON,LD)=VALUE
0218          JDATE(MON,LD)=IDATE
0219     C
0220     C***** SAVE VARIABLE VALUES IN PREPARATION FOR READING NEXT RECORD
0221     C***** SO THAT IF STATION CHANGES, WE HAVE PRESERVED DATA FOR
0222     C***** STATION TO BE LISTED.
0223     C
0224        68 CONTINUE
0225          TD=STATION_NUM
0226          P_YEAR=YEAR
0227          P_TYPE=TYPE
0228          IF(SEL.NE.'Y') GO TO 13
```

```
0229          GO TO 25
0230      C
0231      C***** ENTIRE YEAR FOR ONE STATION READ IN -
0232      C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
0233      C***** IF LAST STATION TITLE HAS BEEN READ, SET NEXT STATION
0234      C***** NUMBER = 99999999 SO THAT SUMS WILL BE PRINTED.
0235      C
0236         98 IEND=1
0237         99 STATION_NUM=99999999
0238        100 IF(ISUM.NE.1.AND.PIS.NE.'Y') GO TO 6
0239            LEAP=0
0240            IF(JMOD(NP1,4).EQ.0) LEAP=1
0241            IF(P_TYPE.NE.'R') GO TO 120
0242            DELTA_S(1)=DV(1,31)-DV(12,31)
0243            DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
0244            DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
0245            DELTA_S(4)=DV(4,30)-DV(3,31)
0246            DELTA_S(5)=DV(5,31)-DV(4,30)
0247            DELTA_S(6)=DV(6,30)-DV(5,31)
0248            DELTA_S(7)=DV(7,31)-DV(6,30)
0249            DELTA_S(8)=DV(8,31)-DV(7,31)
0250            DELTA_S(9)=DV(9,30)-DV(8,31)
0251            DELTA_S(10)=NOVALUE
0252            DELTA_S(11)=DV(11,30)-DV(10,31)
0253            DELTA_S(12)=DV(12,31)-DV(11,30)
0254            DO 119 I=2,12
0255        119 IF(ABS(DELTA_S(I)).GE.8000000.) DELTA_S(I)=NOVALUE
0256        120 DO 135 I=1,12
0257            D=31
0258            IF(I.EQ.2) D=28+LEAP
0259            IF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.9) D=30
0260            CFSD(I)=DV(I,1)
0261            D_ADJ=0
0262            IF(DV(I,1).EQ.NOVALUE) THEN
0263                                   CFSD(I)=0.0
0264                                   D_ADJ=1
0265                                   END IF
0266            MAX(I)=DV(I,1)
0267            IF(DV(I,1).EQ.NOVALUE) MAX(I)=-1.0
0268            MIN(I)=DV(I,1)
0269            DO 130 J=2,D
0270            DZ=DV(I,J)
0271            IF(DV(I,J).EQ.NOVALUE) THEN
0272                                   DZ=0.0
0273                                   D_ADJ=D_ADJ+1
0274                                   END IF
0275            CFSD(I)=CFSD(I)+DZ
0276            IF(DV(I,J).EQ.NOVALUE) THEN
0277                                   GO TO 129
0278                                   END IF
0279            IF(DV(I,J).GT.MAX(I)) MAX(I)=DV(I,J)
0280        129 IF(DV(I,J).LT.MIN(I)) MIN(I)=DV(I,J)
0281        130 CONTINUE
0282            IF(D_ADJ.GE.D) THEN
0283                           CFSD(I)=NOVALUE
0284                           DELTA_S(I)=NOVALUE
0285                           MEAN(I)=NOVALUE
```

```
0286                          GO TO 133
0287                          END IF
0288          MEAN(I)=CFSD(I)/(D-D_ADJ)
0289       133 ACFT(I)=CFSD(I)*1.9835
0290       135 CONTINUE
0291          CALL DATE(RUNDATE)
0292          WRITE(8,801) RUNDATE
0293       801 FORMAT('1',/,120X,A9)
0294          IF(IND.EQ.'F'.OR.IND.EQ.'R') WRITE(8,800)
0295       800 FORMAT(1H+,'PRELIMINARY DATA')
0296          WRITE(8,802) STA_ID,NAME
0297       802 FORMAT(//37X,I8,3X,A70)
0298          IF(P_TYPE .EQ.'R') THEN
0299                               WRITE(8,803) NP,NP1
0300       803 FORMAT(30X,'CONTENTS IN ACRE FEET, WATER YEAR OCTOBER',
0301         *          I5,' TO SEPTEMBER',I5)
0302                               GO TO 160
0303                               ELSE
0304                               WRITE(8,804) NP,NP1
0305       804 FORMAT(30X,'DISCHARGE, CUBIC FEET PER SECOND, WATER YEAR',
0306         *          ' OCTOBER',I5,' TO SEPTEMBER',I5)
0307                               WRITE(8,805)
0308       805 FORMAT(59X,'MEAN VALUES')
0309                               END IF
0310       160 WRITE(8,806)
0311       806 FORMAT(//,1X,'DAY',11X,'OCT',7X,'NOV',7X,'DEC',7X,'JAN',7X,
0312         *          'FEB',7X,'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',
0313         *          7X,'AUG',7X,'SEP')
0314          WRITE(8,807)
0315       807 FORMAT(' ')
0316          DO 180 J=1,31
0317          IF(J.EQ.6.OR.J.EQ.11.OR.J.EQ.16.OR.J.EQ.21.OR.J.EQ.26) THEN
0318                                  WRITE(8,807)
0319                                    END IF
0320          DO 170 I=1,12
0321          II=I-3
0322          IF(I.EQ.1) II=10
0323          IF(I.EQ.2) II=11
0324          IF(I.EQ.3) II=12
0325          IF(II.EQ.10) WRITE(8,807)
0326          IF(II.EQ.10) WRITE(8,811) CNTRL,J
0327       811 FORMAT(A1,I3,'    ')
0328          IF(DV(II,J).EQ.NOVALUE) THEN
0329                                   WRITE(8,809) CNTRL
0330                                   GO TO 170
0331                                   END IF
0332          IF(DV(II,J).GE.10.0) THEN
0333                               IDV=JNINT(DV(II,J))
0334                               WRITE(8,810) CNTRL,IDV
0335       810                     FORMAT(A1,I10)
0336                               GO TO 170
0337                               END IF
0338          IF(DV(II,J).LT.10.0) THEN
0339                               WRITE(8,808) CNTRL,DV(II,J)
0340       808                     FORMAT(A1,F10.1)
0341                               GO TO 170
0342                               END IF
```

```
0343      809                       FORMAT(A1,'       ---')
0344      170 CONTINUE
0345      180 CONTINUE
0346          IF(P_TYPE.EQ.'R') THEN
0347                            WRITE(8,812)
0348      812                   FORMAT('0')
0349                            WRITE(8,813) CNTRL
0350      813                   FORMAT(A1,2X,'MAX  ')
0351                            DO 190 I=1,12
0352                            II=I-3
0353                            IF(I.EQ.1) II=10
0354                            IF(I.EQ.2) II=11
0355                            IF(I.EQ.3) II=12
0356                            IF(MAX(II).EQ.-1.0) THEN
0357                                      WRITE(8,860) CNTRL
0358      860                             FORMAT(A1,'          ')
0359                                      GO TO 190
0360                                      END IF
0361                            IDV=JNINT(MAX(II))
0362                            WRITE(8,810) CNTRL,IDV
0363      190                   CONTINUE
0364                            WRITE(8,807)
0365                            WRITE(8,814) CNTRL
0366      814                   FORMAT(A1,2X,'MIN  ')
0367                            DO 200 I=1,12
0368                            II=I-3
0369                            IF(I.EQ.1) II=10
0370                            IF(I.EQ.2) II=11
0371                            IF(I.EQ.3) II=12
0372                            IF(MIN(II).EQ.NOVALUE) THEN
0373                                                   WRITE(8,860) CNTRL
0374                                                   GO TO 200
0375                                                   END IF
0376                            IDV=JNINT(MIN(II))
0377                            WRITE(8,810) CNTRL,IDV
0378      200                   CONTINUE
0379                            WRITE(8,807)
0380                            WRITE(8,815) CNTRL
0381      815                   FORMAT(A1,1X,'CHNG            ')
0382                            DO 210 I=2,12
0383                            II=I-3
0384                            IF(I.EQ.2) II=11
0385                            IF(I.EQ.3) II=12
0386                            IDV=JNINT(DELTA_S(II))
0387                            IF(DELTA_S(II).EQ.NOVALUE) THEN
0388                                                   WRITE(8,860) CNTRL
0389                                                   GO TO 210
0390                                                   END IF
0391                            WRITE(8,810) CNTRL,IDV
0392      210                   CONTINUE
0393                            GO TO 10
0394                            END IF
0395          WRITE(8,812)
0396          WRITE(8,816) CNTRL
0397      816 FORMAT(A1,'TOTAL  ')
0398          DO 220 I=1,12
0399          II=I-3
```

