

Capítulo 31- Programa Wadiso

31.1 Introdução

O objetivo é mostrar um programa completo feito em Fortran por um dos maiores engenheiros hidráulicos do mundo dr. Thomas Walski.

O programa foi feito para o corpo de engenheiro do exercito dos Estados Unidos e distribuído gratuitamente em todo o mundo.

Thomas M. Walski juntamente com Johannes Gessler e Joh W. Sjostrom fizeram o livro *Water Distribution Systems: simulation and Sizing* de como usar o programa Wadiso.

Usamos durante alguns anos o programa que é fácil de ser usado (amigável) mas possui o inconveniente das unidades inglesas.

No programa temos os cálculos de matrizes esparsas.

O Wadiso deu origem a outros programas como o Epanet, Epanet 2 e o WaterCad e devido a isto é bom saber um pouco da teoria que explicamos no método dos nós no Capítulo 1 deste curso.

APENDICE C- PROGRAMA WADISO

```

C*****
C PROGRAM WADISO DEVELOPED BY ARMY CORPS OF ENGINEERS - WATERWAYS *
C EXPERIMENT STATION - VICKSBURG, MS 1983 - 1987. WADISO IS A *
C PUBLIC DOMAIN PROGRAM DEVELOPED UNDER THE WATER SUPPLY AND *
C CONSERVATION RESEARCH PROGRAM BY: *
C *
C *
C DR. JOHANNES GESSLER *
C DEPARTMENT OF CIVIL ENGINEERING *
C COLORADO STATE UNIVERSITY *
C *
C MR. JOHN SJOSTROM *
C US PEACE CORPS *
C *
C DR. THOMAS WALSKI *
C WYOMING VALLEY SANITARY AUTHORITY *
C *
C THIS VERSION WAS PREPARED SPECIFICALLY FOR LEWIS PUBLISHERS *
C P.O. DRAWER 519 CHELSEA, MI *
C*****
PROGRAM WADISO
CHARACTER JOB*60,ST*90
INTEGER PNL,O
REAL*8 A,S,G,HE
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
COMMON /PRINT/ IPM,IPP,IPE
COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
COMMON /JOB/ JOB
C
C ASSIGNING OF COST DATA WITH BLOCK DATA ASSIGN
C
EXTERNAL ASSIGN
    
```

```
PRINT*, ' WADISO - WATER DISTRIBUTION SYSTEM OPTIMIZATION'
PRINT*, ' VERSION: JANUARY 1, 1990'
PRINT*
PRINT*
WRITE(6,20) PNL,LNL
WRITE(6,30) PNL,LNL
20  FORMAT(' MAX NODE NO = ',I3,' NUMBER OF NODES = ',I3)
30  FORMAT(' MAX PIPE NO = ',I3,' NUMBER OF PIPES = ',I3)
C
C   SETTING PRINT CONTROLS
C   IPM, IPP, AND IPE ARE PRINTING FLAGS.
C   IPM=1 PRINTS MENU
C   IPP=1 PRINTS PROMPTS
C   IPE=1 ECHOES RESPONSES
C
      IPM=1
      IPP=1
      IPE=0
C
C   SETTING DEFAULT VALUES
C
      C3=448.831
      ICL=24
      PRAC=2.
      FLAC=10./C3
      HWMA=100
C
9    OPEN (2,FILE='INPUT')
10   IF (IPM.EQ.0) GOTO 11
C
C   MAIN PROGRAM MENU
C
      PRINT*
      PRINT*, ' PROGRAM CONTROL : '
      PRINT*
      PRINT*, '          SIMULATION          : ENTER 1 PRESS RETURN'
      PRINT*, '          OPTIMIZATION           :          2'
      PRINT*, '          COST DATA              :          3'
      PRINT*, '          TIME SIMULATION         :          4'
      PRINT*, '          TERMINATE PROGRAM      :          9'
      PRINT*
11   READ (*,2,END=2222) ST
      IF (IPE.EQ.1) PRINT*,ST
2    FORMAT (A60)
      IF (ST(1:1).EQ.'1') CALL SIMULA
      IF (ST(1:1).EQ.'2') CALL OPTIMI
      IF (ST(1:1).EQ.'3') CALL COSTDA
      IF (ST(1:1).EQ.'4') CALL TIMENU
      IF (ST(1:1).EQ.'9') CALL TERMIN
12   IF (ST(1:2).EQ.'NM') IPM=0
      IF (ST(1:2).EQ.'NP') IPP=0
      IF (ST(1:2).EQ.'EC') IPE=1
      IF (ST(1:3).EQ.'NOM') THEN
          IPM=0
          IPP=0
          IPE=1
      END IF
      IF (ST(1:2).EQ.'ME') IPM=1
      IF (ST(1:2).EQ.'PR') IPP=1
      IF (ST(1:2).EQ.'NE') IPE=0
```

```
IF (ST(1:3).EQ.'MEN') THEN
  IPM=1
  IPP=1
  IPE=0
END IF
GOTO 10
2222 CALL BLANK
GOTO 10
END

C
C   ASSIGNING OF COST DATA WITH BLOCK DATA SUBPROGRAM
C
  BLOCK DATA ASSIGN
  COMMON/COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
  DATA
SIZ/2,3,4,6,8,10,12,14,16,18,20,24,30,36,42,48,54,60,66,72,78
  +,84,96,108,120/
  DATA
COST/6.29,8.57,10.8,15.1,19.3,28.9,40.5,52.1,59.4,68.6,80.1,1
+06,147,192,242,295,331,396,477,554,642,734,941,1170,1420,3*30,14.5
  +,15.7,16.8,17.7,18.5,19.2,20.0,20.5,21.6,23.1,24.3,25.4,26.4,
  +259*0/
  DATA KS,KC,ENCO,NY,XI/25,2,.1,10,.1/
  END

C
C *****SUBROUTINE SIMULA*****
C *   SUBROUTINE INIALIZES STRINGS AND ARRAYS, PROMPTS USER WITH
C *
C *   SELECTION OF PROGRAM OPTIONS OF NEW JOB OR RETRIEVE DATA,
C *
C *   AND DISPLAYS THE MAIN MENU FOR ALL SIMULATION SUBROUTINES
C *
C *   AND TRANSFERS PROGRAM CONTROL TO SUBROUTINES
C *
C *****
C
  SUBROUTINE SIMULA
  CHARACTER JOB*60,ST*90
  INTEGER PNL,O
  REAL*8 A,S,G,HE
  PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
  COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
  COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
  COMMON /TOPOL/
  IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
  1L)
  COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
  COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
  COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
  COMMON /PRINT/ IPM,IPP,IPE
  COMMON /JOB/ JOB
  CHARACTER KTYPE*4

C
C   *****INITIALIZATION OF VARIABLES*****
C
  IF (O(1).GT.0) GOTO 1990
  O(4)=0
  DO 9 I=1,PNL
```

```

      IBE(I)=0
      IEN(I)=0
      IPI(I)=0
9     INO(I)=0
      KTYPE='JOB'
      O(1)=0
      O(2)=0
      O(4)=0

C
C *****PROGRAM START UP 28-3B*****
C *** USER IS PROMPTED FOR:
C ***     1. NEW JOB - (TRANSFER IS MADE TO SUBROUTINE INPDAT)
C ***     2. RETRIEVE DATA - (TRANSFER IS MADE TO SUBROUTINE
C ***           RETDAT)
C *****
C
5     IF (IPM.EQ.0) GOTO 6
      PRINT*, ' SIMULATION ROUTINE'
      PRINT*
      PRINT *, ' SELECT PROGRAM OPTION      :'
      PRINT*
      PRINT *, '           TO ENTER NEW JOB : ENTER  1  PRESS RETURN'
      PRINT *, '           TO RETRIEVE DATA :           2'
      PRINT*
6     READ (*,2,END=11) ST
      IF (IPE.EQ.1) PRINT*,ST
      IF (ST(1:1).EQ.'1') THEN
        CALL INPDAT(KTYPE)
        GOTO 1990
      END IF
      IF (ST(1:1).EQ.'2') THEN
        CALL RETDAT(IFLG,ST)
        IF (IFLG.EQ.1) THEN
          KTYPE = 'PIPE'
          CALL INPDAT(KTYPE)
        END IF
        GOTO 1990
      END IF
      PRINT *, 'INVALID ENTRY - TRY AGAIN.'
      GOTO 5
11    CALL BLANK
      GOTO 5
1990  IF (IPM.EQ.0) GOTO 1993
      IFLG = 0
      PRINT 1991

C
C *****OPTION MENU*****
C * OPTION MENU IS DISPLAYED AND USER PROMPTED FOR SELECTION.
C * BASED ON SELECTION, PROGRAM CONTROL IS TRANSFERED TO THE
C * APPROPRIATE SUBROUTINE
C *****
C
1991  FORMAT(/,/,' SELECT PROGRAM OPTION :',/)
      PRINT *, '           BALANCE           : ENTER 0 OR 0C PRESS RETURN'
      PRINT *, '           MODIFY SYSTEM      :           1'
      PRINT *, '           PRINT INPUT                   :           2    2C'
      PRINT *, '           STORE DATA                   :           3'
      PRINT *, '           RETRIEVE DATA                :           4'
      IF (O(9).GT.0) PRINT *, '           PRINT OUTPUT                :           6
6C'

```

```
      PRINT *, '          PROGRAM CONTROL      :          8'
      PRINT *, '          TERMINATE PROGRAM    :          9'
      PRINT*
C
C      READING OF INPUT (NUMERIC VALUE, OPTIONALLY FOLLOWED BY A
C      C FOR OPTIONS 0, 2, OR 6 IN ORDER TO PRINT DATA CONTINUOUSLY)
C
1993 READ (*,2,END=1992) ST
      IF(IPE.EQ.1) PRINT*,ST
      IF (ST(1:1).EQ.'0') CALL SIMBAL(ST)
      IF (ST(1:1).EQ.'1') THEN
          CALL MODIFY
C
C.....CHECKING IF MAXIMUM NODE OR NUMBER EXCEEDS LIMIT SET BY PNL
C
      IF (O(1).NE.0.AND.O(2).NE.0) THEN
          IF (INO(O(2)).GT.PNL.AND.INO(O(2)).GE.IPI(O(1))) THEN
              WRITE(6,23) ' NODES', ' NODE',INO(O(2))
23      FORMAT(' TOO MANY',A,' - INCREASE PNL IN SOURCE CODE TO',
>           /' MAXIMUM',A,' NUMBER',I4)
              RETURN
          ELSEIF (IPI(O(1)).GT.PNL) THEN
              WRITE(6,23) ' LINKS', ' LINK',IPI(O(1))
              RETURN
          ENDIF
          KTYPE='PIPE'
          CALL INPDAT(KTYPE)
      END IF
      END IF
      IF (ST(1:1).EQ.'2') CALL PRNINP(ST)
      IF (ST(1:1).EQ.'3') CALL STODAT
      IF (ST(1:1).EQ.'4') CALL RETDAT(IFLG,ST)
      IF (ST(1:1).EQ.'6'.AND.O(9).GT.0) CALL PRNOUT(ST)
      IF (ST(1:1).EQ.'9') CALL TERMIN
      IF (ST(1:1).EQ.'8') THEN
          O(5)=1
          RETURN
      END IF
C
C      IFLG IS SET TO 1 IN SIMRET INDICATING RETURN IS MADE TO
C      THE INPUT PROMPT
C
      IF (IFLG.EQ.1) THEN
          IFLG=0
          CALL INPDAT(KTYPE)
      END IF
      GOTO 1990
1992 CALL BLANK
      GOTO 1990
2      FORMAT(A60)
      END
C      *****SUBROUTINE INPDAT*****
C      * KEYWORDS AND DATA CORRESPONDING TO KEYWORD ARE ENTERED.
C      * INPUT STRING IS DECOMPOSED, AND ANALYZED. OUTPUT OF
C      * ROUTINE IS KEYWORD AND L-1, THE NUMBER OF NUMERIC VALUES
C      * ASSOCIATED WITH KTYPE
C      *****
C
      SUBROUTINE INPDAT(KTYPE)
      CHARACTER JOB*60,ST*90
```

```

INTEGER PNL,R1,O,T3,T4
REAL*8 A,S,G,HE
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /PRINT/ IPM,IPP,IPE
COMMON /JOB/ JOB
COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
DIMENSION VALUE(10),Q0(3),H0(3)
CHARACTER KTYPE*4,KEY1*8,KEY2*32,KEY3*16
CHARACTER T2*2,NUM*10
C3=448.831
T2=', '
C
C *****ASSIGNING OF TEST STRINGS*****
C * KEY1,KEY2, AND KEY3 CONTAIN ALL AVAILABLE KEYWORDS
C *****
C
KEY1='PIPELINE'
KEY2='ELEVOUTPINPUTANKDIAMLENGCOEFACCU'
KEY3='CHECPUMPPRV RATI'
NUM='123456789'
C
C *****PROMPT FOR INPUT 28-3C*****
C * KTYPE, THE KEYWORD IS PRESENTED (INITIALLY AS 'JOB')
C * THE INPUT STRING IS STORED IN STRING ST
C *****
C
WRITE(6,5)
5 FORMAT(/' STEADY STATE SIMULATION INPUT/' ' TYPE KEYW FOR',
>' LIST OF KEYWORDS')
10 IF (IPP.EQ.1) PRINT 1,KTYPE
1 FORMAT (/, ' S. KEYWORD IS ',A4, ' ENTER (KEYWORD) DATA LIST',/)
14 READ (*,2,END=2222) ST
IF (IPE.EQ.1) PRINT*,ST
IF (KTYPE.EQ.'PUM1'.AND.ST.EQ.'END') ST='E'
2 FORMAT (A60)
16 IF (ST.EQ.'END') GOTO 90
IF (ST(1:4).EQ.'CREA') THEN
CALL SIMSTO(ST)
KTYPE = 'PIPE'
GOTO 10
END IF
IF (ST(1:3).EQ.'GET') THEN
CALL SIMRET(IFLG,ST)
IFLG =0
KTYPE = 'PIPE'
GOTO 10
END IF
IF (ST(1:4).EQ.'KEYW') THEN
CALL KEYWRD(1,KTYPE)
KTYPE='PIPE'
GOTO 10
END IF
C
C *****DECOMPOSITION OF INPUT STRING 28-
3*****

```

```
C      * IF FIRST FOUR CHARACTERS (OR CHARACTERS TO FIRST BLANK,  
C      WHICHEVER IS LESS), ARE ALPHA, KTYPE IS ASSIGNED TO THEM  
C      * KEYWORD REMAINS UNCHANGED IF INPUT STRING BEGINS WITH A  
C      * *A NUMERIC VALUE  
C  
*****  
      K=1  
      L=1  
C  
C      COUNTING BLANKS AND COMMAS BEFORE FIRST ENTRY  
C  
15     IF ((ST(K:K).EQ.' ') .OR. (ST(K:K).EQ.', ')) THEN  
        K=K+1  
        IF (K.GE.90) THEN  
          CALL BLANK  
          GOTO 10  
        END IF  
        GOTO 15  
      END IF  
      IF (ST(K:K).LT.':') GOTO 30  
      J=JJ(ST(K:), ' ', T2)  
      IF (ST(K:K+MIN(J-2,3)).EQ.'JOB') THEN  
        JOB=ST(K+J:LEN(ST))  
        KTYPE='PIPE'  
        GOTO 10  
      END IF  
      IF (KTYPE.EQ.'JOB') THEN  
        JOB=ST(K:LEN(ST))  
        KTYPE='PIPE'  
        GOTO 10  
      END IF  
      J=JJ(ST(K:), ' ', T2)  
      KTYPE=ST(K:K+MIN(J-2,3))  
      IF (KTYPE.EQ.'NODE') GOTO 5101  
20     K=K+J  
      IF (ST(K:K+4).EQ.' ') GOTO 45  
25     IF ((ST(K:K).EQ.' ') .OR. (ST(K:K).EQ.', ')) THEN  
        K=K+1  
        IF (K.GE.90) THEN  
          CALL BLANK  
          GOTO 10  
        END IF  
        GOTO 25  
      END IF  
      IF (ST(K:K).GT.'9') THEN  
        J=JJ(ST(K:), ' ', T2)  
        GOTO 20  
      END IF  
30     J=JJ(ST(K:), ' ', T2)  
      IF (L.GT.11) THEN  
        CALL ERROR(1, KTYPE)  
        GOTO 10  
      END IF  
      VALUE(L)=0  
      IP=INDEX(ST(K:K+J-2), '.')+K-1  
      MM=0  
      DO 40 I=K+J-2, K, -1  
      IF ((I.EQ.K+J-2).AND.(IP.EQ.K-1)) THEN  
        IP=I  
      ELSE IF (I.EQ.IP) THEN
```

```
        MM=1
        GOTO 40
    END IF
    IF (ST(I:I).EQ.'-') THEN
        VALUE(L)=-VALUE(L)
        GOTO 40
    END IF
    IF (ST(I:I).EQ.'+') GOTO 40
    K1 = INDEX(NUM,ST(I:I))
    VALUE(L)=VALUE(L)+(K1)*10.**(IP-I-MM)
40    CONTINUE
    L=L+1
    GOTO 20

C
C *****RECOGNIZING KEYWORD 28-3E*****
C * TRANSFER OF PROGRAM EXECUTION BASED ON POSITION OF KTYPE IN
C * KEY2, OR KEY 3, THE STRINGS CONTAINING ALL AVAILABLE KEYWORDS
C * IF KTYPE IS NOT A VALID KEYWORD, AN ERROR MESSAGE IS
DISPLAYED
C * RETURN IS MADE TO INPUT PROMPT
C *****
C
45    IF (KTYPE.EQ.'NODE') GOTO 5104
    IF (KTYPE.EQ.'PUM1'.OR.KTYPE.EQ.'E') GOTO 5903
    IT=INDEX(KEY1,KTYPE)
    IT=(IT+3)/4+1
    IF(IT.GT.1.AND.L.EQ.3.AND.VALUE(2).EQ.0) GOTO 5701

C
C IF HW COEFICIENT WAS NOT ENTERED, DEFAULT HW COEFICIENT OF
C -HWMA IS ASSIGNED, (MINUS SIGN TO FLAG THE DEFAULT VALUE)
C
    IF (L.EQ.6) THEN
        VALUE(6)=-HWMA
        GOTO 49
    END IF
    IF (L.NE.7) GOTO 46
    GOTO (46,49,49)IT
46    IT=INDEX(KEY2,KTYPE)
    IT=(IT+3)/4+1
    IF (KTYPE.EQ.'COEF'.AND.L.EQ.2)GOTO 5702
    IF (KTYPE.EQ.'COEF'.AND.L.EQ.4)GOTO 5704
    IF (KTYPE.EQ.'ACCU'.AND.L.EQ.4) GOTO 70
    IF (L.NE.3) GOTO 47
    R1=VALUE(1)
    J=R1
    R1=J
    GOTO (47,51,52,53,54,55,56,57,70)IT
47    IT=INDEX(KEY3,KTYPE)
    IT=(IT+3)/4+1
    IF (IT.GT.2.AND.L.EQ.3.AND.VALUE(2).EQ.0) GOTO 5701
CIF ((KTYPE.EQ.'PRV ').AND.(L.EQ.5).AND.(MAPS.EQ.1)) GOTO 60
    IF (KTYPE.EQ.'RATI'.AND.L.EQ.2) THEN
        VALUE(3)=VALUE(1)
        VALUE(1)=1
        VALUE(2)=PNL
        GOTO 5106
    END IF
    IF (L.NE.4) THEN
        CALL ERROR(2,KTYPE)
        GOTO 10
```



```
END IF
GOTO (48,58,59,60,5106) IT
48 PRINT *, 'INSUFICIENT DATA FOR KEYWORD ',KTYPE
CALL ERROR(2,KTYPE)
KTYPE='PIPE'
GOTO 10

C
C *****PIPE DATA 28-
3F*****
C * VALUE(1) = LINK NUMBER
C * VALUE(2) = BEGINING NODE
C * VALUE(3) = ENDING NODE
C * VALUE(4) = PIPE DIAMETER
C * VALUE(6) = HAZEN WILLIAMS COEFFICIENT IF
C * ENTERED, IF NOT, DEFAULT OF -100 IS ASSIGNED
C *****
C
49 R1=VALUE(1)
IF (R1.GT.PNL) THEN
PRINT*, ' LINK NUMBER TOO LARGE.'
GOTO 10
END IF
CALL PIPECH(R1)

C
C SWITCHING BEGINNING NODE WITH ENDING NODE, IF BEGINNING NODE
C IS GREATER THAN ENDING NODE NUMBER
C
IF (VALUE(2).LT.VALUE(3)) GOTO 50
IT=VALUE(3)
VALUE(3)=VALUE(2)
VALUE(2)=IT
50 IF (VALUE(2).GT.PNL.OR.VALUE(3).GT.PNL) THEN
PRINT*, ' NODE NUMBER TOO LARGE.'
GOTO 10
END IF
IF (VALUE(5).LE.0.OR.VALUE(5).GE.1000000.) THEN
CALL ERROR(4,KTYPE)
GOTO 10
END IF
IF (VALUE(4).LE.0.) THEN
CALL ERROR(5,KTYPE)
GOTO 10
END IF
IF (ABS(VALUE(6)).LT.1E-5) THEN
CALL ERROR(6,KTYPE)
GOTO 10
END IF

C
C VALUE(4), THE PIPE DIAMETER, IS CONVERTED FROM INCHES TO FEET
C
DI(R1)=VALUE(4)/12
XL(R1)=VALUE(5)
HW(R1)=VALUE(6)
R3=VALUE(6)
IF (R3.LT.0) R3=HWMA

C
C CHARACTERISTIC PIPE COEFFICIENT, CP, ASSIGNED FOR LINK, R1
C
CP(R1)=4.72*ABS(XL(R1))/(R3**1.85*DI(R1)**4.87)
A(R1)=1/CP(R1)
```

```
5001 L=VALUE(2)
      O(4)=0
      IBE(R1)=L
C
C   ARRAY INO SUBSCRIPTED WITH THE USER'S NODE NUMBER IS ASSIGNED
C   A 1 TO FLAG THAT THE NODE HAS BEEN ENTERED
C
      INO(L)=1
      L=VALUE(3)
      IEN(R1)=L
      INO(L)=1
      IF (IPI(R1).EQ.0) O(1)=O(1)+1
      IPI(R1)=1
      IF (KTYPE.EQ.'PUMP') GOTO 5901
      GOTO 10
C
C   *****DIAMETER 28-3G*****
C   * L-1 MUST EQUAL 1, VALUE(1)=LINK NUMBER, VALUE(2)=DIAMETER
C   * IN INCHES. PROGRAM TESTS WHETHER LINK NUMBER WAS PREVIOUSLY
C   * ENTERED AND ASSIGNS THE NEW DIAMETER IN FEET TO ARRAY DI.
C   *****
C
55   IF (IPI(R1).EQ.1) GOTO 5502
5501 PRINT *, 'THIS PIPE WAS NOT ENTERED YET'
      GOTO 10
5502 IF (XL(R1).GT.999999..OR.CP(R1).LT.0) THEN
      PRINT*, ' THIS LINK IS NOT A PIPE.'
      GOTO 10
      END IF
      IF (VALUE(2).LE.0) THEN
      CALL ERROR(5,KTYPE)
      GOTO 10
      END IF
      DI(R1)=VALUE(2)/12
5503 R3=HW(R1)
      IF (R3.LT.0) R3=HWMA
C
C   ARRAYS CP AND A ARE UPDATED
C
      CP(R1)=4.72*ABS(XL(R1))/(R3**1.85*DI(R1)**4.87)
      A(R1)=1/CP(R1)
      GOTO 10
C
C   *****LENGTH 28-3H*****
C   * KEYWORD IS LENGTH. L-1 MUST EQUAL 2.
C   * VALUE(1)=LINK NUMBER
C   * VALUE(2)=LENGTH OF PIPE IN FEET
C   * A TEST IS MADE TO SEE WHETHER LINK NUMBER WAS PREVIOUSLY
C   * ENTERED AS A PIPE AND ASSIGNS NEW LENGTH TO ARRAY XL
C   *****
C
56   R1=VALUE(1)
      IF (IPI(R1).NE.1) GOTO 5501
      IF (XL(R1).GT.999999..OR.CP(R1).LT.0) GOTO 5502
      IF (VALUE(2).LE.0.OR.VALUE(2).GE.1000000.) THEN
      CALL ERROR(4,KTYPE)
      GOTO 10
      END IF
      XL(R1)=VALUE(2)
      GOTO 5503
```

```
C
C *****COEF ONE PIPE 28-3I*****
C *** OPTION CHANGES HAZEN-WILLIAMS COEFFICIENT FOR ONE PIPE
C *** VALUE(1) = LINK NUMBER
C *** VALUE(2) = NEW HAZEN-WILLIAMS COEFFICIENT FOR PIPE
C *****
C
57 IF (IPI(R1).NE.1) GOTO 5501
IF (XL(R1).GT.999999..OR.CP(R1).LT.0) GOTO 5502
IF (VALUE(2).LE.0) THEN
  CALL ERROR(4,KTYPE)
  GOTO 10
END IF
HW(R1)=VALUE(2)
GOTO 5503

C
C *****DELETING LINK
C
5701 R1=VALUE(1)
IF (IPI(R1).EQ.0) GOTO 5501
IPI(R1)=0
T3=IBE(R1)
T4=IEN(R1)
IBE(R1)=0
IEN(R1)=0
O(1)=O(1)-1
O(4)=0
DO 57012 I=1,PNL
IF (IPI(I).EQ.0) GOTO 57012
IF (IBE(I).EQ.T3.OR.IEN(I).EQ.T3) GOTO 57013
57012 CONTINUE
INO(T3)=0
EL(I)=0
O(2)=O(2)-1
57013 DO 57014 I=1,PNL
IF (IPI(I).EQ.0) GOTO 57014
IF (IBE(I).EQ.T4.OR.IEN(I).EQ.T4) GOTO 57015
57014 CONTINUE
INO(T4)=0
EL(I)=0
WRITE(6,500) T4
500 FORMAT(' NODE ',I4,' DELETED.')
O(2)=O(2)-1
57015 GOTO 10

C
C *****COEF DEFAULT VALUE*****
C * OPTION CHANGES THE DEFAULT VALUE OF THE HAZEN-WILLIAMS*
C * COEFFICIENT. *
C * VALUE(1) = NEW DEFAULT COEFFICIENT ASSIGNED TO HW *
C * ALL PREVIOUSLY ENTERED PIPES WITH THE DEFUALT *
C * COEFFICIENT ARE ASSIGNED THE NEW HW COEFFICIENT. *
C *
C * ARRAYS CP AND A ARE UPDATED ACCORDINGLY *
C *****
C
5702 IF (VALUE(1).LE.0) THEN
  CALL ERROR(6,KTYPE)
  GOTO 10
END IF
HWMA=VALUE(1)
```

```

DO 5703 L=1,PNL
IF (IPI(L).EQ.0) GOTO 5703
IF (HW(L).GT.0) GOTO 5703
CP(L)=4.72*ABS(XL(L))/(HWMA**1.85*DI(L)**4.87)
A(L)=1./CP(L)
HW(L)=-HWMA
5703 CONTINUE
GOTO 10
C *****COEFF RANGE*****
C * OPTION CHANGES HAZEN-WILLIAMS COEFFICIENT FOR A RANGE *
C * OF PIPES. *
C * VALUE(1) = FIRST LINK NUMBER *
C * VALUE(2) = LAST LINK NUMBER *
C * VALUE(3) = NEW HW COEF TO BE ASSIGNED TO RANGE OF PIPES *
C * OF PIPES VALUE(1) THROUGH VALUE(2) *
C * ARRAYS CP AND A ARE UPDATED ACCORDINGLY *
C *****
5704 IF (VALUE(1).GT.PNL.OR.VALUE(2).GT.PNL) THEN
PRINT*, ' LINK NUMBER TOO LARGE.'
GOTO 10
END IF
IF (VALUE(3).LE.0) THEN
CALL ERROR(6,KTYPE)
GOTO 10
END IF
C
C NEW HAZEN-WILLIAMS COEFFICIENT IS ASSIGNED
C
DO 5705 L=VALUE(1),VALUE(2)
IF (IPI(L).NE.1) GOTO 5705
IF (CP(L).LT.0.OR.XL(L).GT.999999.) GOTO 5705
HW(L)=VALUE(3)
C
C ARRAYS CP AND A ARE UPDATED
C
CP(L)=4.72*ABS(XL(L))/(HW(L)**1.85*DI(L)**4.87)
A(L)=1./CP(L)
5705 CONTINUE
GOTO 10
C *****NODE DATA
C *****ELEVATION 28-3J*****
C CHANGES OR ASSIGNS THE ELEVATION OF A NODE
C L-1 MUST EQUAL 2
C VALUE(1) = NODE NUMBER
C VALUE(2) = ELEVATION IN FEET
C O(2) IS INCREMENTED IF NODE WAS NOT YET ASSIGNED AN ELEVATION
C *****
C
51 IF (R1.GT.PNL) THEN
PRINT*, ' NODE NUMBER TOO LARGE.'
GOTO 10
END IF
IF (EL(R1).EQ.0) THEN
O(2)=O(2)+1
O(4)=0
END IF
C
C NEW ELEVATION IS ASSIGNED TO TO ARRAY EL, RETAINING SIGN OF
C OLD ELEVATION. (A NEGATIVE ELEVATION INDICATES THAT THERE IS
C A PUMP OR PRV ADJACENT TO THE NODE. )

```

```
C
  IF (VALUE(2).LT.0) THEN
    CALL ERROR(7,KTYPE)
    GOTO 10
  END IF
  IF (VALUE(2).LT..01) VALUE(2)=.01
  EL(R1)=SIGN(VALUE(2),EL(R1))

C
C   ARRAY HE IS INITIALIZED WITH THE ELEVATION.  IF THE NODE WAS
C   PREVIOUSLY DECLARED A TANK (DO>1E10), THE ARRAY HE IS SET TO
C   A VALUE EQUAL TO NODE ELEVATION PLUS WATER LEVEL.
C
  HE(R1)=ABS(EL(R1))
  IF (DO(R1).GT.1E10) HE(R1)=ABS(EL(R1))+DO(R1)/1E10-100
  GOTO 65

C
C   *****OUTPUT 28-3K
C   KEYWORD IS OUTP.  OUTPUT IN GPM IS ASSIGNED TO NODE
C   L-1 MUST EQUAL 2
C   VALUE(1) = NODE NUMBER
C   VALUE(2) = OUTPUT IN GALLONS PER MINUTE
C   TEST IS MADE TO SEE WHETHER NODE WAS PREVIOUSLY DECLARED A
TANK.
C
52  IF (R1.GT.PNL) GOTO 51
    IF (DO(R1).GT.1E10) THEN
      WRITE(6,510) R1
510  FORMAT(1X,I4,' WAS ENTERED AS SUPPLY POINT. ')
      PRINT*,' NEW DATA RETAINED.'
    END IF

C
C   NEW OUTPUT IS ASSIGNED TO ARRAY DO (IN GPM)
C
  DO(R1)=VALUE(2)
  GOTO 65

C
C   *****INPUT 28-3L
C   KEYWORD IS INPU.  INPUT (GPM) IS ASSIGNED TO NODE.
C   VALUE(1) = NODE NUMBER
C   VALUE(2) = INPUT IN GALLONS PER MINUTE
C   THE SIGN OF THE INPUT IS REVERSED SINCE AN INPUT IS
C   THE SAME AS A NEGATIVE OUTPUT
C
53  VALUE(2)=-VALUE(2)
    GOTO 52

C
C   *****TANK 28-3M*****
C   ASSIGNS TANK ELEVATION TO A NODE PREVIOUSLY DECALRED AS A
C   TANK.
C   VALUE(1) INDICATES THE NODE NUMBER
C   VALUE(2) IS THE WATER LEVEL IN FEET
C   TEST IS MADE TO SEE WHETHER NODE WAS PREVIOUSLY DECLARED
C   AS A TANK.  IF NOT A WARNING MESSAGE IS PRINTED.
C
54  IF (R1.GT.PNL) GOTO 51
    IF (ABS(DO(R1)).LT..001.OR.DO(R1).GT.1E10) GOTO 5401
    WRITE(6,520) R1
520  FORMAT(1X,I4,' WAS ENTERED WITH OUTPUT/INPUT. ')
    PRINT*,' NEW DATA RETAINED.'
5401 IF (VALUE(2).LT.0) THEN
```

```
        CALL ERROR(8,KTYPE)
        GOTO 10
    END IF

C
C   ARRAY DO IS ASSIGNED.  ADDING THE CONSTANT OF 100 TO
C   VALUE(2) IS NECESSARY TO PERMIT THE USE OF WATER LEVEL OF 0
C   THE MULTIPLIER 1E10 IDENTIFIES THE NODE AS A TANK.
C
DO(R1)=(VALUE(2)+100)*1E10

C
C   GROUND ELEVATION PLUS WATER LEVEL IS ASSIGN TO ARRAY HE (TOTAL
C   HYDRAULIC HEAD)
C
HE(R1)=EL(R1)+VALUE(2)
65   INO(R1)=1
      GOTO 10

C
C   *****NODE PROMPTS 28-3N
C   NO NUMERIC DATA FOLLOWS THE KEYWORD NODE.
C   USER IS PROMPTED FOR ELEVATION AND OUTPUT FOR ALL NODE NUMBERS
C   PREVIOUSLY USED AS BEGINNING NODES OR ENDING NODES WITH NO
C   ELEVATION YET ENTERED.
C
5101  O(2)=0
      II=1
5102  IF (II.GT.PNL) GOTO 51051
      IF (INO(II).EQ.0) GOTO 51049
      IF (ABS(EL(II)).GT.0) GOTO 5105
      IF (IPP.EQ.1) PRINT 5103,II
5103  FORMAT (/, ' FOR NODE ',I4,' ENTER ELEVATION, OUTPUT',/)
      VALUE(2)=0
      GOTO 14
5104  IF (VALUE(1).LT.0) THEN
      CALL ERROR(7,KTYPE)
      GOTO 5101
    END IF
    IF (VALUE(1).LT..01) VALUE(1)=.01

C
C   VALUE(1), THE ELEVATION IN FEET, IS ASSIGNED TO ARRAY EL
C   AND HE.
C
EL(II)=VALUE(1)
HE(II)=EL(II)

C
C   IF THE NODE WAS PREVIOUSLY DECLARED A TANK, THE ARRAY HE IS
C   GIVEN THE VALUE OF ELEVATION PLUS WATER LEVEL.
C
IF (DO(II).GT.1E10) THEN
    HE(II)=EL(II)+DO(II)/1E10-100
ELSE

C
C   IF THE NODE WAS NOT DECLARED A TANK, VALUE(2), THE OUTPUT (GPM)
C   IS ASSIGNED TO ARRAY DO
C
DO(II)=VALUE(2)
END IF

C
C   NODE COUNTER IS INCREMENTED
C
5105  O(2)=O(2)+1
```

```
O(4)=0
51049 II=II+1
      GOTO 5102
51051 KTYPE='TANK'
      GOTO 10
C
C *****OUTPUT - RATIO 28-30
C KEYWORD IS RATI
C IF L-1 = 1:
C VALUE(3) IS ASSIGNED THE VALUE OF VALUE(1)
C VALUE(1) IS ASSIGNED 1
C VALUE(2) IS ASSIGNED THE HIGHEST PERMISSIBLE NODE NUMBER
C IF L-1=3, VALUE(1) INDICATES THE LOWEST NODE NUMBER AND VALUE(2)
C THE HIGHEST NODE NUMBER FOR WHICH THE OUTPUT, DO, IS TO BE
C MULTIPLIED BY VALUE(3).
C
5106 IF (VALUE(1).GT.PNL.OR.VALUE(2).GT.PNL) GOTO 51
      DO 5107 R1=VALUE(1),VALUE(2)
      IF (INO(R1).NE.1) GOTO 5107
      IF (DO(R1).GT.0.AND.DO(R1).LT.1E12) DO(R1)=VALUE(3)*DO(R1)
5107 CONTINUE
      GOTO 10
C
C *****PRESSURE REDUCING VALVE 28-3P
C KEYWORD IS PRV. L-1 MUST EQUAL 3
C VALUE(1) IS THE LINK NUMBER
C VALUE(2) IS THE UPSTREAM (BEGINNING) NODE OF THE PRV
C VALUE(3) IS THE DOWNSTREAM (ENDING) NODE OF THE PRV
C
60 R1=VALUE(1)
      CALL PIPECH(R1)
CIF (MAPS.EQ.1) GOTO 6001
      IF (IPI(R1).EQ.1) WRITE(6,600) HW(R1)
600 FORMAT(1X,'OLD SETTING ',F6.2,' PSI.')
C
C PROMPT IS DISPLAYED AND PRESSURE SETTING IS READ IN
C
6002 IF (IPP.EQ.1) THEN
      PRINT *,'ENTER PRESSURE SETTING'
      PRINT *
      END IF
      READ*,PSET
      IF (IPE.EQ.1) WRITE(6,610) PSET
610 FORMAT(F6.2)
      IF (PSET.LT.0) THEN
      CALL ERROR(9,KTYPE)
      GOTO 10
      END IF
C
C ARRAY DI IS SET TO 0.001 FT, ARRAY XL IS SET TO 1000000 (AS AN
C IDENTIFIER OF THE PRV), PSET IS ASSIGNED TO ARRAY HW
C ARRAYS CP AND A ARE ASSIGNED VALUES OF 0.0001 AND 0
C
6001 DI(R1)=0.001
      XL(R1)=1000000.
      HW(R1)=PSET
      CP(R1)=.0001
      A(R1)=0
      GOTO 5001
C
```

```
C      *****CHECK VALVE 28-3Q
C      KEYWORD IS CHEC AND L-1 MUST EQUAL 1
C      VALUE(1) INDICATES THE LINK NUMBER
C      VALUE(2) IS THE BEGINNING NODE NUMBER
C      VALUE(3) IS THE ENDING NODE NUMBER
C      THE CHECK VALVE IS OPEN IF FLOW GOES FROM THE BEGINNING TO THE
C      ENDING NODE.
C
58     R1=VALUE(1)
        IF (R1.GT.PNL) GOTO 49
        L=VALUE(2)
        J=VALUE(3)
C
C      A CHECK IS MADE TO SEE THAT LINK EXISTS
C
        IF (IPI(R1).NE.1) GOTO 5501
        IF (XL(R1).GT.999999..OR.CP(R1).LT.0) GOTO 5502
C
C      CHECK IS MADE TO SEE WHETHER BEGINNING AND ENDING NODES
C      MATCH THE ONES PREVIOUSLY ENTERED FOR THE LINK
C
        IF (L.EQ.IBE(R1).AND.J.EQ.IEN(R1)) GOTO 5801
        IF (J.EQ.IBE(R1).AND.L.EQ.IEN(R1)) GOTO 5801
        CALL ERROR(10,KTYPE)
        GOTO 10
C
C      VALUE(2), L, IS ASSIGNED TO IBE AND VALUE(3),J, IS ASSIGNED
C      TO IEN SUBSCRIPTED WITH R1, THE LINK NUMBER
C
5801    IBE(R1)=L
        IEN(R1)=J
C
C      ARRAY XL IS ASSIGNED THE NEGATIVE VALUE OF THE PIPE LENGTH TO
C      RECOGNIZE PIPES WITH CHECK VALVES BY MEANS OF THE NEGATIVE
C      VALUE OF THE PIPE LENGTH
C
        XL(INT(VALUE(1)))=-ABS(XL(INT(VALUE(1))))
        GOTO 10
C
C      *****PUMP 28-3R
C      KEYWORD IS PUMP AND L-1 MUST EQUAL 3
C      VALUE(1) INDICATES THE LINK NUMBER
C      VALUE(2) IS THE BEGINNING NODE NUMBER (SUCTION SIDE OF PUMP)
C      VALUE(3) IS THE ENDING NODE NUMBER (DISCHARGE SIDE OF PUMP)
C      PUMP PARAMETERS ARE ASSIGNED:
C
59     R1=VALUE(1)
        IF (R1.GT.PNL) GOTO 49
        CALL PIPECH(R1)
        GOTO 5001
C
C      PROMPT IS PRINTED THREE TIMES REQUESTING DISCHARGE AND HEAD
C      FOR POINTS ON THE CHARACTERISTIC CURVE OF THE PUMP.
C
5901    IB=0
5904    IB=IB+1
        IF (IPP.EQ.1) PRINT 5902,IB
5902    FORMAT (' POINT',I3,' ON CHARACTERISTIC CURVE: ENTER DISCHARGE,
1HEAD',/)
C
```



```
C      KTYPE IS SET TO PUM1 AS A FLAG SO RETURN IS MADE TO 5903
C
      KTYPE='PUM1'
      GOTO 14
C
C      IF USER ENTERS THE LETTER E AT THE TIME OF THE SECOND POINT ON
C      THE CHARACTERISTIC CURVE IS REQUESTED, THE CHARACTERISTIC CURVE
C      DEFAULTS SO DISCHARGE IS AT 0 AND A HEAD OF 133.334% OF HEAD
C      ENTERED FOR THE FIRST POINT. POINT THREE HAS A HEAD OF
C      0, AT A DISCHARGE TWICE THE DISCHARGE ENTERED FOR POINT ONE.
C
5903  IF (KTYPE.EQ.'E') THEN
      Q0(2)=0
      H0(2)=(1.33334)*H0(1)
      Q0(3)=2*Q0(1)
      H0(3)=0
      GOTO 5905
      END IF
C
C      VALUE(1) IS THE DISCHARGE IN GALLONS PER MINUTE
C      VALUE(2) IS PUMP HEAD IN FEET
C      DISCHARGE, CONVERTED TO CFS, IS ASSIGNED TO ARRAY Q0 AND HEAD
C      IN FEET TO ARRAY H0
C
      Q0(IB)=VALUE(1)/C3
      H0(IB)=VALUE(2)
      IF (IB.NE.3) GOTO 5904
      IF ((Q0(1)-Q0(2))*(Q0(2)-Q0(3))*(Q0(3)-Q0(1)).NE.0) GOTO 5905
      PRINT *, 'REPEAT OF DISCHARGE. START OVER.'
      GOTO 5906
C
C      PROGRAM COMPUTES THE COEFFICIENTS FOR A PARABOLA THROUGH
C      THE THREE POINTS ON THE CHARACTERISTIC CURVE.
C
5905  H2=H0(1)-H0(2)
      H3=H0(1)-H0(3)
      Q2=Q0(1)-Q0(2)
      Q3=Q0(1)-Q0(3)
      Q4=Q0(1)**2-Q0(2)**2
      Q5=Q0(1)**2-Q0(3)**2
      IF (ABS(Q4*Q3-Q5*Q2).LT.1E-10) THEN
          PRINT*, 'DISCHARGES ARE TOO CLOSE TOGETHER'
          GOTO 5907
      END IF
      Y1=(H2*Q3-H3*Q2)/(Q4*Q3-Q5*Q2)
      Y2=(H2-Y1*Q4)/Q2
      Y3=H0(2)-Y1*Q0(2)**2-Y2*Q0(2)
C
C      COEFFICIENTS Y1,Y2, AND Y3 ARE ASSIGNED TO ARRAYS CP, DI
C      AND HW RESPECTIVELY.
C
      CP(R1)=Y1
      DI(R1)=Y2
      HW(R1)=Y3
      WRITE(6,113) R1
113  FORMAT(' PUMP COEFFICIENTS FOR PUMP ',I3,'. Q IN CFS')
      PRINT*, '      Q*Q      Q CONSTANT'
      PRINT 115,CP(R1),DI(R1),HW(R1)
115  FORMAT (F11.4,F10.4,F10.1,/)
C
```

```
C      ARRAY A IS ASSIGNED , CORRESPONDING TO AN OPERATING POINT AT
C      Q0(2)/H0(2)
C
A(R1)=1.85/SQRT(Y2**2-4*Y1*(Y3-H0(2)))
IF (Y1.LT.0.AND.Y3.GT.0.AND.Y2.LE.0) GOTO 5906
IF (Y2.GT.0) THEN
  PRINT*, 'WARNING: CONVERGENCE DURING BALANCING MAY BE '
  PRINT*, 'A PROBLEM DUE TO SHAPE OF CHARACTERISTIC CURVE.'
  GOTO 5906
END IF
IF (Y1.GE.0) THEN
  PRINT*, 'CHARACTERISTIC CURVE IS CONCAVE UP ! '
ELSE
  PRINT*, 'POSITIVE SHUT OFF HEAD REQUIURED.'
END IF
5907 PRINT *, 'START OVER.'
5906 KTYPE='PUMP'
GOTO 10

C
C      *****ACCURACY 28-3S
C      KEYWORD IS ACCU
C      VALUE(1) REPRESENTS THE REQUESTED PRESSURE ACCURACY IS PSI
C      VALUE(2) IS THE REQUESTED FLOW ACCURACY IN GPM
C      VALUE(3) (OPTIONAL) IS THE MAXIMUM NUMBER OF ITERATIONS TO
C      BE CARRIED OUT.
C      VALUE(1) IS ASSIGNED TO PRAC, VALUE(2) IS ASSIGNED TO FLAC,
C      AND VALUE(3) (DEFAULT 24) IS ASSIGNED TO ICL.
C
70  PRAC=VALUE(1)
FLAC=VALUE(2)/C3
IF (L.EQ.4) ICL=VALUE(3)
KTYPE='PIPE'
GOTO 10

2222 CALL BLANK
IF (KTYPE.EQ.'NODE') GOTO 5101
GOTO 10

C
C *****PROCESSING OF INPUT DATA 28-4
C *****PIPE DATA
C
90  J0=0
C
C      PROGRAM TESTS WHETHER EVERY NODE WAS ASSIGNED AN ELEVATION
C      NODES WHICH WERE NOT ASSIGNED AN ELEVATION ARE LISTED.
C
DO 91 J=1,PNL
IF (INO(J).EQ.0) GOTO 91
IF (ABS(EL(J)).LT.1E-5) THEN
  EL(J)=.01
  J0=1
END IF
91  CONTINUE
C
C      PROGRAM TESTS FOR EACH PIPE WHETHER DATA WAS ENTERED FOR
C      ITS BEGINNING AND ENDING NODE
C
92  DO 941 I=1,PNL
IF (IPI(I).EQ.0) GOTO 941
K=IBE(I)
IF (INO(K).GT.0) GOTO 93
```

```
PRINT*, ' DATA FOR NODE ',K,' WAS NOT ENTERED.'
O(2)=O(2)+1
93 INO(K)=2
K=IEN(I)
IF (INO(K).GT.0) GOTO 94
PRINT*, ' DATA FOR NODE ',K,' WAS NOT ENTERED.'
O(2)=O(2)+1
C
C   NODES USED AS BEGINNING OR ENDING NODES ARE MARKED OFF BY
C   ASSIGNING INO, SUBSCRIPTED WITH USER NODE NUMBER, TO 2
C
94 INO(K)=2
941 CONTINUE
C
C   ARRAY INO IS TESTED FOR NODES WHICH WERE NOT USED AS
C   BEGINNING OR ENDING NODES (REMAINING 1S IN ARRAY). SUCH
C   NODES ARE LISTED AS NODES WITHOUT PIPES LEADING TO THEM. THE
C   PROGRAM WILL IGNORE SUCH NODES AND ADJUSTS THE NODE COUNTER
C   O(2).
C
DO 95 I=1,PNL
IF (INO(I).EQ.1) THEN
PRINT*, ' NODE ',I,' HAS NO PIPE LEADING TO IT.'
PRINT*, ' NODE IS IGNORED.'
IF (ABS(EL(I)).GT.1E-5) O(2)=O(2)-1
EL(I)=0
DO(I)=0
HE(I)=0
INO(I)=0
END IF
95 CONTINUE
C
C   *** COMPRESSING OF THE LINK DATA ***
C
98 J=1
O(9)=0
DO 100 I=1,PNL
IF (IPI(I).EQ.0) GOTO 100
IBE(J)=IBE(I)
IEN(J)=IEN(I)
IPI(J)=I
C
C   ARRAY ELEMENTS A, CP, DI, XL ARE SHIFTED FROM THE USER
C   LINK NUMBER TO THE INTERNAL LINK NUMBER I.
C
A(J)=A(I)
IF (A(J).LT.1.E-10) A(J)=1.1E-10
IF (I.NE.J) A(I)=0
CP(J)=CP(I)
DI(J)=DI(I)
XL(J)=XL(I)
HW(J)=HW(I)
C
C   O(10) IS SET TO ONE IN CASE THERE ARE PRV'S (XL(I)>999999)
C   OR CHECK VALVES (XL(I)<0). O(10) WILL BE USED TO INHIBIT
C   OVERRELAXATION IN NUMERIC SOLUTION OF THESE CASES.
C
IF (XL(J).GT.999999..OR.XL(J).LT.0) O(10)=1
J=J+1
IF (J.GT.O(1)) GOTO 10001
```

```
100  CONTINUE
C
C *****COMPRESSING OF NODE DATA BEGINS *****
C
10001 J=1
      J0=0
      DO 110 I=1,PNL
      IF (INO(I).EQ.0) GOTO 110
      INO(J)=I
C
C      ARRAY ELEMENTS HE,EL, AND DO ARE SHIFTED FROM THE USER NODE
C      NUMBER TO THE INTERNAL NODE NUMBER I
C
      HE(J)=HE(I)
      EL(J)=EL(I)
      DO(J)=DO(I)
C
C      IF DO(J)>1E10 (SUPPLY POINT) 'FLAG' J0 IS SET TO 1
C
      IF (DO(J).GT.1E10) J0=1
      J=J+1
      IF (J.GT.O(2)) GOTO 10002
110  CONTINUE
C
C      IF FLAG J0 IS 0 AFTER ALL NODES ARE REASSIGNED, THE MESSAGE
C      BELOW IS PRINTED.
C
      IF (J0.EQ.0) PRINT*,' YOU HAVE NO SUPPLY POINT.'
      IF (O(4).GT.0) RETURN
C
C *****SPECIAL DEVICE FLAGS
C
10002 DO 1102 I=1,O(1)
      DO 11021 J=1,O(2)
11021 IF (IBE(I).EQ.INO(J)) GOTO 11022
11022 IBI(I)=J
      DO 11023 J=1,O(2)
11023 IF (IEN(I).EQ.INO(J)) GOTO 11024
11024 IEI(I)=J
      IF (CP(I).LT.0) GOTO 1101
      IF (XL(I).LT.1000000.) GOTO 1102
C
C      IF A LINK IS A PRV, ITS CORRESPONDING ARRAY ELEMENT A (ENTRY
C      IN COEFFICIENT MATRIX) IS SET TO 10000. (I.E PRV IS ASSUMED
C      COMPLETELY OPEN)
C
      A(I)=10000.
1101  IB=IBI(I)
      IE=IEI(I)
C
C      IF A LINK IS A PRV OR A PUMP, THE ELEVATION AT ITS BEGINNING
C      AND ENDING NODE IS PROVIDED WITH A NEGATIVE SIGN SERVING AS A
C      FLAG IN THE NODE DATA, THAT THIS NODE IS THE CONNECTING POINT
C      TO A PRV OR PUMP AND REQUIRES SPECIAL HANDLING.
C
      EL(IB)=-ABS(EL(IB))
      EL(IE)=-ABS(EL(IE))
1102  CONTINUE
      RETURN
      END
```

```
C
C *****PRINTING OF INPUT*****
C
      SUBROUTINE PRNINP(ST)
      CHARACTER JOB*60,ST*90
      INTEGER O,PNL
      REAL*8 A,S,G,HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /TOPOL/ IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),
1IEI(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /PRINT/ IPM,IPP,IPE
      COMMON /JOB/ JOB
C
C      PRINTING OF OVERALL TABLE HEADING
C
1103 PRINT 101
      101 FORMAT (/,/,5X,' PIPE NETWORK ANALYSIS AND OPTIMIZATION',/)
      PRINT 102,JOB
      102 FORMAT (' JOB: ',A60,/)
      PRINT 103
C
C*****NODE TABLE
C
      103 FORMAT (13X,' NODE',5X,'ELEV.',4X,'DOM. LOAD')
      PRINT 104
      104 FORMAT (14X,' NO',7X,'FT.',8X,'GPM',/)
      DO 120 I=1,O(2)
C
C      A PAUSE IS PROVIDED AFTER PRINTING 18 CONSECUTIVE LINES,
C      UNLESS STRING ST HAS A C IN ITS SECOND POSITION
C
      IF (MOD((I+5),18).EQ.0.AND.ST(2:2).NE.'C') THEN
C
C      INPUT OF ST(2:) ALLOWS USER TO RESET SECOND CHARACTER IN ST
C      TO C OR E. AN ENTRY OF E WILL TERMINATE THE NODE TABLE
PRINTING.
C
      READ (*,2,END=1202) ST(2:)
      IF (ST(2:2).EQ.'E') GOTO 125
      END IF
      GOTO 1203
1202 CLOSE (2)
      OPEN (2,FILE='INPUT')
1203 K=INO(I)
C
C      PROGRAM TESTS ARRAY DO FOR SUPPLY POINTS. IF DO>9E9 THE
C      PROGRAM COMPARES HE AND EL. IF THE HEAD IS LARGER THAN THE
C      ELEVATION (BY MORE THAN 0.5 FT) THE NODE IS LABELED AS A TANK.
C      IF THE HEAD IS LESS THAN 0.5 FT THE NODE IS LABELED AS A
C      RESERVOIR.
C
      IF (DO(I).GT.9E9) THEN
          DIFF = HE(I)-ABS(EL(I))
          IF (DIFF .GT. .5) THEN
              PRINT 1051,K,ABS(EL(I)),HE(I)-ABS(EL(I))
          ELSE
              PRINT 1052,K,ABS(EL(I))
          END IF
      END IF
```

```
1051 FORMAT (12X,I4,5X,F6.1,12X,'WATER LEVEL: ',F5.1)
1052 FORMAT (12X,I4,5X,F6.1,12X,'RESERVOIR ')
105  FORMAT (12X,I4,5X,F6.1,4X,F7.0)
      GOTO 120
      END IF
C
C      IF DO>1E10, THE DATA IS PRINTED WITH THE FOLLOWING
C      FORMAT STATEMENT
C
      PRINT 105,K,ABS(EL(I)),DO(I)
120  CONTINUE
C
C*****PIPE TABLE
C
125  PRINT 106
      ST(1:1)=' '
106  FORMAT (/,3X,'PIPE CONNECTIONS',/)
      PRINT 107
107  FORMAT (3X,'PIPE',5X,'B',7X,'E',6X,'DIAM.',5X,'LENGTH',5X,'H-W-
C')
      PRINT 108
108  FORMAT (4X,'NO',5X,'NODE',4X,'NODE',5X,'IN.',8X,'FT.',/)
      DO 130 I=1,O(1)
C
C      A PAUSE IS PROVIDED AFTER PRINTING 18 CONSECUTIVE LINES
C      UNLESS STRING ST HAS A C IN ITS SECOND POSITION
C
      IF (MOD((O(2)+6+I),18).EQ.0.AND.ST(2:2).NE.'C') THEN
          READ (*,2,END=1302) ST(2:)
          IF (ST(2:2).EQ.'E') RETURN
      END IF
      GOTO 1303
1302 CLOSE (2)
      OPEN (2,FILE='INPUT')
C
C      USER LINK, BEGINNING NODE, AND ENDING NODE NUMBERS ARE
DETERMINED
C
1303 L=IPI(I)
      K=IBE(I)
      J=IEN(I)
C
C      LINK IS DETERMINED TO BE A PIPE (XL<1000000 AND CP>0), OR
C      A PRV (XL>999999), OR A PUMP (CP<0)
C
      IF (XL(I).LT.1000000..AND.CP(I).GT.0) GOTO 128
      IF (XL(I).GT.999999.) GOTO 127
C
C      PUMP DATA IS PRINTED
C
      PRINT 112,L,K,J
      GOTO 130
C
C      PRV DATA IS PRINTED
C
127  PRINT 113,L,K,J,HW(I)
      GOTO 130
C
C      PIPES ARE TESTED FOR PRESENCE OF A CHECK VALVE
C
```

```
128 IF (XL(I).GT.0) GOTO 129
    IF (HW(I).LT.0) GOTO 1281
C
C   IF A CHECK VALVE IS PRESENT (XL<0) AND HAZEN-WILLIAMS
C   COEFFICIENT DOES NOT REPRESENT THE DEFAULT VALUE, DATA
C   IS PRINTED
C
    PRINT 114,L,K,J,DI(I)*12,-XL(I),HW(I)
    GOTO 130
C
C   IF A CHECK VALVE IS PRESENT (XL<0) AND HAZEN-WILLIAMS
C   COEFFICIENT DOES REPRESENT THE DEFAULT VALUE, DATA IS PRINTED
C
1281 PRINT 117,L,K,J,DI(I)*12,-XL(I),-HW(I)
    GOTO 130
129 IF (HW(I).LT.0) GOTO 1291
C
C   IF NO CHECK VALVE IS PRESENT, DATA IS PRINTED
C
    PRINT 109,L,K,J,DI(I)*12,XL(I),HW(I)
    GOTO 130
1291 PRINT 116,L,K,J,DI(I)*12,XL(I),-HW(I)
109 FORMAT (I6,2I8,F9.1,F12.1,F10.0)
116 FORMAT (I6,2I8,F9.1,F12.1,F10.0,' ')
112 FORMAT (I6,2I8,33X,'PUMP')
113 FORMAT (I6,2I8,33X,'PRV AT ',F5.1,' PSI')
114 FORMAT (I6,2I8,F9.1,F12.1,F10.0,' CHECK VALVE')
117 FORMAT (I6,2I8,F9.1,F12.1,F10.0,' * CHECK VALVE')
130 CONTINUE
C
C   PUMP COEFFICIENTS AND PUMP DATA ARE PRINTED
C
    J = 0
    PRINT*
    DO 131 I=1,O(1)
    IF (CP(I).GT.0) GOTO 131
    J = J + 1
    IF (MOD((O(2)+O(1)+6+I),18).EQ.0.AND.ST(2:2).NE.'C') THEN
    READ (*,2,END=1402) ST(2:)
    IF (ST(2:2).EQ.'E') RETURN
1402 CLOSE (2)
    OPEN (2,FILE='INPUT')
    END IF
    WRITE(6,1403) IPI(I)
1403 FORMAT(' PUMP COEFFICIENTS FOR PUMP ',I4)
    PRINT*, '          Q*Q          Q CONSTANT'
    PRINT 115,CP(I),DI(I),HW(I)
115 FORMAT (F11.4,F10.4,F10.1,/)
131 CONTINUE
    IF (IPM.EQ.0) GOTO 135
    DO 133 I = 1, 15
    IF (MOD((O(2)+O(1)+J+6+I),18).EQ.0.AND.ST(2:2).NE.'C')
    >THEN
    READ (*,2,END=1502) ST(2:)
    IF (ST(2:2).EQ.'E') RETURN
1502 CLOSE (2)
    OPEN (2,FILE='INPUT')
    GOTO 135
    END IF
133 CONTINUE
```

```
135 PRINT 111
111 FORMAT (/,/)
  2 FORMAT(A60)
  RETURN
  END
C   SUBROUTINE SIMBAL CREATES THE CHARACTERISTICS OF THE
C   SPARSE MATRIX
C   *****
SUBROUTINE SIMBAL(ST)
  INTEGER O,PNL
  REAL*8 A,S,G,HE
  CHARACTER ST*90
  PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
  COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
  COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
  COMMON /TOPOL/
  IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
  1L)
  COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
  COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
  DIMENSION PT(5,20),EF(20),XP(5,PNL),N3(PNL)
C
C *****SPARSE MATRIX 24-6
C *****STEP 1
C
111  FORMAT(/,/)
      C3=448.831
139  ILILI=MNL
      I6=0
C
C   ELEMENTS IN THE ARRAY A FOR LINKS WITH THE SAME BEGINNING
C   AND ENDING NODE ARE COMBINED
C
      CALL PARALE (A,XL,O,IBI,IEI)
      PRINT 111
C
C   PROGRAM TESTS WHETHER THE SPARSE MATRIX ROUTINE HAS BEEN
C   EXECUTED (O(4)=1)
C
      IF (O(4).GT.0) THEN
        PRINT 498,PRAC,FLAC*C3
        PRINT 499
        M0=1000
        CALL EQSOLV(ST,0,I6,XP,P8,P88,M0,PT,EF,TP,N3,O4)
        IF (I6.EQ.1000) THEN
          I6=1
          GOTO 315
        ENDIF
        RETURN
      ENDIF
C
C   THE TOTAL NUMBER OF ENTRIES IN THE COEFFICIENT, O(3),
C   IS SET EQUAL TO THE NUMBER OF LINKS, O(1)
C
      O(3)=O(1)
C
C   THE ARRAY M IS SET TO ZERO
C
      DO 140 I=1,O(2)
        M(I)=0
```



```
140  CONTINUE
C
C  NON-ZERO ENTRIES ARE ASSIGNED
C
DO 200 I=1,O(1)
IF (ABS(A(I)).LT.1E-10.AND.XL(I).LT.1000000.) GOTO 200
C
C  THE BEGINNING (IB1) AND ENDING (IE1) NODE NUMBERS (INTERNAL)
C  ARE DETERMINED FOR PIPE I.  IB1 REPRESENTS THE LINE NUMBER
C  IN THE COEFFICIENT MATRIX AND IE1 REPRESENTS THE COLUMN
C  NUMBER.
C
IB1=IBI(I)
IE1=IEI(I)
C
C  IB1 AND IE1 ARE SWITCHED IN CASE THAT IB1>IE1 SINCE THE
C  PROGRAM OPERATES ONLY IN THE UPPER TRIANGLE OF THE MATRIX
C
IF (IB1.LT.IE1) GOTO 150
IT=IB1
IB1=IE1
IE1=IT
C
C  THE PRESENT VALUES OF ARRAYS N1, AND N2 ON LINE IB1 ARE
C  TEMPORARILY ASSIGNED TO ARRAYS S AND G, RESPECTIVELY.
C
150  DO 160 I5=1,M(IB1)
S(I5)=N1(IB1,I5)
G(I5)=N2(IB1,I5)
160  CONTINUE
C
C  ENTRY COUNTER M IS INCREMENTED FOR LINE IB1
C
M(IB1)=M(IB1)+1
C
C  NEW VALUES ARE ASSIGNED TO ARRAYS S (THE LINK NUMBER I)
C  AND G (THE ENDING NODE NUMBER/COLUMN NUMBER)
C
S(M(IB1))=I
G(M(IB1))=IE1
C
C  LAST ENTRY IS INSERTED INTO THE PROPER POSITION IN ARRAYS
C  G SUCH THAT THE COLUMN NUMBERS IN ARRAY G INCREASE FROM
C  1 THROUGH THE LAST POSITION M(IB1).
C
DO 170 I3=0,M(IB1)-2
I6=S(M(IB1)-I3-1)
I7=G(M(IB1)-I3-1)
IF (IE1.GT.I7) GOTO 180
S(M(IB1)-I3)=S(M(IB1)-I3-1)
S(M(IB1)-I3-1)=I
G(M(IB1)-I3)=G(M(IB1)-I3-1)
G(M(IB1)-I3-1)=IE1
170  CONTINUE
C
C  MODIFIED ARRAYS S AND G ARE AGAIN ASSIGNED TO THE PROPER
C  LINES IN ARRAYS N1 AND N2, RESPECTIVELY.
C
180  DO 190 I5=1,M(IB1)
N1(IB1,I5)=S(I5)
```

```
      N2( IB1, I5)=G( I5)
190  CONTINUE
200  CONTINUE
C
C *****STEP 2
C     ZERO ENTRIES ARE ESTABLISHED IN ARRAYS N1 AND N2 WHICH
C     WILL BE NEEDED FOR THE GAUSSIAN ELIMINATION PROCEDURE - THE
C     DUMMY LINKS
C     FIRST DO LOOP ASSIGNS THE LINE NUMBER I IN THE MATRIX
C
      DO 310 I=1,O(2)-1
C
C     SECOND DO LOOP ASSIGNS THE ENTRY NUMBER I1 ON THE LINE
C
      DO 300 I1=1,M(I)
      I6=N2( I, I1)
C
C     ENTRY NUMBERS ON LINE I OF THE MATRIX ARE ASSIGNED WITH DO LOOP
C
      DO 290 I2=I1+1,M(I)
      IB=N2( I, I2)
      DO 210 I3=1,M(I6)
      S( I3)=N1( I6, I3)
      G( I3)=N2( I6, I3)
210  CONTINUE
C
C     ENTRY IS TESTED TO SEE WHETHER IT NEEDS TO BE CREATED BY
C     COMPARING COLUMN NUMBERS ON MATRIX LINES I6 AND I
C
      DO 220 I3=1,M(I6)
      IE=G( I3)
      IF ( IE.EQ.IB) GOTO 290
      IF ( IE.GT.IB) GOTO 230
220  CONTINUE
C
C     CREATION OF ENTRY
C
230  M( I6)=M( I6)+1
      IF ( M( I6).EQ.( ILLI+1)) THEN
      PRINT*, 'ALLOCATED MEMORY CAPACITY EXCEEDED, TRY INCREASING'
      PRINT*, 'MNL IN ALL PARAMETER STATEMENTS OR'
      PRINT*, 'RENUMBERING THE SYSTEM.'
      RETURN
      END IF
      DO 240 I3=1,M(I6)-1
      I7=M( I6)-I3
      I8=G( I7)
      IF ( I8.LT.IB) GOTO 250
      S( I7+1)=S( I7)
      G( I7+1)=G( I7)
240  CONTINUE
250  O( 3)=O( 3)+1
      IF ( O( 3).LT.IA+1) GOTO 270
260  PRINT *, 'EXCEEDS ALLOCATED MEMORY CAPACITY TRY TO INCREASE'
      PRINT*, 'IA IN THE PARAMETER STATEMENTS'
      PRINT *, I6, M( I6), O( 3)
      RETURN
C
C     VALUES CORRESPONDING TO ENTRY, THE ARRAYS S AND G ARE ASSIGNED
C
```

```
270 S(M(I6)-I3+1)=O(3)
    G(M(I6)-I3+1)=IB
C
C   ARRAYS S AND G ARE ASSIGNED BACK TO LINE I6 OF ARRAYS N1
C   AND N2, RESPECTIVELY.
C
    DO 280 I3=1,M(I6)
    N1(I6,I3)=S(I3)
    N2(I6,I3)=G(I3)
280 CONTINUE
C
C   VALUE OF ZERO IS ASSIGNED TO THE COEFFICIENT IN THE MATRIX
C   FOR ELEMENT JUST CREATED.
C
    A(O(3))=0
290 CONTINUE
300 CONTINUE
310 CONTINUE
C
C   O(4)=1 SERVES AS A FLAG TO INDICATE THAT THE SPARSE MATRIX
C   ROUTINE HAS BEEN EXECUTED (ARRAYS N1, N2, AND M HAVE BEEN
C   ESTABLISHED.
C
    O(4)=1
C
C   DIAGONOL MEMBERS IN THE MATRIX (ARRAY S) AND THE RIGHT HAND
C   SIDE OF THE CONTINUITY EQUATIONS (ARRAY G) ARE INITIALIZED.
C
315 DO 350 I1=1,O(2)
    IF (DO(I1).LT.9E9) GOTO 320
C
C   FOR SUPPLY POINTS, ARRAY ELEMENT S IS SET TO 1E10 AND ARRAY
C   ELEMENT G IS SET TO THE DESIRED HYDRAULIC GRADE LINE ELEVATION
C   MULTIPLIED BY 1E10.
C
    S(I1)=1E10
    G(I1)=DO(I1)+(ABS(EL(I1))-100)*1E10
    GOTO 350
320 S(I1)=0
    G(I1)=-DO(I1)/C3
350 CONTINUE
    IF (O(10).EQ.0) GOTO 500
    DO 360 I2=1,O(1)
    IF (CP(I2).LT.0.0) GOTO 360
    IF (XL(I2).GT.999999.) THEN
C
C   ARRAY ELEMENT A FOR PRV'S IS INITIALLY SET TO 10000 (PRV
C   COMPLETELY OPEN)
C   FLAG O(10) IS ASSIGNED : 1 - CHECK VALVES ONLY
C                           10 - PRVS ONLY
C                           11 - CHECK VALVES AND PRVS
C
    A(I2)=10000
    O(10)=10+MOD(INT(O(10)),10)
    END IF
    IF (XL(I2).LT.0) THEN
C
C   ARRAY ELEMENT A IS SET TO 1/CP IN EFFECT OPENING THE
C   CHECK VALVE
C