```
0400          IF(I.EQ.1) II=10
0401          IF(I.EQ.2) II=11
0402          IF(I.EQ.3) II=12
0403          IF(CFSD(II).GE.NOVALUE) THEN
0404                                  WRITE(8,860) CNTRL
0405                                  GO TO 220
0406                                  END IF
0407          IDV=JNINT(CFSD(II))
0408          WRITE(8,810) CNTRL,IDV
0409      220 CONTINUE
0410          WRITE(8,807)
0411          WRITE(8,818) CNTRL
0412      818 FORMAT(A1,' MEAN  ')
0413          DO 230 I=1,12
0414          II=I-3
0415          IF(I.EQ.1) II=10
0416          IF(I.EQ.2) II=11
0417          IF(I.EQ.3) II=12
0418          IF(MEAN(II).GE. NOVALUE) THEN
0419                                   WRITE(8,860) CNTRL
0420                                   GO TO 230
0421                                   END IF
0422          IF(MEAN(II).LT.10.0) THEN
0423                                WRITE(8,808) CNTRL,MEAN(II)
0424                                GO TO 230
0425                                END IF
0426          IDV=JNINT(MEAN(II))
0427          WRITE(8,810) CNTRL,IDV
0428      230 CONTINUE
0429          WRITE(8,807)
0430          WRITE(8,819) CNTRL
0431      819 FORMAT(A1,'  MAX  ')
0432          DO 240 I=1,12
0433          II=I-3
0434          IF(I.EQ.1) II=10
0435          IF(I.EQ.2) II=11
0436          IF(I.EQ.3) II=12
0437          IF(MAX(II).EQ.-1.0) THEN
0438                              WRITE(8,860) CNTRL
0439                              GO TO 240
0440                              END IF
0441          IF(MAX(II).LT.10.0) THEN
0442                              WRITE(8,808) CNTRL,MAX(II)
0443                              GO TO 240
0444                              END IF
0445          IDV=JNINT(MAX(II))
0446          WRITE(8,810) CNTRL,IDV
0447      240 CONTINUE
0448          WRITE(8,807)
0449          WRITE(8,820) CNTRL
0450      820 FORMAT(A1,'  MIN  ')
0451          DO 250 I=1,12
0452          II=I-3
0453          IF(I.EQ.1) II=10
0454          IF(I.EQ.2) II=11
0455          IF(I.EQ.3) II=12
0456          IF(MIN(II).GE.NOVALUE) THEN
```

```
0457                                WRITE(8,860) CNTRL
0458                                GO TO 250
0459                                END IF
0460          IF(MIN(II).LT.10.0) THEN
0461                               WRITE(8,808) CNTRL,MIN(II)
0462                               GO TO 250
0463                               END IF
0464          IDV=JNINT(MIN(II))
0465          WRITE(8,810) CNTRL,IDV
0466        250 CONTINUE
0467    C         DO 260 I=1,12
0468    C         IFT(I)=JNINT(ACFT(I)/100.)
0469    C         IFT(I)=IFT(I)*100
0470    C260      CONTINUE
0471          WRITE(8,807)
0472          WRITE(8,821) CNTRL
0473        821 FORMAT(A1,'AC-FT  ')
0474          DO 270 I=1,12
0475          II=I-3
0476          IF(I.EQ.1) II=10
0477          IF(I.EQ.2) II=11
0478          IF(I.EQ.3) II=12
0479          IF(ACFT(II).GE.19834900.0) THEN
0480                                     WRITE(8,860) CNTRL
0481                                     GO TO 270
0482                                     END IF
0483    C         IF(ACFT(II).LE.999.) THEN
0484    C                                 IFT(II)=JNINT(ACFT(II))
0485    C                                 WRITE(8,810) CNTRL,IFT(II)
0486    C                                 GO TO 270
0487    C                                 END IF
0488          IFT(II)=JNINT(ACFT(II))
0489          WRITE(8,810) CNTRL,IFT(II)
0490        270 CONTINUE
0491          YT=0.0
0492          DO 280 I=1,12
0493          IF(CFSD(I) .EQ. NOVALUE) CFSD(I)=0.0
0494        280 YT=YT + CFSD(I)
0495          IYT=JNINT(YT)
0496          IYM=JNINT(YT/(365+LEAP))
0497    C         NT=JNINT(YT*1.9835/100.)
0498    C         IYA=NT*100
0499    C         IF(IYA.LT.9999) IYA=1.9835*YT
0500          IYA=1.9835*YT
0501          WRITE(8,807)
0502          WRITE(8,831) NP1,IYT,IYM,IYA
0503        831 FORMAT(11X,'WATER YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
0504         *          I6,4X,'AC-FT',I8)
0505    C
0506    C***** ENTER SUMMING ROUTINE UNLESS END OF DATA OR SUM HAS JUST
0507    C***** BEEN PRINTED.
0508    C
0509          IF(IEND.EQ.1) GO TO 450
0510          IF(ISUM.EQ.1) GO TO 5
0511          6 DO 460 I=1,12
0512          DO 460 J=1,31
0513          IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).EQ.NOVALUE) GO TO 460
```

```
0514          IF(DV(I,J).GE.0.0.AND.SDV(I,J).EQ.NOVALUE) SDV(I,J)=0.0
0515          IF(DV(I,J).EQ.NOVALUE.AND.SDV(I,J).GT.0.0) DV(I,J)=0.0
0516          SDV(I,J)=SDV(I,J)+DV(I,J)
0517      460 CONTINUE
0518          IF(ISR.EQ.0) GO TO 480
0519          READ(4,301,END=10) IDSUM,SNAME
0520          ISR=0
0521      480 IF(STATION_NUM.LT.IDSUM) GO TO 10 ! GO TO NEXT STATION
0522      485 STA_ID=IDSUM
0523          NAME=SNAME
0524          DO 465 I=1,12
0525          DO 465 J=1,31
0526      465 DV(I,J)=SDV(I,J)
0527          ISUM=1
0528          ISR=1
0529      C
0530      C***** IF FSUM EQUALS 'Y' MAKE A FILE OF SUMMED DATA
0531      C
0532          IF(FSUM.NE.'Y') GO TO 100
0533          DO 885 K=1,12
0534          I=K+9
0535          IF(I.GT.12) I=I-12
0536          JMD=JD(I)
0537          DO 885 J=1,JMD
0538      885 IF(JDATE(I,J).GT.0.AND.SDV(I,J).NE.NOVALUE)
0539         1                  WRITE(9,880) IDSUM,JDATE(I,J),SDV(I,J)
0540      880 FORMAT(I8,'S',I7,F7.0)
0541          GO TO 100  ! ORGANIZE AND PRINT SUMS
0542      450 CONTINUE
0543          IF(ISR.EQ.0) GO TO 485
0544          PRINT *,'  '
0545          PRINT *,'  '
0546          PRINT *,'  REQUESTED STATION LISTINGS AND SUMS COMPLETE '
0547          PRINT *,'  '
0548          PRINT *,'  '
0549      470 PRINT *,'                                        '
0550          PRINT *,'  OUTPUT IS  THE LATEST VERSION OF SNKHSTSUM.OUT'
0551          PRINT *,'  '
0552          STOP
0553          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	6113	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	2228	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	5976	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	14317	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTSUMW$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	CHAR	BLANK		16 23=
**	L*1	CNTRL		12 21= 326 329 334 339 349 357 362 365 373 380 388 391 396 404 411 419 423 427 430 442 446 449 457 461 472 480 489
**	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) 263 266(2) 267(2) 268(2) 270 275(2) 276 279(4) 280(4) 283 285 288(2) 289(2) 320= 321 323 324 351= 352 353 355 367= 368 369 370 382= 383 384 385 398= 400 401 402 413= 414 416 417 432= 433 434 436 451= 452 453 454 474= 475 476 477 478 493(2) 494 511= 513(2) 514(3) 516(3) 524= 526(2) 534= 535(3)= 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=

```
                                                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
                                                353=     354=     355=     356      361
                                                369=     370=     371=     372      376
                                                384=     385=     386      387      399=
                                                401=     402=     403      407      414=
                                                416=     417=     418      422      423
                                                433=     434=     435=     436=     437
                                                442      445      452=     453=     454=
                                                456      460      461      464      475=
                                                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=
                                                326      328      332      333      338
                                                512=     513(2)   514(3)   515(3)   516(3)
                                                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
                                                285      328      372      387      403
                                                456      493      513(2)   514      515

  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= 196 201 225 237= 521
2-00001440	I*4	STA_ID	13 147= 152= 164 165 172 173 174 175 201= 296
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= 288 289 403 407 494
2-000006C0	R*4	DELTA_S		48	(12)	18 242= 243= 244= 245= 246= 247= 248= 249= 251= 252= 253= 255(386 387
2-000000C0	R*4	DV		1488	(12, 31)	17 159= 217= 242(2) 243(2) 244(2) 245(2) 246(2) 247(249(2) 250(2) 252(2) 253(262 266 267 268 271 276 279(2) 280(332 333 338 339 514 515(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)= 356 361 437 441 445
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
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#
1-00000620	3'	124 125#
1-0000059E	4'	40 41# 71 126 129
0-0000075C	5	137# 510
0-0000168C	6	238 511#
1-0000064E	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#
1-0000067E	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#
  1-0000068F  601'                                 175       176#
  0-00000982  651                                  165       174#
  1-00000683  701'                                 170       171#      182       191
  1-000006B3  800'                                 294       295#

  1-000006AA  801'                                 292       293#
  1-000006C9  802'                                 296       297#
  1-000006D4  803'                                 299       300#
  1-00000715  804'                                 304       305#
  1-00000763  805'                                 307       308#

  1-00000773  806'                                 310       311#
  1-000007D1  807'                                 314       315#      318       325       364       379
                                                              410       429       448       471       501
  1-000007E5  808'                                 339       340#      423       442       461
  1-000007EB  809'                                 329       343#
  1-000007E0  810'                                 334       335#      362       377       391       408
                                                              427       446       465       489

  1-000007D5  811'                                 326       327#
  1-000007FA  812'                                 347       348#      395
  1-000007FE  813'                                 349       350#
  1-00000819  814'                                 365       366#
  1-00000825  815'                                 380       381#

  1-0000083C  816'                                 396       397#
  1-00000848  818'                                 411       412#
  1-00000854  819'                                 430       431#
  1-00000860  820'                                 449       450#
  1-0000086C  821'                                 472       473#

  1-00000878  831'                                 502       503#
  1-0000080A  860'                                 357       358#      373       388       404       419
                                                              438       457       480
  1-000008A9  880'                                 538       540#
     **       885                                  533       537       538#
  1-000005CE  900'                                  83        84#

  1-000005BA  901'                                  80        81#
  1-000005D5  903'                                 104       105#
  1-0000056A  904'                                  38        39#
  1-000005A1  905'                                  69        70#
  1-000005E5  907'                                 114       115#

  1-0000061A  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 SNKHSTSUMW

/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.SNAKE]SNKHSTSUMW.LIS;12
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTSUMW.OBJ;2
```

COMPILATION STATISTICS

```
Run Time:           20.71 seconds
Elapsed Time:       26.56 seconds
Page Faults:        1528
Dynamic Memory:     1058 pages
```

```
00001   C  ******************************************************
00002   C     THIS PROGRAM LISTS DATA FROM WD-01 ALLOCATION FILES
00003   C       IN USGS STYLE RECORDS BY IRRIGATION YEAR
00004   C
00005   C              WRITTEN USING VAX FORTRAN
00006   C
00007   C       ADAPTED TO UPPER SNAKE APRIL 1989 - RJS
00008   C
00009   C ******************************************************
00010   C
00011         LOGICAL*1 FILES(40)
00012         BYTE CNTRL
00013         INTEGER TD,STA_ID,STATION_NUM,YEAR,D_ADJ,P_YEAR
00014         REAL MEAN,MAX,MIN,NOVALUE
00015         DIMENSION JD(12),DV(12,31),CFSD(12),MAX(12),MIN(12)
00016         DIMENSION DELTA_S(12),ACFT(12),MEAN(12),IFT(12),PRMTR(16)
00017         CHARACTER NAME*50,ZLCH*1,TYPE*1,P_TYPE*1,RUNDATE*9,TITLE*4
00018         CHARACTER TTL*50,BLANK*1,PRMTR*32,INT*1
00019         DATA JD/31,28,31,30,31,30,31,31,30,31,30,31/
00020         DATA PRMTR/'DISCHARGE, CUBIC FEET PER SECOND',
00021        1           'DAILY STORAGE DIVERSION, CFS    ',
00022        1           'CUMULATIVE STORAGE DIVERTED, AF ',
00023        1           'REMAINING STORAGE, ACRE-FEET    ',
00024        1           'CUMULATIVE TOTAL DIVERTED, AF   ',
00025        1           'NATURAL FLOW DIVERTED, CFS      ',
00026        1           'CUMULATIVE EVAPORATION, AF      ',
00027        1           'DAILY EVAPORATION, CFS          ',
00028        1           'NATURAL FLOW STORED, CFS        ',
00029        1           'CUMULATIVE TOTAL STORED AF      ',
00030        1           'CONTENTS AT HR 2400, ACRE-FEET  ',
00031        1           'NATURAL FLOW, CUBIC FEET/SECOND ',
00032        1           'DISCHARGE, CUBIC FEET PER SECOND',
00033        1           'STORED FLOW, CUBIC FEET/SECOND  ',
00034        1           'AVERAGED REACH GAIN, CFS        ',
00035        1           'PRIORITY DATE OF LAST FILL RIGHT'/
00036         CNTRL='00'X
00037         NOVALUE=9999999.
00038         BLANK=' '
00039         OPEN(UNIT=7,FILE='INPUTDATA',STATUS='OLD')
00040         OPEN(UNIT=8,FILE='SNKALCLST.OUT',STATUS='NEW',RECL=133)
00041       500 PRINT *,'                                                '
00042         PRINT *,'/*************************************************'
00043         PRINT *,'                                                '
00044         PRINT *,'   YOU HAVE A CHOICE:                          '
00045         PRINT *,'                                                '
00046         PRINT *,'   TO LIST DIVERSION DATA, ANSWER  D            '
00047         PRINT *,'   TO LIST RIVER FLOW DATA, ANSWER  F           '
00048         PRINT *,'   TO LIST RESERVOIR DATA, ANSWER  R            '
00049         PRINT *,'   TO QUIT, ANSWER  Q  '
00050         PRINT *,'                                                '
00051         PRINT *,'*************************************************'
00052         PRINT *,'                                                '
00053         PRINT 904
00054         ACCEPT 902, INT
00055         IF(INT.EQ.'Q') GO TO 470
00056         PRINT *,'*************************************************'
00057         PRINT *,'                                                '
```

```
00058         PRINT *,'   YOU HAVE ANOTHER CHOICE:                          '
00059         PRINT *,'                                                     '
00060         PRINT *,'   LIST GROUPS OF DIVERSIONS, FLOWS, OR RESERVOIRS   '
00061         PRINT *,'   (STATION NUMBERS AND TITLES READ IN FROM FILES)   '
00062         PRINT *,'                                                     '
00063         PRINT *,'                              OR                     '
00064         PRINT *,'                                                     '
00065         PRINT *,'   LIST STATIONS INDIVIDUALLY                        '
00066         PRINT *,'   (YOU PROVIDE THE STATION NUMBER AND TITLE)        '
00067         PRINT *,'                                                     '
00068         PRINT *,'                              OR                     '
00069         PRINT *,'                                                     '
00070         PRINT *,'   YOU PROVIDE A FILE NAME OF TITLES TO BE READ IN   '
00071         PRINT *,'                                                     '
00072         PRINT *,'   TO LIST GROUPS , ENTER  G                         '
00073         PRINT *,'   TO LIST INDIVIDUALLY, ENTER  I                    '
00074         PRINT *,'   TO READ IN A FILE OF TITLE NAMES, ENTER  T        '
00075         PRINT *,'   TO QUIT, ENTER  Q                                 '
00076         PRINT *,'                                                     '
00077         PRINT *,'*************************************************'
00078         PRINT *,'                                                     '
00079         PRINT 904
00080     904 FORMAT('        ENTER CHOICE:'$)
00081         ACCEPT 902, IND
00082     902 FORMAT(A1)
00083         IF(IND.EQ.'Q') GO TO 470
00084         IF(INT.NE.'D'.OR.IND.NE.'G') GO TO 910
00085         PRINT *,'                                                     '
00086         PRINT *,'*************************************************'
00087         PRINT *,'                                                     '
00088         PRINT *,'   YOU HAVE STILL ANOTHER CHOICE:                    '
00089         PRINT *,'                                                     '
00090         PRINT *,'   TO LIST ALL 300 DIVERSIONS, ANSWER  D             '
00091         PRINT *,'   TO LIST ALL CANAL DIVERSIONS, ANSWER  C           '
00092         PRINT *,'   TO LIST ALL PUMP DIVERSIONS, ANSWER  P            '
00093         PRINT *,'   TO LIST DIVERSIONS TO LORENZO, ANSWER  L          '
00094         PRINT *,'   TO LIST HENRYS FORK DIVERSIONS, ANSWER  H         '
00095         PRINT *,'   TO LIST DIVERSIONS LORENZO TO BLACKFOOT, ANSWER  B'
00096         PRINT *,'   TO LIST DIVERSIONS BLACKFOOT TO MILNER, ANSWER M  '
00097         PRINT *,'   TO QUIT, ANSWER  Q  '
00098         PRINT *,'                                                    '
00099         PRINT *,'*************************************************'
00100         PRINT *,'                                                    '
00101         PRINT 904
00102         ACCEPT 902, INK
00103         IF(INK.EQ.'Q') GO TO 470
00104     910 IF(INT.NE.'F'.OR.IND.NE.'G') GO TO 911
00105         PRINT *,'                                                     '
00106         PRINT *,'*************************************************'
00107         PRINT *,'                                                     '
00108         PRINT *,'   YOU HAVE STILL ANOTHER CHOICE:                    '
00109         PRINT *,'                                                     '
00110         PRINT *,'   TO LIST ALL FLOW STATIONS,  ANSWER  A             '
00111         PRINT *,'                                                     '
00112         PRINT *,'   TO LIST ONLY THOSE FLOW STATIONS                  '
00113         PRINT *,'   INCLUDED THE ANNUAL WATERMASTER                   '
00114         PRINT *,'   REPORT STORED FLOW TABLES,  ANSWER  R             '
```

```
00115         PRINT *,'   TO QUIT,  ANSWER  Q                              '
00116         PRINT *,'                                                     '
00117         PRINT *,'*****************************************************'
00118         PRINT *,'                                                     '
00119         PRINT 904
00120         ACCEPT 902, INF
00121         IF(INF.EQ.'Q') GO TO 470
00122     911 PRINT *,' '
00123         PRINT *,'   ENTER BEGINNING AND ENDING CALENDAR YEAR OF THE '
00124         PRINT *,'   IRRIGATION YEAR FOR WHICH DATA IS TO BE LISTED -'
00125         PRINT *,' '
00126         PRINT *,'   FOR EXAMPLE - TO LIST DATA FOR IRRIGATION YEAR  '
00127         PRINT *,'   2000, ENTER 1999-2000   '
00128         PRINT *,'                ____-____    '
00129         PRINT 901
00130     901 FORMAT('+                '$)
00131         KY=0