```

```
A(I2)=1/CP(I2)
O(10)=1+INT(O(10)/10)*10
END IF
360 CONTINUE
500 I6=0
PRINT 498,PRAC,FLAC*C3
498 FORMAT (' ACCURACY LIMITS: ',F5.1,' PSI; ',F5.1,' GPM',/)
PRINT 499
499 FORMAT (' ESTIMATED MAXIMUM ERRORS:',/)
IFOP=0
CALL EQSOLV(ST,IFOP,I6,XP,P8,P88,M0,PT,EF,TP,N3,O4)
IF (I6.EQ.1000) THEN
  I6 = 1
  GOTO 315
END IF
RETURN
END

C
C ***** SOLUTION OF LINEAR EQUATIONS
C
SUBROUTINE EQSOLV(ST,IFOP,I6,XP,P8,P88,M0,PT,EF,TP,N3,O4)
INTEGER O,PNL
REAL*8 A,S,G,HE,R2,R8
CHARACTER ST*90
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
DIMENSION PT(5,20),EF(20),XP(5,PNL),N3(PNL)
C3 =448.831

C
C DIAGONOL MEMBERS OF THE MATRIX ARE CALCULATED
C
IF (M0.EQ.1000) GOTO 910
505 DO 510 I1=1,O(1)
  IB=IBI(I1)
  IE=IEI(I1)
  S(IB)=S(IB)+A(I1)
  S(IE)=S(IE)+A(I1)
510 CONTINUE
C
C FIRST LOOP ASSIGNS THE LINE NUMBER IN THE MATRIX, I, OF WHICH
C A MULTIPLE WILL BE ADDED TO A LINE FURTHER DOWN.
C
DO 560 I=1,O(2)-1
C
C SECOND LOOP SELECTS THE ENTRY NUMBER ON THE LINE, I1, AND
C INDIRECTLY THE LINE NUMBER, I5, TO WHICH THE MULTIPLE OF
C LINE I IS TO BE ADDED.
C
DO 550 I1=1,M(I)
  I7=N1(I,I1)
  I5=N2(I,I1)
  IF (S(I).EQ.0) GOTO 980
  R8=A(I7)/S(I)
C
```

```
C      NEW DIAGNOL MEMBER ON LINE I5 IS FOUND
C
      S(I5)=S(I5)-R8*A(I7)
      DO 540 I2=I1+1,M(I)
      IB=N2(I,I2)
      DO 520 I3=1,M(I5)
      IE=N2(I5,I3)
      IF (IB.EQ.IE) GOTO 530
520    CONTINUE
530    IB=N1(I,I2)
      IE=N1(I5,I3)
C
C      NEW OFF-DIAGONAL ENTRIES FOR MATRIX LINE I5 ARE CALCULATED
C
      A(IE)=A(IE)+R8*A(IB)
540    CONTINUE
C
C      RIGHT HAND SIDE OF THE EQUATION IS ADJUSTED
C
      G(I5)=G(I5)+R8*G(I)
550    CONTINUE
560    CONTINUE
      P8 = 1.E38
C
C      THE HYDRAULIC HEAD AT NODE O(2) (HIGHEST NODE NUMBER) IS
C      DETERMINED. R3 REPRESENTS THE ABSOLUTE DIFFERENCE BETWEEN
C      PREVIOUS ESTIMATE OF THIS HEAD AND THE NEW HEAD.
C
      IB=O(2)
      IF (S(IB).EQ.0) GOTO 980
      R2=G(IB)/S(IB)
      R3=ABS(HE(IB)-R2)
      R3M=R3
      I3M=IB
      S(IB)=HE(IB)
      HE(IB)=R2
      IF (IFOP.EQ.1) THEN
        IF (DO(IB).GT.1.E10.OR.N3(IB).NE.0) GOTO 562
        P8=(HE(IB)-ABS(EL(IB)))/2.308-XP(M0,IB)
        IW=IB
      END IF
C
C      LINE NUMBER IN THE MATRIX (FROM BOTTOM TO TOP) IS ASSIGNED
C
562    DO 580 I=IB-1,1,-1
      R2=G(I)
C
C      TERMS FOR WHICH THE HEADS ARE ALREADY KNOWN ARE ACCUMULATED
C      (VARAIBLE R2)
C
      DO 570 I1=1,M(I)
      IE=N1(I,I1)
      I4=N2(I,I1)
      R2=R2+A(IE)*HE(I4)
570    CONTINUE
      IF (S(I).EQ.0) GOTO 980
C
C      VARIABLE R2 IS REASSIGNED TO THE NEW HEAD AT NODE I
C
      R2=R2/S(I)
```

```
C
C     ABSOLUTE DIFFERENCE BETWEEN THE PREVIOUS ESTIMATE OF
C     THE HEAD AND THE NEW VALUE OF THE HEAD IS ACCUMULATED
C     IN VARIABLE R3
C
R3=R3+ABS(HE(I)-R2)
C
C     LARGEST ABSOLUTE DIFFERENCE BETWEEN NEW AND OLD HEAD AND THE
C     CORRESPONDING NODE NUMBER, R3M AND I3M RESPECTIVELY ARE
C     KEPT TRACK OF.
C
IF (ABS(HE(I)-R2).GT.R3M) THEN
  R3M=ABS(HE(I)-R2)
  I3M=I
END IF
S(I)=HE(I)
HE(I)=R2
IF (IFOP.EQ.1) THEN
  IF (DO(I).GT.1.E10) GOTO 580
IF ((R2-ABS(EL(I)))/2.308-XP(M0,I).GT.P8.OR.N3(I).NE.0) GOTO 580
  P8 = (R2-ABS(EL(I)))/2.308-XP(M0,I)
  IW=I
END IF
580  CONTINUE
C
C *****OVERRELAXATION 24-7B
C
IF (IFOP.LE.0) THEN
C
C     OVERRELAXATION IS PERMITTED ONLY IF THERE ARE NO PRVS NOR
C     CHECK VALVES IN THE SYSTEM (O(10)=0), IF THE SUM OF ALL ABSO
C     CORRECTIONS, R3, IS LESS THAN 75% OF THE SUM IN THE PREVIOUS
C     ITERATION (C4), AND IF THE ITERATION COUNTER I6 IS LARGER THAN
C     OR EQUAL 2 AND LESS THAN OR EQUAL TO 10.
C
IF (O(10).GT.0.OR.R3.GT..75*C4.OR.I6.LT.2.OR.I6.GT.10) GOTO 600
C
C     DO LOOP USES THE OLD HEAD, S, AND THE NEW HEAD, HE, TO
C     CALCULATE THE OVERRELAXED HEAD HE
C
DO 590 I=1,O(2)
  HE(I)=1.85*HE(I)-.85*S(I)
590  CONTINUE
  R3=R3/2
END IF
C
C     ITERATION COUNTER IS INCREMENTED
C
600  I6=I6+1
  C4=R3
C
C *****REEVALUATION OF COEFFICIENT MATRIX 24-7C
C
O(9)=0
DO 610 I1=O(1)+1,O(3)
C
C     COEFFICIENT MATRIX FOR THE DUMMY LINKS IS RESET TO 0
C
610  A(I1)=0
  DO 620 I1=1,O(2)
```

```
C
C   THE NET OUTFLOW IS INITIALIZED WITH THE DOMESTIC LOAD
C   IN CFS.   AT SUPPLY POINTS, THE OUTFLOW IS INITIALIZED
C   WITH ZERO
C
C   S(I1)=-DO(I1)/448.831
620  IF (DO(I1).GT.9E9) S(I1)=0
      TP = 0
      IPMP=0
      DO 790 I1=1,O(1)
C
C   INTERNAL NODE NUMBERS OF THE BEGINNING AND ENDING NODES
C   OF THE LINK ARE DETERMINED, IB AND IE, RESPECTIVELY.
C
      IB=IBI(I1)
      IE=IEI(I1)
C
C   HEAD DIFFERENCE BETWEEN ENDING AND BEGINNING NODE, S1
C   IS CALCULATED. (POSITIVE IF FLOW GOES FROM BEGINNING
C   NODE TO ENDING NODE)
C
      S1=HE(IE)-HE(IB)
      IF (CP(I1).GT.0.0) GOTO 740
C
C *****PUMPS*****
C
      IPMP=IPMP+1
      IF (IFOP.EQ.-1.AND.XL(I1).LT.0) THEN
        A(I1)=0
        D=0
        GOTO 780
      ENDIF
      Y1=CP(I1)
      Y2=DI(I1)
      Y3=HW(I1)
      IF (S1.LT.0.OR.S1.GT.Y3) O(9)=-2
      IF (S1.GT.0) GOTO 660
      IF (I6.LT.3) GOTO 650
C
C   IF HEAD DIFFERENCE, S1, IS NEGATIVE, AND IF THE ITERATION
C   COUNTER IS LARGER THAN 2, THIS WARNING IS DISPLAYED
C
      IF (IFOP.LE.0) THEN
        PRINT *, 'PUMP ', IPI(I1), ' CANNOT DELIVER DISCHARGE.'
      END IF
      IF (I6.EQ.ICL) GOTO 690
C
C   S1 IS SET TO 5% OF Y3, SINCE A NEGATIVE S1 IS NOT POSSIBLE.
C
650  S1=.05*Y3
      GOTO 720
660  IF (S1.LE.Y3) GOTO 720
      IF (I6.LT.3) GOTO 690
C
C   IF HEAD DIFFERENCE S1 IS LARGER THAN Y3 (THE PUMP HEAD AT
C   A FLOW OF ZERO), AND IF THE ITERATION COUNTER IS LARGER
C   THAN 2, THIS WARNING IS PRINTED.
C
      IF (IFOP.LE.0) THEN
        WRITE(6,689) IPI(I1)
```

```
689   FORMAT(' PUMP ',I4,' CANNOT DELIVER HEAD. ')
      END IF
690   IF (I6.LT.ICL) GOTO 700
      O(9) = IPI(I1)
      IF (IFOP.LE.0) THEN
        PRINT *, ' YOU MUST CHANGE PUMP. '
        IFOP=999
      END IF
      RETURN

C
C     S1 IS SET TO A VALUE BETWEEN 80% AND 100% OF Y3,
C     DEPENDING ON THE ITERATION NUMBER I6, SINCE S1>Y3 IS
C     NOT POSSIBLE.
C
700   S1=(.8+FLOAT(I6)/FLOAT(ICL)*.2)*Y3
C
C     THE NEW ENTRY IN THE COEFFICIENT MATRIX, A IS CALCULATED
C     AS WELL AS THE FLOW RATE THROUGH THE PUMP, D.
C
720   Y0=Y2**2-4*Y1*(Y3-S1)
      A(I1)=1.85/SQRT(Y0)
      D=-((1.85/A(I1)+Y2)/2)/Y1
      IF (IFOP.LE.0) GOTO 780
      XL(I1)=ABS(D)
      IF (N3(IE).EQ.0) TP=TP+ABS(D)*ABS(S1)*PT(M0,IPMP)/EF(IPMP)
      GOTO 780
C *****CHECK VALVES*****
740   IF (XL(I1).GT.0) GOTO 750
      IF (S1.LT.0) GOTO 750
      A(I1)=0
      GOTO 790
750   IF (XL(I1).GT.999999.) GOTO 760
      T0=ABS(S1)
C
C     FLOW RATE, D, IN REGULAR PIPES (CHECK VALVE OPEN) IS CALCULATED
C
      D=-((T0/CP(I1))**.54*SIGN(1.,S1)
      IF (ABS(D).GT..0001) GOTO 770
C
C     IN CASE OF PRVS, AND PIPES WITH FLOW RATES OF LESS THAN
C     0.0001 CFS, THE MEMBER OF THE A ARRAY IS SET TO 1 E-6
C
760   A(I1)=.000001
      GOTO 790
C
C     NEW ENTRY IN THE COEFFICIENT MATRIX FOR THE LINK IS CALCULATED
C
770   A(I1)=(T0/CP(I1))**.54/T0
C
C     OUTFLOW IS ACCUMULATED AT THE BEGINNING AND ENDING NODE
C     OF THE LINK.
C
780   S(IB)=S(IB)-D
      S(IE)=S(IE)+D
790   CONTINUE
      X8 = R3/O(2)
C
C     DO LOOP CALCULATES THE RIGHT HAND SIDE MEMBER OF THE
C     CONTINUITY EQUATION, G, AS IT WILL BE USED IN THE NEXT
ITERATION
```



```
C
DO 860 I1=1,0(2)
IF (DO(I1).LT.9E9) GOTO 800

C
C   G FOR CONSTANT HEAD NODES IS DETERMINED
C
G(I1)=DO(I1)+(ABS(EL(I1))-100)*1E10
IF (DO(I1).GT.1E20) G(I1)=HE(I1)*1E10
GOTO 860

C
C   G FOR REGULAR NODES IS ASSIGNED
C
800 G(I1)=-DO(I1)/C3
IF (EL(I1).GE.0) GOTO 860

C
C   PROGRAM SEARCHES FOR LINKS WITH BEGINNING OR ENDING
C   NODE NUMBER EQUAL TO THE NODE PRESENTLY UNDER CONSIDERATION
C
DO 850 I2=1,0(1)
IB=IBI(I2)
IE=IEI(I2)
IF (CP(I2).GT.0) GOTO 820
IF (A(I2).EQ.0) GOTO 850
Y4=(-DI(I2)-1.85/A(I2))/(2*CP(I2))
IF (IB.EQ.I1) GOTO 810
IF (IE.NE.I1) GOTO 850

C
C   G IS UPDATED IN CASE A PUMP IS ADJACENT
C
G(I1)=G(I1)+A(I2)*(HE(I1)-HE(IB))+Y4
GOTO 850

C
C   G IS UPDATED IN CASE A PUMP IS ADJACENT
C
810 G(I1)=G(I1)-A(I2)*(HE(IE)-HE(I1))-Y4
GOTO 850

C
C*****EVALUATION OF PRV STATUS*****
C
820 IF (XL(I2).LT.1000000.) GOTO 850
IF (IB.NE.I1.AND.IE.NE.I1) GOTO 850
HSET=-EL(IE)+HW(I2)*2.308
IF (IB.EQ.I1) GOTO 840

C
C   EVALUATING PRV STATUS BASED ON DOWNSTREAM NODE
C
C   THE A VALUE FOR THE PRV IS SET TO ZERO, CLOSING THE PRV
C
A(I2)=0
IF ((HE(IE).GT.HSET+.01).AND.ABS(HE(IB)-HE(IE)).LT..01) GOTO 829
IF (HE(IE).GT.HSET+.01) GOTO 850
IF (HE(IB).LT.HSET-.01.AND.ABS(HE(IE)-
HSET).LT..01.AND.S(IE).LT.0)
1GOTO 830
IF (HE(IE).GT.HE(IB)) GOTO 850
IF (HE(IB).LT.HSET-.01) GOTO 830
IF ((S(IE).GE.-.001).AND.(ABS(HE(IE)-HSET).LT..01)) GOTO 850

C
C   IN CASE A PRV WHICH REDUCES THE PRESSURE, THE G VALUE
C   IS SET AS FOR A SUPPLY POINT AT THE DOWNSTREAM NODE
```

```
C      PRV IS SET TO ACTIVE STATE
C
829    G(IE)=HSET*1E10
      GOTO 850
C
C      IN CASE OF PRVS WHICH ARE COMPLETELY OPEN (UPSTREAM PRESSURE>
C      PRESSURE SETTING AND NO REVERSE FLOW), THE A VALUE IS SET TO
C      10000, AND THE G VALUES AT THE TWO ENDS NEED NO ADJUSTMENT
C
830    A(I2)=10000
      GOTO 850
C
C      EVALUATING THE PRV STATUS BASE ON THE UPSTREAM NODE
C
840    IF ((HE(IE).GT.HSET+.01).AND.ABS(HE(IB)-HE(IE)).LT..01) GOTO 849
      IF (HE(IE).GT.HSET+.01) GOTO 850
      IF (HE(IB).LT.HSET-.01.AND.ABS(HE(IE)-
HSET).LT..01.AND.S(IE).LT.0)
1GOTO 850
      IF (HE(IE).GT.HE(IB)) GOTO 850
      IF (HE(IB).LT.HSET-.01) GOTO 850
      IF (S(IE).GE.0) GOTO 850
C
C      G VALUE IS INCREMENTED BY THE NET OUTFLOW AT THE DOWNSTREAM
C      NODE
C
849    G(IB)=G(IB)+S(IE)
850    CONTINUE
860    CONTINUE
C
C *****ACCURACY CHECK/TERMINATION
C      LOOP SEARCHES ALL LINKS FOR PRVS AND COMBINES THE NET
C      OUTFLOW AT THE BEGINNING AND ENDING NODES OF EACH PRV.
C
      IF (IFOP.EQ.1) GOTO 10
      DO 950 I1=1,O(1)
      IF (XL(I1).LT.1000000.) GOTO 950
      IB=IBI(I1)
      IE=IEI(I1)
      S(IB)=S(IB)+S(IE)
      S(IE)=S(IB)
950    CONTINUE
      R4M=0.
C
C      LOOP DETERMINES THE MAXIMUM FLOW IMBALANCE IN THE SYSTEM (R4M)
C      AND THE CORRESPONDIN NODE NUMBER (I4M)
C
      DO 960 I1=1,O(2)
      IF (DO(I1).GT.9E9) GOTO 960
      IF (ABS(S(I1)).GT.R4M) THEN
      R4M=ABS(S(I1))
      I4M=I1
      END IF
960    CONTINUE
C
C      PRINTING OF ITERATION NUMBER, MAXIMUM PRESSURE ADJUSTMENT
C      WITH CORRESPONDING NODE NUMBER, AND MAXIMUM FLOW IMBALANCE
C      WITH CORRESPONDING NODE NUMBER.
C
      IF (IFOP.EQ.-1) GOTO 503
```

```
      IF (R3M/2.308.LT.100000..AND.R4M*C3.LT.1000000..AND.I4M.GT.0)
THEN
  PRINT 501,I6,R3M/2.308,INO(I3M),R4M*C3,INO(I4M)
  ELSE
  PRINT 502,I6
  END IF
501  FORMAT (' ITERATION #',I3,' :',F8.1,' PSI AT NODE',I5,';',F8.0,'
G
1PM AT NODE',I5)
502  FORMAT (' ITERATION #',I3)
503  IF (I6.LT.3) GOTO 910
C
C    ITERATIVE SCHEME IS TERMINATED IF MAXIMUM PRESSURE ADJUSTMENT
C    (R3M/2.308) AND MAXIMUM FLOW IMBALANCE (R4M) ARE LESS THAN
C    THE VALUES SPECIFIED UNDER THE KEYWORD ACCU.
C
      IF (R3M/2.308.LT.PRAC.AND.R4M.LT.FLAC.AND.O(9).EQ.0.AND.I6.GT.2)
G
1OTO 965
      IF (R3M/2.308.LT.PRAC.AND.R4M.LT.FLAC.AND.O(9).LT.0.AND.I6.GT.2)
G
1OTO 965
      IF (I6.GT.ICL.AND.O(10).GT.100) GOTO 930
C
C    AN EXIT IS PROVIDED IN CASE OF NON-CONVERGENCE
C
      IF (I6.GT.20.AND.O(10).GT.9.AND.O(10).LT.12) THEN
C
C    IN CASE OF NON CONVERGENCE, THE PROGRAM IS CAUSED TO START
C    OVER AGAIN, WITH NO OVERREALXATION IN THE SECOND TIME THROUGH
C    THE ITERATIVE SCHEME.
C
      O(10)=O(10)+100
      I6=1000
      RETURN
      END IF
C
C    THE S ARRAY IS INITIALIZED
C
910  IF (I6.GE.ICL) GOTO 930
      DO 920 I1=1,O(2)
      S(I1)=0
      IF (G(I1).LT.9.9E7) GOTO 920
      S(I1)=1E10
920  CONTINUE
C
C    THE ELEMENTS IN THE A ARRAY FOR LINKS WITH SAME BEGINNING
C    AND ENDING NODE ARE COMBINED.
      CALL PARALE (A,XL,O,IBI,IEI)
C
C    CONTROL IS TRANSFERRED BACK TO NEXT ITERATION
C
      GOTO 505
C
C    PRINTING OF MESSAGES IN CASE THE ITERATIVE SCHEME IS
C    TERMINATED BECAUSE COMPUTATIONS HAVE CONVERGED OR BECAUSE
C    OF NON-CONVERGENCE.
C
930  PRINT 931
931  FORMAT (/, ' SYSTEM MAY NOT BE PROPERLY BALANCED !',/)
```

```
O(9)=1
GOTO 970
965 IF (IFOP.EQ.-1) RETURN
PRINT 961
961 FORMAT (/, ' SYSTEM IS BALANCED' ,/)
O(9)=3
970 O(6)=I6
GOTO 989
980 PRINT*, ' DIVISION BY ZERO WHILE SOLVING THE LINEARIZED
EQUATIONS.'
WRITE(6,981) I
981 FORMAT(' CHECK DATA AROUND NODE',I4,' FOR POSSIBLE ERRORS.')
RETURN
989 IF (IFOP.EQ.-1) THEN
  IFOP=999
  RETURN
ELSE
  CALL PRNOUT(ST)
ENDIF
RETURN
10 DO 20 I1=1,O(2)
  S(I1)=0
  IF (G(I1).LT.9.9E7) GOTO 20
  S(I1)=1E10
20 CONTINUE
  IF (I6.GT.15) THEN
    O(9)=.5
    GOTO 50
  END IF
  IF (O(12).LT.-1.9) GOTO 30
  IF (IW0.NE.IW) GOTO 940
  IF (I6.LT.2) GOTO 940
  IF (X8.GT.ABS(P8/2)) GOTO 30
  IF (X8.GT.ABS(P8/2)) GOTO 30
  IF (ABS(P88-P8).LT.X8/4) GOTO 30
928 IF (P88*P8.LE.0) GOTO 30
  IF (ABS(P8).GT.ABS(P88).AND.ABS(P88-
P8).LT.ABS(P8).AND.O(9).EQ.0)
  1GOTO 50
  IF (ABS(P8).LT.ABS(P88).AND.ABS(P88-
P8).LT.ABS(P8/5).AND.O(9).EQ.0
  1) GOTO 50
30 IF (X8.LT.04.AND.O(9).EQ.0) GOTO 50
940 P88=P8
  IW0=IW
  CALL PARALE(A,XL,O,IBI,IEI)
  GOTO 505
50 P88=P8
  RETURN
END

C
C *****OUTPUT - PRNOUT *****
C PRINTS NODE AND LINK TABLES AFTER A SYSTEM IS BALANCED
SUBROUTINE PRNOUT(ST)
CHARACTER JOB*60,ST*90
REAL*8 A,S,G,HE
INTEGER PNL,O
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
```

```
COMMON /TOPOL/
IBE(PNL), IEN(PNL), IPI(PNL), INO(PNL), IBI(PNL), IEI(PN
1L)
COMMON /JOB/ JOB
COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
990 O(7)=-1
C PRINTING OF WARNING MESSAGES IF THE NETWORK IS NOT PROPERLY
C BALANCED.
IF (O(9).GT.0) GOTO 1000
PRINT 1001
1001 FORMAT (/, ' THIS SYSTEM IS NOT BALANCED',/)
RETURN
1000 IF (O(9).GT.1) GOTO 1020
PRINT 1002,ICL
1002 FORMAT (' THIS SYSTEM WAS NOT PROPERLY BALANCED AFTER',I4,'
ITERATI
1ONS')
1020 C3=448.831
C
C*****PRINTING OF NODE TABLE*****
C
PRINT 101
PRINT 102,JOB
IP=1
PRINT 1005,IP
1005 FORMAT (9X,' NODE DATA',45X,'PAGE ',I3)
CALL NOTAHE
L0=11
DO 1080 I1=1,O(2)
C
C *****PAGE CONTROL*****
C
IF (L0.LT.61.OR.O(2)-I1.LE.3) GOTO 1040
PRINT 1111
IP=IP+1
DO 1030 K=L0+1,66
1030 PRINT *
IF (ST(2:2).NE.'C') THEN
C
C USER IS ALLOWED TO SWITCH TO 'CONTINUOUS' MODE BY ENTERING A
C C, OR TO EXIT THE NODE TABLE PRINTING BY ENTERING A E
C ANY OTHER ENTRY WILL CONTINUE THE PAUSE MODE AT THE END OF
C EACH PAGE
C
READ (*,2,END=1031) ST(2:)
IF (ST(2:2).EQ.'E') GOTO 1091
END IF
GOTO 1032
1031 CLOSE (2)
OPEN (2,FILE='INPUT')
1032 PRINT 102,JOB
PRINT 1112,IP
L0=9
CALL NOTAHE
C
C INCREMENTING OF LINE COUNTER
C
1040 L0=L0+1
C
C DETERMINATION OF USER NODE NUMBER
```

```
C
      IT=INO(I1)
      ELE=ABS(EL(I1))
      H1=HE(I1)-ELE
C
C      CONTROLLING OF PRINT FORMATS FOR REGULAR NODE WITH OR WITHOUT
C      OUTPUT, TANK NODE, AND RESERVOIR NODE.
C
      IF (DO(I1).GT.9E9) GOTO 1060
      IF (ABS(DO(I1)).GT.0.5) GOTO 1050
C
C      PRINTING OF REGULAR NODE DATA FOR NODES WITHOUT OUTPUT
C
      PRINT 1113,IT,ELE,HE(I1),H1,H1/2.308
      GOTO 1080
C
C      PRINTING OF REGULAR NODE DATA FOR NODES WITH OUTPUT
C
1050 PRINT 1114,IT,ELE,DO(I1),HE(I1),H1,H1/2.308
      GOTO 1080
1060 IF (ABS(H1).GT.0.5) GOTO 1070
C
C      PRINTING OF NODE DATA FOR A RESERVOIR, LISTING THE FLOW
C      INTO THE NETWORK AS A NEGATIVE OUTPUT
C
      PRINT 1115,IT,ELE,S(I1)*C3,HE(I1)
      GOTO 1080
C
C      PRINTING OF NODE DATA FOR A TANK, LISTING THE FLOW INTO THE
C      NETWORK AS A NEGATIVE OUTPUT.
1070 PRINT 1116,IT,ELE,S(I1)*C3,HE(I1),H1,H1/2.308
1080 CONTINUE
      PRINT 1111
1111 FORMAT (5X,'I-----')
--
      1-----I')
1112 FORMAT (9X,'NODE DATE (CONTINUED)',33X,'PAGE ',I3)
1113 FORMAT (5X,'I',I7,F11.1,10X,3F9.1,10X,'I')
1114 FORMAT (5X,'I',I7,F11.1,F10.0,3F9.1,10X,'I')
1115 FORMAT (5X,'I',I7,F11.1,F10.0,F9.1,19X,'RESERVOIRI')
1116 FORMAT (5X,'I',I7,F11.1,F10.0,3F9.1,3X,'SUPPLY I')
      IF (O(1)+O(2).GT.43) GOTO 1090
C
C*****PRINTING OF PIPE TABLE*****
C
      PRINT 1117
1117 FORMAT (/,/,9X,'PIPE DATA')
      L0=L0+10
      GOTO 1110
C
C      PAGE CONTROL
C
1090 DO 1100 K=L0+1,69
1100 PRINT *
      IF (ST(2:2).NE.'C') THEN
      READ (*,2,END=1092) ST(2:)
      IF (ST(2:2).EQ.'E')RETURN
      END IF
      GOTO 1093
1092 CLOSE (2)
      OPEN (2,FILE='INPUT')
```

```
1093 IP=IP+1
1091 PRINT 102,JOB
      PRINT 1118,IP
1118 FORMAT (9X,'PIPE DATA',45X,'PAGE ',I3)
      L0=9
1110 CALL PITAHE
      DO 1190 I1=1,O(1)
      IF (L0.LT.61.OR.O(1)-I1.LE.3) GOTO 1130
      PRINT 1111
      IP=IP+1
      DO 1120 K=L0+1,67
1120 PRINT *
      IF (ST(2:2).NE.'C') THEN
      READ (*,2,END=1094) ST(2:)
      IF (ST(2:2).EQ.'E') RETURN
      END IF
      GOTO 1095
1094 CLOSE (2)
      OPEN (2,FILE='INPUT')
1095 PRINT 102,JOB
      PRINT 1119,IP
      L0=9
      CALL PITAHE
1130 L0=L0+1
C
C      DETERMINATION OF USER NUMBERS FOR BEGINNING AND ENDING NODE
C      OF THE LINK
C
      IB=IBI(I1)
      IE=IEI(I1)
      IB1=IBE(I1)
      IE1=IEN(I1)
      IP1=IPI(I1)
      H1=HE(IB)-HE(IE)
      IF (CP(I1).GT.0) GOTO 1140
C
C      CALCULATION AND PRINTING OF DATA FOR PUMPS
C
      Y1=CP(I1)
      Y2=DI(I1)
C
C      IF A(I1)=0 THE PUMP IS OFF, NO FLOW, SO D1=0
C
      IF (A(I1).EQ.0) THEN
      D1=0
      ELSE
      D1=(-Y2-1.85/A(I1))/2/Y1
      END IF
      P9=ABS(H1)*D1/8.814
      PRINT 1127,IP1,IB1,IE1,ABS(H1),D1*C3,P9
      GOTO 1190
1140 IF (XL(I1).GT.0) GOTO 1145
C
C      PRINTING OF INFORMATION FOR CLOSED CHECK VALVES
C
      IF (H1.GT.0.0) GOTO 1145
      PRINT 1121,IP1,IB1,IE1
      GOTO 1190
1145 IF (XL(I1).LT.1000000.) GOTO 1160
C
```

```
C      PRINTING OF DATA FOR PRVS ACCORDING TO THE STATUS
C      OF THE VALVE.
C
      IF (A(I1).LT.9999) GOTO 1150
      PRINT 1128,IP1,IB1,IE1,HW(I1)
      GOTO 1190
1150  IF (ABS(HE(IE)+EL(IE)-HW(I1)*2.308).LT.0.1) GOTO 1155
      PRINT 1123,IP1,IB1,IE1,HW(I1)
      GOTO 1190
1155  PRINT 1122,IP1,IB1,IE1,HW(I1)
      GOTO 1190
C
C      *****PRINTING OF PIPE DATA*****
C      CALCULATION OF FLOW RATE
C
1160  D1=(ABS(H1)/CP(I1))**.54
C
C      DETERMINATION OF FLOW DIRECTION
C
      IF (H1.GE..0) GOTO 1170
      IT=IE1
      IE1=IB1
      IB1=IT
C
C      CALCULATION OF VELOCITY
C
1170  IF (DI(I1).LE.1E-5) THEN
      PRINT 1165,IP1,IB1,IE1,DI(I1)*12,XL(I1)
      GOTO 1190
      END IF
      VEL=1.273*D1/(DI(I1))**2
      IF (XL(I1).LT.0.0) GOTO 1180
      IF (HW(I1).LT.0) GOTO 1175
C
C      PRINTING OF PIPE DATA FOR PIPES WITH A NON-DEFAULT FRICTION
C      COEFFICIENT
C
      PRINT 1124,IP1,IB1,IE1,DI(I1)*12,XL(I1),HW(I1),D1*C3,VEL,ABS(H1)
      GOTO 1190
C
C      PRINTING OF PIPE DATA FOR PIPES WITH THE DEFAULT VALUE FOR THE
C      FRICTION COEFFICIENT (MARKED BY A *).
C
1175  PRINT 1125,IP1,IB1,IE1,DI(I1)*12,XL(I1),-
      HW(I1),D1*C3,VEL,ABS(H1)
      GOTO 1190
1180  IF (HW(I1).LT.0) GOTO 1185
C
C      PRINTING OF DATA FOR A PIPE WITH AN OPEN CHECK VALVE AND A
C      A NON-DEFAULT FRICTION COEFFICIENT
C
      PRINT 1126,IP1,IB1,IE1,DI(I1)*12,-
      XL(I1),HW(I1),D1*C3,VEL,ABS(H1)
      GOTO 1190
C
C      PRINTING OF DATA FOR A PIPE WITH AN OPEN CHECK VALVE AND THE
C      DEFAULT VALUE FOR THE FRICTION COEFFICIENT
C
1185  PRINT 1129,IP1,IB1,IE1,DI(I1)*12,-XL(I1),-
      HW(I1),D1*C3,VEL,ABS(H1)
```



```
1190 CONTINUE
      PRINT 1111
2      FORMAT(A60)
101     FORMAT(/,/,5X,'PIPE NETWORK ANALYSIS AND OPTIMIZATION',/)
102     FORMAT(' JOB: ',A60,/)
1119    FORMAT (9X,'PIPE DATA (CONTINUED)',33X,'PAGE ',I3)
1121    FORMAT (5X,'I',3I6,3X,'CHECK VALVE',10X,'CLOSED',17X,'I')
1122    FORMAT (5X,'I',3I6,3X,'PRV AT',F7.1,' PSI',4X,'ACTIVE',17X,'I')
1123    FORMAT (5X,'I',3I6,3X,'PRV AT',F7.1,' PSI',4X,'CLOSED',17X,'I')
1124    FORMAT (5X,'I',3I6,F7.1,F9.1,F7.0,F7.0,F6.1,F8.1,3X,'I')
1125    FORMAT (5X,'I',3I6,F7.1,F9.1,F7.0,'*',F6.0,F6.1,F8.1,3X,'I')
1126    FORMAT (5X,'I',3I6,F7.1,F9.1,F7.0,F7.0,F6.1,F8.1,' CVI')
1129    FORMAT (5X,'I',3I6,F7.1,F9.1,F7.0,'*',F6.0,F6.1,F8.1,' CVI')
1127    FORMAT (5X,'I',3I6,3X,'PUMP HEAD',F7.1,' FT ',F6.0,'
POWER',F6.0
1, 'HP I')
1128    FORMAT (5X,'I',3I6,3X,'PRV AT',F7.1,' PSI',4X,'OPEN',19X,'I')
1165    FORMAT (5X,'I',3I6,F7.1,F9.1,' LINK REMOVED ',11X,'I')
      RETURN
      END
C      SUBROUTINE MODIFY EXPANDS THE NODE AND PIPE DATA INTO THE
C      ORIGINAL FORMAT USED IN THE DATA INPUT.
C
      SUBROUTINE MODIFY
      INTEGER PNL,O
      REAL*8 A,S,G,HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
      IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
C
C.....CHECKING IF MAXIMUM NODE OR NUMBER EXCEEDS LIMIT SET BY PNL
C
      IF (O(2).EQ.0.OR.O(1).EQ.0) RETURN
      IF (INO(O(2)).GT.PNL.OR.IPI(O(1)).GT.PNL) RETURN
C
      *****EXPANDING NODE DATA*****
1999 DO 2000 I=O(2),1,-1
C
      COMPUTATION OF THE STARTING POSITION OF THE USER NODE
      DETERMINATION OF THE USER NODE NUMBER, TO BE USED AS SUBSCRIPT
      IN TRANSFERRING THE NODE DATA FROM THE INTERNAL NUMBER TO THE
      USER NUMBER. IF THESE TWO NUMBERS COINCIDE, NO SWITCH TAKES
C
      IB=INO(I)
      IF (IB.EQ.I) GOTO 2000
C
      EXECUTION OF SWITCH
C
      EL(IB)=EL(I)
      DO(IB)=DO(I)
      HE(IB)=HE(I)
C
      RESETTING DATA FOR THE INTERNAL NUMBER TO 0
C
      EL(I)=0
      DO(I)=0
```

```
      HE(I)=0
C
C      ERASING OF USER NODE NUMBER AT THE INTERNAL NODE
C      NUMBER POSITION IN ARRAY INO
C
      INO(I)=0
C
C      FLAGGING OF THE USER NODE NUMBER WITH A 1 IN ARRAY INO
C
2000  INO(IB)=1
C
C*****EXPANDING PIPE DATA
C
      DO 2010 I=O(1),1,-1
C
C      COMPUTATION OF STARTING POSITION OF THE USER LINK
C      NUMBER IN ARRAY IPI
C      DETERMINATION OF USER LINK NUMBER, TO BE USED AS SUBSCRIPT
C      IN TRANSFERRING THE LINK DATA FROM THE INTERNAL LINK NUMBER
C      TO THE USER LINK NUMBER IF THESE TWO NUMBERS COINCIDE,
C      NO SWITCH TAKES PLACE
C
      IB=IPI(I)
      IF (IB.EQ.I) GOTO 2010
C
C      SWITCH IS EXECUTED
C
      DI(IB)=DI(I)
      XL(IB)=XL(I)
      HW(IB)=HW(I)
      CP(IB)=CP(I)
      A(IB)=A(I)
C
C      RESETTING OF DATA FOR THE INTERNAL LINK NUMBER TO 0
C
      DI(I)=0
      XL(I)=0
      HW(I)=0
      CP(I)=0
      A(I)=0
C
C      ERASING OF USER LINK NUMBER AT THE INTERNAL POSITION
C      IN ARRAY IPI, AND USER NODE NUMBERS AT THE INTERNAL
C      NODE NUMBER POSTION IN ARRAYS IBE AND IEN
C
      IBE(IB)=IBE(I)
      IEN(IB)=IEN(I)
      IBE(I)=0
      IEN(I)=0
      IPI(I)=0
C
C      FLAGGING OF THE USER LINK NUMBER WITH A 1 IN ARRAY IPI
C
2010  IPI(IB)=1
      RETURN
      END
C
C***** STORING DATA *****
C
C      STODAT STORES NODE AND LINK DATA FROM THE SIMULATION ROUTINE
```

```
C
SUBROUTINE STODAT
CHARACTER JOB*60,ST*90
INTEGER PNL,O
REAL*8 A,S,G,HE
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
COMMON /PRINT/ IPM,IPP,IFE
COMMON /JOB/ JOB
3000 IF (IPP.EQ.1) PRINT 3001
3001 FORMAT (/, ' ENTER FILE NAME ',/)
C
C READING IN OF USER SELECTED FILE NAME
C
READ (*,2,END=3002) ST
IF (IFE.EQ.1) PRINT*,ST
2 FORMAT(A60)
GOTO 3003
3002 CALL BLANK
RETURN
C
C OPENING FILE ST
C
3003 OPEN(1,ERR=3060,FILE=ST,RECL=63)
IFLGG = O(4)
O(4) = 0
C
C WRITING DATA TO FILE
C
WRITE (1,20) (O(II),II=1,15)
20 FORMAT(15I4)
O(4) = IFLGG
WRITE(1,30,ERR=3060)(HE(II),DO(II),EL(II),II=1,O(2))
30 FORMAT(3E21.14)
WRITE(1,30,ERR=3060)(A(II),CP(II),DI(II),II=1,O(1))
WRITE(1,40,ERR=3060)(XL(II),HW(II),II=1,O(1))
40 FORMAT(2E21.14)
WRITE(1,50,ERR=3060) JOB
50 FORMAT(A60)

WRITE(1,70,ERR=3060)(IBE(J),IEN(J),IBI(J),IEI(J),IPI(J),J=1,O(1))
70 FORMAT(5I4)
WRITE(1,80,ERR=3060) (INO(J),J=1,O(2))
80 FORMAT(10I4)
C
C REWINDING AND CLOSING THE FILE
C
REWIND 1
CLOSE (1)
RETURN
3060 PRINT*, ' PROGRAM CANNOT ACCESS THIS FILE. '
RETURN
END
C
C *****SIMSTO*****
C
```

```
C      SIMSTO STORES A PARTIAL SYSTEM CREATED IN SUBROUTINE INPUT
C
      SUBROUTINE SIMSTO(ST)
      CHARACTER JOB*60,ST*90
      INTEGER O,PNL,00,01,02
      REAL*8 A,S,G,HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /JOB/ JOB
      K=4
5      K = K +1
      IF (ST(K:K).NE.' ') GOTO 5
6      IF (ST(K:K).EQ.' ') THEN
      IF (K.GE.LEN(ST)) THEN
      PRINT*, 'PARTIAL SYSTEM WILL BE STORED ON FILE "SYSDA" '
      ST='SYSDA'
      GOTO 8
      END IF
      K = K +1
      GOTO 6
      END IF
      ST = ST(K:K+15)
C
C      SEARCHING FOR HIGHEST NODE AND LINK NUMBERS
C
8      DO 10 I = 1,PNL
      IF (IPI(I).EQ.0) O1 = I
      IF (INO(I).EQ.0) O2 = I
10     CONTINUE
3021  O0=1000
C
C      OPENING FILE SYSDA ON WHICH PARTIAL SYSTEM DATA WILL BE STORED
C
      OPEN(1,ERR=3060,FILE=ST,RECL=63 )
      REWIND 1
      WRITE (1,15,ERR=3060) O0,(O(J),J=1,14)
15     FORMAT(15I4)
      WRITE(1,20,ERR=3060) O(15),01,02
20     FORMAT(3I4)
      WRITE (1,30,ERR=3060) (HE(J),DO(J),EL(J),J=1,02)
30     FORMAT(3E21.14)
      WRITE (1,30,ERR=3060) (A(J),CP(J),DI(J),J=1,01)
      WRITE(1,40,ERR=3060) (XL(J),HW(J),J=1,01)
40     FORMAT(2E21.14)
      WRITE (1,50,ERR=3060) JOB
50     FORMAT(A60)
      WRITE (1,60,ERR=3060) (IBE(I),IEN(I),IPI(I),INO(I),I=1,02)
60     FORMAT(4I4)
C
C      REWINDING AND CLOSING FILE
C
      REWIND 1
      CLOSE (1)
      RETURN
3060  PRINT*, ' PROGRAM CANNOT ACCESS THIS FILE.'
      PRINT*
```

```

        RETURN
        END
C
C ***** RETRIEVING DATA *****
C DATA FROM A LOCAL FILE, PREVIOUSLY STORED USING SUBROUTINE
C STODAT, IS RETRIEVED
C
        SUBROUTINE RETDAT(IFLG,ST)
        CHARACTER JOB*60,ST*90
        INTEGER O,PNL
        REAL*8 A,S,G,HE
        PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
        COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
        COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
        COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
        COMMON /PRINT/ IPM,IPP,IPE
        COMMON /JOB/ JOB
        IFLG=0
3030 IF (IPP.EQ.1) PRINT 3001
C
C PROMPTING USER AND READING IN RESPONSE
C
3001 FORMAT(/,' ENTER FILE NAME',/)
        READ (*,2,END=3031) ST
        IF (IPE.EQ.1) PRINT*,ST
2        FORMAT(A60)
C
C OPENING FILE
C
        OPEN(1,ERR=3060,FILE=ST,STATUS='OLD',RECL=63)
        REWIND 1
        READ(1,20,ERR=3060) (O(II),II=1,15)
        IF (O(1).EQ.1000) THEN
            PRINT*, ' THIS FILE WAS CREATED AS A BACKUP FILE DURING THE '
            PRINT*, ' MODIFY MODE OF THE SIMULATION ROUTINE. '
            PRINT*, ' AFTER READING THE FILE, CONTROL RETURNS TO THE INPUT
1PROMPT. '
            REWIND 1
            CLOSE (1)
            CALL SIMRET(IFLG,ST)
            RETURN
        END IF
20        FORMAT(15I4)
C
C...CHECKING IF SYSTEM HAS MORE NODES OR LINKS THAN ALLOWED BY LNL
C
        IF (O(1).GT.LNL.AND.O(1).GE.O(2)) THEN
            WRITE(6,23) ' LINKS', ' LINKS',O(1)
23        FORMAT(' CANNOT RETRIEVE SYSTEM - TOO MANY',A,'. TO RETREIVE,'
>,' INCREASE '/' LNL IN SOURCE CODE TO MAXIMUM NUMBER OF',A,I4)
            RETURN
        ELSE IF (O(2).GT.LNL) THEN
            WRITE(6,23) ' NODES', ' NODES',O(2)
            RETURN
        END IF
C
C REINITIALIZING VARIABLES
C

```

```
DO 9 I = 1, PNL
  IBE(I) = 0
  IEN(I) = 0
  IPI(I) = 0
  INO(I) = 0
  IBI(I) = 0
  IEI(I) = 0
  HE(I)   = 0
  DO(I)   = 0
  EL(I)   = 0
  HW(I)   = 0
9  CONTINUE
  READ(1,30,ERR=3060,END=3060) (HE(II),DO(II),EL(II),II=1,O(2))
30  FORMAT(3E21.14)
  READ(1,30,ERR=3060,END=3060) (A(II),CP(II),DI(II),II=1,O(1))
  READ(1,40,ERR=3060,END=3060) (XL(II),HW(II),II=1,O(1))
40  FORMAT(2E21.14)
  READ(1,50,ERR=3060,END=3060) JOB
50  FORMAT(A60)
  READ(1,60,ERR=3060,END=3060) (IBE(J),IEN(J),IBI(J),IEI(J),
> IPI(J),J=1,O(1))
60  FORMAT(5I4)
  READ(1,70,ERR=3060,END=3060) (INO(J),J=1,O(2))
70  FORMAT(10I4)
C
C  REWINDING AND CLOSING FILE
C
3053 REWIND 1
      CLOSE (1)
      IFLG=0
      RETURN
C
C  ERROR MESSAGE IF FILE IS NOT ACESIBLE
C
3060 PRINT*, ' PROGRAM CANNOT ACCESS THIS FILE.'
      PRINT*
      RETURN
3031 CALL BLANK
      RETURN
      END
C
C  *****SIMRET*****
C
SUBROUTINE SIMRET(IFLG,ST)
CHARACTER JOB*60,ST*90
INTEGER O,PNL,O0,O1,O2
REAL*8 A,S,G,HE
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
COMMON /JOB/ JOB
K=3
5  K = K +1
   IF (ST(K:K).NE.' ') GOTO 5
6  IF (ST(K:K).EQ.' ') THEN
   IF (K.GE.LEN(ST)) THEN
     PRINT*, 'PARTIAL SYSTEM WILL BE READ FROM FILE "SYSDA"'
```

```

        ST='SYSDA'
        GOTO 8
    END IF
    K = K +1
    GOTO 6
    END IF
    ST = ST(K:K+15)
8    OPEN(1,ERR=3102,FILE=ST,STATUS='OLD',RECL=63)
    REWIND 1
    READ (1,20) O0,(O(J),J=1,14)
20   FORMAT(15I4)
    IF (O0.NE.1000) THEN
        PRINT*,' THIS FILE WAS NOT CREATED AS A BACKUP FILE DURING THE'
        PRINT*,' MODIFY MODE OF THE SIMULATION ROUTINE.'
        REWIND 1
        CLOSE (1)
        IFLG=1
        RETURN
    END IF
    READ (1,30,ERR=3102) O(15),O1,O2
30   FORMAT(3I4)
C
C    PROMPTING USER AND READING IN RESPONSE
C
3001  FORMAT(/,' ENTER FILE NAME',/)
C
C....CHECKING IF SYSTEM HAS MORE NODES OR LINKS THAN ALLOWED BY LNL
C
    IF (O(1).GT.LNL.AND.O(1).GE.O(2)) THEN
        WRITE(6,23) ' LINKS',' LINKS',O(1)
23   FORMAT(' CANNOT RETRIEVE SYSTEM - TOO MANY',A,' . TO RETREIVE,'
>,' INCREASE '/' LNL IN SOURCE CODE TO MAXIMUM NUMBER OF',A,I4)
        RETURN
    ELSE IF (O(2).GT.LNL) THEN
        WRITE(6,23) ' NODES',' NODES',O(2)
        RETURN
    END IF
C
C    REINITIALIZING VARIABLES
C
    DO 9 I = 1, PNL
        IBE(I) = 0
        IEN(I) = 0
        IPI(I) = 0
        INO(I) = 0
        IBI(I) = 0
        IEI(I) = 0
        HE(I)   = 0
        DO(I)   = 0
        EL(I)   = 0
        HW(I)   = 0
9    CONTINUE
    READ (1,40,ERR=3102,END=3102) (HE(J),DO(J),EL(J),J=1,O2)
40   FORMAT(3E21.14)
    READ (1,40,ERR=3102,END=3102) (A(J),CP(J),DI(J),J=1,O1)
    READ(1,50,ERR=3102,END=3102) (XL(J),HW(J),J=1,O1)
50   FORMAT(2E21.14)
    READ (1,60,ERR=3102,END=3102) JOB
60   FORMAT(A60)
    READ (1,70,ERR=3102,END=3102) (IBE(I),IEN(I),IPI(I),

```

```
> INO(I),I=1,02)
70  FORMAT(4I4)
    REWIND 1
    CLOSE (1)
    IFLG=1
    RETURN
3102 PRINT*, ' NO BACKUP FILE PRESENT. '
    RETURN
2   FORMAT(A60)
    END

C
C   SUBROUTINE PARALE LOCATES LINKS WITH THE SAME BEGINNING AND
C   ENDING NODES, TO ADD TO THE CORRESPONDING COEFFICIENTS A IN
C   THE COEFFICIENT MATRIX AND TO STORE THE SUM UNDER THE
C   COEFFICIENT OF THE LINK WITH THE HIGHER INTERNAL LINK NUMBER
C   EQUAL TO ZERO
C
C   SUBROUTINE PARALE (A,XL,O,IBI,IEI)
C   INTEGER O,PNL
C   REAL*8 A
C   PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
C   DIMENSION A(IA),XL(PNL),O(15),IBI(PNL),IEI(PNL)
C   DO 20 I=1,O(1)-1
C
C   RECOGNIZING LINKS WHICH WERE ALREADY COMBINED WITH A LOWERED
C   NUMBERD LINK
C
C   IF (A(I).EQ.0.AND.XL(I).LT.1000000.) GOTO 20
C
C   CHECKING FOR LINKS WITH DIFFERENT NUMBER BUT SAME
C   BEGINNING AND ENDING NODES NUMBERS.
C
C   DO 10 J=I+1,O(1)
C   IF (IBI(I).EQ.IBI(J).AND.IEI(I).EQ.IEI(J)) THEN
C
C   IF ENDING NODES ARE IDENTICAL TWO A VALUES ARE ADDED AND
C   STORED UNDER THE A VALUE OF THE LINK WITH THE LOWER LINK NUMBER
C
C   A(I)=A(I)+A(J)
C   A(J)=0
C   END IF
10  CONTINUE
20  CONTINUE
    RETURN
    END

C *****
C   SUBROUTINE PIPECH CHECKS WHETHER A LINK WITH THE SAME NUMBER
C   WAS PREVIOUSLY ENTERED
C *****
C   SUBROUTINE PIPECH(II)
C   INTEGER PNL
C   PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
C   COMMON
C /TOPOL/IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PNL)
C   1)
C
C   CHECKING FOR A 1 AT THE APPROPRIATE LOCATION OF ARRAY IPI
C
C   IF (IPI(II).EQ.1) THEN
C
```



```
C      DETERMINING USER NUMBER OF BEGINNING NODE OF THE PREVIOUSLY
C      ENTERED LINK
C
      K=IBE(II)
C
C      DETERMINING USER NUMBER OF ENDING NODE OF THE PREVIOUSLY
C      ENTERED LINK
C
      J=IEN(II)
      PRINT 11,INT(II),K,J
11     FORMAT (' ELEMENT ',I4,' WAS PREVIOUSLY ENTERED FROM ',I4,' TO
',I
14)
      PRINT*,' NEW DATA RETAINED.'
      END IF
      RETURN
      END
*****OPTIMIZATION*****
C OPTIMI INITIALIZES OPTIMIZATION VARIABLES, DISPLAYS OPTIMIZATION
C MENU, AND TRANSFERS CONTROL TO SUBROUTINE DEPENDING ON THE
C OPTION SELECTED BY USER
C
      SUBROUTINE OPTIMI
      CHARACTER ST*90
      REAL*8 A,S,G,HE
      INTEGER O,PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
      COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
      COMMON /PRINT/ IPM,IPP,IPE
      COMMON /SOLUT/ ISS(20,15),IBS(15)
      COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
      COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
      COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
C
C      15 GROUPS, 10 SIZES/GROUP
C      (PNL) PIPES, (PNL) NODES
C
C ***** INITIALIZATION *****
C      TESTING WHETHER OPTIMIZATION PARAMETERS ARE PRESENT
C      IF DATA IS PRESENT THE PROGRAM BYPASSES THE REST OF THE
C      INITIALIZATION
C
      O(9)=0
      IF (IGR.GT.0) GOTO 11
C
C      SETTING OF DEFAULT VALUES FOR PRESSURE INCREMENT, COST
C      COST PERCENTAGE, AND HAZEN-WILLIAMS COEFFICIENT FOR PIPE
C      CLEANING/LINING
C
      R22=-3
      R23=1.03
      HWCC=120
      DO 49 I=1,PNL
```

```

        IGROU(I)=0
        ICATE(I)=1
49      N3(I)=0
        DO 51 I=1,20
51      EF(I)=100
C
C      SETTING OF ALL FIVE LOADING PATTERNS EQUAL TO THE LOADING
C      PATTERN USED IN THE SIMULATION ROUTINE AND SETTING THE
C      REQUIRED MINIMUM PRESSURE AT ALL NODES FOR ALL PATTERNS
C      TO 1E-10
C
        DO 50 I=1,5
        DO 50 J=1,5
        DM(J,I)=DO(I)
50      XP(J,I)=-1E10
        O(12)=0
C
C      SUBROUTINE PUMP IS CALLED TO DETERMINE PUMPS WHICH HAVE
C      A FLOW RATE DICTATED BY CONTINUITY
C
11      CALL PUMP(N3,IBI,IEI,O,CP,DO)
C
C *****OPTION MENU*****
C
1      IF (IPM.EQ.0) GOTO 20
        WRITE(6,3)
3      FORMAT (/,/,/,/, ' SELECT PROGRAM OPTION : '/
1/'          OPTIMIZE                : ENTER 0 OR 0E PRESS RETURN')
        WRITE(6,87)
87     FORMAT('          MODIFY OPT. DATA          :          1'/
1         '          PRINT OPT. DATA              :          2'/
2         '          STORE OPT. DATA              :          3'/
3         '          RETRIEVE OPT. DATA          :          4'/
4         '          ENTER/MODIFY COST DATA      :          5'/
5         '          PROGRAM CONTROL              :          8'/
6         '          TERMINATE                    :          9'//)
20     READ(*,5,END=999) ST
        IF (IPE.EQ.1) PRINT*,ST
5      FORMAT (A60)
        IF (ST(1:1).LT.'0'.OR.ST(1:1).GT.'9') GOTO 1
        IF (ST(1:1).EQ.'0') CALL OPTRUN(ST)
        IF (ST(1:1).EQ.'1') CALL OPTMOD
        IF (ST(1:1).EQ.'2') CALL OPTPRN
        IF (ST(1:1).EQ.'3') CALL
OPTSTO(MUNU,IGR,R22,R23,XP,DM,EF,PT,SI,NS
1,HWCC,IGROU,ICATE,O)
        IF (ST(1:1).EQ.'4') CALL
OPTRET(MUNU,IGR,R22,R23,XP,DM,EF,PT,SI,NS
1,HWCC,IGROU,ICATE,O)
        IF (ST(1:1).EQ.'5') CALL COSTDA
        IF ((ST(1:1).EQ.'6').OR.(ST(1:1).EQ.'7')) THEN
PRINT*, 'INVALID ENTRY - TRY AGAIN'
        GOTO 1
        ENDIF
        IF (ST(1:1).EQ.'8') RETURN
        IF (ST(1:1).EQ.'9') CALL TERMIN
        GOTO 1
999    CALL BLANK
        GOTO 1
        END

```

```
C
C ***** DATA INPUT *****
C ENTERING OF THE OPTIMIZATION KEYWORDS AND NUMERICAL DATA
C
      SUBROUTINE OPTMOD
      CHARACTER ST*90
      REAL*8 A,S,G,HE
      INTEGER PNL,O
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /PRINT/ IPM,IPP,IPE
      COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
      COMMON /SOLUT/ ISS(20,15),IBS(15)
      COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
      COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
      COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
      DIMENSION VALUE(30)
      CHARACTER TYPE*4,TYPF*4,KEY*36
      CHARACTER KTYPE*4,KTYPF*4
      KEY='END GROUHWCCCLIMCLIMPLOADPRICSIZE'
      HWCC=120
      WRITE(6,5)
5  FORMAT(/' OPTIMIZATION INPUT'/' TYPE KEYW FOR LIST OF KEYWORDS')
99  IG=1
      IF (IGR.EQ.0) IGR=1
C
C   TYPE IS INITIALLY SET TO GROU
C
      TYPE='GROU'
      TYPF=' '
100 IF (IPP.EQ.1) PRINT 7,TYPE,IG
7  FORMAT (/,' O. KEYWORD IS (' ,A4,' ',I2,') ENTER (KEYWORD) DATA
1LIST',/)
      READ(*,2,END=999)ST
      IF (IPE.EQ.1) PRINT*,ST
2  FORMAT (A60)
      KTYPE=' '
      IF (ST(1:4).EQ.'KEYW') THEN
          CALL KEYWRD(3,KTYPE)
          GOTO 99
      END IF
C
C   CALLING OF SUBROUTINE DECINP TO DECOMPOSE THE INPUT STRING
C
      CALL DECINP (ST,VALUE,KTYPE,KTYPF,L)
      IF (KTYPE.EQ.'END') RETURN
      IF (INDEX(KEY,KTYPE).EQ.0) THEN
      IF (INDEX('OUTPRATIMINIPUMP',KTYPE).NE.0.AND.L.NE.0) GOTO 15
          PRINT*,'ILLEGAL KEYWORD'
          GOTO 99
      END IF
      IF (L.EQ.0) THEN
          CALL KEYWRD(30,KTYPE)
          GOTO 100
      END IF
```

```

C
C   DIRECTING OF PROGRAM EXECUTION ACCORDING TO KEYWORD TYPE
C
C   *****KEYWORD HWCC*****
C   KEYWORD MUST BE FOLLOWED BY A SINGLE NUMERIC VALUE WHICH IS
C   ACCEPTED AS THE HAZEN WILLIAMS COEFFICIENT FOR PIPES AFTER
C   CLEANING/LINING
C
C   IF (KTYPE.EQ.'HWCC') THEN
C   IF (L.EQ.0) GOTO 100
C   HWCC=VALUE(1)
C   GOTO 100
C   END IF
C
C   *****KEYWORD LIMP*****
C   KEYWORD MUST BE FOLLOWED BY A SINGLE NUMERIC VALUE WHICH IS
C   ACCEPTED AS THE PRESSURE INCREMENT TO BE USED IN THE FORMATION
C   OF THE SOLUTION QUEUE OF PARETO OPTIMAL SOLUTIONS.
C
C   IF (KTYPE.EQ.'LIMP') THEN
C   IF (L.EQ.0) GOTO 100
C   R22=-ABS(VALUE(1))
C   GOTO 100
C   END IF
C
C   *****KEYWORD LIMC*****
C   KEYWORD MUST BE FOLLOWED BY A SINGLE NUMERIC VALUE WHICH IS
C   ACCEPTED AS THE COST PERCENTAGE TO BE USED IN THE FORMATION
C   OF THE SOLUTION QUEUE OF PARETO OPTIMAL SOLUTIONS.
C
C   IF (KTYPE.EQ.'LIMC') THEN
C   IF (L.EQ.0) GOTO 100
C   R23=1.+ABS(VALUE(1))/100.
C   GOTO 100
C   END IF
C
C   RESETTING THE FIRST AND SECOND KEYWORD ACCORDING TO THE FIRST
C   AND SECOND KEYWORD IN THE INPUT STRING. IF NO KEYWORD WAS
C   ENTERED IN THE INPUT STRING KTYPE IS SET TO THE SECOND KEYWORD
C   OF THE PREVIOUS ENTRY
C
15 IF (INDEX('GROULOADPRICSIZE',KTYPE).GT.0) TYPE=KTYPE
IF (INDEX('OUTPRATIMINIPUMP',KTYPE).GT.0) TYPF=KTYPE
IF (INDEX('OUTPRATIMINIPUMP',KTYPE).GT.0) TYPF=KTYPE
IF (KTYPE.EQ.' ') KTYPE=TYPF
IF (L.EQ.0) THEN
  IG=1
  IF (TYPE.EQ.'PRIC') IG=2
  GOTO 100
END IF
IF
(INDEX('GROULOADPRICSIZE',KTYPE).GT.0.AND.L.EQ.1.AND.KTYPF.NE.'
1DELE') THEN
  IG=VALUE(1)
  GOTO 100
END IF
C
C   TRANSFERRING OF CONTROL ACCORDING TO THE PRIMARY KEYWORD TYPE
C
C   IF (TYPE.EQ.'GROU') GOTO 102

```

```
      IF (TYPE.EQ.'LOAD') GOTO 182
      IF (TYPE.EQ.'PRIC') GOTO 202
      IF (TYPE.EQ.'SIZE') GOTO 302
C
C ***** GROUP ASSIGNMENTS *****
C
102   J=0
C
C*****KEYWORD GROU*****
C
      IF (KTYPE.EQ.'GROU') THEN
          IG=VALUE(1)
          IF (IG.GT.IGR) IGR=IG
C
C      TESTING OF THE VALIDITY OF THE GROUP NUMBER, WHICH MUST BE
C      IN THE RANGE 0-15
C
          IF (IG.GE.0.AND.IG.LT.16) GOTO 110
          PRINT *, ' ILLEGAL GROUP #.'
          GOTO 100
110   IF (L.EQ.1.AND.KTYPE.NE.'DELE'.AND.KTYPF.NE.'DELE') GOTO 100
          J=1
          END IF
          IF (KTYPE.EQ.'ALL'.OR.KTYPF.EQ.'ALL') THEN
C
C      TESTING WHETHER TWO LINK NUMBERS ENTERED ARE PRESENT
C
          DO 1140 II=1,O(1)
1140  IF (VALUE(J+1).EQ.IPI(II)) GOTO 115
114   PRINT *, ' ILLEGAL PIPE #.'
          GOTO 100
C
C      COMPUTING OF IF, THE FIRST INTERNAL LINK NUMBER AND IL, THE
C      LAST INTERNAL LINK NUMBER IN THE SPECIFIED RANGE
C
115   IF=II
          DO 1150 II=1,O(1)
1150  IF (VALUE(J+2).EQ.IPI(II)) GOTO 1151
          GOTO 114
1151  IL=II
          DO 120 I=IF,IL
C
C      LISTING OF THE LINKS IN THE RANGE WHICH ARE NOT PIPES (I.E.
C      PUMPS OR PRVS)
C
          IF (XL(I).GT.999999..OR.CP(I).LT.0) THEN
              WRITE(6,10) IPI(I)
10    FORMAT(' LINK ',I4,' IS NOT A PIPE. LINK IS IGNORED.')
              GOTO 120
          END IF
C
C      SETTING OF THE PROPER GROUP IN ARRAY IGROU
C
          IGROU(I)=IG
120   CONTINUE
          IG=IG+1
          GOTO 100
          END IF
C
C      ELIMINATING GROUP ASSIGNMENTS
```

```
C
  IF (KTYPE.EQ.'DELE'.OR.KTYPF.EQ.'DELE') THEN
    KTYPF=' '
    DO 125 I = 1 ,O(1)
125   IF (IGROU(I).EQ.IG) IGROU(I) = 0
      NS(IG) = 0
      GOTO 100
    END IF
C
C   ENUMERATING THE NUMERIC VALUES IN THE INPUT STRING
C
  IF (IG.GT.IGR) IGR=IG
  DO 140 I=1+J,L
  DO 1400 II=1,O(1)
C
C   TESTING WHETHER LINK EXISTS
C
1400 IF (VALUE(I).EQ.IPI(II)) GOTO 1401
    WRITE(6,25) INT(VALUE(I))
    25  FORMAT(' NO PIPE # ',I4)
    GOTO 140
1401 IF=II
C
C   TESTING WHETHER LINK IS A PRV OR PUMP
C
  IF (XL(IF).GT.999999..OR.CP(IF).LT.0) THEN
    WRITE(6,10) IPI(IF)
    GOTO 140
  END IF
C
C   SETTING THE PROPER GROUP IN ARRAY IGROU
C
  IGROU(IF)=IG
140  CONTINUE
    IG=IG+1
    GOTO 100
C   *****KEYWORD LOAD*****
182  J=0
    IF (KTYPE.EQ.'LOAD') THEN
      IG=VALUE(1)
C
C   TESTING THE VALIDITY OF THE PATTERN NUMBER ,IG, WHICH MUST
C   BE IN THE RANGE 5 INCLUSIVE
C
  IF (IG.GT.5.OR.IG.LT.1) THEN
    PRINT*,' ILLEGAL LOAD PATTERN #'
    GOTO 100
  END IF
  J=1
  END IF
C
C   RESETTING OF THE NUMBER OF PATTERNS, MUNU, IF NECESSARY
C
  IF (MUNU.LT.IG) MUNU=IG
C
C   CHECKING SECOND KEYWORD FOR MINI > MINIMUM PRESSURES ARE TO
C   BE ASSIGNED
C
  IF (KTYPE.EQ.'MINI'.OR.KTYPF.EQ.'MINI') THEN
    IF (L.EQ.J+1) THEN
```

```
C
C      ONE NUMERIC VALUE AFTER KEYWORD MINI
C
      VALUE(3+J)=VALUE(1+J)
      K0=1
      K1=O(2)
      ELSE IF (L.EQ.J+2) THEN
C
C      TWO NUMERIC VALUES AFTER KEYWORD MINI
C
      VALUE(3+J)=VALUE(2+J)
      DO 1821 II=1,O(2)
1821  IF (VALUE(1+J).EQ.INO(II)) GOTO 1822
      GOTO 189
1822  K0=II
      K1=K0
      ELSE
C
C      THREE NUMERIC VALUES AFTER KEYWORD MINI
C
      DO 1823 II=1,O(2)
1823  IF (VALUE(1+J).EQ.INO(II)) GOTO 1824
      GOTO 189
1824  K0=II
      DO 1825 II=1,O(2)
1825  IF (VALUE(2+J).EQ.INO(II)) GOTO 1826
      GOTO 189
1826  K1=II
      END IF
C
C      ENUMERATING THE INTERNAL NODE NUMBERS IN THE RANGE KO
C      THROUGH K1.  NODES WHICH ARE SUPPLY POINTS ARE LISTED
C      AND EXCLUDED FROM ASSIGNMENT OF A MINIMUM OF A MINIMUM
C      PRESSURE
C
      DO 184 I=K0,K1
      IF (DM(IG,I).GT.1E12) THEN
      IF (L.GT.1) WRITE(6,20) INO(I)
20    FORMAT(' NODE ',I4,' IS A SUPPLY POINT.')
      GOTO 184
      END IF
      XP(IG,I)=VALUE(3+J)
184  CONTINUE
      GOTO 100
      END IF
C
C      IF SECOND KEYWORD IS PUMP, PERCENT RUNNING TIME AND EFFICIENCY
C      IN PERCENT ARE TO BE ASSIGNED
C
      IF (KTYPE.EQ.'PUMP'.OR.KTYPF.EQ.'PUMP') THEN
C
C      DETERMINING THE INTERNAL LINK NUMBER FOR THE ELEMENT SPECIFIED
C
      IPMP=0
      DO 1841 II=1,O(1)
      IF (CP(II).LT.0) IPMP=IPMP+1
1841  IF (VALUE(1+J).EQ.IPI(II).AND.CP(II).LT.0) GOTO 1842
      PRINT*,' NO LINK WITH THIS NUMBER, OR ELEMENT IS NOT A PUMP.'
      GOTO 100
1842  K0=II
```

```
C
C   ASSIGNING OF PERCENT TIME RUNNING TO ARRAY PT, SUBSCRIPTED
C   WITH THE PATTERN NUMBER AND THE INTERNAL LINK NUMBER.
C   IF A SECOND NUMERIC VALUE IS SPECIFIED THE SECOND VALUE IS
C   ASSIGNED TO THE EFFICIENCY EF, SUBSCRIPTED WITH THE INTERNAL
C   LINK NUMBER.
C
C   IF (IPMP.GT.20) THEN
C       WRITE(6,1844)
1844   FORMAT(1X,' 20 PUMPS ALREADY DEFINED - ENTRY IGRNORED.')
C       GOTO 100
C   ELSE
C       PT(IG,IPMP)=VALUE(2+J)
C       IF (L.EQ.J+3) EF(IPMP)=VALUE(3+J)
C   END IF
C   GOTO 100
C END IF

C
C *****KEYWORD OUTP*****
C
C   IF (KTYPE.EQ.'OUTP'.OR.KTYPF.EQ.'OUTP') THEN
C   DO 1845 II=1,O(2)
1845   IF (VALUE(1+J).EQ.INO(II)) GOTO 1846
C       GOTO 189
1846   K0=II
C       IF (DO(K0).GT.9E9) THEN
C           WRITE(6,30) INT(VALUE(1+J))
30     FORMAT(' NODE ',I4,' IS A SUPPLY POINT.')
C           GOTO 100
C       END IF
C       DM(IG,K0)=VALUE(2+J)
C       GOTO 100
C   END IF

C
C *****KEYWORD RATI*****
C
C   IF (KTYPE.EQ.'RATI'.OR.KTYPF.EQ.'RATI') THEN
C   IF (L.EQ.J+1) THEN
C       VALUE(3+J)=VALUE(1+J)
C       K0=1
C       K1=O(2)
C   ELSE
C
C   IF THREE NUMERIC VALUES ARE SPECIFIED AFTER KEYWORD RATI
C   INTERNAL NODE NUMBERS FOR NODES SPECIFIED IN THE INPUT STRING
C   K0,K1 ARE DETERMINED
C
C       DO 1861 II=1,O(2)
1861   IF (VALUE(1+J).EQ.INO(II)) GOTO 1862
C           GOTO 189
1862   K0=II
C       DO 1863 II=1,O(2)
1863   IF (VALUE(2+J).EQ.INO(II)) GOTO 1864
C           GOTO 189
1864   K1=II
C   END IF

C
C   ENUMERATING THE INTERNAL NODE NUMBERS FROM K0 TO K1.
C
C   DO 194 I=K0,K1
```



```

        IF (DO(I).LT..001.OR.DO(I).GT.9E9) THEN
            WRITE(6,40) INO(I)
40         FORMAT(' NODE ',I4,' IS NOT AN OUTPUT POINT. ')
            GOTO 194
        END IF
C
C     MULTIPLYING THE PREVIOUSLY SPECIFIED OUPUT BY THE THIRD
C     NUMERIC VALUE LISTED AFTER THE KEYWORD RATI IN THE INPUT STRING
C
        DM(IG,I)=VALUE(3+J)*DM(IG,I)
194       CONTINUE
            GOTO 100
        END IF
        PRINT*,' ERROR IN INPUT'
        GOTO 100
189      PRINT*,' NO SUCH NODE NUMBER.'
        GOTO 100
C
C ***** PRICE FCT. ASSIGNMENTS *****
C
202      J=0
        IF (KTYPE.EQ.'PRIC') THEN
            IG=VALUE(1)
C
C     TESTING THE VALIDITY OF THE PRICE FUNCTION NUMBER, WHICH MUST
C     BE IN THE RANGE 0-15 (INCLUSIVE)
C
            IF (IG.GE.0.AND.IG.LT.16) GOTO 210
            PRINT *,' ILLEGAL PRICE FCT. #'
            GOTO 100
210      IF (L.EQ.1) GOTO 100
            J=1
        END IF
C
C     ASSIGNING THE PRICE FUNCTION NUMBER IG TO ALL PIPES IN THE
C     RANGE OF USER LINK NUMBERS INDICATED BY THE BY THE TWO NUMERIC
C     VALUES FOLLOWING THE WORD ALL
C
        IF (KTYPE.EQ.'ALL'.OR.KTYPF.EQ.'ALL') THEN
C
C     TESTING WHETHER TWO LINK NUMBERS ARE PRESENT
C
            DO 2131 II=1,O(1)
2131     IF (VALUE(J+1).EQ.IPI(II)) GOTO 215
214     PRINT *,' ILLEGAL PIPE #'
            GOTO 100
215     IF=II
            DO 2141 II=1,O(1)
2141     IF (VALUE(J+2).EQ.IPI(II)) GOTO 2142
            GOTO 214
2142     IL=II
C
C     ENUMERATING THE INTERNAL LINK NUMBERS IN THE SPECIFIED RANGE
C
            DO 220 I=IF,IL
                IF (XL(I).GT.999999..OR.CP(I).LT.0) THEN
                    WRITE(6,10) IPI(I)
                    GOTO 220
                END IF
C

```

```
C      SETTING THE PRICE FUNCTION IN ARRAY ICATE
C
      ICATE(I)=IG
220    CONTINUE
      GOTO 100
    END IF

C
C      ASSIGNING PIPES LISTED INDIVIDUALLY IN THE INPUT STRING TO
C      PRICE FUNCTION IG
C
      DO 240 I=1+J,L
      DO 2391 II=1,O(1)
2391   IF (VALUE(I).EQ.IPI(II)) GOTO 2392
      WRITE(6,25) INT(VALUE(I))
      GOTO 240
2392   IF=II

C
C      TESTING WHETHER LINK IS A PRV OR PUMP
C
      IF (XL(IF).GT.999999..OR.CP(IF).LT.0) THEN
        WRITE(6,10) IPI(IF)
        GOTO 240
      END IF
      ICATE(IF)=IG
240    CONTINUE
      GOTO 100

C
C ***** SIZES *****
C
302    J=0
      IF (KTYPE.EQ.'SIZE') THEN
        IG=VALUE(1)

C
C      TESTING OF THE VALIDITY OF THE GROUP NUMBER, WHICH MUST BE IN
C      THE RANGE 1 THROUGH IGR (THE HIGHEST GROUP NUMBER USED UNDER
C      THE KEYWORD GROU)
C
      DO 303 II=1,O(1)
303    IF (IGROU(II).EQ.IG) GOTO 304
        PRINT*,' ILLEGAL GROUP #.'
        IF (IG.EQ.0) IG=1
        GOTO 100
304    J=1
      END IF
      J0=NS(IG)

C
C      IF THE SECOND KEYWORD IS IN THE INPUT STRING IS DELE, THE
C      SIZE COUNTER NS OF GROUP IG IS SET BACK TO ZERO
C
      IF (KTYPE.EQ.'DELE'.OR.KTYPF.EQ.'DELE') THEN
        KTYPF=' '
        NS(IG)=0
        GOTO 100
      END IF

C
C      ENUMERATING THE NUMERIC VALUES IN THE INPUT STRING
C
      DO 340 K=1+J,L

C
C      NUMERIC VALUES ARE ASSIGNED TO ARRAY SI, SUBSCRIPTED WITH
```

```

C      THE GROUP NUMBER AND A SIZE COUNTER FOR THIS GROUP
C
340    SI(IG,K+J0-J)=VALUE(K)
C
C      UPDATING OF SIZE COUNTER FOR THE GROUP
C      IF THE LIST OF SIZES CONTAINS THE LETTER C FOR CLEANING/LINING
C      THE SIZE COUNTER IS INCREMENTED BY ONE AND A VALUE OF -1 IS
C      ASSIGNED TO THE NEXT ELEMENT IN ARRAY SI
C
      NS(IG)=J0+L-J
      IF (KTYPE.EQ.'C'.OR.KTYPF.EQ.'C') THEN
        NS(IG)=NS(IG)+1
        SI(IG,NS(IG))=-1
        KTYPF='      '
      END IF
      IG=IG+1
      DO 343 I = 1, O(1)
        IF (IGROU(I).EQ.IG) GOTO 100
343    CONTINUE
      IG=IG-1
      GOTO 100
999    CALL BLANK
      GOTO 100
      END
C
C      *****PRINTING OF OPTIMIZATION DATA*****
C
      SUBROUTINE OPTPRN
      REAL*8 A,S,G,HE
      INTEGER O,PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
      IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /SOLUT/ ISS(20,15),IBS(15)
      COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
      COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
      COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
      CHARACTER OUTP*90
C
C*****PRINTING OF GROUP ASSIGNMENTS*****
C
150    PRINT*
      PRINT*,' OPTIMIZATION PARAMETERS '
      PRINT*
      PRINT*,' GROUP ASSIGNMENTS '
C
C      ENUMERATING OF ALL GROUPS
C
      DO 180 I=1,IGR
      DO 181 II=1,O(1)
181    IF (IGROU(II).EQ.I) GOTO 179
      GOTO 180
179    L=1
C
C ADDING USER LINK NUMBERS WHICH BELONG TO THE GROUP
C TO STRING OUTP

```

```
C
      PRINT 159,I
159  FORMAT (/, ' PIPES IN GROUP ',I3, ' :')
      DO 160 IL=1,O(1)
      IF (IGROU(IL).NE.I) GOTO 160
      CALL OUT(L,IL,IPI,OUTP)
      IF (L.EQ.18) THEN
C
C PRINTING OF THE STRING OUTP IF IT IS FULL
C
      PRINT*,OUTP(1:72)
      L=1
      ELSE
      L=L+1
      END IF
160  CONTINUE
      IF (L.EQ.1) GOTO 180
      PRINT*,OUTP(1:4*L-4)
180  CONTINUE
      PRINT *
C
C PRINTING OF PRICE FUNTION ASSIGNMENTS
C
250  PRINT*
      PRINT*, ' PRICE FUNCTION ASSIGNMENTS'
      ICA=0
C
C ENUMERATING ALL GROUPS AND LINKS
C
      DO 280 I=1,15
      DO 281 II=1,O(1)
281  IF (ICATE(II).EQ.I) GOTO 282
      GOTO 280
282  L=1
C
C ACCUMULATING USER LINK NUMBERS WHICH BELONG TO THE PRICE
C FUNCTION IN STRING OUTP
C
      PRINT 259,I
259  FORMAT (/, ' PIPES IN PRICE FCT. ',I3, ' :')
      ICA=ICA+1
      DO 260 IL=1,O(1)
      IF (ICATE(IL).NE.I.OR.IGROU(IL).EQ.0) GOTO 260
      CALL OUT(L,IL,IPI,OUTP)
      IF (L.EQ.18) THEN
C
C PRINTING OF STRING IF IT IS FULL
C
      PRINT*,OUTP(1:72)
      L=1
      ELSE
      L=L+1
      END IF
260  CONTINUE
      PRINT*,OUTP(1:4*L-4)
280  CONTINUE
C
C PRINTING LIST OF SIZES SPECIFIED FOR EACH GROUP
C
      PRINT *
```

```
350 PRINT*
PRINT*, ' SIZE ASSIGNMENTS '
PRINT*
PRINT*, ' GROUP #   SIZES ASSIGNED: '
C
C ENUMERATING SIZES ASSIGNED TO TO EACH GROUP
C
DO 360 I=1,IGR
DO 361 II=1,O(1)
IF (IGROU(II).NE.0) THEN
C
C CALLING SUBROUTINE SIZEL TO GENERATE LIST OF SIZES TO PRINT
C
CALL SIZEL(SI,I,NS,OUTP)
IF (NS(I).NE.0) WRITE(6,355)I,OUTP(1:6*NS(I))
355 FORMAT(5X,I2,3X,A)
GOTO 360
END IF
361 CONTINUE
360 CONTINUE
C
C PRINTING OF OPTIMIZATION PARAMETERS RELATED TO LOADING PATTERNS
C
PRINT*
195 PRINT*
PRINT*, ' LOADING PATTERNS '
PRINT*
PRINT*, '                LOADS IN GPM AND MIN. PRESSURE IN PSI '
PRINT 193,(I,I=1,MUNU)
193 FORMAT (1X,'PATTERN #',I7,5X,'*',4(I8,5X,'*'))
PRINT 200,('GPM','PSI *',I=1,MUNU)
200 FORMAT (1X,'NODE # *',5(3X,A3,3X,A5))
C
C ENUMERATING OF NODES
C
DO 198 I=1,O(2)
DO 196 J=1,MUNU
IF (DM(J,I).GT.1E10) GOTO 198
196 IF (ABS(DO(I)-DM(J,I)).GT..001.OR.XP(J,I).GT.0) GOTO 197
GOTO 198
197 PRINT 199,INO(I),(DM(K0,I),XP(K0,I),K0=1,MUNU)
199 FORMAT (1X,I4,3X,'*',5(F7.0,F6.1,'*'))
198 CONTINUE
C
C ENUMERATING OF LINKS TO DETERMINE WHICH ARE PUMPS
C
IPMP=0
DO 1981 I=1,O(1)
IF (CP(I).GT.0) GOTO 1981
IPMP=IPMP+1
IF (IPMP.EQ.1) THEN
PRINT*
PRINT*, '                PUMP EFFICIENCY % AND % TIME RUNNING '
PRINT 192,(K,K=1,MUNU)
192 FORMAT (1X,'PATTERN #',10X,I8,2X,'*',4(I9,2X,'*'))
PRINT 191,('*',K=1,MUNU)
191 FORMAT (1X,'PUMP #   EFFIC.',13X,5(1X,A,10X))
END IF
PRINT 1991,IPI(I),EF(IPMP),(PT(K0,IPMP),K0=1,MUNU)
1991 FORMAT (1X,I4,'   *',F6.1,3X,'*',5(F10.1,'*'))
```

```
1981 CONTINUE
C
C PRINTING OF HAZEN-WILLIAMS COEFFICIENT OF CLEANED/LINED
C PIPES, THE PRESSURE TOLERANCE, AND THE COST TOLERANCE
C
      PRINT*
      WRITE(6,5) HWCC
5     FORMAT(' COEF. FOR CLEANING ',F5.1)
      PRINT*
      WRITE(6,10) R22
      WRITE(6,20) (R23-1)*100
10    FORMAT(' PRESSURE TOLERANCE ',F5.1,' PSI')
20    FORMAT(' COST TOLERANCE      +',F4.1,' %')
      PRINT*
      RETURN
      END
C*****OPTIMIZING*****
C SUBROUTINE OPTRUN CALLS ALL PROCEDURES WHICH ARE NECESSARY
C FOR THE OPTIMIZATION PROCESS
C
      SUBROUTINE OPTRUN(ST)
      CHARACTER ST*90
      REAL*8 A,S,G,HE
      INTEGER O,PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
      COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
      COMMON /SOLUT/ ISS(20,15),IBS(15)
      COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
      COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
      COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
      DIMENSION IFS(100,15)
      IF (O(4).EQ.0) THEN
          PRINT*,'PRIOR TO OPTIMIZING, THIS SYSTEM MUST BE BALANCED'
          CALL SIMBAL(ST)
      END IF
      J = 0
      DO 30 I = 1, O(1)
          IF (IGROU(I).GT.0) GOTO 10
          J = J + 1
          IF (J.GE.O(1)) THEN
              PRINT*,'NO GROUPS ASSIGNED !'
              RETURN
          END IF
10         IF (IGROU(I).NE.0) THEN
              IF (NS(IGROU(I)).EQ.0.AND.IGROU(I).NE.0) THEN
                  WRITE(6,20) IGROU(I)
20                 FORMAT(' NO SIZES ASSIGNED TO GROUP ',I2)
                  RETURN
              END IF
          END IF
30        CONTINUE
          IF (MUNU.EQ.0) THEN
              WRITE(6,40)
```

```
40  FORMAT(' NO LOADINGS ASSIGNED !')
    RETURN
    ENDIF
    CALL OPTINT(IFLG,PAT,FAC,IFS)
    IF (IFLG.EQ.1) RETURN
    CALL ENUMER(PAT,FAC,IFS,ST)
    CALL OPTERM(PAT)
    RETURN
    END
C
C ***** COST ACCUMULATION *****
C
    SUBROUTINE OPTINT(IFLG,PAT,FAC,IFS)
    CHARACTER PATT*5
    REAL*8 A,S,G,HE
    INTEGER O,PNL
    PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
    COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
    COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
    COMMON /TOPOL/
    IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
    1L)
    COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
    COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
    COMMON /SOLUT/ ISS(20,15),IBS(15)
    COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
    COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
    COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
    1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
    DIMENSION ICS(15),IFS(100,15)
    IFLG=1
C
C CALCULATING OF THE FACTOR USED IN CONVERTING ANNUAL COST
C TO PRESENT WORTH
C
390  FAC=(1.+XI)**NY
    FAC=(FAC-1.)/XI/FAC
C
C CHECKING LOAD ASSIGNMENTS TO SEE THAT THEY DID NOT CHANGE
C THE SUPPLY POINT CHARACTERISTICS, BY SETTING ARRAY DM EQUAL TO
C ARRAY DO AT ALL SUPPLY POINTS, AND FOR ALL LOADING PATTERNS
C
    DO 3030 J=1,O(2)
    IF (DO(J).LT.9E9) GOTO 3030
    DO 3020 I=1,5
3020  DM(I,J)=DO(J)
3030  CONTINUE
    DO 492 I=1,IGR
C
C DETERMINING PIPE OR CLEANING/LINING COST IN EACH GROUP FOR
C FOR EACH SIZE SPECIFIED, AND SORTING THE SIZES ACCORDING TO
C COST, ELIMINATING DUPLICATIONS AND ELIMINATING SIZES WHICH
C ARE MORE EXPENSIVE THAN THE NEXT LARGER SIZE IN THE GROUP.
C LOOP ENUMERATES THE GROUP NUMBERS
C
    TC(I,1)=0
C
C ENUMERATING SIZES IN GROUPS I
C
    DO 460 J=1,NS(I)
```

```
C
C INITIALIZING COST FOR GROUP I AND SIZE J WITH ZERO
C
      TC(I,J)=0
C
C CHECKING FOR ELIMINATION
C
      IF (ABS(SI(I,J)).LT..01) GOTO 460
      DMIN=10000.
C
C CHECKING FOR CLEANING
C
      IF (SI(I,J).LT.-.01) GOTO 420
C
C LOOP SEARCHES COST TABLE FOR THE SIZE EQUAL TO SIZE J IN
C GROUP I
C
      DO 400 K=1,KS
      IF (ABS(SIZ(K)-ABS(SI(I,J))).LT.0.01) GOTO 420
400  CONTINUE
      PRINT*,' NO SIZE ',SI(I,J),' IN COST TABLE.'
      RETURN
420  DO 450 IL=1,O(1)
C
C SEARCHING ARRAY IGROU FOR THE NEXT PIPE IN GROUP I
C
      IF (IGROU(IL).NE.I) GOTO 450
C
C DETERMINING PRICE CATEGORY OF THE PIPE WITH INTERNAL NUMBER IL
C
      L=ICATE(IL)
C
C CHECK FOR CLEANING
C
      IF (SI(I,J).LT.-.01) GOTO 441
C
C ACCUMUALTING COST
C
      TC(I,J)=TC(I,J)+XL(IL)*COST(K,L)
      GOTO 450
C
C IN CASE OF CLEANING/LINING A SEARCH FOR PARALLEL PIPE TO BE
C CLEANED/LINED IS MADE
C
441  DO 442 IP=1,O(1)
442  IF (IBE(IP).EQ.IBE(IL).AND.IEN(IP).EQ.IEN(IL).AND.IP.NE.IL)
1GOTO 4421
      PRINT*,' PIPE ',IPI(IL),' DOES NOT HAVE A PARALLEL PIPE.'
      PRINT*,' CLEANING CANNOT BE SPECIFIED FOR GROUP ',I,'.'
      RETURN
C
C CALCULATING OF THE INTERNAL NUMBER OF THE PARALLEL PIPE, IP
C CALCULATING THE DIAMETER WHICH WOULD HAVE TO BE ASSIGNED TO
C THE NEW PIPE SUCH THAT THE OLD PIPE AND THE NEW PIPE WITH THIS
C DIAMETER ARE EQUIVALENT TO THE CLEANED/LINED OLD PIPE ALONE
C
4421  DECL=((HWCC-ABS(HW(IP)))/ABS(HW(IL)))**.38*DI(IP)
C
C SELCTION OF THE SMALLEST DIAMETER WITHIN THE GROUP
C
```



```

      IF (DECL.LT.DMIN) DMIN=DECL
C
C SEARCHING THE COST TABLE FOR THE DIAMETER OF THE OLD PIPE
C WITH INTERNAL LINK NUMBER IP
C
      DO 443 K0=1,KS
      IF (ABS(SIZ(K0)-DI(IP)*12).LT..01) GOTO 444
443  CONTINUE
      PRINT*,' NO SIZE ',DI(IP)*12,' FOR CLEANING IN COST TABLE.'
      RETURN
C
C ACCUMULATING CLEANING COST
C
444  TC(I,J)=TC(I,J)+XL(IP)*COST(K0,2)
450  CONTINUE
      IF (SI(I,J).GT.-.01) GOTO 460
C
C SETTING SI FOR THE GROUP AND SIZE TO THE NEGATIVE VALUE OF DMIN
C
      SI(I,J)=-DMIN*12.
460  CONTINUE
C
C SORTING OF ARRAYS TC AND SI, ACCORDING TO TC WITH THE MOST
C EXPENSIVE COST ASSIGNED TO THE FIRST LOCATION
C
470  DO 484 I0=1,NS(I)-1
      DO 482 J0=I0+1,NS(I)
      IF (ABS(SI(I,J0)).LT..01) GOTO 484
      IF (TC(I,I0).GE.TC(I,J0)) GOTO 482
      X1=SI(I,I0)
      X2=TC(I,I0)
      SI(I,I0)=SI(I,J0)
      TC(I,I0)=TC(I,J0)
      SI(I,J0)=X1
      TC(I,J0)=X2
      GOTO 470
482  CONTINUE
484  CONTINUE
C
C ELIMINATING DUPLICATE SIZES, AND SIZES IN ARRAY SI WHICH ARE
C MORE EXPENSIVE THAN THE NEXT LARGER SIZE (INCLUDING CLEANING
C OPTION, USING DMIN AS THE SIZE)
C
      I00=1
      DO 490 I0=2,NS(I)
      I2=I00
      IF (ABS(SI(I,I0)).GE.ABS(SI(I,I00))) THEN
486  I2=I2-1
      IF (I2.EQ.0) GOTO 488
      IF (ABS(SI(I,I0)).GE.ABS(SI(I,I2))) GOTO 486
488  I00=I2+1
      SI(I,I00)=SI(I,I0)
      TC(I,I00)=TC(I,I0)
      ELSE
      I00=I00+1
      SI(I,I00)=SI(I,I0)
      TC(I,I00)=TC(I,I0)
      END IF
490  CONTINUE
      NS(I)=I00