00132         ACCEPT 900, NP,NP1
00133     900 FORMAT(I4,1X,I4)
00134   C
00135   C***** COMPUTE ENCOMPASSING DATES FOR DESIRED PERIOD
00136   C
00137   C     NP=NP+1900
00138   C     NP1=NP1+1900
00139         IF(JMOD(NP,4).EQ.0) KY=1
00140         ID=(NP*1000)+305+KY
00141         KY=0
00142         IF(JMOD(NP1,4).EQ.0) KY=1
00143         ID1=(NP1*1000)+304+KY
00144   C
00145   C***** ENTER INDIVIDUAL STATIONS NOT SUPPLIED FROM FILE
00146   C
00147         IF(IND.EQ.'I') THEN
00148                        OPEN(UNIT=3,STATUS='SCRATCH',RECL=80,INITIALSIZE=5)
00149       1                PRINT *,'   ENTER STATION NUMBER AND TITLE '
00150                        PRINT *,'   TO END   ENTER   99999999  '
00151                        PRINT *,'                 ________ '
00152                        PRINT 903
00153   903                  FORMAT('+              '$)
00154                        ACCEPT 905, ISTA,TTL
00155   905                  FORMAT(I8,A50)
00156                        IF(ISTA.EQ.99999999) GO TO 9
00157                        WRITE(3,906) ISTA,TTL
00158   906                  FORMAT(I8,A50)
00159                        GO TO 1
00160       9                REWIND 3
00161                        GO TO 11
00162                        END IF
00163   C
00164   C***** ENTER FILE WITH TITLE LIST IF NOT A FILE ALREADY IDENTIFIED
00165   C
00166         IF(IND.NE.'T') GO TO 8
00167         PRINT 907
00168     907 FORMAT('     ENTER FILE NAME:'$)
00169         ACCEPT 908, IC,(FILES(I),I=1,IC)
00170     908 FORMAT(Q,40A1)
00171         CALL ASSIGN(3,FILES)
```

```
00172         GO TO 11
00173   C
00174   C***** CHOOSE EXISTING FILE OF TITLES TO BE LISTED
00175   C
00176       8 IF(INT.NE.'D') GO TO 3
00177         IF(INK.EQ.'D') OPEN(UNIT=3,FILE='SNKDIV.TTL',STATUS='OLD')
00178         IF(INK.EQ.'C') OPEN(UNIT=3,FILE='SNKCAN.TTL',STATUS='OLD')
00179         IF(INK.EQ.'P') OPEN(UNIT=3,FILE='SNKPMP.TTL',STATUS='OLD')
00180         IF(INK.EQ.'L') OPEN(UNIT=3,FILE='SNKTOL.TTL',STATUS='OLD')
00181         IF(INK.EQ.'H') OPEN(UNIT=3,FILE='SNKHEN.TTL',STATUS='OLD')
00182         IF(INK.EQ.'B') OPEN(UNIT=3,FILE='SNKLTB.TTL',STATUS='OLD')
00183         IF(INK.EQ.'M') OPEN(UNIT=3,FILE='SNKBTM.TTL',STATUS='OLD')
00184         GO TO 11
00185       3 IF(INT.NE.'F') GO TO 2
00186         IF(INF.EQ.'A') OPEN(UNIT=3,FILE='SNKFLO.TTL',STATUS='OLD')
00187         IF(INF.EQ.'R') OPEN(UNIT=3,FILE='SNKFLR.TTL',STATUS='OLD')
00188         GO TO 11
00189       2 IF(INT.EQ.'R') OPEN(UNIT=3,FILE='SNKRES.TTL',STATUS='OLD')
00190   C
00191   C***** CHOOSE FIELD (PARAMETER) TO BE LISTED
00192   C
00193      11 PRINT *,' '
00194         PRINT *,'*****************************************************'
00195         PRINT *,'                                                     '
00196         PRINT *,'   YOU MUST NOW CHOOSE PARAMETER TO BE LISTED.       '
00197         PRINT *,'                                                     '
00198         IF(INT.EQ.'D') THEN
00199           PRINT *,'   TO LIST:                                        '
00200           PRINT *,'   DISCHARGE, CUBIC FEET PER SECOND, ANSWER   A    '
00201           PRINT *,'   DAILY STORAGE DIVERSION, CFS      ANSWER   B    '
00202           PRINT *,'   CUMULATIVE STORAGE DIVERTED, AF   ANSWER   C    '
00203           PRINT *,'   REMAINING STORAGE, ACRE-FEET      ANSWER   D    '
00204           PRINT *,'   CUMULATIVE TOTAL DIVERTED, AF     ANSWER   E    '
00205           PRINT *,'   NATURAL FLOW DIVERTED, CFS        ANSWER   F    '
00206           PRINT *,'   TO QUIT                           ANSWER   Q    '
00207           PRINT *,'                                                   '
00208           PRINT *,'*****************************************************'
00209           PRINT 904
00210           ACCEPT 902, INP
00211           IF(INP.EQ.'Q') GO TO 470
00212           GO TO 12
00213           END IF
00214         IF(INT.EQ.'R') THEN
00215           PRINT *,'   TO LIST:                                        '
00216           PRINT *,'   CUMULATIVE EVAPORATION, AF        ANSWER   A    '
00217           PRINT *,'   DAILY EVAPORATION, CFS            ANSWER   B    '
00218           PRINT *,'   DAILY NATURAL FLOW STORED, CFS    ANSWER   C    '
00219           PRINT *,'   CUMULATIVE TOTAL STORED AF        ANSWER   D    '
00220           PRINT *,'   CONTENTS AT HR 2400, ACRE-FEET    ANSWER   E    '
00221           PRINT *,'   TO QUIT                           ANSWER   Q    '
00222           PRINT *,'                                                   '
00223           PRINT *,'*****************************************************'
00224           PRINT 904
00225           ACCEPT 902, INP
00226           IF(INP.EQ.'Q') GO TO 470
00227           GO TO 12
00228           END IF
```

```
00229         IF(INT.EQ.'F') THEN
00230           PRINT *,'   TO LIST:                                          '
00231           PRINT *,'   NATURAL FLOW, CFS                 ANSWER   A    '
00232           PRINT *,'   DISCHARGE, CUBIC FEET PER SECOND  ANSWER   B    '
00233           PRINT *,'   STORED FLOW, CFS                  ANSWER   C    '
00234           PRINT *,'   AVERAGED REACH GAIN, CFS          ANSWER   D    '
00235           PRINT *,'   PRIORITY DATE OF LAST FILL RIGHT  ANSWER   E    '
00236           PRINT *,'   TO QUIT                           ANSWER   Q    '
00237           PRINT *,'                                      '
00238           PRINT *,'*************************************************'
00239           PRINT 904
00240           IF(INP.EQ.'Q') GO TO 470
00241           ACCEPT 902, INP
00242           END IF
00243        12 IF(INP.EQ.'A') INPCODE=1
00244           IF(INP.EQ.'B') INPCODE=2
00245           IF(INP.EQ.'C') INPCODE=3
00246           IF(INP.EQ.'D') INPCODE=4
00247           IF(INP.EQ.'E') INPCODE=5
00248           IF(INP.EQ.'F') INPCODE=6
00249           INPCODE2=INPCODE
00250           IF(INT.EQ.'R') INPCODE2=INPCODE+6
00251           IF(INT.EQ.'F') INPCODE2=INPCODE+11
00252      C
00253      C***** READ NEXT STATION TITLE
00254      C
00255        10 READ(3,301,END=450) STA_ID,NAME
00256       301 FORMAT(I8,A50)
00257      C
00258      C***** FILL ARRAY WITH MISSING VALUE INDICATORS
00259      C
00260           DO 20 I=1,12
00261           DO 20 J=1,31
00262        20 DV(I,J)=NOVALUE
00263      C
00264      C*****READ EITHER TITLES OR DATA UNTIL A MATCH IS FOUND
00265      C
00266           IF(STA_ID.EQ.STATION_NUM) GO TO 60
00267           IF(STA_ID.LT.STATION_NUM) GO TO 10
00268        25 IF(INPCODE.EQ.1)
00269          1 READ(7,701,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
00270       701 FORMAT(I8,A1,I4,I3,F10.0)
00271           IF(INPCODE.EQ.2)
00272          1 READ(7,702,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
00273       702 FORMAT(I8,A1,I4,I3,10X,F10.0)
00274           IF(INPCODE.EQ.3)
00275          1 READ(7,703,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
00276       703 FORMAT(I8,A1,I4,I3,20X,F10.0)
00277           IF(INPCODE.EQ.4)
00278          1 READ(7,704,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
00279       704 FORMAT(I8,A1,I4,I3,30X,F10.0)
00280           IF(INPCODE.EQ.5)
00281          1 READ(7,705,END=100) STATION_NUM,TYPE,YEAR,ND,VALUE
00282       705 FORMAT(I8,A1,I4,I3,40X,F10.0)
00283           IF(INPCODE.EQ.6) THEN
00284                READ(7,706,END=100) STATION_NUM,TYPE,YEAR,ND,VT,VS
00285       706      FORMAT(I8,A1,I4,I3,2F10.0)
```

```
00286              VALUE=VT-VS
00287              ENDIF
00288         IF(TYPE.NE.INT) GO TO 25
00289         IF(STA_ID.LT.STATION_NUM) THEN
00290                                   IF(TD.NE.STA_ID) THEN
00291                                                    PRINT 601,STA_ID
00292     601 FORMAT(1X,'STATION ',I8,' NOT FOUND')
00293                                                    GO TO 10
00294                                                    END IF
00295                                   GO TO 30
00296                                   END IF
00297         IF(STA_ID.EQ.STATION_NUM) GO TO 60
00298         IF(STA_ID.GT.STATION_NUM) GO TO 25
00299      30 IF(STA_ID.NE.STATION_NUM.AND.TD.EQ.STA_ID) GO TO 100