```

```
492  CONTINUE
C
C*****TEST ON SIZE RANGE*****
C ROUTINE WILL TEST WHETHER SOME SMALL SIZES IN EACH GROUP CAN BE
C RULED OUT
C
C INITIALIZING THE NUMBER OF ENTRIES IN SOQU WITH ZERO
C
      K0=0
C
C ASSIGNING THE MAXIMUM SIZE TO EACH ELEMENT IN ARRAY ICS
C
      DO 494 I=1,IGR
494  ICS(I)=1
C
C CALL SUBROUTINE SIZE TO CHANGE PIPE SIZES AND CHARACTERISTIC
C PIPE COEFFICIENTS
C
      CALL SIZE (O,DI,XL,HW,CP,A,IBE,IEN,IGROU,ICS,SI,IGR,HWCC)
      POLD=1.E38
      C8=1.E38
C
C LOOP ENUMERATES THE LOADING PATTERNS AND CHECKS WHETHER THE
C MINIMUM PRESSURE REQUIREMENT IS MET FOR ALL LOADING PATTERNS
C
      DO 495 M0=1,MUNU
C
C INNER LOOP ASSIGNS THE OUTPUTS FOR THE PARTICULAR PATTERN
C
      DO 491 I=1,O(2)
491  DO(I)=DM(M0,I)
C
C CALLING OF SUBROUTINE OPTBAL TO CALCULATE THE PRESSURE
C DISTRIBUTION AND RETURN WITH THE LOWEST PRESSURE, P8 AT ANY
C NODE CHECKED
C
      CALL OPTBAL (XP,P8,P88,M0,PT,EF,TP,N3)
      CALL PUMPH(N3,IBI,IEI,IPI,O,CP,XL,HE,EL,TP,PT,EF,XP,M0)
C
C CALCULATING OF THE PRESENT WORTH OF PUMPING COST
C
      TP0=741.6*ENCO*TP*FAC
      IF (P8.LT.R22) THEN
        PRINT*, ' MAXIMUM SIZES ARE INSUFFICIENT IN PATTERN ',M0
        RETURN
      END IF
495  CONTINUE
C
C LOOP ENUMERATES THE POSITIONS IN ARRAY ICS FROM LOWER TO HIGHER
C
      DO 499 I=1,IGR
C
C INNER LOOP ENUMERATES THE SIZES, STARTING WITH SMALLEST SIZE
C
      DO 498 J=NS(IGR-I+1),2,-1
        ICS(I)=J
C
C ASSIGNING OF THE SIZE IN EACH GROUP
C
      CALL SIZE (O,DI,XL,HW,CP,A,IBE,IEN,IGROU,ICS,SI,IGR,HWCC)
```

```
C
C INITIALIZING PUMPING COST WITH ZERO
C
      TP1=0
C
C LOOP ENUMERATES LOADING PATTERNS
C
      DO 496 M0=1,MUNU
C
C INNER LOOP ASSIGNS OUTPUTS
C
      DO 4981 K=1,O(2)
4981  DO(K)=DM(M0,K)
      CALL OPTBAL (XP,P8,P88,M0,PT,EF,TP,N3)
      CALL PUMPH(N3,IBI,IEI,IPI,O,CP,XL,HE,EL,TP,PT,EF,XP,M0)
C
C ACCUMULATING PUMPING COSTS FOR ALL PATTERNS
C
      TP1=TP1+741.6*ENCO*TP*FAC
C
C IF THE MINIMUM PRESSURE IS LESS THAN THE PRESSURE TOLERANCE,
C AND THE SIZE OF THE GROUP HAS REACHED THE SECOND LARGEST SIZE,
C NS FOR THIS GROUP IS SET TO 1
C
      IF (P8.LT.R22.AND.J.EQ.2) THEN
        NS(IGR-I+1)=1
        GOTO 497
      END IF
      IF (P8.LT.R22) GOTO 497
      IF (P8.LT.POLD) THEN
        POLD=P8
        MOLD=M0
      END IF
496  CONTINUE
      IF (POLD.GE.0) THEN
C
C ACCUMULATING THE TOTAL COST (PRESENT WORTH OF PUMPING COST
C PLUS PIPE COST) OF THE SYSTEM (VARIABLE C9)
C
        C9=TP1
        DO 4961 M0=1,IGR
          J0=ICS(IGR+1-M0)
4961  C9=C9+TC(M0,J0)
C
C IF C9 IS LESS THAN C8 (THE COST SO FAR OF THE LEAST EXPENSIVE
C SOLUTION , WHICH MET ALL REQUIREMENTS) THE COMBINATION ICS IS
C RECORDED UNDER IBS AND PATTERN NUMBER UNDER PAT
C
        IF (C9.LT.C8) THEN
          DO 4962 J2=1,IGR
4962  IBS(J2)=ICS(J2)
          BESTP=POLD
          C8=C9
          PAT=MOLD
C
C THE PRESENT WORTH OF PUMPING COST IS RECORDED UNDER TP2
C
          TP2=TP1
C
C THE SOLUTION IS TESTED FOR INCLUSION IN THE SOLUTION QUEUE
```

```
C
      CALL SOQU( ISQ, POLD, C8, MOLD, IGR, ICS, SPR, SCO, ISP, ISS)
      END IF
      END IF
C
C ADJUSTING OF THE NUMBER OF SIZES
C
      NS(IGR-I+1)=J
      GOTO 499
497  IF (SI(IGR-I+1,J).GT.-.1) WRITE(6,500) IGR-I+1,SI(IGR-I+1,J)
500  FORMAT(' IN GROUP ',I2,': SIZE ',F4.1,' ELIMINATED')
      IF (SI(IGR-I+1,J).LT.-.01) WRITE(6,510) IGR-I+1
510  FORMAT(' IN GROUP ',I2,': CLEANING ELIMINATED')
498  CONTINUE
499  ICS(I)=1
C
C*****INITIALIZE QUEU OF BAD COMBINATIONS*****
C LOOP ENUMERATES THE ARRAY ICS, STARTING AT THE
C HIGHEST SIZE NUMBER
C
      DO 5010 I=1,IGR
      ICS(I)=1
      J=2
C
C INITIALIZING FLAG, IF1, WITH 1. FLAG WILL BE SET TO 0 WHEN ALL
C SIZES HAVE REACHED THEIR MINIMUM VALUE
C
5002  IF1=1
C
C LOOP ASSIGNS J TO ALL ELEMENTS IN ARRAY ICS, EXCEPT
C FOR ELEMENT I
C
      DO 5004 K=1,IGR
      IF (K.EQ.I) GOTO 5004
      IF (J.LE.NS(IGR-K+1)) THEN
      ICS(K)=J
      IF1=0
      ELSE
C
C IF J EXCEEDS NS OF A GROUP, THE MINIMUM SIZE IS ASSIGNED
C
      ICS(K)=NS(IGR-K+1)
      END IF
5004  CONTINUE
      IF (IF1.EQ.1) GOTO 5010
C
C ASSIGNING OF ACTUAL PIPE SIZES WITH SUBROUTINE SIZE
C
      CALL SIZE (O,DI,XL,HW,CP,A,IBE,IEN,IGROU,ICS,SI,IGR,HWCC)
      IF2=1
      IF3=1
      POLD=1.E38
      TP1=0
C
C LOOP ASSIGNS CHARACTER '0' TO ALL POSITIONS
C IN STRING PATT. STRING IS USED TO FLAG LOADING PATTERNS WHICH HAVE
C FAILED BY ASSIGNING CHARACTER '1' TO ITS POSITION.
C
      DO 5006 M0=1,MUNU
5006  PATT(M0:M0)='0'
```

```
C
C ENUMERATING LOADING PATTERNS
C
      DO 5008 M0=1,MUNU
C
C TESTING WHETHER PATTERN HAS ALREADY FAILED.  IF SO FLAG IF2 IS
C SET TO 0
C
      IF (PATT(M0:M0).EQ.'1') THEN
        IF2=0
        GOTO 5008
      END IF
C
C ASSIGNING OUTPUTS FOR LOADING PATTERN
C
      DO 5007 K=1,O(2)
5007  DO(K)=DM(M0,K)
C
C CALCULATING PRESSURE DISTRIBUTION WITH OPTBAL
C
      CALL OPTBAL (XP,P8,P88,M0,PT,EF,TP,N3)
C
C ADJUSTING PRESSURE DISTRIBUTION WITH PUMPH
C
      CALL PUMPH(N3,IBI,IEI,IPI,O,CP,XL,HE,EL,TP,PT,EF,XP,M0)
C
C ACCUMULATING TOTAL PUMPING COST FOR VARIOUS PATTERNS
C
      TP1=TP1+741.6*ENCO*TP*FAC
      IF (P8.LT.R22) THEN
        DO 50071 K=1,IGR
50071  IFS(K0+1,K)=ICS(K)
        IF (K0+1.LE.100) K0=K0+1
        PATT(M0:M0)='1'
        GOTO 5010
      END IF
      IF (P8.LT.0) IF2=0
      IF (P8.GE.R22) IF3=0
      IF (P8.LT.POLD) THEN
        POLD=P8
        MOLD=M0
      END IF
5008  CONTINUE
      IF (IF2.EQ.1) THEN
C
C ACCUMULATING TOTAL SYSTEM COST, C9 INLUDING PUMPING COST
C
        C9=TP1
        DO 5009 M0=1,IGR
          J0=ICS(IGR+1-M0)
5009  C9=C9+TC(M0,J0)
C
C IF C9 IS LESS THAN THE SO FAR BEST SOLUTION, C8 THE SIZE
C COMBINATION IS STORED AS THE BEST SOLUTION
C
          IF (C9.LT.C8) THEN
            DO 5011 J2=1,IGR
5011  IBS(J2)=ICS(J2)
            BESTP=POLD
            C8=C9
```

```

        PAT=MOLD
        TP2=TP1
        CALL SOQU( ISQ, POLD, C8, MOLD, IGR, ICS, SPR, SCO, ISP, ISS)
    END IF
END IF
IF (IF3.EQ.1) GOTO 5010
C
C INCREMENTING      J
C
        J=J+1
        GOTO 5002
5010 CONTINUE
        IFLG=0
        RETURN
        END
C
C*****ENUMERATOR*****
C SUBROUTINE ENUMER ENUMERATES AND TESTS ALL POSSIBLE COMINATIONS
C
        SUBROUTINE ENUMER(PAT,FAC,IFS,ST)
        CHARACTER ST*90,OUTP*90
        REAL*8 A,S,G,HE
        INTEGER O,PNL
        PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
        COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
        COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
        COMMON /TOPOL/
IBE(PNL), IEN(PNL), IPI(PNL), INO(PNL), IBI(PNL), IEI(PN
1L)
        COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
        COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
        COMMON /SOLUT/ ISS(20,15),IBS(15)
        COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
        COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
        COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
        1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
        DIMENSION ICS(15),IFS(100,15)
        CHARACTER RSN*8
        INTEGER*4 N,N0
        N=1
        XN0=.1
        N0=0
        O(9)=0
C
C PRINTING OF THE NUMBER OF SIZES REMAINING IN EACH GROUP AND THE
C TOTAL NUMBER OF POSSIBLE COMBINATIONS
C
        PRINT*
        DO 500 I=1,IGR
        DO 5001 II=1,O(1)
5001 IF (IGROU(II).NE.0) GOTO 4999
        GOTO 500
4999 N=N*NS(I)
        WRITE(6,300) I,NS(I)
300 FORMAT(1X,' GROUP ',I2,', # OF SIZES: ',I2)
500 ICS(I)=1
        PRINT*
        WRITE(6,301) N
301 FORMAT(1X,I7,' COMBINATIONS WILL BE TESTED.')
        PRINT*
```

```
C
C   ***INITIALIZING***
C
      K=1
      ICOM=1
      ISQ=0
      J=1
      L=IGR
      GOTO 503
501  J=1
      IF (ST(2:2).EQ.'E') THEN
      CALL RESUL (O,SI,IGR,ICS,IGROU,OUTP)
      IF (ICOM.EQ.1) WRITE(6,10)
10   FORMAT(' COMB. #    COST    MIN. PRESS.  REASON FAILED    ',
>'GROUP SIZES')
      IF (IFLAG.GT.0) THEN
      WRITE(6,20) ICOM,C9,RSN,OUTP(1:6*IGR)
20   FORMAT(1X,I5,1X,F10.0,3X,'- - - -',7X,A8,2X,A)
      ELSE
      WRITE(6,30) ICOM,C9,P8,RSN,OUTP(1:6*IGR)
30   FORMAT(1X,I5,1X,F10.0,F9.1,8X,A8,2X,A)
      ENDIF
      ENDIF
      IF (IFLAG.EQ.-2) ICS(IGR)=NS(1)
      IFLAG=0
      ICOM=ICOM+1
C
C CALCULATING L , THE CORRESPONDING ELEMENT IN ARRAY ICS
C
502  L=IGR+1-J
C
C INCREMENTING COLLATING WEIGHT OF CHARACTER L BY ONE
C
      K=ICS(L)+1
      IF (K.LE.NS(J)) GOTO 503
C
C IF K EXCEEDS THE NUMBER OF SIZES, ELEMENT L IN ICS IS FILLED
C WITH A 1 AND GROUP NUMBER, J, IS INCREMENTED BY ONE
C
      ICS(L)=1
      J=J+1
      IF (J.LE.IGR) GOTO 502
C
C IF J IS LARGER THAN THE TOTAL NUMBER OF GROUPS, THE
C ENUMERATIONS ARE COMPLETED AND OPTIMIZATION TERMINATED
C
      RETURN
503  ICS(L)=K
C
C COUNTING THE NUMBER OF COMBINATIONS GENERATED
C
      N0=N0+1
      IF (ST(2:2).NE.'E') THEN
      IF ((REAL(N0))/(REAL(N)).GT.XN0) THEN
      WRITE(6,4444) XN0*100.
4444  FORMAT (1X,'JOB IS',F5.0,' % COMPLETE.')
      CALL INTER (C8,IGR,O,SI,IBS,IGROU,BESTP)
      XN0=XN0+.1
      END IF
      END IF
```

```
C
C*****COST*****
C
      C9=0
C
C ACCUMUALTING SYSTEM COST
C
      DO 600 I=1,IGR
      J=ICS(IGR+1-I)
600   C9=C9+TC(I,J)
C
C TESTING SUM OF PIPING COST AND LOWEST POSSIBLE PUMPING COST
C AGAINST THE COST OF THE BEST FUNCTIONAL SOLUTION TIMES R23
C
      IF (C9+TP0.GE.R23*C8) THEN
      C9 = C9 + TP0
      IFLAG=1
      RSN = 'COST      '
      GOTO 501
      END IF
C
C*****QUEU*****
C
      CALL QUEU (IQ,K0,IFS,IGR,ICS)
C
C A VALUE OF 1 FOR IQ INDICATES THE TEST AGAINST NONFUNCTIONAL
C SYSTEMS HAS FAILED
C
      IF (IQ.EQ.1) THEN
      IFLAG=1
      RSN = 'SIZE      '
      GOTO 501
      END IF
C
C*****SIZE*****
C
      CALL SIZE (O,DI,XL,HW,CP,A,IBE,IEN,IGROU,ICS,SI,IGR,HWCC)
C
C*****PRESSURE COMPUTATIONS*****
C
      POLD=1.E38
      TP1=0
C
C LOOP ENUMERATES LOADING PATTERNS
C
      DO 620 M0=1,MUNU
C
C INNER LOOP ENUMERATES THE NODES AND ASSIGNS OUTPUTS ACCORDING
C TO THE LOADING PATTERN
C
      DO 609 K=1,O(2)
609   DO(K)=DM(M0,K)
C
C DETERMINING PRESSURE DISTRIBUTIONS WITH OPTBAL
C
      CALL OPTBAL (XP,P8,P88,M0,PT,EF,TP,N3)
      IF (O(9).GT.0.5) THEN
      CALL TROUBL(IGR,SI,ICS,IGROU,O)
      O(9)=0
      RSN = 'PUMP      '

```



```
        GOTO 501
      END IF
C
C UPDATING PRESSURE DISTRIBUTION WITH PUMPH
C
      CALL PUMPH(N3,IBI,IEI,IPI,O,CP,XL,HE,EL,TP,PT,EF,XP,M0)
C
C ACCUMULATING PRESENT WORTH OF THE PUMPIN COST FOR ALL
C LOADING PATTERNS
C
      TP1=TP1+741.6*ENCO*TP*FAC
      C9=C9+741.6*ENCO*TP*FAC
C
C IF TOTAL COST EXCEEDS THE COST OF THE BEST FUNCTIONAL
C SOLUTION MULTIPLIED BY R23, CONTROL RETURNS TO ENUMERATOR
C
      IF (C9.GE.R23*C8) THEN
        RSN = 'COST      '
        GOTO 501
      END IF
      IF (P8.GE.R22) GOTO 610
C
C IF MINIMUM PRESSURE IS LESS THAN THE PRESSURE TOLERANCE, THE
C COMBINATION IS ENTERED INTO THE ARRAY OF NON-FUNCTIONAL
C SOLUTIONS
C
      DO 6091 K=1,IGR
6091  IFS(K0+1,K)=ICS(K)
      IFLAG=-2
      IF (K0+1.LE.100) K0=K0+1
      RSN = 'PRESSURE'
      GOTO 501
610  IF (P8.GT.POLD) GOTO 620
C
C IF MINIMUM PRESSURE IS LESS THAN POLD, POLD AND THE PATTERN
C NUMBER MOLD ARE UPDATED
C
      POLD=P8
      MOLD=M0
620  CONTINUE
      IF (POLD.LT.0.OR.C9.GT.C8) GOTO 640
      DO 6201 J2=1,IGR
6201  IBS(J2)=ICS(J2)
      BESTP=POLD
      C8=C9
      PAT=MOLD
      TP2=TP1
C
C TESTING OF SOLUTION FOR INCLUSION IN THE QUEUE OF PARETO
C OPTIMAL SOLUTIONS BY SOQU
C
640  CALL SOQU(ISQ,POLD,C9,MOLD,IGR,ICS,SPR,SCO,ISP,ISS)
      RSN = '          '
      GOTO 501
      END
C
C***** TERMINATION *****
C SUBROUTINE OPTERM ASSIGNS PIPE SIZES OF THE FUNCTIONAL
C COMBINATION, AND THE OUTPUTS FOR THE PATTERN WHICH GENERATED
C THE LOWEST PRESSURE
```

```
C
SUBROUTINE OPTERM(PAT)
REAL*8 A,S,G,HE
INTEGER O,PNL
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
COMMON /SOLUT/ ISS(20,15),IBS(15)
COMMON /INTEGR/ ISQ,IGR,MUNU,IGROU(PNL),ICATE(PNL),N3(PNL)
COMMON /REALS/ C8,HWCC,R22,R23,TP2,BESTP,TP0
COMMON /ARRAYS/ SPR(14),SCO(14),DM(5,PNL),EF(20),PT(5,20)
1,SI(15,10),XP(5,PNL),NS(15),TC(15,10),ISP(14)
CHARACTER OUTP*90
DIMENSION ICS(15)

C
C ASSIGNING OF IBS, THE SIZE COMBINATION FOR THE BEST FUNCTIONAL
C SOLUTION TO THE COMBINATION ARRAY ICS
C
700 DO 7001 K=1,IGR
7001 ICS(K)=IBS(K)
C
C O(12)=-2 INDICATES TO OPTBAL THAT THIS WILL BE THE FINAL PASS
C
O(12)=-2
CALL SIZE (O,DI,XL,HW,CP,A,IBE,IEN,IGROU,ICS,SI,IGR,HWCC)
M0=PAT

C
C LOOP ENUMERATES ALL NODES AND ASSIGNS OUPUTS FOR THE PATTERN
C
DO 702 K=1,O(2)
702 DO(K)=DM(M0,K)
CALL OPTBAL (XP,P8,P88,M0,PT,EF,TP,N3)
CALL PUMPH(N3,IBI,IEI,IPI,O,CP,XL,HE,EL,TP,PT,EF,XP,M0)

C
C PRINTING OF PIPE SIZES FOR ALL GROUPS FOR THE OPTIMUM SOLUTION
C
PRINT*
PRINT*, ' OPTIMUM SOLUTION: '
IF (C8.GT..3E38) THEN
PRINT*, 'NO SOLUTION WHICH MEETS PRESSURE REQUIREMENT'
GOTO 710
END IF
PRINT 711,(I,I=1,IGR)
711 FORMAT (/, ' GROUP ',I5,14I6)
CALL RESUL (O,SI,IGR,ICS,IGROU,OUTP)
PRINT*, 'DIAM. ',OUTP(1:6*IGR)
PRINT 698,C8
PRINT 699,BESTP
PRINT 697,INT(PAT)
698 FORMAT ( ' AT COST OF ',F10.0)
699 FORMAT ( ' MIN. PRESSURE ',F11.1)
697 FORMAT ( ' IN PATTERN ',I11)
IF (O(12).LT.-2.9) THEN
PRINT 721,TP2
721 FORMAT (1X, ' PRESENT WORTH OF PUMPING COST ',F12.0)
PRINT*, ' CHARACTERISTIC CURVES MUST BE ASSIGNED TO PUMPS ',
```

```
1 'LISTED ABOVE'
  PRINT*,' BEFORE RUNNING SIMULATION AGAIN.'
  END IF
710 PRINT*
  PRINT*,' ALTERNATIVE SOLUTIONS:'
  PRINT*

C
C LOOP ENUMERATES ALL COMBINATIONS IN THE QUEUE OF PARETO OPTIMAL
C SOLUTIONS.
C
  DO 720 I=1,ISQ
  DO 717 K=1,IGR
717  ICS(K)=ISS(I,K)
  DO 718 K=1,IGR
718  IF (ICS(K).NE.IBS(K)) GOTO 719
  GOTO 720

C
C CALLING RESUL TO GENERATE THE OUTPUT STRING OUTP
C
719  IF (SCO(I).GT.R23*C8) GOTO 720
  CALL RESUL (O,SI,IGR,ICS,IGROU,OUTP)

C
C PRINTING OF THE OUTPUT STRING, THE MINIMUM PRESSURE, THE
C CORRESPONDING PATTERN NUMBER, AND THE COST OF THE COMBINATION
C
  PRINT*,'DIAM. ',OUTP(1:6*IGR)
  PRINT 701,SPR(I),ISP(I),SCO(I)
701  FORMAT (' MIN.PR. ',F5.1,' IN PATTERN ',I2,' COST ',F10.0,/)
720  CONTINUE
  O(12)=0
  RETURN
  END

C
C SUBROUTINE OPTSTO STORES OPTIMIZATION PARAMETERS
C
  SUBROUTINE
OPTSTO(MUNU,IGR,R22,R23,XP,DM,EF,PT,SI,NS,HWCC,IGROU,IC
  LATE,O)
  COMMON /PRINT/ IPM,IPP,IPE
  CHARACTER ST*90
  INTEGER O,PNL
  PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
  DIMENSION XP(5,PNL),DM(5,PNL),EF(20),PT(5,20),SI(15,10),NS(15),O
  1(15)
  DIMENSION IGROU(PNL),ICATE(PNL)

C
C PROMPTING AND READING OF FILE NAME UNDER WHICH DATA IS TO BE STORED
C
3000 IF (IPP.EQ.1) PRINT 3001
3001 FORMAT (/, ' ENTER FILE NAME',/)
  READ(*,2,END=999) ST
  IF (IPE.EQ.1) PRINT*,ST
2  FORMAT(A60)
  OPEN (1,ERR=99,FILE=ST,RECL=56)

C
C WRITING OF OPTIMIZATION PARAMETERS INTO THE FILE
C
  WRITE (1,10,ERR=99) MUNU,IGR,R22,R23
10  FORMAT(2I3,2E14.7)
  WRITE (1,20,ERR=99) ((XP(I,J),DM(I,J),I=1,5),J=1,O(2))
```

```
20   FORMAT(4E14.7)
    WRITE (1,20,ERR=99) (EF(J), (PT(I,J), I=1,5), J=1,20)
    WRITE (1,20,ERR=99) ((SI(I,J), I=1,15), J=1,10)
    WRITE (1,30,ERR=99) (NS(I), I=1,15), HWCC
30   FORMAT(15I3, E10.3)
    WRITE (1,40,ERR=99) (IGROU(I), ICATE(I), I=1, O(1))
40   FORMAT(10I3)
C
C   REWINDING AND CLOSING FILE
C
    REWIND 1
    CLOSE (1)
    RETURN
99   PRINT*, 'PROGRAM CANNOT ACCESS THIS FILE'
    RETURN
999  CALL BLANK
    GOTO 3000
    END
C
C*****RETRIEVING DATA*****
C SUBROUTINE OPTRET RETREIVES THE OPTIMIZATION PARAMETERS
C
    SUBROUTINE
OPTRET(MUNU, IGR, R22, R23, XP, DM, EF, PT, SI, NS, HWCC, IGROU, IC
1ATE, O)
    CHARACTER ST*90
    COMMON /PRINT/ IPM, IPP, IPE
    INTEGER O, PNL
    PARAMETER (PNL=800, LNL=750, MNL=40, IA=7500)
    DIMENSION XP(5, PNL), DM(5, PNL), EF(20), PT(5, 20), SI(15, 10), NS(15), O
1(15)
    DIMENSION IGROU(PNL), ICATE(PNL)
C
C   PROMPTING FOR FILE NAME
C
3010 IF (IPP.EQ.1) PRINT 3001
3001 FORMAT (/, ' ENTER FILE NAME ', /)
    READ(*, 2, END=999) ST
    IF (IPE.EQ.1) PRINT*, ST
2   FORMAT(A60)
C
C   OPENING FILE ST
C
    OPEN (1, ERR=99, FILE=ST, STATUS='OLD', RECL=56)
    REWIND 1
    READ (1, 10, ERR=99) MUNU, IGR, R22, R23
10  FORMAT(2I3, 2E14.7)
C
C   REINITIALIZING VARIABLES PRIOR TO READING
C
    DO 8 J=1, 20
        EF(J) = 0
        DO 8 I=1, 5
            PT(I, J) = 0
8   CONTINUE
    DO 9 J = 1, PNL
        IGROU(J) = 0
        ICATE(J) = 0
        DO 9 I = 1, 5
            XP(I, J) = 0
```

```
          DM(I,J) = 0
9      CONTINUE
      READ (1,20,ERR=99,END=99) ((XP(I,J),DM(I,J),I=1,5),J=1,O(2))
20     FORMAT(4E14.7)
      READ (1,20,ERR=99,END=99) (EF(J),(PT(I,J),I=1,5),J=1,20)
      READ (1,20,ERR=99,END=99) ((SI(I,J),I=1,15),J=1,10)
      READ (1,30,ERR=99,END=99) (NS(I),I=1,15),HWCC
30     FORMAT(15I3,E10.3)
      READ (1,40,ERR=99,END=99) (IGROU(I),ICATE(I),I=1,O(1))
40     FORMAT(10I3)
C
C      REWINDING AND CLOSING FILE
C
      REWIND 1
      CLOSE (1)
      RETURN
C
C      ERROR MESSAGE IF FILE IS NOT ACCESSIBLE
C
99     PRINT*, 'PROGRAM CANNOT ACCESS THIS FILE'
      RETURN
999    CALL BLANK
      GOTO 3010
      END
      SUBROUTINE OPTBAL(XP,P8,P88,M0,PT,EF,TP,N3)
      INTEGER O,PNL
      REAL*8 A,S,G,HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
      IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /MATRI/ M(LNL),N1(LNL,MNL),N2(LNL,MNL)
      DIMENSION PT(5,20),EF(20),XP(5,PNL),N3(PNL)
      DO 400 I1=1,O(2)
          G(I1)=DO(I1)+(ABS(EL(I1))-100)*1.E10
          S(I1)=1.E10
          IF (DO(I1).GT.9.E9) GOTO 400
          G(I1)=-DO(I1)/448.831
          S(I1)=0
400    CONTINUE
      P88=0
      I6=0
      O4=.1
      CALL PARALE (A,XL,O,IBI,IEI)
      IFOP=1
      CALL EQSOLV(ST,IFOP,I6,XP,P8,P88,M0,PT,EF,TP,N3,O4)
      RETURN
      END
C
C      SUBROUTINE RESUL PUTS THE SIZE OF EACH GROUP INTO STRING OUTP
C
      SUBROUTINE RESUL (O,SI,IGR,ICS,IGROU,OUTP)
      INTEGER O,PNL,POS
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION O(15),SI(15,10),IGROU(PNL),ICS(15)
      CHARACTER OUTP*90,NUMB*11
      NUMB='0123456789:'
```

```
C INITIALIZING STRING OUTF WITH BLANKS
C
      OUTF(1:6)='      '
      DO 109 I=2,IGR
109   OUTF(I*6-5:I*6)=OUTF(1:6)
C
C ENUMERATING POSITIONS OF ARRAY ICS FROM LOWER TO HIGHER
C
      DO 120 I=1,IGR
      DO 121 II=1,O(1)
121   IF (IGROU(II).EQ.I) GOTO 122
      GOTO 120
C
C      ASSIGNING THE SIZE NUMBER FOR THE GROUP TO K
C
122   K=ICS(IGR+1-I)
C
C -.01 FLAGS CLEANING
C
      IF (SI(I,K).LT.-.01) THEN
        OUTF(I*6:I*6)='C'
        GOTO 120
      END IF
C
C ELIMINATION
C
      IF (ABS(SI(I,K)).LT..01) THEN
        OUTF(I*6:I*6)='E'
        GOTO 120
      END IF
C
C DETERMINING THE DIGITS TO THE RIGHT OF THE DECIMAL POINT
C
      A=MOD(SI(I,K),1.)
      M=0
C
C ASSIGNING OF THE CHARACTER WHICH IS EQUAL TO THE DIGIT TO THE
C FAR RIGHT POSITION
C
      POS = INT(1.5+A*10)
      OUTF(I*6:I*6)=NUMB(POS:POS)
      IF (OUTF(I*6:I*6).EQ.':') THEN
        OUTF(I*6:I*6)='0'
C
C M=1 INDICATES ROUNDING NEEDS TO TAKE PLACE
C
      M=1
      END IF
C
C SETTING OF THE DECIMAL POINT
C
      OUTF(I*6-1:I*6-1)='.'
C
C J IS THE POSITION IN FRONT OF THE DECIMAL POINT
C
      J=I*6-2
C
C N REPRESENTS THE REMAINING DIGITS TO BE INSERTED INTO STRING OUTF
C
      N=INT(SI(I,K))
```

```
110   IF (N+M.EQ.0) GOTO 120
C
C DETERMINING LAST DIGIT IN N
C
      K=MOD(N,10)
      POS=1+K+M
      OUTP(J:J)=NUMB(POS:POS)
      M=0
      IF (OUTP(J:J).EQ.':') THEN
        OUTP(J:J)='0'
        M=1
      END IF
      N=N/10
      J=J-1
      IF (MOD(J,6).NE.0) GOTO 110
C
C * INDICATES A FIELD OVEFLOW
C
      OUTP(J+1:J+1)='*'
120   CONTINUE
      RETURN
      END
C
C*****OUT*****
C SUBROUTINE OUT PLACES THE USER LINK NUMBER OF THE LINK WITH
C INTERNAL LINK NUMBER IL INTO THE OUTPUT STRING AS THE L'TH ENTRY
C
      SUBROUTINE OUT(L,IL,IPI,OUTP)
      INTEGER PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION IPI(PNL)
      CHARACTER OUTP*90,NUMB*10
      NUMB='0123456789'
C
C ASSIGNING OF THE ENDING FIELD NUMBER TO VARIABLE J
C
      J=4*L
C
C INITIALIZING L'TH FIELD WITH BLANKS
C
      OUTP(J-3:J)='   '
C
C DETERMINING USER LINK NUMBER
C
      N=IPI(IL)
100   IF (N.EQ.0) GOTO 110
C
C ASSIGNING ALL DIGITS WITHIN THE ORIGINAL USER LINK NUMBER
C
      K=MOD(N,10)
      OUTP(J:J)=NUMB(1+K:1+K)
      N=N/10
      J=J-1
      GOTO 100
110   RETURN
      END
C
C*****SIZE*****
C SUBROUTINE SIZE ASSIGNS THE PROPER SIZES TO THE ARRAY DI FOR
C EACH PIPE TO BE SIZED, IN ACCORDANCE WITH THE SIZE NUMBERS
```

```
C STORED IN ARRAY ICS
C
      SUBROUTINE SIZE (O,DI,XL,HW,CP,A,IBE,IEN,IGROU,ICS,SI,IGR,HWCC)
      REAL*8 A
      INTEGER O,PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION DI(PNL),XL(PNL),HW(PNL),CP(PNL)
      DIMENSION IBE(PNL),IEN(PNL),IGROU(PNL)
      DIMENSION A(IA)
      DIMENSION O(15)
      DIMENSION SI(15,10),ICS(15)
C
C LOOP ENUMERATES ALL LINKS
C
      DO 120 I=1,O(1)
C
C CHECKING FOR LINKS WHICH ARE PRV'S OR PUMPS. THEY ARE SKIPPED
C
      IF (XL(I).GT.999999.) GOTO 120
      IF (CP(I).LT.0) GOTO 120
C
C DETERMINING GROUP NUMBER OF THE LINK
C
      J=IGROU(I)
      IF (J.EQ.0) GOTO 120
C
C DETERMINING THE SIZE NUMBER, K
C
      K=ICS(IGR+1-J)
      IF (ABS(DI(I)-SI(J,K)/12).LT..001.AND.O(12).GT.-1.9) GOTO 120
C
C SMALL ENTRIES IN THE COEFFICIENT MATRIX ARE RESET TO 1
C
      IF (A(I).LT.1.E-5) THEN
      A(I)=1
      GOTO 100
      END IF
      A(I)=A(I)*CP(I)
C
C W7 IS THE PIPE DIAMETER TO BE ASSIGNED IN FEET
C
100   W7=ABS(SI(J,K)/12)
C
C Q0 IS THE HAZEN WILLIAMS COEFFICIENT OF THE PIPE
C
      Q0=ABS(HW(I))
      IF (SI(J,K).LT.-.001) THEN
      IP=0
C
C SEARCHING ARRAY IBE AND IEN FOR A PIPE WITH THE SAME BEGINNING
C AND ENDING NODES (BUT DIFFERENT LINK NUMBERS)
C
      DO 110 IP=1,O(1)
110   IF (IBE(IP).EQ.IBE(I).AND.IEN(IP).EQ.IEN(I).AND.IP.NE.I) GOTO
1101
C
C Q1 IS THE HAZEN-WILLIAMS COEFFICIENT OF THE PIPE TO BE CLEANED
C
1101  Q1=ABS(HW(IP))
      IF (O(12).LT.-1.9) THEN
```



```
C
C HAZEN-WILLIAMS COEFICIENT FOR THE CLEANED PIPE IS ASSIGNED TO
C THE OLD PIPE IP
C
      HW(IP)=HWCC
C
C CALCULATING NEW CHARACTERISTIC PIPE COEFFICIENT AND THE
C CORRESPONDING ENTRY INTO THE COEFFICIENT MATRIX
C
      CP(IP)=4.72*ABS(XL(IP))/(HWCC**1.85*DI(IP)**4.87)
      A(IP)=1/CP(IP)
C
C CALCULATING NEW CHARACTERISTIC PIPE COEFFICIENT AND CORRESPONDING
C ENTRY INTO THE COEFFICIENT MATRIX
C
      CP(I)=4.72*ABS(XL(I))/(ABS(HW(I))**1.85*(1./1000.))**4.87)
      A(I)=1.E-6
      DI(I)=1./1000.
      GOTO 120
      END IF
C
C CALCULATING PIPE DIAMETER OF THE NEW PIPE, I, SUCH THAT THE
C OLD AND NEW PIPE COMBINED WILL HAVE THE SAME EFFECT AS THE OLD
C PIPE ALONE AFTER CLEANING
C
      W7=((HWCC-Q1)/Q0)**.38*DI(IP)
      END IF
      IF (ABS(W7).LT..001) W7=.001
C
C ASSINGING OF THE NEW CHARACTERISTIC PIPE COEFFICIENT, THE ENTRY
C INTO THE COEFFICIENT MATRIX, AND THE DIAMETER
C
      CP(I)=4.72*ABS(XL(I))/(Q0**1.85*W7**4.87)
      A(I)=A(I)/CP(I)
      DI(I)=SI(J,K)/12.
120  CONTINUE
      RETURN
      END
C
C *****PUMP*****
C SUBROUTINE PUMP FINDS ZONES WITHIN A NETWORK WHICH ARE FED BY
C A SINGLE PUMP AND DO NOT HAVE A SUPPLY POINT FEEDING DIRECTLY
C INTO THIS ZONE
C
      SUBROUTINE PUMP(N3,IBI,IEI,O,CP,DO)
      INTEGER O,PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION O(15),CP(PNL),DO(PNL)
      DIMENSION IBI(PNL),IEI(PNL)
      DIMENSION N1(PNL),N3(PNL)
      I=1
C
C LOOP ENUMERATES ALL LINKS
C
      DO 300 J0=1,O(1)
      IF (CP(J0).GT.0) GOTO 300
C
C K0 IS THE ENDING NODE (USER NUMBER) OF THE PUMP FEEDING ZONE J0
C
      K0=IEI(J0)
```

```
      IF (DO(K0).GT.9E9) GOTO 300
      N3(K0)=I
C
C SEARCHING FOR LINKS WHICH HAVE A BEGINNING NODE EQUAL TO K0
C
40    DO 50 J1=1,O(1)
      IF (J1.EQ.J0) GOTO 50
      IF (IBI(J1).NE.K0) GOTO 50
      IF (CP(J1).LT.0) GOTO 50
C
C K1 IS THE ENDING USER NODE NUMBER WITH CORRESPONDING
C INTERNAL NUMBER J1
C
      K1=IEI(J1)
C
C TESTING FOR SUPPLY POINT
C
      IF (DO(K1).GT.9E9) GOTO 250
C
C ASSIGNING ZONE NUMBER TO NODE
C
      N3(K1)=I
      O(14)=1
50    CONTINUE
C
C SEARCHING FOR LINKS WITH ENDING NODE EQUAL TO K0
C
100   DO 150 J1=1,O(1)
      IF (J1.EQ.J0) GOTO 150
      IF (IEI(J1).NE.K0) GOTO 150
C
C TESTING FOR A PUMP PUMPING INTO THE ZONE
C
      IF (CP(J1).LT.0) GOTO 250
      K1=IBI(J1)
      IF (DO(K1).GT.9E9) GOTO 250
      N3(K1)=I
150   CONTINUE
C
C IF NO MORE LINKS WITH ENDING NODE EQUAL TO ST, NODE K0 IS
C FLAGGED WITH A 1
C
200   N1(K0)=1
C
C SEARCHING FOR NODES IN THE SAME ZONE WHICH HAVE NOT YET
C BEEN CHECKED
C
      DO 210 K0=1,O(2)
210   IF (N3(K0).EQ.I.AND.N1(K0).EQ.0) GOTO 211
      GOTO 270
211   GOTO 40
C
C REMOVING ELEMENTS EQUAL TO I FROM ARRAYS N1 AND N3
C
250   DO 260 L=1,O(2)
      IF (N3(L).EQ.I) THEN
        N3(L)=0
        N1(L)=0
      END IF
260   CONTINUE
```

```
GOTO 300
C
C INCREMENTING I BY 1 AFTER A ZONE IS COMPLETED
C
270 I=I+1
300 CONTINUE
RETURN
END
C
C *****PUMPH*****
C SUBROUTINE PUMPH SHIFTS THE PRESSURE LEVEL IN THE ZONES AS
C DETERMINED IN SUBROUTINE PUMP SUCH THAT THE LOWEST PRESSURE
C IS EQUAL TO 0
C
SUBROUTINE PUMPH(N3, IBI, IEI, IPI, O, CP, XL, HE, EL, TP, PT, EF, XP, M0)
REAL*8 HE
INTEGER O, PNL
PARAMETER (PNL=800, LNL=750, MNL=40, IA=7500)
DIMENSION O(15), CP(PNL), XL(PNL), HE(PNL), PT(5, 20), EF(20), XP(5, PNL
1)
DIMENSION EL(PNL), IBI(PNL), IEI(PNL), IPI(PNL)
DIMENSION N3(PNL)
I=1
10 II=0
C
C P8M IS THE LOWEST PRESSURE IN A ZONE
C
P8M=1E10
DO 50 J1=1, O(2)
C
C SEARCHING ALL NODES WHICH BELONG TO ZONE I, DETERMINING PRESSURE
C P8 AT THE NODE, AND UPDATE P8M IF P8 IS LESS THAN P8M
C
IF (N3(J1).NE.I) GOTO 50
P8=HE(J1)-(ABS(EL(J1))+XP(M0, J1)*2.308)
IF (P8.LT.P8M) P8M=P8
II=1
50 CONTINUE
IF (II.EQ.0) GOTO 200
C
C SEARCHING ALL NODES WHICH BELONG TO ZONE I, AND SHIFTING
C HEADS BY P8M
C
DO 110 J1=1, O(2)
IF (N3(J1).NE.I) GOTO 110
HE(J1)=HE(J1)-P8M
110 CONTINUE
150 I=I+1
GOTO 10
200 I=0
C
C LOOP ENUMERATES ALL LINKS
C
IPMP=0
DO 210 J1=1, O(1)
C
C TESTING FOR A LINK AS A PUMP
C
IF (CP(J1).GT.0.) GOTO 210
IPMP=IPMP+1
```

```
C
C DETERMINING ENDING NODE OF THE PUMP
C
      K1=IEI(J1)
      IF (N3(K1).EQ.0) GOTO 210
C
C BEGINNING NODE IS DETERMINED
C
      K0=IBI(J1)
      IF (HE(K1)-HE(K0).LT.0.) GOTO 210
C
C ACCUMULATING PUMPING ENERGY
C
      TP=TP+XL(J1)*(HE(K1)-HE(K0))*PT(M0,IPMP)/EF(IPMP)
      IF (O(12).LT.-1.9) THEN
        IF (I.EQ.0) THEN
C
C IF LAST PASS THROUGH SUBROUTINE OPTBAL,PUMP NUMBER, FLOW, AND
C HEAD FOR ALL PUMPS ARE PRINTED
C
          PRINT*
          PRINT*,' PUMPS WITH UNSPECIFIED CHARACTERISTIC CURVE:'
          I=1
          O(12)=-3
          END IF
          PRINT 201,IPI(J1),XL(J1)*448.831,HE(K1)-HE(K0)
201    FORMAT (1X,' PUMP # ',I5,' FLOW ',F5.0,' HEAD ',F7.1)
          END IF
210    CONTINUE
          RETURN
          END
C
C PRINTING OF BEST SOLUTION
C
      SUBROUTINE INTER (C8,IGR,O,SI,IBS,IGROU,BESTP)
      INTEGER PNL
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION SI(15,10),IGROU(PNL),IBS(15)
      CHARACTER OUTP*90
      PRINT*,' BEST SOLUTION:'
      IF (C8.GT..3E38) THEN
        PRINT*,' NO SOLUTION YET.'
        GOTO 10
      END IF
      PRINT 711,(I,I=1,IGR)
711    FORMAT (6X,' GROUP ',I5,14I6)
      CALL RESUL (O,SI,IGR,IBS,IGROU,OUTP)
      PRINT*,'          DIAM. ',OUTP(1:6*IGR)
      PRINT 698,C8
      PRINT 699,BESTP
698    FORMAT (7X,'AT COST OF      ',F10.0)
699    FORMAT (7X,'MIN. PRESSURE ',F11.1)
10     PRINT*
      RETURN
      END
C
C*****SOQU*****
C SUBROUTINE SOQU TESTS WHETHER THE COMBINATION ICS IS PARETO
C OPTIMAL AND IF IT IS PARETO OPTIMAL WHETHER IT ELMINATES OTHER
C COMBINATIONS PREVIOUSLY STORED IN THE SOLUTION QUEUE
```

```
C
      SUBROUTINE SOQU( ISQ, POLD, C9, MOLD, IGR, ICS, SPR, SCO, ISP, ISS)
      DIMENSION SPR(14), SCO(14), ISP(14), ICS(15), ISS(20,15)
C
C TESTING WHETHER THERE IS ALREADY AN ENTRY IN ISS
C
      IF (ISQ.EQ.0) THEN
        I=1
        GOTO 130
      END IF
C
C IC IS A FLAG INDICATING THAT A SOLUTION WAS REMOVED FROM THE
C QUEUE (IC=1). INITIALIZED WITH 0.
C
100   IC=0
C
C STORING THE NUMBER OF ENTRIES IN THE QUEUE UNDER VARIABLE
C I00
C
      I00=ISQ
C
C LOOP ENUMERATES ALL PREVIOUSLY STORED ENTRIES IN THE SOLUTION
C QUEUE
C
      DO 120 I=1,I00
C
C CONDITIONALS
C
      DO 115 K=1,IGR
115   IF (ISS(I,K).NE.ICS(K)) GOTO 116
      GOTO 180
116   IF (SPR(I).GE.POLD.AND.SCO(I).LE.C9) GOTO 140
      IF (SPR(I).GT.POLD.AND.SCO(I).GT.C9) GOTO 120
      IF (SPR(I).LT.POLD.AND.SCO(I).GE.C9) THEN
C
C ELIMINATING THE PREVIOUSLY STORED COMBINATION, WHICH IS NO
C LONGER PARETO OPTIMAL
C
      ISQ=ISQ-1
      DO 102 I0=I,I00-1
        SPR(I0)=SPR(I0+1)
        SCO(I0)=SCO(I0+1)
        ISP(I0)=ISP(I0+1)
      DO 102 K=1,IGR
102   ISS(I0,K)=ISS(I0+1,K)
      IC=1
      GOTO 100
      END IF
C
C IF NONE OF THE CONDITIONALS ARE TRUE, AN OPENING IN THE QUEUE
C IS CREATED
C
      DO 110 I0=I00,I,-1
        SPR(I0+1)=SPR(I0)
        SCO(I0+1)=SCO(I0)
        ISP(I0+1)=ISP(I0)
      DO 110 K=1,IGR
110   ISS(I0+1,K)=ISS(I0,K)
      GOTO 130
120   CONTINUE
```

```
C
C ASSIGNING THE NEW COMBINATION TO THE QUEU
C
      I=I-IC
130   SPR(I)=POLD
      SCO(I)=C9
      ISP(I)=MOLD
      DO 135 K=1,IGR
135   ISS(I,K)=ICS(K)
      ISQ=ISQ+1
C
C IF QUEU LENGTH IS LESS THAN 14 CONTROL RETURNS TO THE CALLING
C PROGRAM
C
140   IF (ISQ.LT.14) GOTO 180
      ISQ=13
C
C LOOP COUNTS THE NUMBER OF SOLUTIONS WITH MINIMUM PRESSURE
C LARGER THAN ZERO
C
      DO 150 I=1,14
      IF (SPR(I).LT.0) GOTO 160
150   CONTINUE
160   IF (I.LT.11) GOTO 180
C
C IF 10 OR MORE COMBINATIONS HAVE MINIMUM PRESSURE LARGER THAN
C ZERO, THE SOLUTION WITH THE HIGHEST PRESSURE AND HIGHEST COST
C IS ELIMINATED
C
      DO 170 I=1,13
      SPR(I)=SPR(I+1)
      SCO(I)=SCO(I+1)
      ISP(I)=ISP(I+1)
      DO 170 K=1,IGR
170   ISS(I,K)=ISS(I+1,K)
180   RETURN
      END
      SUBROUTINE TROUBL (IGR,SI,ICS,IGROU,O)
      DIMENSION SI(15,10),O(15),ICS(15)
      CHARACTER OUTF*90
      PRINT*
      PRINT*,' THE FOLLOWING COMBINATION FAILED BECAUSE OF PUMP ',O(9)
      CALL RESUL (O,SI,IGR,ICS,IGROU,OUTP)
      PRINT 711,(I,I=1,IGR)
711   FORMAT (6X,' GROUP ',I5,14I6)
      PRINT*,'          DIAM. ',OUTP(1:6*IGR)
      O(9)=0
      RETURN
      END
      SUBROUTINE TERMIN
      STOP
      END
C
C*****SIZEL*****
C SUBROUTINE SIZEL ASSIGNS ALL SIZES IN ONE GROUP TO OUTPUT STRING
C OUTF. WORKS LIKE SUBROUTINE RESUL
C
      SUBROUTINE SIZEL(SI,I,NS,OUTP)
      DIMENSION SI(15,10),NS(15)
      INTEGER POS
```

```
CHARACTER OUTF*90,NUMB*11
NUMB = '0123456789:'
C
C INITIALIZING STRING OUTF WITH BLANKS
C
      OUTF(1:6)='      '
      DO 109 K=2,NS(I)
109   OUTF(K*6-5:K*6)=OUTF(1:6)
      DO 120 K=1,NS(I)
C
C CLEANING
C
      IF (SI(I,K).LT.-.01) THEN
          OUTF(K*6:K*6)='C'
C
C ELIMINATION
C
      ELSE IF (ABS(SI(I,K)).LT..01) THEN
          OUTF(K*6:K*6)='E'
      ELSE
C
C DETERMINING DIGITS TO THE RIGHT OF THE DECIMAL POINT
C
          A=MOD(SI(I,K),1.)
          M=0
C
C ASSIGNING THE CHARACTER WHICH IS EQUAL TO THE DIGIT TO THE FAR
C RIGHT POSITION
C
          POS=1.5+A*10
          OUTF(K*6:K*6)=NUMB(POS:POS)
          IF (OUTF(K*6:K*6).EQ.':') THEN
              OUTF(K*6:K*6)='0'
              M=1
          END IF
C
C SETTING DECIMAL POINT
C
          OUTF(K*6-1:K*6-1)='.'
          J=K*6-2
          N=INT(SI(I,K))
110   IF (N+M.EQ.0) GOTO 120
          L=MOD(N,10)
          POS = 1+L+M
          OUTF(J:J)=NUMB(POS:POS)
          M=0
          IF (OUTF(J:J).EQ.':') THEN
              OUTF(J:J)='0'
              M=1
          END IF
          N=N/10
          J=J-1
          IF (MOD(J,6).NE.0) GOTO 110
          OUTF(J+1:J+1)='*'
          END IF
120   CONTINUE
      RETURN
      END
C
C ++++++COST++++++
```

```
C
SUBROUTINE COSTDA
COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
COMMON /PRINT/ IPM,IPP,IPE
CHARACTER ST*90
IF (KS.GT.0) GOTO 400
KS=0
KC=1
10 IF (IPM.EQ.0) GOTO 11
PRINT*,' COST DATA ROUTINE '
PRINT*
PRINT*,' SELECT PROGRAM OPTION:'
PRINT*
PRINT*,'          TO ENTER NEW COST DATA : ENTER  1  PRESS RETURN'
PRINT*,'          TO RETRIEVE COST DATA   :          2'
PRINT*
11 READ (*,2,END=59) ST
2  FORMAT (A60)
IF (IPE.EQ.1) PRINT*,ST
IF (ST(1:1).NE.'1'.OR.ST(1:1).NE.'2') THEN
  PRINT*,' INVALID ENTRY - TRY AGAIN'
  GOTO 10
END IF
IF (ST(1:1).EQ.'1') CALL INPCST
IF (ST(1:1).EQ.'2') CALL RETCST
GOTO 10

C
C*****PRICE-OPTION MENU*****
C
400 IF (IPM.EQ.0) GOTO 402
PRINT 401
401 FORMAT(//,' SELECT PROGRAM OPTION :'/)
PRINT*,'          MODIFY DATA      : ENTER  1  PRESS RETURN'
PRINT*,'          PRINT DATA        :          2'
PRINT*,'          STORE DATA         :          3'
PRINT*,'          RETRIEVE DATA      :          4'
PRINT*,'          PROGRAM CONTROL    :          8'
PRINT*,'          TERMINATE PROGRAM  :          9'
PRINT*
402 READ (*,2,END=61) ST
IF (IPE.EQ.1) PRINT*,ST
IF (ST(1:1).LT.'1'.OR.ST(1:1).GT.'9') GOTO 400
IF (ST(1:1).EQ.'1') CALL INPCST
IF (ST(1:1).EQ.'2') CALL PRNCST
IF (ST(1:1).EQ.'3') CALL STOCST
IF (ST(1:1).EQ.'4') CALL RETCST
IF (ST(1:1).EQ.'8') RETURN
IF (ST(1:1).EQ.'9') CALL TERMIN
GOTO 400
59 CALL BLANK
GOTO 10
61 CALL BLANK
GOTO 400
END

C
C *****ENTERING OF COST
DATA*****
C
SUBROUTINE INPCST
COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
```



```
COMMON /PRINT/IPM,IPP,IPE
DIMENSION VALUE(20)
CHARACTER KTYPE*4,KTYPF*4,TYPE*4,ST*90,IKEY1*32
IKEY1='SIZEPRICENERYEARINTEDELEEND ENR '
WRITE(6,6)
6  FORMAT(/' COST INPUT '/' TYPE KEYW FOR LIST OF KEYWORDS')
89  KTYPE(1:4)='SIZE'
    KTYPF='      '
90  IF (IPP.EQ.1) PRINT 1,KTYPE
1   FORMAT (/, ' C. KEYWORD IS ',A4, ' ENTER (KEYWORD) DATA LIST',/)
    READ (*,2,END=63) ST
    IF (IPE.EQ.1) PRINT*,ST
2   FORMAT (A60)
    IF (ST(1:4).EQ.'KEYW') THEN
        CALL KEYWRD(2,KTYPE)
        GOTO 89
    END IF
    TYPE=KTYPE

C
C   CALLING OF SUBROUTINE DECINP TO DECOMPOSE INPUT STRING
C
CALL DECINP(ST,VALUE,KTYPE,KTYPF,L)
IF (KTYPE.NE.'DELE') TYPE=KTYPE
IF (KTYPE.EQ.'END') RETURN
IPK1 = INDEX (IKEY1,KTYPE)
IF (IPK1.EQ.0) THEN
    PRINT*, 'ILLEGAL KEYWORD'
    GOTO 89
END IF
IF (L.EQ.0) THEN
    CALL KEYWRD(20,KTYPE)
    GOTO 90
END IF

C
C   KEYWORD SIZE
C
IF (TYPE.EQ.'SIZE') THEN
    LL=L
    IF (KTYPE.EQ.'DELE'.OR.KTYPF.EQ.'DELE') THEN

C
C   SEARCHING FOR THE SPECIFIED SIZE
C
        DO 91 J=1,KS
            IF (ABS(SIZ(J)-VALUE(1)).LT..001) GOTO 92
91         CONTINUE
            WRITE(6,85) VALUE(1)
85         FORMAT(' NO SIZE ',F3.0,' IN COST DATA BASE.')
            GOTO 89
92         KS=KS-1
        C
        C   ARRAYS SIZ AND COST ARE MOVED BEYOND THE SPECIFIED SIZE
        C   BY ONE LINE
        C
            DO 93 K=J,KS
                SIZ(K)=SIZ(K+1)
                DO 93 M=1,KC
93             COST(K,M)=COST(K+1,M)
                LL=LL-1
                GOTO 89
            END IF
```

```
DO 100 I=1,L
C
C TESTING FOR RENRTRY OF A SIZE VALUE ALREADY IN THE LIST
C REPEATED VALUES ARE IGNORED
C
DO 95 J=1,KS
IF (ABS(SIZ(J)-VALUE(I)).LT..001) THEN
LL=LL-1
GOTO 100
END IF
95 CONTINUE
C
C ASSIGNING OF THE VALUE ENTERED TO THE NEXT ELEMENT
C IN THE ARRAY SIZ
C
SIZ(KS+I)=VALUE(I)
100 CONTINUE
C
C UPDATING OF THE NUMBER OF SIZES
C
KS=KS+LL
GOTO 90
END IF
C
C *****KEYWORD PRIC*****
C
IF (TYPE.EQ.'PRIC') THEN
C
C IF ONLY TWO NUMERIC VALUES ARE ENTERED THE PROGRAM WILL
C TREAT THE ENTRIES AS A PIPE SIZE AND CORRESPONDING COST,
C LEAVCING THE PRICE FUNCTION UNCHANGED
C
IF (L.EQ.2) THEN
VALUE(3)=VALUE(2)
VALUE(2)=VALUE(1)
VALUE(1)=IC
L=3
END IF
C
C ASSIGNING THE COST FUNCTION TO IC, AND IF NECESSARY, UPDATE
C THE NUMBER OF PRICE FUNCTIONS, KC
C
IC=VALUE(1)
IF (IC.GT.KC) KC=IC
C
C RESETTING OF ARRAY COST TO ZERO FOR THE SPECIFIED PRICE
FUNCTION
C
IF (L.EQ.1) THEN
IF (KTYPE.EQ.'DELE'.OR.KTYPF.EQ.'DELE') THEN
DO 120 M=1, KS
120 COST(M,IC)=0
KTYPE='PRIC'
GOTO 90
END IF
C
C PROMPTING USER FOR THE COST OF THE DISPLAYED SIZE, FOR THE
C SPECIFIED PRICE FUNCTION (AS PRINTED)
C
30 IF (IPP.EQ.1) WRITE(6,5) IC
```

```
5      FORMAT(' FOR PRICE FUNCTION ',I2,' ENTER COST/FT FOR SIZES:')
      IF (IPP.EQ.1) PRINT*
      I=1
131     IF (SIZ(I).EQ.1) THEN
          PRINT*,' FOR CLEANING'
      ELSE
          IF (IPP.EQ.1) THEN
              WRITE(6,15) SIZ(I)
15      FORMAT(' FOR SIZE ',F3.0)
          PRINT*
          END IF
          END IF
          READ (*,2,END=132) ST
          IF (IPE.EQ.1) PRINT*,ST
C
C      CALLING OF SUBROUTINE DECINP TO DECOMPOSE INPUT STRING
C
      CALL DECINP(ST,VALUE,KTYPE,KTYPF,L)
      IF (KTYPE.EQ.'END') THEN
          CALL PRNCST
          RETURN
      END IF
C
C      ASSIGNING OF THE INPUT VALUE TO ARRAY COST
C
140     COST(I,IC)=VALUE(1)
      I = I + 1
      IF (I.LE.KS) GOTO 131
      GOTO 90
132     CALL BLANK
      GOTO 131
      END IF
      IF (L.EQ.3) THEN
C
C      SEARCHING FOR THE SPECIFIED SIZE, AND PRINTING OF WARNING
C      MESSAGE IF IF THE SIZE IS NOT PRESENT IN ARRAY SIZ.
C
      DO 150 J=1,KS
          IF (ABS(SIZ(J)-VALUE(2)).LT..001) GOTO 152
150     CONTINUE
          WRITE(6,16) VALUE(I)
16      FORMAT(' NO SIZE ',F3.0,' IN COST DATA BASE.')
          GOTO 90
C
C      ASSIGNING OF THE INPUT VALUE TO ARRAY COST
C
152     COST(J,IC)=VALUE(3)
          GOTO 90
      END IF
      END IF
C
C*****KEYWORD ENER*****
C
      IF (TYPE.EQ.'ENER'.AND.L.EQ.1) THEN
C
C      ENERGY COST IS ASSIGNED TO VARIABLE ENCO. NUMBER OF ENTRIES
C      MUST BE ONE
C
      ENCO=VALUE(1)
      GOTO 90
```

```
END IF
C
C *****KEYWORD YEAR*****
C
IF (TYPE.EQ.'YEAR'.AND.L.EQ.1.AND.VALUE(1).GE.1) THEN
  NY=VALUE(1)
  GOTO 90
END IF
C
C*****KEYWORD INTE*****
C
IF (TYPE.EQ.'INTE'.AND.L.EQ.1.AND.VALUE(1).GE.0) THEN
  XI=VALUE(1)/100.
  GOTO 90
END IF
C
C*****KEYWORD ENR*****
C
IF (TYPE.EQ.'ENR '.AND.L.EQ.2.AND.VALUE(2).GT.0) THEN
  IF (VALUE(1).GE.1.AND.VALUE(1).LE.KC) THEN
    J = VALUE(1)
    DO 10 I = 1,KS
10    COST(I,J) = COST(I,J)*VALUE(2)
  ELSE
    PRINT*,'ILLEGAL PRICE FCT #.'
  END IF
  GOTO 90
END IF
C
C ERROR MESSAGE - IF AN ILLEGAL FORMAT WAS USED IN RESPONSE
C TO THE INPUT PROMPT
C
PRINT*,' ERROR IN INPUT'
GOTO 90
63 CALL BLANK
GOTO 90
END
C
C *****SUBROUTINE PRNCST - PRINTING OF COST DATA*****
SUBROUTINE PRNCST
COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
80 PRINT*
PRINT*,' PRICE FUNCTIONS'
PRINT 1001,(I,I=1,KC)
1001 FORMAT (1X,' SIZE ',8I7)
PRINT*
DO 85 I=1,KS
85 PRINT 1002,SIZ(I),(COST(I,J),J=1,KC)
1002 FORMAT (1X,F5.1,5X,8F7.1)
PRINT 1003,ENCO
1003 FORMAT (/, ' ENERGY COST ',F6.3, ' $/KWH' )
PRINT 1004,NY
1004 FORMAT (/, ' TIME PERIOD',I6, ' YEARS' )
PRINT 1005,XI*100
1005 FORMAT (/, ' INTEREST RATE',F5.1, ' %' )
RETURN
END
C*****STORING COST DATA*****
SUBROUTINE STOCST
COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
```

```
COMMON /PRINT/ IPM,IPP,IPE
CHARACTER ST*90
C
C   PROMPTING OF USER FOR FILE NAME UNDER WHICH THE DATA IS
C   TO BE STORED
C
510  IF (IPP.EQ.1) PRINT 5
5   FORMAT(/,' ENTER FILE NAME ',/)
   READ (*,2,END=61) ST
2   FORMAT (A60)
   IF (IPE.EQ.1) PRINT*,ST
   OPEN(1,ERR=62,FILE=ST,RECL=72)
C
C   STORING OF THE NUMBER OF SIZES, THE NUMBER OF PRICE FUNCTIONS,
C   ARRAY SIZ AND COST, THE ENERGY COST, NUMBER OF YEARS, AND
C   INTEREST RATE
C
   WRITE(1,10,ERR=62) KS,KC
10  FORMAT(2I3)
   WRITE(1,30,ERR=62) (SIZ(I),(COST(I,J),J=1,KC),I=1,KS)
30  FORMAT(6E12.5)
   WRITE (1,40,ERR=62) ENCO,NY,XI
40  FORMAT(E12.5,I3,E12.5)
C
C   CLOSING AND REWINDING FILE
C
   REWIND 1
   CLOSE(1)
   RETURN
61  CALL BLANK
   RETURN
62  PRINT*,' PROGRAM CANNOT ACCESS THIS FILE.'
   RETURN
   END
C*****RETRIEVING OF COST DATA*****
SUBROUTINE RETCST
COMMON /COST/ SIZ(25),COST(25,12),KS,KC,ENCO,NY,XI
COMMON /PRINT/ IPM,IPP,IPE
CHARACTER ST*90
C
C   PROMPTING FOR FILE NAME
C
50  IF (IPP.EQ.1) PRINT 5
5   FORMAT(/,' ENTER FILE NAME ',/)
   READ (*,2,END=61) ST
2   FORMAT (A60)
   IF (IPE.EQ.1) PRINT*,ST
C
C   OPENING AND REWINDING FILE
C
   OPEN(1,ERR=62,FILE=ST,STATUS='OLD',RECL=72)
   REWIND 1
   READ(1,10,ERR=62,END=62) KS,KC
10  FORMAT(2I3)
C
C   REINITIALIZING VARIABLES PRIOR TO READING
C
   DO 9 I = KS, 25
   SIZ(I) = 0
   DO 9 J = KC , 12
```

```
      COST(I,J) = 0
9      CONTINUE
      READ(1,30,ERR=62,END=62) (SIZ(I),(COST(I,J),J=1,KC),I=1,KS)
30     FORMAT(6E12.5)
      READ (1,40,ERR=62,END=62) ENCO,NY,XI
40     FORMAT(E12.5,I3,E12.5)
C
C     REWINDING AND CLOSING FILE
C
      REWIND 1
      CLOSE(1)
      RETURN
61     CALL BLANK
      RETURN
62     PRINT*,' PROGRAM CANNOT ACCESS THIS FILE.'
      RETURN
      END
C*****SUBROUTINE DECINP*****
C     SUBROUTINE DECOMPOSES THE INPUT STRING USED IN SUBROUTINE
C     INPCST AND OPTMOD INTO KEYWORD(S) AND NUMERIC VALUES
C*****
      SUBROUTINE DECINP(ST,VALUE,KTYPE,KTYPF,L)
      DIMENSION VALUE(30)
      CHARACTER ST*60,KTYPE*4,KTYPF*4,T2*1,NUM*9
      NUM='123456789'
      T2=', '
      K=1
      L=1
15     IF ((ST(K:K).EQ.' ').OR.(ST(K:K).EQ.', ')) THEN
          K=K+1
          IF (K.GE.60) THEN
              CALL BLANK
              RETURN
          END IF
          GOTO 15
      END IF
      IF (ST(K:K).LT.':') GOTO 30
      J=JJ(ST(K:),' ',T2)
C
C     IF INPUT STRING STARTS WITH AN ALPHA CHARACTER, THE FIRST
C     FOUR CHARACTERS (OR CHARACTERS TO THE FIRST BLANK OR COMMA,
C     WHICHEVER IS LESS) OF THE WORD ARE ACCEPTED AS THE KEYWORD
C     AND ASSIGNED TO KTYPE
C
      KTYPE=ST(K:K+MIN(J-2,3))
C
C     IF A SECOND ALPHA INPUT IS ENCOUNTERED, ITS FIRST FOUR LETTERS
C     ARE ACCEPTED AS KEYWORD KTYPF
C
      K=K+J
      IF (ST(K:K+4).EQ.' ') GOTO 45
25     IF ((ST(K:K).EQ.' ').OR.(ST(K:K).EQ.', ')) THEN
          K=K+1
          GOTO 25
      END IF
      IF (ST(K:K).GT.'9') THEN
          J=JJ(ST(K:),' ',T2)
          KTYPF=ST(K:K+MIN(J-2,3))
          GOTO 20
      END IF
```

```
C
C      CONVERTING OF NUMERIC INPUTS ENTERED AS STRINGS AND
C      ASSIGNING TO ARRAY VALUE
C
30  J=JJ(ST(K:), ' ', T2)
    VALUE(L)=0
    IP=INDEX(ST(K:K+J-2), '.')+K-1
    MM=0
    DO 40 I=K+J-2,K,-1
    IF ((I.EQ.K+J-2).AND.(IP.EQ.K-1)) THEN
        IP=I
    ELSE IF (I.EQ.IP) THEN
        MM=1
        GOTO 40
    END IF
    IF (ST(I:I).EQ.'-') THEN
        VALUE(L)=-VALUE(L)
        GOTO 40
    END IF
    IF (ST(I:I).EQ.'+') GOTO 40
    K1 = INDEX(NUM,ST(I:I))
    VALUE(L)=VALUE(L)+(K1)*10.** (IP-I-MM)
40  CONTINUE
    L=L+1
    GOTO 20
45  L=L-1
    RETURN
    END

C
C      *****SUBROUTINE KEYWRD*****
C
SUBROUTINE KEYWRD(ISCOT,KTYPE)
CHARACTER KTYPE*4
IF (ISCOT.EQ.1) THEN
10  WRITE(6,10) ' SIMULATION KEYWORDS '
    FORMAT(/,A40,/)
    WRITE(6,20) ' ACCU', ' CHEC', ' COEF', ' CREA', ' DIAM',
1  ' ELEV', ' END ', ' GET ', ' INPU', ' JOB ', ' LENG', ' LINE',
2  ' NODE', ' OUTP', ' PIPE', ' PRV ', ' PUMP', ' RATI', ' TANK'
20  FORMAT(19(A5,/))
    ELSEIF (ISCOT.EQ.2) THEN
        WRITE(6,10) ' COST KEYWORDS '
        WRITE(6,40) ' DELE', ' END ', ' ENER', ' ENR ', ' INTE', ' PRIC'
1  ' SIZE', ' YEAR '
40  FORMAT(8(A5,/))
    ELSEIF (ISCOT.EQ.3) THEN
        WRITE(6,10) ' OPTIMIZATION KEYWORDS '
        WRITE(6,60) ' END ', ' GROU', ' HWCC', ' LIMC',
1  ' LIMP', ' LOAD', ' PRICE', ' SIZE '
60  FORMAT(10(A5,/))
    ELSEIF (ISCOT.EQ.4) THEN
        WRITE(6,10) ' TIME SIM. KEYWORDS '
        WRITE(6,70) ' BEGI', ' DELTA', ' DURA', ' END ', ' EXCL', ' FIRE'
>  ' INCL '
1  ' PLIN', ' PNOD', ' PUMP', ' RATI', ' STEP', ' SPAT', ' TANK', ' USAG'
70  FORMAT(15(A5,/))
    ELSE
        WRITE(6,75) KTYPE
75  FORMAT(/,/, ' CORRECT FORMAT FOR KEYWORD ',A4, ' IS:')
    END IF
```

```
      IF (ISCOT.EQ.20) THEN
      IF (KTYPE.EQ.'SIZE') WRITE(6,77)
77  FORMAT(/,'  XX XX XX ...',/, '  LIST OF SIZES, IN.',/, ' OR',
1  ' XX      DELETE',/, ' SIZE, IN. ')
      IF (KTYPE.EQ.'PRIC') WRITE(6,79)
79  FORMAT(/,' PRIC  XX ',/, '  PRICE FCT. # ',/, ' OR',/,
1  '   ' PRIC  XX      XX      XX ',/,
2  '   ' PRICE FCT. #  SIZE IN.  PRICE $/FT',/, ' OR',/,
3  '   ' PRIC  XX      DELETE ',/, '  PRICE FCT. #')
      IF (KTYPE.EQ.'YEAR') WRITE(6,81)
81  FORMAT(/,' YEAR  XX ',/, '  NUMBER OF YEARS ')
      IF (KTYPE.EQ.'INTE') WRITE(6,83)
83  FORMAT(/,' INTE  X.X',/, '  INTEREST IN %')
      IF (KTYPE.EQ.'ENER') WRITE(6,85)
85  FORMAT(/,' ENER  X.XXX',/, '  ENERGY COST $/KWH')
      IF (KTYPE.EQ.'DELE') THEN
      WRITE(6,87)
87  FORMAT(/,' SIZE  XX      DELETE ',/, '  SIZE IN.',/,
1  '   ' OR',/, ' PRICE  XX      DELETE ',/, '  PRICE FCT.
#')
      KTYPE = 'SIZE'
      END IF
      IF (KTYPE.EQ.'ENR ') WRITE(6,89)
89  FORMAT(/,'  XX      XX',/, ' FCT. #      MULT. FACTOR')
      ELSEIF (ISCOT.EQ.30) THEN
      IF (KTYPE.EQ.'GROU') WRITE(6,80)
80  FORMAT(/,' XX      XXX XXX XXX ...',/, ' GROUP # LIST OF',
1  ' LINK #',/, ' OR',/, ' XX      ALL      XXX      XXX',/, ' GROUP#',
2  '   ' FIRST LINK#      LAST LINK#')
      IF (KTYPE.EQ.'HWCC') WRITE(6,90)
90  FORMAT(/,' HWCC  XXX ',/, '  COEF. FOR CLEANING')
      IF (KTYPE.EQ.'LIMC') WRITE(6,100)
100 FORMAT(/,' LIMC  X.X ',/, '  % OF MINIMUM COST')
      IF (KTYPE.EQ.'LIMP') WRITE(6,110)
110 FORMAT(/,' LIMP  X.X ',/, '  PRESSURE INCREMENT, PSI')
      IF (KTYPE.EQ.'LOAD') WRITE(6,120)
120 FORMAT(/,' LOAD  XX  MINI  XX.X',/,
1  '   ' PATTERN #  MIN. PRESS. PSI',/, ' OR ',/
2  '   ' LOAD  XX  MINI  XXX      XX.X',/
3  '   ' PATTERN #  NODE #  MIN. PRESS. PSI',/, ' OR ',/
4  '   ' LOAD  XX  MINI  XXX      XXX      XX.X',/
5  '   ' PATTERN #  1ST NODE #  LAST NODE #  MIN. PRESS. PSI.'
6  '   ',/, ' OR',/
7  '   ' LOAD  XX  OUTPUT  XXX      XXX ',/
8  '   ' PATTERN #  NODE #  OUTPUT,GPM',/, ' OR ',/
9  '   ' LOAD  XX  PUMP  XXX      XX.X      (XX.X) ',/
+  '   ' PATTERN #  LINK #  TIME RUNNING %  EFFICIENCY %'
+  '   ',/, ' OR',/
+  '   ' LOAD  XX  RATIO  X.XX',/
+  '   ' PATTERN #  RATIO',/, ' OR',/
+  '   ' LOAD  XX  RATIO  XXX      XXX      X.XX',/
+  '   ' PATTERN #  1ST NODE #  LAST NODE #  RATIO')
      IF (KTYPE.EQ.'PRIC') WRITE(6,130)
130 FORMAT(/,' PRIC  XX      XXX XXX XXX ...',/
1  '   ' PRICE FCT. #  LIST OF LINK #',/, ' OR',/
2  '   ' PRIC  XX  ALL  XXX      XXX',/
3  '   ' PRIC FCT. #  1ST LINK #  LAST LINK #')
      IF (KTYPE.EQ.'SIZE') WRITE(6,140)
140 FORMAT(/,' SIZE  XX  XX.X  XX.X  XX.X ...',/
1  '   ' GROUP #  LIST OF SIZES, IN.')
```



```

ELSE
  IF (KTYPE.EQ.'DURA') WRITE(6,141)
141  FORMAT(/,' DURA   XX.X',/, '          HOURS OF SIMULATION')
  IF (KTYPE.EQ.'STEP') WRITE(6,143)
143  FORMAT(/,' STEP   XX.X',/, '          NUMBER OF STEPS')
  IF (KTYPE.EQ.'RATI') WRITE(6,145)
145  FORMAT(/,' RATI   XX.X',/, '          CORRECTION FOR OUTPUT FLOWS')
  IF (KTYPE.EQ.'USAG') WRITE(6,147)
147  FORMAT(/,' USAG   XX.X   XX.X1   XX.X2',/
1,    '          PATTERN #   TIME STEP1   TIME STEP2')
  IF (KTYPE.EQ.'TANK') WRITE(6,148)
148  FORMAT(/,' TANK   XX   XX.X   XX.X   XX.X   (XX.X)',/
1,    '          NODE MAX HT,FT  MIN HT,FT  AREA,SQFT  INIT. HT'
2,/, ' OR',/, ' TANK   XX ',/, '          NODE')
  IF (KTYPE.EQ.'FIRE') WRITE(6,150)
150  FORMAT(/,' FIRE   XX   XX.X   XX.X   XX.X',/
1,    '          NODE  BEG. STEP  DURA. STEP  FLOW,GPM',/, ' OR'
2,/, ' FIRE   XX ',/, '          NODE')
  IF (KTYPE.EQ.'EXCL') WRITE(6,152)
152  FORMAT(/,' EXCL   XX   TIME   XX   XX',/
1,    '          LINK #   START HOUR  END HOUR')
  IF (KTYPE.EQ.'INCL') WRITE(6,154)
154  FORMAT(/,' INCL   XX   XX   XX . . .',/, '          LIST OF LINK #')
  IF (KTYPE.EQ.'SPAT') WRITE(6,156)
156  FORMAT(/,' SPAT   XX   RANGE   XX   XX',/
1,    '          PATTERN#   B-NODE E-NODE',/, ' OR',/
2, ' SPAT   XX   NODES   XX XX XX XX',/
3, '          PATTERN#   LIST OF NODES')
  IF (KTYPE.EQ.'PUMP') WRITE(6,158)
158  FORMAT(/,' PUMP   XX   LEFT   XX   XX   XX',/
1,    '          LINK #   TANK #   ON LVL,FT  OFF LVL,FT',/
2, ' OR',/, ' PUMP   XX   LEPSI  XX   XX   XX',/
3,    '          LINK #   NODE #   ON PSI   OFF PSI',/
4, ' OR',/, ' PUMP   XX   TIME   XX   XX',/
5,    '          LINK #   START HOUR  END HOUR' /
6, ' OR' / ' PUMP   XX')
  IF (KTYPE.EQ.'PNOD') WRITE(6,160)
160  FORMAT(/,' PNODE   XX   XX   XX   XX . . .',/
1,    '          LIST OF NODES',/, ' OR',/
2,    ' PNODE   XX   XX   XX  DELETE ',/
3,    '          LIST OF NODES')
  IF (KTYPE.EQ.'PLIN') WRITE(6,163)
163  FORMAT(/,' PLINK   XX   XX   XX   XX . . .',/
1,    '          LIST OF LINKS',/, ' OR',/
2,    ' PLINK   XX   XX   XX  DELETE ',/
3,    '          LIST OF LINKS')
  IF (KTYPE.EQ.'DETA') WRITE(6,165)
165  FORMAT(/,' DETAIL  ON'/' OR'/' DETAIL  OFF')
  IF (KTYPE.EQ.'BEGI') WRITE(6,167)
167  FORMAT(/,' BEGIN   XX   ON '/'          LINK #'/' OR'/'
1,    ' BEGIN   XX   OFF'/'          LINK #')
END IF
RETURN
END
C *****ERROR*****
C IF AN ERROR WAS MADE BY THE USER WHILE ENTERING DATA,
C THE APPROPRIATE MESSAGE IS DISPLAYED, DEPENDING ON THE ERROR
C DISPLAYED
C *****
C