00300    C
00301    C***** CALCULATE MONTH, DAY FROM JULIAN DAY NUMBER
00302    C
00303       60 IDATE=(YEAR*1000)+ND
00304          IF(IDATE.LT.ID.OR.IDATE.GT.ID1) GO TO 25
00305          JD(2)=28
00306          IF(JMOD(YEAR,4).EQ.0) JD(2)=29
00307          LD=ND
00308          MON=0
00309       62 MON=MON+1
00310          IF(MON.GT.12) GO TO 68
00311          LD=LD-JD(MON)
00312          IF(LD.GT.0) GO TO 62
00313          LD=LD+JD(MON)
00314          DV(MON,LD)=VALUE
00315    C
00316    C***** SET VARIABLES TO PRESENT VALUES IN PREPARATION FOR
00317    C***** READING NEXT RECORD
00318    C
00319       68 CONTINUE
00320          TD=STATION_NUM
00321          P_YEAR=YEAR
00322          P_TYPE=TYPE
00323          GO TO 25
00324    C
00325    C***** ENTIRE YEAR FOR ONE STATION READ IN -
00326    C***** CALCULATE STORAGE CHANGE, CFS DAYS, ETC.
00327    C
00328      100 LEAP=0
00329          IF(JMOD(NP1,4).EQ.0) LEAP=1
00330          IF(INPCODE2.NE.11) GO TO 120
00331          DELTA_S(1)=DV(1,31)-DV(12,31)
00332          DELTA_S(2)=DV(2,28+LEAP)-DV(1,31)
00333          DELTA_S(3)=DV(3,31)-DV(2,28+LEAP)
00334          DELTA_S(4)=DV(4,30)-DV(3,31)
00335          DELTA_S(5)=DV(5,31)-DV(4,30)
00336          DELTA_S(6)=DV(6,30)-DV(5,31)
00337          DELTA_S(7)=DV(7,31)-DV(6,30)
00338          DELTA_S(8)=DV(8,31)-DV(7,31)
00339          DELTA_S(9)=DV(9,30)-DV(8,31)
00340          DELTA_S(10)=DV(10,31)-DV(9,30)
00341          DELTA_S(11)=NOVALUE
00342          DELTA_S(12)=DV(12,31)-DV(11,30)
```

```
00343         DO 119 I=2,12
00344      119 IF(ABS(DELTA_S(I)).GE.8000000.) DELTA_S(I)=NOVALUE
00345      120 DO 135 I=1,12
00346         D=31
00347         IF(I.EQ.2) D=28+LEAP
00348         IF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.9) D=30
00349         CFSD(I)=DV(I,1)
00350         D_ADJ=0
00351         IF(DV(I,1).EQ.NOVALUE) THEN
00352                                CFSD(I)=0.0
00353                                D_ADJ=1
00354                                END IF
00355         MAX(I)=DV(I,1)
00356         IF(DV(I,1).EQ.NOVALUE) MAX(I)=-1.0
00357         MIN(I)=DV(I,1)
00358         DO 130 J=2,D
00359         DZ=DV(I,J)
00360         IF(DV(I,J).EQ.NOVALUE) THEN
00361                                DZ=0.0
00362                                D_ADJ=D_ADJ+1
00363                                END IF
00364         CFSD(I)=CFSD(I)+DZ
00365         IF(DV(I,J).EQ.NOVALUE) THEN
00366                                GO TO 129
00367                                END IF
00368         IF(DV(I,J).GT.MAX(I)) MAX(I)=DV(I,J)
00369      129 IF(DV(I,J).LT.MIN(I)) MIN(I)=DV(I,J)
00370      130 CONTINUE
00371         IF(D_ADJ.GE.D) THEN
00372                        CFSD(I)=NOVALUE
00373                        DELTA_S(I)=NOVALUE
00374                        MEAN(I)=NOVALUE
00375                        GO TO 133
00376                        END IF
00377         MEAN(I)=CFSD(I)/(D-D_ADJ)
00378      133 ACFT(I)=CFSD(I)*1.9835
00379      135 CONTINUE
00380    C
00381    C***** BEGIN PRINTOUT
00382    C
00383         CALL DATE(RUNDATE)
00384         WRITE(8,801) RUNDATE
00385      801 FORMAT('1',//120X,A9)
00386         WRITE(8,802) TD,NAME
00387      802 FORMAT(//,40X,I8,6X,A50)
00388         WRITE(8,804) PRMTR(INPCODE2),NP,NP1
00389      804 FORMAT(30X,A32,', IRRIGATION YEAR'
00390        1          ' NOVEMBER',I5,' TO OCTOBER',I5)
00391      160 WRITE(8,806)
00392      806 FORMAT(//,1X,'DAY',11X,'NOV',7X,'DEC',7X,'JAN',7X,'FEB',7X,
00393        1          'MAR',7X,'APR',7X,'MAY',7X,'JUN',7X,'JUL',7X,'AUG',
00394        2          7X,'SEP',7X,'OCT')
00395         WRITE(8,807)
00396      807 FORMAT(' ')
00397         DO 180 J=1,31
00398         IF(J.EQ.6.OR.J.EQ.11.OR.J.EQ.16.OR.J.EQ.21
00399        1     .OR.J.EQ.26) THEN
```

```
00400                           WRITE(8,807)
00401                           END IF
00402          DO 170 I=1,12
00403          II=I-2
00404          IF(I.EQ.1) II=11
00405          IF(I.EQ.2) II=12
00406          IF(II.EQ.11) WRITE(8,807)
00407          IF(II.EQ.11) WRITE(8,811) CNTRL,J
00408      811               FORMAT(A1,I3,'    ')
00409          IF(DV(II,J).EQ.NOVALUE) THEN
00410                                  WRITE(8,809) CNTRL
00411      809                         FORMAT(A1,'       ---')
00412                                  GO TO 170
00413                                  END IF
00414          IF(ABS(DV(II,J)).LT.10.0) THEN
00415                                 WRITE(8,808) CNTRL,DV(II,J)
00416      808                        FORMAT(A1,F10.1)
00417                                 GO TO 170
00418                                 END IF
00419          IDV=JNINT(DV(II,J))
00420          WRITE(8,810) CNTRL,IDV
00421      810 FORMAT(A1,I10)
00422      170 CONTINUE
00423      180 CONTINUE
00424    C
00425    C***** PRINT MAX AND MIN, CHANGE IN STORAGE FOR RESERVOIR CONTENTS
00426    C
00427          IF(INPCODE2.EQ.11) THEN
00428                           WRITE(8,812)
00429      812                 FORMAT('0')
00430                          WRITE(8,813) CNTRL
00431      813                 FORMAT(A1,2X,'MAX  ')
00432                          DO 190 I=1,12
00433                          II=I-2
00434                          IF(I.EQ.1) II=11
00435                          IF(I.EQ.2) II=12
00436                          IF(MAX(II).EQ.-1.0) THEN
00437                                 WRITE(8,860) CNTRL
00438      860                        FORMAT(A1,'          ')
00439                                 GO TO 190
00440                                 END IF
00441                          IDV=JNINT(MAX(II))
00442                          WRITE(8,810) CNTRL,IDV
00443      190                 CONTINUE
00444                          WRITE(8,807)
00445                          WRITE(8,814) CNTRL
00446      814                 FORMAT(A1,2X,'MIN  ')
00447                          DO 200 I=1,12
00448                          II=I-2
00449                          IF(I.EQ.1) II=11
00450                          IF(I.EQ.2) II=12
00451                          IF(MIN(II).EQ.NOVALUE) THEN
00452                                                 WRITE(8,860) CNTRL
00453                                                 GO TO 200
00454                                                 END IF
00455                          IDV=JNINT(MIN(II))
00456                          WRITE(8,810) CNTRL,IDV
```

```
00457     200                   CONTINUE
00458                           WRITE(8,807)
00459                           WRITE(8,815) CNTRL
00460     815                   FORMAT(A1,1X,'CHNG            ')
00461                           DO 210 I=2,12
00462                           II=I-2
00463                           IF(I.EQ.2) II=12
00464                           IDV=JNINT(DELTA_S(II))
00465                           IF(DELTA_S(II).EQ.NOVALUE)THEN
00466                                           WRITE(8,860) CNTRL
00467                                           GO TO 210
00468                                           END IF
00469                           WRITE(8,810) CNTRL,IDV
00470     210                   CONTINUE
00471                           GO TO 10
00472                           END IF
00473   C
00474   C***** FOR DATA IN CFS, PRINT MONTH END SUMMARY, OTHERWISE GO TO NEXT
00475   C***** STATION
00476   C
00477         IF(INPCODE2.GT.2.AND.INPCODE2.LT.6) GO TO 10
00478         IF(INPCODE2.EQ.7) GO TO 10
00479         IF(INPCODE2.EQ.10.OR.INPCODE2.EQ.11) GO TO 10
00480         IF(INPCODE2.EQ.16) GO TO 10
00481         WRITE(8,812)
00482         WRITE(8,816) CNTRL
00483     816 FORMAT(A1,'TOTAL  ')
00484         DO 220 I=1,12
00485         II=I-2
00486         IF(I.EQ.1) II=11
00487         IF(I.EQ.2) II=12
00488         IF(CFSD(II).GE.NOVALUE) THEN
00489                                 WRITE(8,860) CNTRL
00490                                 GO TO 220
00491                                 END IF
00492         IDV=JNINT(CFSD(II))
00493         WRITE(8,810) CNTRL,IDV
00494     220 CONTINUE
00495         WRITE(8,807)
00496         WRITE(8,818) CNTRL
00497     818 FORMAT(A1,' MEAN  ')
00498         DO 230 I=1,12
00499         II=I-2
00500         IF(I.EQ.1) II=11