```

```

SUBROUTINE ERROR(NERR,KTYPE)
CHARACTER KTYPE*4,KEY1*8,KEY2*32,KEY3*16
KEY1='PIPELINE'
KEY2='ELEVOUTPINPUTANKDIAMLENGCOEFACCU'
KEY3='CHECPUMPPRV RATI'
GOTO (1,2,3,4,5,6,7,8,9,10) NERR
1 PRINT*, 'MORE DATA ENTERED THAN REQUIRED FOR KEYWORD ',KTYPE
RETURN
2 GOTO 20
3 PRINT*, 'YOU MUST ENTER A KEYWORD'
RETURN
4 PRINT*, 'PIPE LENGTH RANGE IS FROM 0 TO 10000 FEET'
RETURN
5 PRINT*, 'PIPE DIAMETER MUST BE GREATER THAN THAN ZERO'
RETURN
6 PRINT*, 'HAZEN WILLIAMS COEFFICIENT IS TOO SMALL'
RETURN
7 PRINT*, 'ELEVATION MUST BE GREATER THAN OR EQUAL TO ZERO'
RETURN
8 PRINT*, 'WATER LEVEL MUST BE GREATER THAN ZERO'
RETURN
9 PRINT*, 'PRESSURE SETTING MUST BE GREATER THAN ZERO'
RETURN
10 PRINT*, 'BEGINNING OR ENDING NODE DOES NOT MATCH THIS PIPE'
RETURN
20 IN1=(INDEX(KEY1,KTYPE)+3)/4
IN2=(INDEX(KEY2,KTYPE)+3)/4
IN3=(INDEX(KEY3,KTYPE)+3)/4
IN=IN1+IN2+IN3
IF (IN.EQ.0) THEN
PRINT*, 'INVALID KEYWORD ',KTYPE
KTYPE='PIPE'
RETURN
END IF
C
C IF A VALID KEYWORD WAS ENTERED AND NOT FOLLOWED BY ANY NUMERIC
C DATA, THE INPUT FORMAT FOR THE KEYWORD IS DISPLAYED
C
PRINT*, 'CORRECT FORMAT FOR KEYWORD ',KTYPE, ' IS:'
IF (IN1.NE.0) THEN
PRINT*, 'LINK # B-NODE E-NODE DIAMETER(IN) LENGTH(FT) HW
C
10EF'
RETURN
END IF
IF (IN2.NE.0) GOTO(21,22,23,24,25,26,27,28)IN2
IF (IN3.NE.0) GOTO(29,29,29,30)IN3
21 PRINT*, 'NODE # ELEVATION(FT)'
RETURN
22 PRINT*, 'NODE # OUTPUT(GPM)'
RETURN
23 PRINT*, 'NODE # INPUT(GPM)'
RETURN
24 PRINT*, 'NODE # WATER LEVEL'
RETURN
25 PRINT*, 'LINK # DIAMETER(IN)'
RETURN
26 PRINT*, 'LINK # LENGHT(FT)'
RETURN
27 PRINT*, 'NEW HW DEFAULT OR'
```

```
      PRINT*, 'LINK #   HW COEFFICIENT   OR'
      PRINT*, 'LINK #1   LINK #2   HW COEFFICIENT'
      RETURN
28  PRINT*, ' PRESSURE ACCURACY(PSI)   FLOW ACCURACY(GPM)   #OF
ITERATI
      IONS '
      RETURN
29  PRINT*, 'LINK #   B-NODE   E-NODE '
      RETURN
30  PRINT*, 'B-NODE   E-NODE   RATIO   OR'
      PRINT*, 'RATIO'
      RETURN
      END
C
C      FUNCTION JJ LOCATES IN STRING ST THE FIRST OCCURENCE OF
C      STRING T1 AND T2 AND SELECTS THE SMALLER OF THE TWO VALUES
C
      FUNCTION JJ(ST,T1,T2)
      CHARACTER ST*60,T1*1,T2*1
      J1=INDEX(ST,T1)
      J2=INDEX(ST,T2)
      J=MIN(J1,J2)
C
C      IF ONE OF THE TWO VALUES IS ZERO, THE NON-ZERO VALUE IS
C      SELECTED
C
      IF (J1*J2.EQ.0) J=MAX(J1,J2)
C
C      IF BOTH VALUES ARE ZERO, THE LENGTH OF STRING ST PLUS 1 IS
C      SELECTED
C
      IF (J.EQ.0) J=LEN(ST)+1
      JJ=J
      RETURN
      END
C
C      SUBROUTINE NOTHAE PRINTS THE TABLE HEADING OF THE NODE TABLE
C      WHEN PRINTING THE OUTPUT
C
      SUBROUTINE NOTAHE
      PRINT 1151
      PRINT 1152
      PRINT 1153
      PRINT 1154
      PRINT 1155
1151  FORMAT (5X, '-----')
--
      1-----')
1152  FORMAT (5X, 'I',65X, 'I')
1153  FORMAT (5X, 'I   NODE   ELEV.   OUTPUT   E.G.L.   PR.HEAD
PRES
      1SURE',8X, 'I')
1154  FORMAT (5X, 'I   NO.   FT.   GPM   FT.   FT.
PS
      1I',11X, 'I')
1155  FORMAT (5X, 'I-----')
--
      1-----I')
      RETURN
      END
```

```
C      SUBROUTINE PITAHE PRINTS THE TABLE HEADING OF THE LINK TABLE
C      WHEN PRINTING THE OUTPUT
      SUBROUTINE PITAHE
      PRINT 1161
      PRINT 1162
      PRINT 1163
      PRINT 1164
      PRINT 1165
1161  FORMAT (5X,'-----
--
      1-----')
1162  FORMAT (5X,'I',65X,'I')
1163  FORMAT (5X,'I  PIPE      NODES      DIAM.  LENGTH  COEF  FLOW
VE
      1L.  HEAD',3X,'I')
1164  FORMAT (5X,'I  NO.  FROM TO      IN.      FT.      GPM
FT/
      1SEC  LOSS',3X,'I')
1165  FORMAT (5X,'-----
--
      1-----')
      RETURN
      END
      SUBROUTINE BLANK
      PRINT*,' INPUT REQUIRED.'
      CLOSE (2)
      OPEN (2,FILE='INPUT')
      RETURN
      END

C
C
C
C*****QUEU*****
C SUBROUTINE QUEU TESTS A SIZE COMBINATION AGAINST COMBINATIONS
C WHICH WERE PREVIOUSLY TESTED AND FAILED TO MEET THE PRESSURE
C REQUIREMENTS
C
      SUBROUTINE QUEU (IQ,K0,IFS,IGR,ICS)
      DIMENSION ICS(15),IFS(100,15)
      IQ=0
      IF (K0.EQ.0) GOTO 140

C
C LOOP ENUMERATES ALL COMBINATIONS IN IFS, STARTING WITH THE LAST
C COMBINATION ENTERED IN IFS.
C
      DO 130 J0=K0,1,-1

C
C COMBINATION IFS TO BE TESTED AGAINST ICS
C INNER LOOP ENUMERTATES THE POSITIONS IN ARRAY ICS
C
      DO 110 J1=1,IGR
      IF (ICS(J1).LT.IFS(J0,J1)) GOTO 130
110  CONTINUE
C
C IQ=1 INDICATES THAT WILL NOT BE NECESSARY TO CALCULATE
C PRESSURE DISTRIBUTION
C
      IQ=1
      GOTO 140
130  CONTINUE
```

```
140  RETURN
      END
      SUBROUTINE TIMENU
C PRINTS MAIN TIME SIMULATION MENU AND
C CALLS OTHER TIME SIMULATION ROUTINES
C
      INTEGER PNL,O
      REAL*8 A,S,G,HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      CHARACTER ST*60,NUM*7,NOLI*4,UNIT*3,PUBE*3
      COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XLLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
      COMMON /NAME/ NOLI(11),UNIT(11),PUBE(20)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
      COMMON /PRINT/ IPM,IPP,IPE
      COMMON /ACCU/ PRAC,FLAC,HWMA,ICL
      DIMENSION DOTMP(PNL),ELTMP(20)
      NUM='0123489'
      IF (O(1).EQ.0) THEN
        PRINT*,'SYSTEM DATA FOR STEADY STATE SIMULATION MUST '
        PRINT*,'BE ENTERED PRIOR TO THE TIME SIMULATION. '
        RETURN
      END IF
100  IF(IPM.EQ.0)GO TO 20
      WRITE(6,10)
10   FORMAT(/' TIME SIMULATION MENU'/
1/' SELECT PROGRAM OPTION: '/
2/' BEGIN SIMULATION : ENTER 0, PRESS RETURN'
3/' MODIFY,ENTER DATA : 1'
4/' PRINT TIME DATA : 2'
5/' STORE TIME DATA : 3'
6/' RETRIEVE TIME DATA: 4'
7/' PROGRAM CONTROL : 8'
8/' TERMINATE : 9'/)
20  READ(*,21,END=999)ST(1:60)
21  FORMAT(A60)
      IF(IPE.EQ.1)WRITE(6,22)ST(1:60)
22  FORMAT(1X,A60)
      IF (INDEX(NUM,ST(1:1)).EQ.0) WRITE(6,23)
23  FORMAT(' INVALID RESPONSE TO MENU')
      IF(ST(1:1).EQ.'0')THEN
        CALL TMPRAR(1,NTANK,NPUMP,NOPU,HEON,HEOF,UPL,XLLOL,INO,IPI
>,DOTMP,ELTMP)
        CALL TIMSIM(DOTMP,ELTMP)
        CALL TMPRAR(-1,NTANK,NPUMP,NOPU,HEON,HEOF,UPL,XLLOL,INO,IPI
>,DOTMP,ELTMP)
      END IF
      IF(ST(1:1).EQ.'1')CALL TIMEIN
      IF(ST(1:1).EQ.'2')CALL TIMEOT
      IF(ST(1:1).EQ.'3')CALL STOTIM(O)
      IF(ST(1:1).EQ.'4')CALL RETIME
      IF(ST(1:1).EQ.'8')RETURN
      IF(ST(1:1).EQ.' ')WRITE(6,25)
```

```
25   FORMAT(' VALUE MUST BE IN COLUMN 1')
    IF(ST(1:1).EQ.'9') CALL TERMIN
    GO TO 100
999  CALL BLANK
    GO TO 100
    END
    SUBROUTINE TIMEIN
C
C DATA ENTRY ROUTINE FOR TIME SIMULATION
C
    INTEGER PNL,O
    REAL*8 A,S,G,HE
    PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
    DIMENSION VALUE(30),XDEFAL(24)
    CHARACTER NOLI*4,UNIT*3,PUBE*3
    COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1INTANK(20),UPL(20),XLLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
    COMMON /NAME/ NOLI(11),UNIT(11),PUBE(20)
    COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
    COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
    COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
    COMMON /PRINT/ IPM,IPP,IPE
    CHARACTER KTYPE*4,KTYPF*4,ST*90,KW*68,NW*10
    DATA XDEFAL/1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.5,1.4,1.3,1.2,
11.1,1.0,0.9,0.8,0.7,0.6,0.5,0.4,0.5,0.6,0.7,0.8,0.9/
C
C INITIALIZING VALUES FOR VALID LEADING CHARACTERS
C
    KW(1:68)='DURASTEPRATIUSAGTANKFIREEXCLINCLKEYWSPATPUMPDELEEND '
    >/'PNODPLINDETABEGI'
    NW(1:10)='1234567890'
    IF(IPP.EQ.1) WRITE(6,50)
50   FORMAT('/' TIME SIMULATION INPUT'/
1' TYPE KEYW FOR LIST OF KEYWORDS')
55   KTYPE='DURA'
    KTYPF='      '
C
C MAIN PROMPT AND READ
C
100  CONTINUE
    IF(IPP.EQ.1)WRITE(6,101)KTYPE
101  FORMAT('/' T. KEYWORD IS ('A4,') ENTER (KEYWORD) DATA LIST',/)
    READ(*,102,END=999)ST(1:60)
102  FORMAT(A60)
    IF(IPE.EQ.1)WRITE(6,107)ST(1:60)
107  FORMAT(1X,A60)
C
C TESTING IF LEADING CHARACTERS ARE CORRECT
C
    DO 108 I=1,10
    IF(ST(1:1).EQ.NW(I:I))GO TO 106
108  CONTINUE
    DO 103 I=1,17
    IBEG=(I-1)*4+1
    IEND=IBEG+3
```

```
IF(ST(1:4).EQ.KW(IBEG:IEND))GO TO 106
103 CONTINUE
WRITE(6,104)ST(1:4)
104 FORMAT(1X,A4,' IS INVALID KEYWORD--ENTRY IGNORED')
GO TO 100
106 CONTINUE
KTYPF=' '
CALL DECINP(ST,VALUE,KTYPE,KTYPF,L)
IF(L.EQ.0.AND.KTYPE.NE.'END'.AND.KTYPE.NE.'KEYW'.AND.
>KTYPE.NE.'DETA') THEN
CALL KEYWRD(40,KTYPE)
GO TO 100
ENDIF
C
C ++++++ DURATION ++++++
C
IF(KTYPE.EQ.'DURA')THEN
TMAX=VALUE(1)
XSTEP=NSTEP
IF(NSTEP.NE.0)TSLF=TMAX/XSTEP
IF(TSLF.NE.0)WRITE(6,111)TSLF
111 FORMAT(' TIME STEP SIZE',F7.2,' HOURS')
GO TO 100
C
C ++++++ STEPS ++++++
C
ELSEIF(KTYPE.EQ.'STEP')THEN
NSTEP=VALUE(1)
IF(NSTEP.LT.1) THEN
WRITE(6,113)
113 FORMAT(1X,'NUMBER OF STEPS MUST BE AT LEAST ONE.')
GOTO 100
END IF
XSTEP=NSTEP
IF(TMAX.GT.0)TSLF=TMAX/XSTEP
IF(TSLF.GT.0)WRITE(6,111)TSLF
GO TO 100
C
C ++++++ RATIO ++++++
C
ELSEIF(KTYPE.EQ.'RATI')THEN
IF(L.GT.1)THEN
WRITE(6,121)
121 FORMAT(' ONLY 1 VALUE CAN BE SPECIFIED WITH RATIO'
1 /' ENTRY IGNORED')
GO TO 100
ELSE
RATIO=VALUE(1)
GO TO 100
ENDIF
C
C ++++++ USAGE ++++++
C
ELSEIF(KTYPE.EQ.'USAG')THEN
IF(L.LE.1.OR.L.GT.3)THEN
WRITE(6,131)
131 FORMAT(' MUST BE 2 OR 3 ENTRIES FOR USAGE'
1 /' ENTRY IGNORED')
GO TO 100
ENDIF
```

```
C
C TESING FOR VALID BEGINING AND ENDING TIME STEPS
C
      IUSE=VALUE(1)
      IBEG=VALUE(2)
      IEND=VALUE(3)
      IF (IBEG.GT.57.OR.IEND.GT.57)THEN
133         WRITE(6,133)
            FORMAT(' TIME STEP MUST BE LESS THAN 57')
            GO TO 100
      ENDIF
      IF(L.EQ.2)IEND=IBEG
      IF (IBEG.GT.IEND)THEN
135         WRITE(6,135)
            FORMAT(' BEGINNING TIME MUST BE LESS THAN ENDING')
            GO TO 100
      ENDIF
      IF (IUSE.LT.1.OR.IUSE.GT.5)THEN
137         WRITE(6,137)
            FORMAT(' USE PATTERN MUST BE BETWEEN 1 AND 5')
            GO TO 100
      ENDIF
      IVAL=IEND-IBEG+1
      IF (IVAL.GT.15)THEN
139         WRITE(6,139)
            FORMAT(' TOO MANY VALUES PER LINE')
            IVAL=15
      ENDIF
C
C PROMPTING FOR WATER USE PATTERN FACTORS
C
      IF (IPP.EQ.1) WRITE(6,141) IVAL
141     FORMAT(' ENTER',I5,' VALUES')
      READ(5,143)ST(1:80)
      IF (IPE.EQ.1)WRITE(6,144)ST(1:79)
143     FORMAT(A80)
144     FORMAT(1X,A79)
      CALL DECINP(ST,VALUE,KTYPE,KTYPF,L)
      INUM=IVAL
      IF(L.LT.IVAL)INUM=L
      IF(L.NE.IVAL)WRITE(6,147)INUM
147     FORMAT(I5,' VALUES USED')
      DO 149 I=1,INUM
          II=I-1+IBEG
          XLOF(II,IUSE)=VALUE(I)
149     CONTINUE
          GO TO 100
C
C ++++++ TANK ++++++
C
      ELSEIF(KTYPE.EQ.'TANK')THEN
          IF(L.NE.1.AND.L.NE.4.AND.L.NE.5)THEN
              WRITE(6,161)
161         FORMAT(' MUST SPECIFY NODE, UPPER AND LOWER LEVELS'
1          /' AREA AND, OPTIONALLY, INITIAL LEVEL')
              GO TO 100
          ENDIF
          NOTANK=VALUE(1)
C
C TESTING IF NODE SPECIFIED IS A TANK
```



```
C
      DO 163 I=1,0(2)
      IF(INO(I).EQ.NOTANK)GO TO 165
163  CONTINUE
      WRITE(6,164)NOTANK
164  FORMAT(I5,' HAS NOT BEEN DEFINED AS NODE '
1      /' ENTRY IGNORED')
      GO TO 100
165  IF(DO(I).LT.1E10)THEN
      WRITE(6,167)NOTANK
167  FORMAT(I5,' IS NOT A TANK--ENTRY IGNORED')
      GO TO 100
      ENDIF

C
C PUTTING TANK DATA IN ARRAY
C
      DO 171 J=1,20
      IF(NOTANK.EQ.NTANK(J))GO TO 181
      IF(NTANK(J).EQ.0)GO TO 181
171  CONTINUE
      WRITE(6,173)
173  FORMAT(' 20 TANKS ALREADY DEFINED--ENTRY IGNORED')
      GO TO 100
181  IF(L.EQ.4.OR.L.EQ.5)THEN
      NTANK(J)=NOTANK
      UPL(J)=VALUE(2)
      XLOL(J)=VALUE(3)
      ART(J)=VALUE(4)
      IF(L.EQ.5) THEN
      HETI(J)=VALUE(5)
      ELSE
      HETI(J)=HE(I)-ABS(EL(I))
      END IF

C
C ELIMINATING TANK SPECIFICATIONS
C
      ELSE
      DO 185 JJ=J,19
      NTANK(JJ)=NTANK(JJ+1)
      UPL(JJ)=UPL(JJ+1)
      XLOL(JJ)=XLOL(JJ+1)
      ART(JJ)=ART(JJ+1)
      HETI(JJ)=HETI(JJ+1)
185  CONTINUE
      NTANK(20)=0
      UPL(20)=0.
      XLOL(20)=0.
      ART(20)=0.
      HETI(20)=0.
      ENDIF
      GO TO 100

C
C ++++++ FIRE ++++++
C
      ELSEIF(KTYPE.EQ.'FIRE')THEN
      IF(L.NE.4.AND.L.NE.1)THEN
      WRITE(6,191)
191  FORMAT(' FOUR ENTRIES REQUIRED FOR FIRE '
1      /' NODE, START, END, FLOW')
      GO TO 100
```

```
        ENDIF
        NOFIRE=VALUE(1)
C
C CHECKING IF NODE IS DEFINED
C
        DO 193 I=1,0(2)
        IF(INO(I).EQ.NOFIRE)GO TO 195
193     CONTINUE
        WRITE(6,194)NOFIRE
194     FORMAT(I5,' WAS NOT DEFINED AS NODE--ENTRY IGNORED')
        GO TO 100
C
C PUTTING FIRE DATA IN ARRAY
C
195     IF(DO(I).GT.1E10)THEN
        WRITE(6,197)NOFIRE
197         FORMAT(I5,' IS A TANK--ENTRY IGNORED')
        GO TO 100
        ENDIF
        DO 201 J=1,5
        IF(NOFIRE.EQ.NFIRE(J))GO TO 205
        IF(NFIRE(J).EQ.0)GO TO 205
201     CONTINUE
        WRITE(6,203)
203     FORMAT(' FIVE FIRES ALREADY DEFINED'
1         /' DELETE ONE BEFORE ADDING ANOTHER')
        GO TO 100
205     IF(L.EQ.4)THEN
        NFIRE(J)=NOFIRE
        ITF(J)=VALUE(2)
        IDF(J)=VALUE(3)
        FF(J)=VALUE(4)
        ELSE
C
C ELIMINATING FIRE SPECIFICATIONS
C
        DO 209 JJ=J,4
        NFIRE(JJ)=NFIRE(JJ+1)
        ITF(JJ)=ITF(JJ+1)
        IDF(JJ)=IDF(JJ+1)
        FF(JJ)=FF(JJ+1)
209     CONTINUE
        NFIRE(5)=0
        ITF(5)=0
        IDF(5)=0
        FF(5)=0.
        ENDIF
        GO TO 100
C
C ++++++ EXCLUDE ++++++
C
        ELSEIF(KTYPE.EQ.'EXCL')THEN
        IEXP=VALUE(1)
        DO 211 I1=1,0(1)
211     IF (IPI(I1).EQ.IEXP) GOTO 215
        WRITE(6,214) IEXP
214     FORMAT(I5,' IS NOT LINK - ENTRY IGNORED.')
        GOTO 100
215     IF (CP(I1).LT.0) THEN
        WRITE(6,217) IEXP
```

```
217     FORMAT(1X,'USE PUMP KEYWORD TO EXCLUDE PUMP',I4)
      GOTO 100
      END IF
      IF (XL(I1).GT.999999..OR.XL(I1).LT.0) THEN
        WRITE(6,219)
219     FORMAT(1X,'NO PRV OR CHECK VALVE CAN BE EXCLUDED. ')
      GOTO 100
      END IF
      IF (MOD(L-1,2).NE.0.OR.L.EQ.1) THEN
        WRITE(6,273)
        GOTO 100
      END IF
      DO 220 I = 1,15
220     IF (IPEX(I).EQ.IEXP.OR.IPEX(I).EQ.0) GOTO 224
        WRITE(6,271)' 15 PIPES'
        GOTO 100
224     ICPE=I
        DO 227 K=2,L,2
          DO 226 K2=1,7
226     IF (TEE(I,K2).LE.0) GOTO 228
            WRITE(6,467)' PIPE',IEXP
            GOTO 100
228     TEB(I,K2)=VALUE(K)
            TEE(I,K2)=VALUE(K+1)
227     CONTINUE
        IPEX(I)=IEXP
        GOTO 100
C
C ++++++ INCLUDE ++++++
C
      ELSEIF(KTYPE.EQ.'INCL')THEN
        DO 229 I=1,L
          INCL=VALUE(I)
          JMAX=ICPE
C
C CHECKING IF NODE WAS EXCLUDED
C
          DO 221 J=1,JMAX
            IF(INCL.EQ.IPEX(J))GO TO 223
221     CONTINUE
            WRITE(6,222)INCL
222     FORMAT(I5,' WAS NOT ORIGINALLY EXCLUDED')
            GO TO 100
223     ICPE=ICPE-1
            DO 225 JJ=J,15
              IPEX(JJ)=IPEX(JJ+1)
              DO 232 K=1,7
                TEB(JJ,K)=TEB(JJ+1,K)
232     TEE(JJ,K)=TEE(JJ+1,K)
225     CONTINUE
229     CONTINUE
C
C ++++++ KEYWORDS ++++++
C
      ELSEIF(KTYPE.EQ.'KEYW')THEN
        CALL KEYWRD(4,KTYPE)
        GO TO 100
C
C ++++++ SPATIAL ++++++
C
```

```

ELSEIF(KTYPE.EQ.'SPAT')THEN
  NPATT=VALUE(1)
  IF(NPATT.LT.1.OR.NPATT.GT.5)THEN
    WRITE(6,230)
230    FORMAT(' USE PATTERN MUST BE BETWEEN 1 AND 5 ')
    GO TO 100
  ENDIF
  IF(KTYPF.EQ.' ')THEN
    WRITE(6,231)
231    FORMAT(' SECOND KEYWORD NEEDED ')
    GO TO 100
  ENDIF
C
C ++++++ RANGE ++++++
C
  IF(KTYPF.EQ.'RANG')THEN
    IF(L.NE.3)THEN
      WRITE(6,233)
233    FORMAT(' BEGIN AND END OF RANGE NEEDED '
1      /'ENTRY IGNORED')
      GO TO 100
    ENDIF
    IBEG=VALUE(2)
    IEND=VALUE(3)
    IF( IBEG.GT.IEND)THEN
      WRITE(6,235)
235    FORMAT(' BEGINNING MUST BE LESS THAN END ')
      GO TO 100
    ENDIF
C
C CHECKING IF NODES EXIST IN INO
C
    DO 239 I=IBEG,IEND
    DO 236 J=1,O(2)
    IF(INO(J).EQ.I)GO TO 237
236    CONTINUE
    GO TO 239
237    IF(DO(J).GE.1E10)GO TO 239
C
C ASSIGNING NODE TO PATTERN
C
    IPAT(J)=NPATT
239    CONTINUE
    GO TO 100
C
C ++++++ NODE ++++++
C
    ELSEIF(KTYPF.EQ.'NODE')THEN
      DO 252 K=2,L
      INOD=VALUE(K)
      DO 248 J=1,O(2)
      IF(INO(J).EQ.INOD)GO TO 250
248    CONTINUE
      WRITE(6,249)INOD
249    FORMAT(I5,' NOT DEFINED AS NODE ')
      GO TO 252
250    IF(DO(J).GE.1E10)THEN
      WRITE(6,251)INOD
251    FORMAT(I5,' IS TANK--ENTRY IGNORED ')
      GO TO 100