00501         IF(I.EQ.2) II=12
00502         IF(MEAN(II).GE.NOVALUE) THEN
00503                                 WRITE(8,860) CNTRL
00504                                 GO TO 230
00505                                 END IF
00506         IF(ABS(MEAN(II)).LT.10.0) THEN
00507                                 WRITE(8,808) CNTRL,MEAN(II)
00508                                 GO TO 230
00509                                 END IF
00510         IDV=JNINT(MEAN(II))
00511         WRITE(8,810) CNTRL,IDV
00512     230 CONTINUE
00513         WRITE(8,807)
```

```
00514         WRITE(8,819) CNTRL
00515      819 FORMAT(A1,'  MAX  ')
00516         DO 240 I=1,12
00517         II=I-2
00518         IF(I.EQ.1) II=11
00519         IF(I.EQ.2) II=12
00520         IF(MAX(II).EQ.-1.0) THEN
00521                             WRITE(8,860) CNTRL
00522                             GO TO 240
00523                             END IF
00524         IF(ABS(MAX(II)).LT.10.0) THEN
00525                             WRITE(8,808) CNTRL,MAX(II)
00526                             GO TO 240
00527                             END IF
00528         IDV=JNINT(MAX(II))
00529         WRITE(8,810) CNTRL,IDV
00530      240 CONTINUE
00531         WRITE(8,807)
00532         WRITE(8,820) CNTRL
00533      820 FORMAT(A1,'  MIN  ')
00534         DO 250 I=1,12
00535         II=I-2
00536         IF(I.EQ.1) II=11
00537         IF(I.EQ.2) II=12
00538         IF(MIN(II).GE.NOVALUE) THEN
00539                                WRITE(8,860) CNTRL
00540                                GO TO 250
00541                                END IF
00542         IF(ABS(MIN(II)).LT.10.0) THEN
00543                             WRITE(8,808) CNTRL,MIN(II)
00544                             GO TO 250
00545                             END IF
00546         IDV=JNINT(MIN(II))
00547         WRITE(8,810) CNTRL,IDV
00548      250 CONTINUE
00549    C       DO 260 I=1,12
00550    C       IFT(I)=JNINT(ACFT(I)/100.)
00551    C       FT(I)=IFT(I)*100
00552    C   260 CONTINUE
00553         WRITE(8,807)
00554         WRITE(8,821) CNTRL
00555      821 FORMAT(A1,'AC-FT  ')
00556         DO 270 I=1,12
00557         II=I-2
00558         IF(I.EQ.1) II=11
00559         IF(I.EQ.2) II=12
00560         IF(ACFT(II).GE.19834900.) THEN
00561                                   WRITE(8,860) CNTRL
00562                                   GO TO 270
00563                                   END IF
00564    C        IF(ACFT(II).LE.999.) THEN
00565    C                               IFT(II)=JNINT(ACFT(II))
00566    C                              WRITE(8,810) CNTRL,IFT(II)
00567    C                               GO TO 270
00568    C                               END IF
00569         IFT(II)=JNINT(ACFT(II))
00570         WRITE(8,810) CNTRL,IFT(II)
```

```
00571      270 CONTINUE
00572          YT=0.0
00573          DO 280 I=1,12
00574          IF(CFSD(I).EQ.NOVALUE) CFSD(I)=0.0
00575      280 YT=YT + CFSD(I)
00576          IYT=JNINT(YT)
00577          IYM=JNINT(YT/(365+LEAP))
00578    C      NT=JNINT(YT*1.9835/100.)
00579    C      IYA=NT*100
00580    C      IF(IYA.LT.9999) IYA=1.9835*YT
00581          IYA=1.9835*YT
00582          WRITE(8,807)
00583          WRITE(8,831) NP1,IYT,IYM,IYA
00584      831 FORMAT(11X,'IRRIGATION YEAR',I5,4X,'TOTAL',I10,4X,'MEAN',
00585         1         I6,4X,'AC-FT',I8)
00586          GO TO 10      ! RETURN TO START ANOTHER STATION
00587      450 REWIND 7
00588          CLOSE (UNIT=3)
00589          PRINT *,'  '
00590          PRINT *,'  '
00591          PRINT *,'  REQUESTED OF STATIONS COMPLETE - RETURN TO START'
00592          PRINT *,'  '
00593          PRINT *,'  '
00594          GO TO 500
00595      470 PRINT *,'                                        '
00596          PRINT *,'  OUTPUT IS  THE LATEST VERSION OF SNKALCLST.OUT'
00597          STOP
00598          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	8183	PIC CON REL LCL SHR EXE RD NOWRT QUAD
1	$PDATA	3865	PIC CON REL LCL SHR NOEXE RD NOWRT QUAD
2	$LOCAL	4020	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	Total Space Allocated	16068	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKALCLST$MAIN	

VARIABLES

Address	Type	Name	Attributes	References
**	CHAR	BLANK		18 38=
**	L*1	CNTRL		12 36= 407 410 415 420 430 437 442 445 452 456 459 466 469 482 489 493 496 503 507 511 514 521 525 529 532 539 543 547 554 561 570
**	R*4	D		346= 347= 348= 358 371 377
**	R*4	DZ		359= 361= 364
**	I*4	D_ADJ		13 350= 353= 362(2)= 371 377
**	I*4	I		169(2)= 260= 262 343= 344(2) 345= 347 348(3) 349(2) 351 352 355(2) 356(2) 357(2) 359 360 364(2) 365 368(4) 369(4) 372 373 374 377(2) 378(2) 402= 403 404 405 432= 433 434 435 447= 448 449 450 461= 462 463 484= 485 486 487 498= 499 500 501 516= 517 518 519 534= 535 536 537 556= 557 558 559 573= 574(2) 575
2-00000A10	I*4	IC		169(2)=
**	I*4	ID		140= 304
**	I*4	ID1		143= 304
**	I*4	IDATE		303= 304(2)
**	I*4	IDV		419= 420 441= 442 455= 456 464= 469 492= 493 510= 511 528= 529 546= 547
**	I*4	II		403= 404= 405= 406 407 409 414 415 419 433= 434= 435= 436 441 448= 449= 450= 451 455 462= 463= 464 465 485= 486= 487= 488 492 499= 500=

		501= 502 506 507 510 517=	
		518= 519= 520 524 525 528	
		535= 536= 537= 538 542 543	
		546 557= 558= 559= 560 569(2)	
		570	
2-000009F8	I*4 IND	81= 83 84 104 147 166	
2-00000A00	I*4 INF	120= 121 186 187	
2-000009FC	I*4 INK	102= 103 177 178 179 180	
		181 182 183	
2-00000A18	I*4 INP	210= 211 225= 226 240 241=	
		243 244 245 246 247 248	
2-00000A1C	I*4 INPCODE	243= 244= 245= 246= 247= 248=	
		249 250 251 268 271 274	
		277 280 283	
**	I*4 INPCODE2	249= 250= 251= 330 388 427	
		477(2) 478 479(2) 480	
2-000009E6	CHAR INT	18 54= 55 84 104 176	
		185 189 198 214 229 250	
		251 288	
2-00000A0C	I*4 ISTA	154= 156 157	
**	I*4 IYA	581= 583	
**	I*4 IYM	577= 583	
**	I*4 IYT	576= 583	
**	I*4 J	261= 262 358= 359 36D 365	
		368(2) 369(2) 397= 398(5) 407 409	
		414 415 419	
**	I*4 KY	131= 139= 140 141= 142= 143	
**	I*4 LD	307= 311(2)= 312 313(2)= 314	
**	I*4 LEAP	328= 329= 332 333 347 577	
**	I*4 MON	308= 309(2)= 310 311 313 314	
2-00000978	CHAR NAME	17 255= 386	
2-00000A20	I*4 ND	268= 271= 274= 277= 280= 284=	
		303 307	
**	R*4 NOVALUE	14 37= 262 341 344 351	
		356 360 365 372 373 374	
		409 451 465 488 502 538	
		574	
2-00000A04	I*4 NP	132= 139 140 388	
2-00000A08	I*4 NP1	132= 142 143 329 388 583	
**	CHAR P_TYPE	17 322=	
**	I*4 P_YEAR	13 321=	
2-000009AB	CHAR RUNDATE	17 383A 384	
2-000009F0	I*4 STATION_NUM	13 266 267 268= 271= 274=	
		277= 280= 284= 289 297 298	
		299 320	
2-000009EC	I*4 STA_ID	13 255= 266 267 289 290	
		291 297 298 299(2)	
2-000009E8	I*4 TD	13 290 299 320= 386	
**	CHAR TITLE	17	
2-0DDD09B4	CHAR TTL	18 154= 157	
2-000009AA	CHAR TYPE	17 268= 271= 274= 277= 280=	

			284=	288	322				
2-00000A24	R*4	VALUE	268=	271=	274=	277=	280=	286=	
			314						
2-00000A2C	R*4	VS	284=	286					
2-00000A28	R*4	VT	284=	286					
2-000009F4	I*4	YEAR	13	268=	271=	274=	277=	280=	
			284=	303	306	321			
**	R*4	YT	572=	575(2)=	576	577	581		
**	CHAR	ZLCH	17						

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-000006F0	R*4	ACFT		48	(12)	16	378=	560	569	
2-00000690	R*4	CFSD		48	(12)	15	349=	352=	364(2)=	372=
						377	378	488	492	574(2)=
						575				
2-000006C0	R*4	DELTA_S		48	(12)	16	331=	332=	333=	334=
						335=	336=	337=	338=	339=
						340=	341=	342=	344(2)=	373=
						464	465			
2-000000C0	R*4	DV		1488	(12, 31)	15	262=	314=	331(2)	332(2)
						333(2)	334(2)	335(2)	336(2)	337(2)
						338(2)	339(2)	340(2)	342(2)	349
						351	355	356	357	359
						360	365	368(2)	369(2)	409
						414	415	419		
2-00000750	L*1	FILES		40	(40)	11	169=	171A		
2-00000720	I*4	IFT		48	(12)	16	569=	570		
2-00000090	I*4	JD		48	(12)	15	19D	305=	306=	311
						313				
2-00000030	R*4	MAX		48	(12)	14	15	355=	356=	368(2)=
						436	441	520	524	525
						528				
2-00000000	R*4	MEAN		48	(12)	14	16	374=	377=	502
						506	507	510		
2-00000060	R*4	MIN		48	(12)	14	15	357=	369(2)=	451
						455	538	542	543	546
2-00000778	CHAR	PRMTR		512	(16)	16	18	20D	388	

LABELS

Address	Label	References					
0-00000A0C	1	149#	159				
0-00000C18	2	185	189#				
0-00000BE8	3	176	185#				
0-00000B54	8	166	176#				
0-00000ADC	9	156	160#				
0-00001124	10	255#	267	293	471	477	478

```
  0-00000C28  11
  0-000010BB  12
      **      20
  0-000011E3  25

  0-00001437  30
  0-00001184  60
  0-000011BC  62
  0-000011DE  68
  0-00001448  100


      **      119
  0-000014FF  120
  0-000015A0  129
      **      130
  0-000015E1  133

      **      135
      **      160
  0-000017EF  170
      **      180
  0-000018AF  190

  0-0000195F  200
  0-00001A05  210
  0-00001AE7  220
  0-00001BD7  230
  0-00001CCB  240

  0-00001DBF  250
  0-00001E79  270
      **      280
  1-00000070  301'
  0-00001F14  450

  0-00001FB8  470

  0-0000002C  500
  1-000000C7  601'
  1-00000075  701'
  1-00000081  702'

  1-0000008F  703'
  1-0000009D  704'
  1-000000AB  705'
  1-000000B9  706'
  1-000000E2  801'

  1-000000EC  802'
  1-000000F7  804'
  1-0000012B  806'
  1-00000189  807'

  1-0000D1A7  8D8'
```

```
  479       480       586
  161       172       184       188       193#
  212       227       243#
  260       261       262#
  268#      288       298       304       323

  295       299#
  266       297       303#
  309#      312
  310       319#
  268       271       274       277       280       284
  299       328#

  343       344#
  330       345#
  366       369#
  358       370#
  375       378#

  345       379#
  391#
  402       412       417       422#
  397       423#
  432       439       443#

  447       453       457#
  461       467       470#
  484       490       494#
  498       504       508       512#
  516       522       526       530#

  534       540       544       548#
  556       562       571#
  573       575#
  255       256#
  255       587#

   55        83       103       121       211       226
  240       595#
   41#      594
  291       292#
  268       270#
  271       273#

  274       276#
  277       279#
  280       282#
  284       285#
  384       385#

  386       387#
  388       389#
  391       392#
  395       396#      400       406       444       458
  495       513       531       553       582
  415       416#      507       525       543
```

1-00000198	809'	410	411#				
1-000001AD	810'	420	421#	442	456	469	493
		511	529	547	570		
1-0000018D	811'	407	408#				
1-000001B2	812'	428	429#	481			
1-000001B6	813'	430	431#				
1-000001D1	814'	445	446#				
1-000001DD	815'	459	460#				
1-000001F4	816'	482	483#				
1-00000200	818'	496	497#				
1-0000020C	819'	514	515#				
1-00000218	820'	532	533#				
1-00000224	821'	554	555#				
1-00000230	831'	583	584#				
1-000001C2	860'	437	438#	452	466	489	503
		521	539	561			
1-00000030	900'	132	133#				
1-0000001C	901'	129	130#				
1-00000019	902'	54	81	82#	102	120	210
		225	241				
1-00000037	9D3'	152	153#				
1-00000000	904'	53	79	80#	101	119	209
		224	239				
1-00000047	905'	154	155#				
1-0000004C	906'	157	158#				
1-00000051	907'	167	168#				
1-0000006A	908'	169	170#				
0-000006AA	910	84	104#				
0-0000088D	911	104	122#				

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References					
R*4	ABS	344	414	506	524	542	
	ASSIGN	171					
	DATE	383					
I*4	JMOD	139	142	306	329		
I*4	JNINT	419	441	455	464	492	510
		528	546	569	576	577	

```
+-----------------------------------------------+
|           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 SNKALCLST

  /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.SNAKE]SNKALCLST.LIS;1
  /OBJECT=HYDRO:[BSUTTER.SNAKE]SNKALCLST.OBJ;1
```

COMPILER: DEC Fortran V6.3-141

COMPILATION STATISTICS

```
  Run Time:           1.83 seconds
  Elapsed Time:       2.78 seconds
  Page Faults:        5402
  Dynamic Memory:     2176 pages
```

SNKHSTZERO.LIS

```
0001    C
0002    C***** THIS PROGRAM TAKE DATA RETRIEVED FROM HYDROMET AND ELIMINATES
0003    C***** ANY DIVERSION RECORDS WITH 0.00 OR ALL BLANKS IN THE FIRST
0004    C***** FIELD IF THE DATA ARE BEFORE MAY 1 OR AFTER OCTOBER 31.
0005    C*****               RJS - MAY 15, 1989.
0006    C
0007          REAL*8 ADIFF
0008          CHARACTER*28 VALU
0009          CHARACTER*7 VAL(4)
0010          EQUIVALENCE (VALU,VAL)
0011          DATA IDAY1/121/,IDAY2/304/
0012          OPEN(UNIT=1,FILE='INFILE',TYPE='OLD')
0013          OPEN(UNIT=2,FILE='OUTFILE',TYPE='NEW',CARRIAGECONTROL='LIST')
0014          IDROP=0
0015    C
0016    C***** READ A RECORD FROM OLD FILE
0017    C
0018        1 READ(1,10,END=5)  ID,S,IYR,IDAY,VALU
0019       10 FORMAT(I8,A1,I4,I3,A28)
0020          IF(S.NE.'D') GO TO 2
0021    C
0022    C***** CHECK FOR LEAP YEAR CORRECTION, THEN CHECK IF DAY IS BEFORE
0023    C***** MAY 1, OR AFTER OCT 31.  IF IT IS, THEN ZERO DIVERSION RECORDS
0024    C***** CAN BE ELIMINATED.
0025    C
0026          ILEAP=0
0027          ADIFF=MOD(IYR,4)
0028          IF(ADIFF.EQ.0) ILEAP=1
0029          ID1=IDAY1+ILEAP
0030          ID2=IDAY2+ILEAP
0031          IF(IDAY.GE.ID1.AND.IDAY.LE.ID2) GO TO 2
0032    C
0033    C***** CHECK IF DIVERSION DISCHARGE IS EQUAL TO 0.00, IF SO, DO
0034    C***** NOT WRITE RECORD TO NEW FILE.
0035    C
0036          IF(VAL(1).EQ.'   0.00'.OR.VAL(1).EQ.'       ') THEN
0037                                                          IDROP=IDROP+1
0038                                                          GO TO 1
0039                                                          ENDIF
0040    C
0041    C***** WRITE DATA TO NEW FILE, THEN RETURN TO GET ANOTHER RECORD.
0042    C
0043        2 WRITE(2,12) ID,S,IYR,IDAY,VALU
0044       12 FORMAT(I8,A1,I4,I3,A28)
0045          GO TO 1
0046        5 TYPE *, ' '
0047          TYPE *, ' DIVERSION DATA FROM HYDROMET HAS BEEN CHECKED FOR '
0048          TYPE *, ' ZERO OR MISSING DATA IN NON-IRRIGATION SEASON -   '
0049          TYPE 3, IDROP
0050        3 FORMAT(/1X,' NUMBER OF RECORDS DROPPED = ',I8)
0051          TYPE *,  ' '
0052          STOP
0053          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	415	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	191	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	152	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	758	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKHSTZERO$MAIN	

VARIABLES

Address	Type	Name	Attributes	References			
**	R*8	ADIFF		7	27=	28	
2-0000001C	I*4	ID		18=	43		
**	I*4	ID1		29=	31		
**	I*4	ID2		30=	31		
2-00000028	I*4	IDAY		18=	31(2)	43	
**	I*4	IDAY1		11D	29		
**	I*4	IDAY2		11D	30		
**	I*4	IDROP		14=	37(2)=	49	
**	I*4	ILEAP		26=	28=	29	30
2-00000024	I*4	IYR		18=	27	43	
2-00000020	R*4	S		18=	20	43	
2-00000000	CHAR	VALU	EQUIV	8	10	18=	43

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
2-00000000	CHAR	VAL	EQUIV	28	(4)	9	10	36(2)

LABELS

Address	Label	References		
0-0000001C	1	18#	38	45
0-000000C0	2	20	31	43#
1-0000008C	3'	49	50#	
0-0000010C	5	18	46#	
1-00000076	10'	18	19#	
1-00000081	12'	43	44#	

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	FOR$OPEN	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 SNKHSTZERO

/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.SNAKE]SNKHSTZERO.LIS;7
/OBJECT=HYDRO:[SUTTER.SNAKE]SNKHSTZERO.OBJ;1
```

COMPILATION STATISTICS

```
Run Time:           2.81 seconds
Elapsed Time:       4.44 seconds
Page Faults:        659
Dynamic Memory:     357 pages
```

```
0001    C*************************************************************************
0002    C
0003    C       PROGRAM TO CHANGE VALUES IN DATA SET SNKRRT.IND.
0004    C        THESE VALUES ARE THE RESERVOIR RIGHT STORAGE VOLUMES.
0005    C                       RJS - MAY 1989.
0006    C
0007    C*************************************************************************
0008
0009          DIMENSION RESRTNM(16),MR(16),RST(16)
0010          CHARACTER*17 RESRTNM
0011          CALL ASSIGN (1,'SNKRRT.IND')
0012          OPEN (UNIT=2,NAME='SNKRRT.TMP',TYPE='NEW',CARRIAGECONTROL='LIST')
0013          READ(1,2,END=6) (RESRTNM(L),MR(L),RST(L),L=1,16)
0014        2 FORMAT(A17,I5,F10.0)
0015        5 PRINT *,' '
0016          PRINT *,' ENTER ANY CHANGES YOU WISH FOR THESE RESERVOIR RIGHT VOL
0017         1UMES  '
0018          PRINT *,' THESE VALUES ARE THE ACTUAL WATER RIGHTS TO BE USED IN T
0019         1HE '
0020          PRINT *,' ACCOUNTING AND NORMALLY DO NOT CHANGE FROM YEAR TO YEAR'
0021          PRINT *,' '
0022          DO 101 I=1,16
0023          TYPE 11, RESRTNM(I),RST(I)
0024       11 FORMAT(' ',A17,' =',F10.1,' NEW VALUE? Y/N '$)
0025          ACCEPT 102, IANS
0026      102 FORMAT(A1)
0027          IF(IANS.NE.'Y') GO TO 101
0028          PRINT 103
0029      103 FORMAT(' ENTER NEW VALUE   ='$)
0030          ACCEPT 4, RST(I)
0031        4 FORMAT(F10.0)
0032      101 CONTINUE
0033          PRINT 8
0034        8 FORMAT(/' DO YOU WISH TO REVIEW/CHECK RESERVOIR RIGHTS? Y/N '$)
0035          ACCEPT 3, IANS
0036        3 FORMAT(A1)
0037          IF(IANS.EQ.'Y') GO TO 5
0038          WRITE(2,1) (RESRTNM(L),MR(L),RST(L),L=1,16)
0039        1 FORMAT(A17,I5,F10.1)
0040          PRINT *,' '
0041          PRINT *,'  FILE SNKRRT.IND HAS BEEN UPDATED WITH NEW VALUES '
0042          PRINT *,' '
0043          STOP
0044        6 PRINT *,' '
0045          PRINT *,'  ******** END OF FILE REACHED ON SNKRRT.IND ******'
0046          PRINT *,'              FILE SNKRRT.IND NOT UPDATED '
0047          PRINT *,' '
0048          STOP
0049          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	779	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	487	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	584	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1850	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SNKCHGRR$MAIN	

VARIABLES

Address	Type	Name	Attributes	References				
**	I*4	I		22=	23(2)	30		
2-00000194	I*4	IANS		25=	27	35=	37	
2-00000190	I*4	L		13(4)=	38(4)=			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000000	I*4	MR		64	(16)	9	13=	38		
2-00000080	CHAR	RESRTNM		272	(16)	9	10	13=	23	38
2-00000040	R*4	RST		64	(16)	9	13=	23	30=	38

LABELS

Address	Label	References		
1-000001DF	1'	38	39#	
1-0000015D	2'	13	14#	
1-000001DC	3'	35	36#	
1-000001A0	4'	30	31#	
0-00000070	5	15#	37	
0-00000294	6	13	44#	
1-000001A4	8'	33	34#	
1-00000165	11'	23	24#	
0-0000019C	101	22	27	32#
1-00000185	102'	25	26#	
1-00000188	103'	28	29#	