```

```

                ENDIF
C
C ASSIGNING NODE TO PATTERN
C
                IPAT(J)=NPATT
252             CONTINUE
                GO TO 100
            ELSE
                WRITE(6,255)
255             FORMAT(' SECOND KEYWORD MUST BE NODE OR RANGE' )
                GO TO 100
            ENDIF
C
C ++++++ PUMP ++++++
C
            ELSEIF (KTYPE.EQ.'PUMP') THEN
                IF (KTYPF.NE.'LEFT'.AND.KTYPF.NE.'LEPS'.AND.
                >          KTYPF.NE.'TIME'.AND.L.NE.1) THEN
                    WRITE(6,260)
260             FORMAT(' SECOND KEYWORD MUST BE LEFT, LEPSI, OR TIME' )
                    GOTO 100
                ENDIF
                IPUMP = VALUE(1)
                DO 263 I=1,0(1)
                    IF (IPI(I).EQ.IPUMP.AND.CP(I).LT.0) GOTO 267
263             CONTINUE
                WRITE(6,265) IPUMP
265             FORMAT(I5,' IS NOT A PUMP - ENTRY IGNORED' )
                GOTO 100
267             DO 269 I = 1, 20
                    IF (NPUMP(I).EQ.IPUMP.OR.NPUMP(I).EQ.0) GOTO 272
269             CONTINUE
                WRITE(6,271)' 20 PUMPS'
271             FORMAT(1X,A9,' ALREADY DEFINED - ENTRY IGNORED' )
                GOTO 100
C
C DELETING A PUMP FROM ARRAYS
C
272             IF (L.EQ.1) THEN
                    DO 283 J = I,20
                        NPUMP(J) = NPUMP(J+1)
                        NOPU(J) = NOPU(J+1)
                        HEON(J) = HEON(J+1)
                        HEOF(J) = HEOF(J+1)
                        DO 463 K = 1,7
                            TPB(J,K) = TPB(J+1,K)
                            TPE(J,K) = TPE(J+1,K)
463                     CONTINUE
283                 CONTINUE
                    GOTO 100
                END IF
C
C TIME CONTROLLED PUMPS
C
                IF (KTYPF.EQ.'TIME') THEN
                    IF (MOD(L-1,2).NE.0) THEN
                        WRITE(6,273)
273                 FORMAT(' EVERY ON TIME MUST HAVE AN OFF TIME. ENTRIES '
                >                    , ' IGNORED.' )
                        GOTO 100
                    
```

```
        ENDIF
        DO 465 K = 2,L,2
        DO 464 K2 = 1,7
464      IF (TPE(I,K2).LE.0) GOTO 469
        WRITE(6,467) ' PUMP',IPUMP
467      FORMAT( A,I4,' ALREADY HAS 7 ON/OFF TIMES DEFINED. '
>        , ' DELETE ONE '/' BEFORE ADDING ANOTHER.')
        GOTO 100
469      TPB(I,K2) = VALUE(K)
        TPE(I,K2) = VALUE(K+1)
465      CONTINUE
        ELSE
        IF (L.NE.4) THEN
        WRITE(6,274)
274      FORMAT(' NODE #, ON LEVEL, AND OFF LEVEL MUST',
>        ' FOLLOW SECOND KEYWORD')
        GOTO 100
        ENDIF
        NODE = VALUE(2)
        DO 275 J = 1, O(2)
        IF (INO(J).EQ.NODE) GOTO 279
275      CONTINUE
        WRITE(6,164) NODE
        GOTO 100
C
C PUMPS CONTROLLED BY TANK LEVEL
C
279      IF (KTYPF.EQ.'LEFT') THEN
        IF (DO(J).LT.1E10) THEN
        WRITE(6,167)
        GOTO 100
        ENDIF
        NOPU(I)=NODE
        HEON(I)=VALUE(3)
        HEOF(I)=VALUE(4)
        ELSE
C
C PUMPS CONTROLLED BY PRESSURE AT A NODE
C
        NOPU(I)=NODE
        HEON(I)=VALUE(3)*2.308
        HEOF(I)=VALUE(4)*2.308
        ENDIF
        IF (HEON(I).GT.HEOF(I)) WRITE(6,281)
281      FORMAT(' ON LEVEL IS HIGHER THAN OFF LEVEL')
        ENDIF
        NPUMP(I)=IPUMP
        GOTO 100
C
C ++++++PNOD+++++
C
        ELSEIF (KTYPE.EQ.'PNOD') THEN
        IF (KTYPF.EQ.'DELE') THEN
        KTYPF=' '
        DO 299 I = 1,L
        INUM = VALUE(I)
        DO 291 J = 1,10
291      IF (INUM.EQ.NUSR(J).AND.NOLI(J).EQ.'NODE') GOTO 295
        WRITE(6,293) INUM
293      FORMAT(1X,I4,' HAS NOT BEEN ASSIGNED FOR PRINTING OR ',
```

```
> 'IS NOT A NODE' )
GOTO 299
295 DO 297 K = J,10
NUSR(K) = NUSR(K+1)
NOLI(K) = NOLI(K+1)
UNIT(K) = UNIT(K+1)
297 CONTINUE
299 CONTINUE
ELSE
DO 409 I1 = 1,L
INUM = VALUE (I1)
DO 405 J = 1,O(2)
405 IF (INUM.EQ.INO(J)) GOTO 407
WRITE(6,249) INUM
GOTO 409
407 DO 411 I = 1,10
411 IF ((NUSR(I).EQ.INUM.AND.NOLI(I).EQ.'NODE').OR.
> (NUSR(I).EQ.0)) GOTO 415
WRITE(6,413)
413 FORMAT(' ONLY 10 ENTRIES CAN BE PRINTED - DELETE ONE ',
> 'BEFORE ADDING ANOTHER')
GOTO 100
415 NUSR(I) = INUM
NOLI(I) = 'NODE'
IF (DO(J).GE.1E10) THEN
UNIT(I) = ' FT'
ELSE
UNIT(I) = 'PSI'
END IF
409 CONTINUE
ENDIF
C
C ++++++PLIN+++++
C
ELSEIF (KTYPE.EQ.'PLIN') THEN
IF (KTYPF.EQ.'DELE') THEN
KTYPF=' '
DO 419 I = 1,L
INUM = VALUE(I)
DO 427 J = 1,10
427 IF (INUM.EQ.NUSR(J).AND.NOLI(J).EQ.'LINK') GOTO 421
WRITE(6,426) INUM
426 FORMAT(1X,I4,' HAS NOT BEEN ASSIGNED FOR PRINTING OR',
> ' IS NOT A LINK')
GOTO 419
421 DO 423 K = J,10
NUSR(K) = NUSR(K+1)
NOLI(K) = NOLI(K+1)
UNIT(K) = UNIT(K+1)
423 CONTINUE
419 CONTINUE
ELSE
DO 439 I1 = 1,L
INUM = VALUE (I1)
DO 429 J = 1,O(1)
429 IF (INUM.EQ.IPI(J)) GOTO 431
WRITE(6,214) INUM
GOTO 439
431 DO 433 I = 1,10
IF ((NUSR(I).EQ.INUM.AND.NOLI(I).EQ.'LINK').OR.
```

```
>(NUSR(I).EQ.0)) GOTO 435
433 CONTINUE
WRITE(6,413)
GOTO 100
435 NUSR(I) = INUM
NOLI(I) = 'LINK'
IF (XL(J).GT.999999.) THEN
    UNIT(I) = ' '
ELSE
    UNIT(I) = 'GPM'
END IF
439 CONTINUE
ENDIF

C
C ++++++DETA+++++
C
ELSEIF (KTYPE.EQ.'DETA') THEN
    IF (KTYPF.EQ.'ON') THEN
        IDTL=1
    ELSEIF (KTYPF.EQ.'ALL') THEN
        IPR =1
    ELSEIF (KTYPF.EQ.'OFF') THEN
        IDTL=2
        IPR =2
    ENDIF
    KTYPF=' '

C
C ++++++BEGI+++++
C
ELSEIF (KTYPE.EQ.'BEGI') THEN
    IF (KTYPF.NE.'ON'.AND.KTYPF.NE.'OFF') THEN
        WRITE(6,441)
441 FORMAT(' SECOND KEYWORD MUST BE "ON" OR "OFF" ')
        GOTO 100
    END IF
    IPUMP = VALUE(1)
    DO 443 I = 1 , O(1)
443 IF (IPI(I).EQ.IPUMP.AND.CP(I).LT.0) GOTO 445
        WRITE(6,265) IPUMP
        GOTO 100
445 DO 447 I = 1,20
447 IF (NPUMP(I).EQ.IPUMP.OR.NPUMP(I).EQ.0) GOTO 449
        WRITE(6,271)' 20 PUMPS'
        GOTO 100
449 IF (KTYPF.EQ.'ON') THEN
        PUBE(I) = 'ON '
    ELSE
        PUBE(I) = 'OFF'
    END IF
C ++++++ END ++++++
C
ELSEIF(KTYPE.EQ.'END') THEN
    ISTEP=0
    IF(TSLF.LE.0.OR.TMAX.LE.0) ISTEP=1

C
C SETTING DEFAULT TIME STEP AND DURATION
C
IF(TMAX.EQ.0)TMAX=24
IF(NSTEP.EQ.0)NSTEP=24
IF(ISTEP.EQ.1)THEN
```



```

        XSTEP=NSTEP
        TSLF=TMAX/XSTEP
        WRITE(6,301)NSTEP,TSLF,TMAX
301     FORMAT(I5,' TIME STEPS OF',F6.2,' HRS FOR',F6.2,' HRS')
        ENDIF
C
C TESTING IF USE PATTERNS ARE SPECIFIED
C
        DO 302 I=1,5
        LPAT(I)=0
302     CONTINUE
        DO 305 I=1,5
        DO 303 J=1,NSTEP
        IF (XLOF(J,I).NE.0) THEN
            LPAT(I)=1
            GOTO 305
        END IF
303     CONTINUE
305     CONTINUE
        DO 309 I=1,5
        IF(LPAT(I).EQ.0)GO TO 309
        DO 306 J=1,NSTEP
        IF(XLOF(J,I).LE.0)GO TO 307
306     CONTINUE
        GO TO 309
307     WRITE(6,308)I
308     FORMAT('/' ZERO ENTRIES IN LOADING',I4)
309     CONTINUE
C
C ASSIGNING DEFAULT USE PATTERN TO 1
C
        IF(LPAT(1).EQ.0)THEN
            LPAT(1)=1
            DO 311 I=1,24
            XLOF(I,1)=XDEFAL(I)
311     CONTINUE
            WRITE(6,312)
312     FORMAT(' DEFAULT ASSIGNED TO PATTERN 1')
        ENDIF
C
C CHECKING IF ALL NODES ASSIGNED TO A PATTERN
C
        DO 319 I=1,O(2)
        IEX=INO(I)
        IF(DO(I).LE.0)GO TO 319
        IF(DO(I).GE.1E10)GO TO 315
        IF(IPAT(I).GT.5.OR.IPAT(I).LT.1)THEN
            WRITE(6,313)IEX
313     FORMAT('/' NODE',I5,' NOT ASSIGNED TO USE PATTERN'
1         '/' USE PATTERN 1 WILL BE USED')
            IPAT(I)=1
            IF(LPAT(1).EQ.0)LPAT(1)=1
        ENDIF
        GO TO 319
C
C CHECKING IF TANK NODES PROPERLY ASSIGNED
C
315     DO 316 J=1,20
        IF(NTANK(J).EQ.IEX)GO TO 318
316     CONTINUE
```

```
        WRITE(6,317) IEX
317    FORMAT(/' TANK AT NODE ',I4,' WILL BE ASSIGNED CONSTANT '
>      , ' HEAD')
        DO 442 J1 = 1 ,20
442    IF (NTANK(J1).EQ.0) GOTO 448
        WRITE(6,451)
451    FORMAT(' ONLY 20 SUPPLY POINTS ALLOWED.  DELETE EXCESS')
        GOTO 319
448    ART(J1) = 1E10
        NTANK(J1) = IEX
        GOTO 319
318    IF(ART(J).LE.0)THEN
        ART(J) = 1E10
        WRITE(6,320)IEX
320    FORMAT(/' TANK',I5,' IS CONSTANT HEAD TANK')
        ENDIF
        IF(UPL(J).LT.XLOL(J))WRITE(6,321)UPL(J),NTANK(J),XLOL(J)
321    FORMAT(/' UPPER LIMIT',F6.1,' FT OF TANK',I5,
1      /' MUST BE LARGER THAN LOWER LIMIT',F6.1,' FT')
        IF(HETI(J).GT.UPL(J).OR.HETI(J).LT.XLOL(J))WRITE(6,323)
1      HETI(J),NTANK(J),UPL(J),XLOL(J)
323    FORMAT(/' INITIAL LEVEL (' ,F6.1,') OF TANK',I5,
1      /' MUST BE BETWEEN UPPER',F6.1,' AND LOWER ',F6.1,'
LIMITS')
319    CONTINUE
C
C    SORTING TANKS BY SMALLEST TO LARGEST AREA
C
        DO 501 I = 1, 19
        K = I + 1
        IF (NTANK(K).EQ.0) GOTO 503
        DO 500 J = K,20
            IF (NTANK(J).EQ.0) GOTO 501
            IF (ART(I).GT.ART(J)) THEN
                TM1 = ART(I)
                NTM = NTANK(I)
                TM2 = UPL(I)
                TM3 = XLOL(I)
                TM4 = HETI(I)
                ART(I) = ART(J)
                NTANK(I)=NTANK(J)
                UPL(I) = UPL(J)
                XLOL(I) =XLOL(J)
                HETI(I) =HETI(J)
                ART(J) = TM1
                NTANK(J)= NTM
                UPL(J) = TM2
                XLOL(J) = TM3
                HETI(J) = TM4
            END IF
500    CONTINUE
501    CONTINUE
C
C    ASSIGNING DEFAULT RATIO
C
503    IF(RATIO.LE.0)THEN
        RATIO=1.
        WRITE(6,322)
322    FORMAT(' RATIO SET TO 1')
        ENDIF
```

```
C
C CHECK IF FIRE SPECIFICATION CONSISTENT
C
      DO 327 I=1,5
      IF(NFIRE(I).LE.0)GO TO 327
      IF(ITF(I).GE.NSTEP)WRITE(6,325)ITF(I),NFIRE(I),NSTEP
325  FORMAT(' START OF FIRE AT TIME',I5,' AT NODE',I5,
      1      ' IS LATER THAN NUMBER OF STEPS',I5)
      IF(IDF(I).GT.NSTEP)WRITE(6,326)IDF(I),NFIRE(I),NSTEP
326  FORMAT(' END OF FIRE AT TIME',I5,' AT NODE',I5,
      1      ' IS LATER THAN NUMBER OF STEPS',I5)
327  CONTINUE
C
C SETTING LAST TIME LOADING EQUAL TO FIRST
C
      DO 330 I=1,5
      XLOF(NSTEP+1,I)=XLOF(1,I)
330  CONTINUE
C
C.....CHECKING IF EXCLUDE TIME DATA IS CONSISTENT
C
      DO 335 I = 1,ICPE
      DO 333 K=1,7
      IF (TEB(I,K).GE.TMAX) WRITE(6,343)' EXCLUDING',TEB(I,K),'
LINK',
      > IPEX(I),TMAX
      IF (TEE(I,K).GT.TMAX) WRITE(6,344) ' EXCLUDING',TEE(I,K),'
LINK',
      > IPEX(I),TMAX
      IF(TEE(I,K).LT.TEB(I,K))WRITE(6,324)' EXCLUDING',IPEX(I),'
PIPE'
C
C TEB AND TEE ARE SET TO -1 IF TEE NOT GREATER THAN 0
C
      IF (TEE(I,K).LE.0) THEN
      TEB(I,K)=-1.
      TEE(I,K)=-1.
      ENDIF
333  CONTINUE
335  CONTINUE
C
C..... CHECKING IF PUMP DATA IS CONSISTENT
C
      DO 355 I = 1,20
      IF (NPUMP(I).EQ.0) GOTO 356
C
C IF INITIAL PUMP STATUS HAS NOT BEEN SET, IT IS INITIALLY TURNED ON
C
      IF (PUBE(I).NE.'ON '.AND.PUBE(I).NE.'OFF') PUBE(I)='ON '
      DO 350 K = 1,7
      IF (TPB(I,K).GE.TMAX) WRITE(6,343)' PUMPING',TPB(I,K),' PUMP',
      > NPUMP(I),TMAX
343  FORMAT(' START OF',A,' AT TIME',I5,' WITH',A,I5,
      > ' IS LATER THAN DURATION ',F6.0,' HOURS')
      IF (TPE(I,K).GT.TMAX) WRITE(6,344) ' PUMPING',TPE(I,K),' PUMP',
      > NPUMP(I),TMAX
344  FORMAT(' END OF',A,' AT TIME',I5,' WITH',A,I5,
      1      ' IS LATER THAN DURATION ',F6.0,' HOURS')
      IF(TPE(I,K).LT.TPB(I,K))WRITE(6,324)' PUMPING',NPUMP(I),' PUMP'
324  FORMAT(' ENDING TIME OF',A,' IS EARLIER THAN BEGINNING TIME'
```

```
> , ' FOR' ,A,I5)
C
C IF PUMP IS NOT CONTROLLED BY TIME, TPE AND TPB ARE SET TO -1
C AS A FLAG
C
      IF (TPE(I,K).LE.0) THEN
        TPE(I,K)=-1.
        TPB(I,K)=-1.
      ENDIF
      IF (TPB(I,K).EQ.0) PUBE(I)='ON '
350  CONTINUE
      DO 347 J = 1,20
C
C CHECKING PUMPS CONTROLLED BY TANK LEVEL
C
      IF (NOPU(I).EQ.NTANK(J)) THEN
351  IF (HEON(I).GT.UPL(J).OR.HEON(I).LT.XLOL(J).OR.HEOF(J)
>      .GT.UPL(J)) WRITE(6,353) NTANK(J),NPUMP(I)
353  FORMAT(' OFF AND ON TANK LEVELS OF TANK',I4,
>      ' CONTROLLING' ,
>      ' PUMP',I5,/, ' ARE NOT CONSISTENT WITH UPPER AND LOWER' ,
>      ' TANK LEVELS')
      ENDIF
347  CONTINUE
355  CONTINUE
356  RETURN
      ELSE
        WRITE(6,331)KTYPE
331  FORMAT(A4,' NOT RECOGNIZED--ENTRY IGNORED' )
      ENDIF
      GO TO 100
999  CALL BLANK
      GO TO 100
      END
C
C TMRPAR TO ASSIGN ORIGINAL LOADINGS TO TEMPORARY VARIABLES
C SO THEY REMAIN UNCHANGED WHEN RETURNING FROM TIMSIM
C
      SUBROUTINE TMRPAR(NBA,NTANK,NPUMP,NOPU,HEON,HEOF,UPL,XLOL
> ,INO,IPI,DOTMP,ELTMP)
      INTEGER PNL,O
      REAL*8 A,S,G,HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      DIMENSION DOTMP(PNL),ELTMP(20)
> ,NPUMP(21),NOPU(21),HEON(21),HEOF(21),UPL(20),XLOL(20),INO(PNL),
> IPI(PNL),NTANK(20)
C
C ASSIGNING ORIGINAL VALUES TO TEMPORARY VARIABLES
C
      IF (NBA.EQ.1) THEN
        DO 20 I = 1, O(2)
          DOTMP(I) = DO(I)
20  CONTINUE
        DO 25 I1 = 1,20
          IF (NTANK(I1).EQ.0) GOTO 45
          DO 23 J1 = 1, O(2)
            IF (NTANK(I1).EQ.INO(J1)) THEN
              ELTMP(I1) = EL(J1)
```

```
                END IF
23      CONTINUE
25      CONTINUE
        ELSE
C
C  ASSIGNING TEMPORARY VALUES BACK TO ORIGINAL VARIABLES
C
        DO 30 I1 = 1 , O(1)
            IF (CP(I1).LT.0.OR.XL(I1).GT.999999.) GOTO 30
            CP(I1) = 4.72*ABS(XL(I1))/(ABS(HW(I1))**1.85*DI(I1)**4.87)
            A(I1) = 1/CP(I1)
30      CONTINUE
        DO 40 I = 1, O(2)
            DO(I) = DOTMP(I)
40      CONTINUE
        END IF
45      DO 70 I1 = 1, 20
            IF (NTANK(I1).EQ.0) GOTO 75
            DO 60 J1 = 1,O(2)
                IF (NTANK(I1).EQ.INO(J1)) THEN
                    EL(J1) = ELTMP(I1)
                    UPL(I1) = UPL(I1) + ABS(EL(J1))*NBA
                    XLOL(I1)= XLOL(I1)+ ABS(EL(J1))*NBA
                END IF
60      CONTINUE
70      CONTINUE
75      DO 90 I1 = 1,20
            IF (NPUMP(I1).EQ.0) GOTO 100
            DO 80 J1 = 1,O(1)
                IF (IPI(J1).EQ.NPUMP(I1)) THEN
                    DO 77 J2 = 1,O(2)
                        IF (INO(J2).EQ.NOPU(I1)) THEN
                            HEON(I1) = HEON(I1) + ABS(EL(J2))*NBA
                            HEOF(I1) = HEOF(I1) + ABS(EL(J2))*NBA
                        END IF
77      CONTINUE
                    END IF
80      CONTINUE
90      CONTINUE
100     RETURN
        END
C
C  PRINTING OF TIME DATA INPUT
C
        SUBROUTINE TIMEOT
        REAL*8 A,S,G,HE
        INTEGER PNL,O
        PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
        CHARACTER NOLI*4,UNIT*3,PUBE*3
        DIMENSION LIST(5)
        COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
        COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
        COMMON /TOPOL/
        IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
1L)
        COMMON /PRINT/ IPM,IPP,IPE
        COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
```

```
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
COMMON /NAME/ NOLI(11),UNIT(11),PUBE(20)
WRITE(6,10)
10  FORMAT(//' TIME SIMULATION PARAMETERS')
C
C TIME STEP PARAMETERS
C
WRITE(6,20)TMAX,TSLF,NSTEP
20  FORMAT('/' SIMULATION DURATION',F6.2,' HRS'
1/' TIME STEP SIZE      ',F6.2,' HRS'
2/' NUMBER OF TIME STEPS', I5)
WRITE(6,25)RATIO
25  FORMAT(' RATIO OF OUTPUTS FROM SIMULATION',F6.2)
C
C LOADING PATTERN OUTPUT
C
NP=0
DO 31 I=1,5
IF(LPAT(I).EQ.1)THEN
NP=NP+1
LIST(NP)=I
ENDIF
31  CONTINUE
WRITE(6,29)
29  FORMAT(//,10X,' PATTERN')
WRITE(6,32)(LIST(J),J=1,NP)
32  FORMAT(' STEP',5I8)
DO 40 I=1,NSTEP
WRITE(6,38)I,(XLOF(I,LIST(J)),J=1,NP)
38  FORMAT(I5,5F8.2)
40  CONTINUE
C
C TANK LEVEL AND AREA OUTPUT
C
WRITE(6,45)
45  FORMAT(//' TANK DATA'/
1/' NODE      MAX HT      MIN HT      AREA      INIT HT'
2/'          FT          FT          SQFT      FT')
DO 49 I=1,20
IF(NTANK(I).EQ.0)GO TO 60
IF (ART(I).LT.1E10) THEN
WRITE(6,47)NTANK(I),UPL(I),XL0L(I),ART(I),HETI(I)
47  FORMAT(I5,4F9.1)
ELSE
WRITE(6,48)NTANK(I)
48  FORMAT(I5,'      CONSTANT HEAD      1E10')
END IF
49  CONTINUE
C
C FIRE OUTPUTS
C
60  IF(NFIRE(1).EQ.0)THEN
WRITE(6,61)
61  FORMAT(//' NO FIRES SPECIFIED')
ELSE
WRITE(6,65)
65  FORMAT(//' FIRE EVENT DATA'/
1      '/' NODE      START ENDING      FLOW'
2      '/'          STEP  STEP          GPM')
DO 70 I=1,5
```

```
        IF(NFIRE(I).LE.0)GO TO 72
        FLOW=FF(I)
        WRITE(6,67)NFIRE(I),ITF(I),IDF(I),FLOW
67      FORMAT(I4,I8,I6,6X,F6.1)
70      CONTINUE
      ENDIF
C
C EXCLUDING LINKS
C
72     IF(ICPE.LE.0)THEN
        WRITE(6,74)
74     FORMAT(//' NO LINKS OUT OF SERVICE')
        ELSE
        WRITE(6,73)
73     FORMAT(//' LINKS OUT OF SERVICE')
        WRITE(6,75)
75     FORMAT(10X,'START      END'/2X,'PIPE #    HOUR      HOUR')
        DO 79 I=1,ICPE
          IF (TEB(I,1).GE.0) WRITE(6,76)IPEX(I),TEB(I,1),TEE(I,1)
76     FORMAT(2X,I4,4X,2(F6.2,2X))
          DO 77 K=2,7
87     IF (TEB(I,K).GE.0) WRITE(6,78) TEB(I,K),TEE(I,K)
78     FORMAT(10X,2(F6.2,2X))
79     CONTINUE
      ENDIF
C
C LIST OF SPATIAL NODE ASSIGNMENTS
C
      IF(LPAT(1).EQ.1.AND.LPAT(2).EQ.0.AND.LPAT(3).EQ.0.
1AND.LPAT(4).EQ.0.AND.LPAT(5).EQ.0)THEN
        WRITE(6,80)
80     FORMAT(//' ALL NODES ASSIGNED USE PATTERN 1')
        ELSE
        WRITE(6,82)
82     FORMAT(//' NODE USAGE'/5X,' PATTERN')
        DO 88 I=1,PNL
          DO 84 J=1,O(2)
            IF(INO(J).EQ.I)GO TO 86
84     CONTINUE
          GO TO 88
86     IF(DO(J).GE.1E10.OR.DO(J).EQ.0)GO TO 88
          WRITE(6,87)INO(J),IPAT(J)
87     FORMAT(2I5)
88     CONTINUE
      ENDIF
C
C PUMP CONTROLS
C
      PRINT*
      IF (NPUMP(1).EQ.0) THEN
        WRITE(6,89)
89     FORMAT(/,' NO PUMP CONTROLS SPECIFIED')
        ELSE
        WRITE(6,90)
90     FORMAT(/,' PUMP CONTROLS')
        PRINT*
        WRITE(6,91)
91     FORMAT(10X,'START      END          ON          OFF          INITIAL'
>/2X,'PUMP #    HOUR      HOUR      NODE #    LEVEL    LEVEL    ',
>'STATUS')
```

```
DO 99 I = 1,20
  IF (NPUMP(I).EQ.0) GOTO 99
  IF (TPB(I,1).LT.0) THEN
    WRITE(6,93) NPUMP(I),NOPU(I),HEON(I),HEOF(I),PUBE(I)
93    FORMAT(2X,I4,20X,I4,4X,2(F6.2,4X),2X,A3)
  ELSEIF (HEON(I).EQ.0) THEN
    WRITE(6,95) NPUMP(I),TPB(I,1),TPE(I,1),PUBE(I)
95    FORMAT(2X,I4,4X,2(F6.2,2X),32X,A3)
    DO 96 K = 2,7
96      IF (TPB(I,K).GE.0) WRITE(6,98) TPB(I,K),TPE(I,K)
98      FORMAT(10X,2(F6.2,2X))
    ELSE
      WRITE(6,97) NPUMP(I),TPB(I,1),TPE(I,1),NOPU(I),
>      HEON(I),HEOF(I),PUBE(I)
97      FORMAT(2X,I4,4X,2(F6.2,2X),I4,4X,2(F6.2,4X),4X,A3)
      DO 111 K = 2,7
111      IF (TPB(I,K).GE.0) WRITE(6,98) NPUMP(I),TPB(I,K),TPE(I,K)
    ENDIF
99    CONTINUE
    END IF
C
C  NODE AND LINK OUTPUT
C
  PRINT*
  IF (NUSR(1).EQ.0) THEN
    WRITE(6,100)
100  FORMAT(' NO NODES OR LINKS ASSIGNED FOR PRINT OUT')
  ELSE
    J = 0
    DO 103 I = 1, 10
      IF ( NOLI(I) .EQ. 'NODE') THEN
        IF (J.EQ.0) WRITE(6,101)
101      FORMAT(' NODES ASSIGNED FOR PRINT OUT')
        J = 1
        WRITE(6,102) NUSR(I)
102      FORMAT(5X,I4)
      END IF
103      CONTINUE
    IF (J.EQ.0) WRITE(6,104)
104  FORMAT(' NO NODES ASSIGNED FOR PRINT OUT')
    PRINT*
    J = 0
    DO 109 I = 1,10
      IF ( NOLI(I) .EQ. 'LINK') THEN
        IF (J.EQ.0) WRITE(6,105)
105      FORMAT(' LINKS ASSIGNED FOR PRINT OUT')
        J = 1
        WRITE(6,102) NUSR(I)
      ENDIF
109      CONTINUE
    IF (J.EQ.0) WRITE(6,106)
106  FORMAT(' NO LINKS ASSIGNED FOR PRINT OUT')
    ENDIF
    IF (IDTL.EQ.1) THEN
      WRITE(6,107) 'ON '
    ELSE
      WRITE(6,107) 'OFF'
    ENDIF
107  FORMAT(//' DETAIL PRINTING OF TIME STEPS: ',A3)
  RETURN
```



```
END
C
C  STORING TIME DATA
C
      SUBROUTINE STOTIM(O)
      CHARACTER ST*9,NOLI*4,UNIT*3,PUBE*3
      INTEGER PNL,O
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XLLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
      COMMON /NAME/ NOLI(11),UNIT(11),PUBE(20)
      COMMON /PRINT/ IPM,IPP,IPE
      DIMENSION O(15)
      IF (IPP.EQ.1) WRITE(6,10)
10  FORMAT(/,' ENTER FILE NAME ',/)
      READ(*,20,END=99) ST
      IF (IPE.EQ.1) WRITE(6,21) ST
20  FORMAT(A9)
21  FORMAT(1X,A9)
      OPEN(1,FILE=ST,ERR=999,RECL=72)
      WRITE(1,30,ERR=999) TMAX,TSLF,NSTEP,RATIO
30  FORMAT(2F7.3,I4,F7.3)
      WRITE(1,60,ERR=999) O(2),(IPAT(J),J=1,O(2))
60  FORMAT(16I4)
      NP=0
      DO 65 I = 1,5
      DO 63 J = 1,O(2)
      IF (IPAT(J).EQ.I) THEN
      NP=NP+1
      GOTO 65
      END IF
63  CONTINUE
65  CONTINUE
      WRITE(1,70,ERR=999)NP,((XLOF(J,I),J=1,NSTEP),I=1,NP)
70  FORMAT(I1,(5F7.3))
      WRITE(1,80,ERR=999) (LPAT(I),I=1,5)
80  FORMAT(5I4)
      NF=5
      DO 85 I = 1,5
85  IF (NFIRE(I).EQ.0) GOTO 86
86  IF (I.NE.5)NF=I-1
      WRITE(1,90,ERR=999)NF,(NFIRE(I),ITF(I),IDF(I),FF(I),I=1,NF)
90  FORMAT(I1,(3I4,E16.9))
      NTN=20
      DO 95 I=1,20
95  IF (NTANK(I).EQ.0) GOTO 97
97  IF (I.NE.20)NTN=I-1
      WRITE(1,100,ERR=999) NTN,(NTANK(I),UPL(I),XLLOL(I),ART(I),
>HETI(I),I=1,NTN)
100  FORMAT(I4,1(I4,4E16.9))
      WRITE(1,110,ERR=999) ICPE,(IPEX(I),I=1,ICPE)
110  FORMAT(16I4)
      WRITE(1,112,ERR=999)((TEB(I,K),TEE(I,K),K=1,7),I=1,ICPE)
112  FORMAT(12F6.2)
      NPUC=20
      DO 115 I=1,20
115  IF (NPUMP(I).EQ.0) GOTO 117
```

```
117 IF (I.NE.20)NPUC=I-1
WRITE(1,120,ERR=999) NPUC,(NPUMP(I),NOPU(I),PUBE(I),I=1,NPUC)
120 FORMAT(I4,5(2I4,A3))
WRITE(1,130,ERR=999) ((TPB(I,K),TPE(I,K),K=1,7),I=1,NPUC)
130 FORMAT(12F6.2)
WRITE(1,133) (HEON(I),HEOF(I),I=1,NPUC)
133 FORMAT(4E16.9)
WRITE(1,140,ERR=999) (NUSR(I),I=1,10)
140 FORMAT(10I4)
WRITE(1,150,ERR=999) (NOLI(I),UNIT(I),I=1,10)
150 FORMAT(2(A4,A3))
REWIND 1
CLOSE(1)
RETURN
99 CALL BLANK
RETURN
999 PRINT*,'PROGRAM CANNOT ACCESS THIS FILE'
CLOSE(1)
RETURN
END

C
C *****RETRIEVING OF TIME SIMULATION DATA*****
C
SUBROUTINE RETIME
CHARACTER ST*9,NOLI*4,UNIT*3,PUBE*3
INTEGER PNL
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XLLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
COMMON /NAME/ NOLI(11),UNIT(11),PUBE(20)
COMMON /PRINT/ IPM,IPP,IPE
IF (IPP.EQ.1) WRITE(6,10)
10 FORMAT(/,' ENTER FILE NAME ',/)
READ(*,20,END=99) ST
IF (IPE.EQ.1) WRITE(6,21) ST
20 FORMAT(A9)
21 FORMAT(1X,A9)
OPEN(1,ERR=999,FILE=ST,STATUS='OLD',RECL=72)
REWIND 1
READ(1,30,ERR=999) TMAX,TSLF,NSTEP,RATIO
30 FORMAT(2F7.3,I4,F7.3)
DO 40 I = 1,PNL
40 IPAT(I)=0
READ(1,60,ERR=999,END=999) I,(IPAT(J),J=1,I)
60 FORMAT(16I4)
READ(1,70,ERR=999,END=999)NP,((XLOF(J,I),J=1,NSTEP),I=1,NP)
70 FORMAT(I1,(5F7.3))
READ(1,80,ERR=999,END=999) (LPAT(I),I=1,5)
80 FORMAT(5I4)
DO 85 I=1,5
85 NFIRE(I)=0

READ(1,90,ERR=999,END=999)NF,(NFIRE(I),ITF(I),IDF(I),FF(I),I=1,NF)
90 FORMAT(I1,(3I4,E16.9))
READ(1,100,ERR=999,END=999) NTN,(NTANK(I),UPL(I),XLLOL(I),ART(I),
+HETI(I),I=1,NTN)
100 FORMAT(I4,1(I4,4E16.9))
```

```
      IF (NTN.NE.20) THEN
      DO 105 I=NTN+1,20
105  NTANK(I)=0
      END IF
      READ(1,110,ERR=999,END=999) ICPE,(IPEX(I),I=1,ICPE)
110  FORMAT(16I4)
      READ(1,112,ERR=999,END=999)((TEB(I,K),TEE(I,K),K=1,7),I=1,ICPE)
112  FORMAT(12F6.2)
      IF (ICPE.NE.15) THEN
      DO 115 I=ICPE+1,15
          IPEX(I)=0
          DO 113 K=1,7
              TEB(I,K)=0
113          TEE(I,K)=0
115  CONTINUE
      END IF
      READ(1,120,ERR=999,END=999) NPUC,(NPUMP(I),NOPU(I),PUBE(I),
>I=1,NPUC)
120  FORMAT(I4,5(2I4,A3))
      IF (NPUC.NE.20) THEN
      DO 125 I=NPUC+1,20
          NPUMP(I)=0
          DO 123 K=1,7
              TPB(I,K)=0
123          TPE(I,K)=0
125  CONTINUE
      END IF
      READ(1,130,ERR=999,END=999)((TPB(I,K),TPE(I,K),K=1,7),I=1,NPUC)
130  FORMAT(12F6.2)
      READ(1,133,ERR=999,END=999) (HEON(I),HEOF(I),I=1,NPUC)
133  FORMAT(4E16.9)
      READ(1,140,ERR=999,END=999) (NUSR(I),I=1,10)
140  FORMAT(10I4)
      READ(1,150,ERR=999,END=999) (NOLI(I),UNIT(I),I=1,10)
150  FORMAT(2(A4,A3))
      REWIND 1
      CLOSE(1)
      RETURN
99  CALL BLANK
      RETURN
999  PRINT*, 'PROGRAM CANNOT ACCESS THIS FILE'
      CLOSE(1)
      RETURN
      END
      SUBROUTINE TIMSIM(DOTMP,ELTMP)
C
      INTEGER PNL,0
      REAL*8 A,S,G,HE
      CHARACTER ST*3,RES1*10,RES2*2
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      CHARACTER NOLI*4,UNIT*3,PUBE*3
      COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XLLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
      COMMON /NAME/ NOLI(11),UNIT(11),PUBE(20)
      COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
```

```

COMMON /TOPOL/
IBE(PNL), IEN(PNL), IPI(PNL), INO(PNL), IBI(PNL), IEI(PN
1L)
COMMON /MATRI/ M(LNL), N1(LNL, MNL), N2(LNL, MNL)
COMMON /ACCU/ PRAC, FLAC, HWMA, ICL
COMMON /PRINT/ IPM, IPP, IPE
DIMENSION TAH(20, 114), DO(20), XLP(57), ILP(57), PT(5, 20),
>EF(20), N3(PNL), PUST(20, 114), DOTMP(PNL), ELTMP(20)
RES1='0123456789'
RES2='NR'
C
C CHECKING IF TIME DATA HAS BEEN ENTERED (TMAX>0)
C
IF (TMAX.LE.0) THEN
PRINT*, 'PRIOR TO RUNNING AN EXTENDED PERIOD SIMULATION, '
PRINT*, 'TIME DATA MUST BE ENTERED'
RETURN
END IF
C
C CHECKING IF SPARSE MATRIX IS SET UP (O(4)=1) AND SYSTEM IS
C BALANCED (O(9)=3)
IF (O(4).EQ.0.OR.O(9).LT.0) THEN
PRINT*, 'PRIOR TO RUNNING AN EXTENDED PERIOD SIMULATION, '
PRINT*, 'THE SYSTEM MUST BE BALANCED'
CALL SIMBAL('0C')
END IF
C
C3=448.831
C
C COUNTING NUMBER OF TANKS ASSIGNED FOR EPS - RECORDED IN ITN
C NUMBER OF FIXED HEAD TANKS (RESERVOIRS) RECORDED IN ITNC
C
ITNC=0
DO 5 J1=1, 20
IF (ART(J1).GE.1E10) ITNC=ITNC+1
5 IF (NTANK(J1).EQ.0) GOTO 6
6 ITN=J1-1
C
C SEARCHING FOR SUPPLY POINTS
C
DO 20 I1=1, O(2)
IF (DO(I1).GT.9E9) THEN
DO 10 J1=1, ITN
IF (NTANK(J1).EQ.0) GOTO 20
IF (NTANK(J1).EQ.INO(I1)) THEN
C
C ASSIGNING INITIAL TANK LEVEL TO TAH FOR FIRST TIME STEP
C
HE(I1) = HETI(J1) + ABS(EL(I1))
TAH(J1, 1)=HE(I1)
C
C FLAGGING VARIABLE TANK BY MULTIPLYING DO BY 1E10
C
DO(I1)=1E10*DO(I1)
GOTO 20
END IF
10 CONTINUE
ELSE
C
C MULTIPLYING NON SUPPLY POINTS BY RATIO

```

```
C
      DO(I1)=DO(I1)*RATIO
      END IF
20  CONTINUE
C
C  INITIALIZING TIME STEP VARIABLES
C
      ILC = 0
      ILF=0
      NUM = 0
      I66 = 0
      ITC = 2
      DT=.01
      TOT=-.01
      ITFL1=0
C
C  CALLING PIPEX TO EXCLUDE PIPES WHICH USER HAS IDENTIFIED
C
      CALL PIPEX (O,A,CP,XL,HW,DI,IPEX,ICPE,IPI,TOT,TEB,TEE)
C
C  CALLING PUMPCO TO TURN ON OR OFF PUMPS CONTROLLED BY TIME STEP
C
      CALL PUMPCO(O,HE,IPI,ILC,INO,TOT,PUBE,PUST)
C
C  INCREMENTING TIME STEP COUNTER
C
30  ILC=ILC+1
C
C  ASSIGNING LOADS
C
      CALL LOAD (NFIRE,ITF,O,INO,FF,DO,IPAT,XLOF,IDF,ILC,DOTMP,RATIO
      >,TOT)
C
C  IF VARIABLE TIME STEP IS TOO SMALL, IT IS ASSIGNED TO 2.5 MINUTES
C
      IF (ABS(DT-.01).GT.1E-5) DT=2.5
50  DO 70 I1=1,0(2)
      IF (DO(I1).LT.9E9) GOTO 60
C
C  ASSIGNING ELEMENTS S AND G FOR SUPPLY POINTS
C
      S(I1)=1E10
      G(I1)=DO(I1)+(ABS(EL(I1))-100)*1E10
C
C  ASSINING G FOR VARIABLE LEVEL TANK
C
      IF (DO(I1).GT.1E20) G(I1)=HE(I1)*1E10
      GOTO 70
C
60  S(I1)=0
      G(I1)=-DO(I1)/C3
70  CONTINUE
C
      IF (O(10).EQ.0) GOTO 90
C
      DO 80 I2=1,0(1)
C
C  CHECKING FOR PUMPS
C
      IF (CP(I2).LT.0) GOTO 80
```

```
C
C CHECKING FOR PRV'S
C
      IF (XL(I2).GT.999999.) THEN
C
C OPENING PRV BY ASSIGNING A TO 10000
C
      A(I2)=10000
      O(10)=10+MOD(INT(O(10)),10)
      END IF
C
C CHECKING FOR CHECK VALVES
C
      IF (XL(I2).LT.0) THEN
      A(I2)=1/CP(I2)
      O(10)=1+INT(O(10)/10)*10
      END IF
80 CONTINUE
C
90 I6=0
   IFOP=-1
C
C...CALLING ON EQSOLV TO BALANCE SYSTEM
C
      CALL EQSOLV(ST,IFOP,I6,XP,P8,P88,M0,PT,EF,TP,N3,O4)
C
C I6 INDICATES NON CONVERGENCE WILL RETURN AND TRY ONE MORE TIME
C
      IF (I6.EQ.1000) THEN
      I66 = I66 + 1
      IF (I66.LT.2) GOTO 50
      I66 = 0
      IFOP = 999
      END IF
C
C...CHECK FOR SYSTEM NON BALANCE, IFOP=999
C
      IF (IFOP.EQ.999.OR.NUM.NE.0) THEN
      O(9)=3
      IF (ILC.LT.NSTEP.AND.IFOP.EQ.999) THEN
      WRITE(6,97) ILC*TSLF
      ILC = ILC-1
      ELSEIF (IFOP.EQ.999) THEN
      WRITE(6,97) ITC*TSLF
97   FORMAT('/' SYSTEM NOT BALANCED AT TIME ',F6.2,' HOURS',/
>         ' CHECK FOR PORTIONS OF SYSTEM DISCONNECTED FROM
TANKS'//)
      ELSE
C
C.....DISPLAYING LOADINGS , FLOWS AND PRESSURES, FOR INDICATED TIME
C
      WRITE(6,102)
102   FORMAT('/'          LOADINGS  '// ' PATTERN  LOADING FACTOR')
      DO 88 I = 1,5
      DO 84 J = 1,O(2)
      IF (IPAT(J).EQ.I) THEN
      WRITE(6,103) IPAT(J),XLOF(ILF,IPAT(J))
103   FORMAT(1X,I5,5X,F8.2)
      GOTO 88
      END IF
```

```
84      CONTINUE
88      CONTINUE
      WRITE(6,98)
98      FORMAT(' FLOWS AND PRESSURES  '//)
      END IF
      CALL PRNOUT('0C')
      GOTO 200
      END IF
C
C...UPDATING TIME, INITIALLY TOT=-.01, DT=.01 SO TOT+DT=0
C
      TOT=TOT+DT
C
C...CALLING ON TSOUT TO DISPLAY FLOWS AND HEADS FROM PREVIOUS BALANCE
C
      IF (ILC.GT.NSTEP) GOTO 200
      ENDTM=AMOD(TOT+1E-5,60.*TSLF)
      IF (ENDTM.LT.1E-4.OR.ABS(ENDTM-60.*TSLF).LE.1E-3.OR.
>IDTL.EQ.1) CALL TSOUT(ILC,TOT,ITFL)
C
C.....TANK CONTROL
C
      ITFL=0
      DO 130 I1=1,O(1)
      IB=IBI(I1)
      IE=IEI(I1)
      IF (DO(IB).LT.1E19.AND.DO(IE).LT.1E19) GOTO 130
100     IF (DO(IB).GT.1E19) THEN
      DO 110 J1=1,ITN
      IF (ART(J1).GE.1E10) THEN
      TAH(J1,ILC+1)=HE(IB)
      ITFL = 1
      GOTO 130
      END IF
C
C      SEARCHING FOR USER'S TANK NUMBER
C
      IF (NTANK(J1).EQ.INO(IB)) GOTO 120
110     CONTINUE
120     H1 = HE(IE)-HE(IB)
C
C      UPDATING HEAD
C
      D0(J1)=(ABS(HE(IE)-HE(IE))/CP(I1))**.54*SIGN(1.,H1)
      HE(IE)=HE(IE)+D0(J1)*DT*60/ART(J1)
      IF ((ENDTM.LT.1E-4.OR.ABS(ENDTM-60.*TSLF).LE.1E-3).AND.
>      TOT.GT.1E-4) TAH(J1,ILC+1)=HE(IE)
C
C      CHECKING IF PIPE CONNECTED TO TANK IS OPEN
C
      IF (CP(I1).LT.1E19) THEN
C
C      ASSIGNING TANK LEVEL AS UPPER LEVEL AND CLOSING PIPE
C      CONNECTED TO TANK
C
      IF (HE(IE).GT.UPL(J1).AND.D0(J1).GT.0) THEN
      WRITE(6,125) TOT/60.,NTANK(J1),' FULL'
      CP(I1)=1E20
      HE(IE)=UPL(J1)
      END IF
```

```
C
C ASSIGNING TANK LEVEL AS LOWER LEVEL AND CLOSING PIPE
C CONNECTED TO TANK
C
      IF (HE(IB).LT.XLOL(J1).AND.DO(J1).LT.0) THEN
        WRITE(6,125) TOT/60.,NTANK(J1),' EMPTY'
        CP(I1)=1E20
        HE(IB)=XLOL(J1)
      END IF
125   FORMAT(1X,F6.2,6X,'TANK',I4,' : DISCONNECTED',A6)
      ELSE
C
C OPENING PIPE CONNECTED TO TANK AND RESETTING TANK LEVELS
C
      IF (HE(IE).LT.HE(IB).AND.HE(IB).GT.UPL(J1)) THEN
        WRITE(6,127) TOT/60.,NTANK(J1)
        CP(I1)=4.72*ABS(XL(I1))/(ABS(HW(I1))**1.85*DI(I1)**4.87)
        HE(IB)=UPL(J1)
      END IF
      IF (HE(IE).GT.HE(IB).AND.HE(IB).LT.XLOL(J1)) THEN
        WRITE(6,127) TOT/60.,NTANK(J1)
        CP(I1)=4.72*ABS(XL(I1))/(ABS(HW(I1))**1.85*DI(I1)**4.87)
        HE(IB)=XLOL(J1)
      END IF
127   FORMAT(1X,F6.2,6X,'TANK',I4,' : CONNECTED')
      END IF
C
C WHAT'S HAPPENING HERE ?
C
      IF (EL(IB).GE.0) THEN
        EL(IB)=ABS(HE(IB))
      ELSE
        EL(IB)=-ABS(HE(IB))
      END IF
      DO(IB)=1E22
      IF (IE.EQ.IBI(I1)) GOTO 130
      END IF
C
C.....INDICATING TANK IS CONNECTED BY SETTING TANK FLAG, ITFL, TO 1
C
      IF (CP(I1).LT.1E19.OR.ABS(HE(IB)-UPL(J1)).LT.1E-2) ITFL =1
C
C.....CHECKING IF ENDING NODE OF PIPE CONNECTED TO TANK IS THE TANK
C NODE
      IF (DO(IE).GT.1E19) THEN
        IE=IBI(I1)
        IB=IEI(I1)
        GOTO 100
      END IF
C
130  CONTINUE
C
C...PUMP CONTROLS
C
      CALL PUMPCO(O,HE,IPI,ILC,INO,TOT,PUBE,PUST)
C
C...EXCLUDING PIPES
C
      CALL PIPEX (O,A,CP,XL,HW,DI,IPEX,ICPE,IPI,TOT,TEB,TEE)
C
```



```
C....CHECKING IF TANKS ARE EMPTY, ITFL=0 & ITFL1=0
C
  IF (ITFL.EQ.0.AND.ITFL1.EQ.0) THEN
    PRINT*, 'ALL TANKS ARE EMPTY'
    ITFL1=1
    GOTO 200
  END IF
C
C....CHECKING FOR END OF TIME STEP, RECORD PUMP STATUS, AND MIN PR.
C
  IF (ENDTM.LT.1E-4.OR.ABS(ENDTM-60.*TSLF).LE.1E-3) THEN
    IF (IPR.EQ.1) THEN
      O(9) =3
      CALL PRNOUT('0C')
    END IF
    CALL LOWP (O,DO,HE,EL,XLP,ILP,ILC,TOT,TSLF)
    IF (NPUMP(1).NE.0) THEN
      DO 91 K1 = 1 ,20
        IF (NPUMP(K1).NE.0) THEN
          DO 93 K2 = 1, O(1)
93          IF (IPI(K2).EQ.NPUMP(K1)) PUST(K1,ITC) = XL(K2)
          END IF
91          CONTINUE
        END IF
        ITC = ITC + 1
        IF (TOT.GT..1) GOTO 30
        DT=2.5
        GOTO 50
      END IF
C
C....CHECK FOR BEGINNING OF TIME STEP, RECORD PUMP STATUS, MIN PR.
C
  IF (AMOD(TOT-2.49999,60.*TSLF).LT.1E-4.AND.TOT.GT.3.) THEN
    IF (IPR.EQ.1) THEN
      O(9) =3
      CALL PRNOUT('0C')
    END IF
    CALL LOWP (O,DO,HE,EL,XLP,ILP,ILC,TOT,TSLF)
    IF (NPUMP(1).NE.0) THEN
      DO 95 K1 = 1 ,20
        IF (NPUMP(K1).NE.0) THEN
          DO 94 K2 = 1, O(1)
94          IF (IPI(K2).EQ.NPUMP(K1)) PUST(K1,ITC) = XL(K2)
          END IF
95          CONTINUE
        END IF
        ITC = ITC + 1
      END IF
C
C....ASSIGNING VARIABLE TIME STEP WITH TIMSTE
C
  CALL TIMSTE (DO,DT,ITN,TOT,ILC)
C
  GOTO 50
C
200  IF (ILF.GT.0) GOTO 220
    O(9)=3
    O(7)=-1
C
C....DISPLAYING MIN PRESSURES AND TANK LEVELS FOR EACH TIME STEP
```

```
C
  PRINT*
  PRINT*
198 WRITE(6,199)
199 FORMAT(' TANK WATER LEVELS AND MINIMUM PRESSURES '
>/' OVER DURATION OF TIME SIMULATION')
  PRINT*
  PRINT*, ' NUMBER   TIME,HR   MIN.PR.   NODE   TANK WATER LEVELS '
  PRINT 201,(NTANK(I),I=1,ITN-ITNC)
201  FORMAT (/ ,28X,6I9)
  PRINT*
  PRINT 209,1, .0,XLP(1),INO(ILP(1)),(TAH(I,1)-ABS(ELTMP(I))
>,I=1,ITN-ITNC)
209  FORMAT (1X,I2,7X,F6.2,'+',F8.1,I5,11F9.1)
  IF (ILC.LT.NSTEP) THEN
    ISTEP=ILC
  ELSE
    ISTEP=NSTEP
  END IF
  DO 210 J=2,2*ISTEP,2
    IF (ILP(J).NE.0) I1=INO(ILP(J))
    PRINT 211,J,(J/2)*TSLF,XLP(J),I1,(TAH(I,J/2+1)-ABS(ELTMP(I))
>,I=1,ITN-ITNC)
    IF (J.EQ.2*NSTEP) GOTO 220
211  FORMAT (1X,I2,7X,F6.2,'-',F8.1,I5,11F9.1)
    IF (ILP(J+1).NE.0) I1=INO(ILP(J+1))
210  PRINT 212,J+1,(J/2)*TSLF,XLP(J+1),I1
212  FORMAT (1X,I2,7X,F6.2,'+',F8.1,I5)
C
C...PROMPTING FOR NUMBER CORRESPONDING TO TIME STEP
C
220  PRINT*
227  IF (IPP.EQ.1) WRITE(6,230)
230  FORMAT(' ENTER NUMBER CORRESPONDING TO TIME STEP TO VIEW '
>,' FLOWS AND PRESSURES '/' FOR ENTIRE SYSTEM (ENTER N FOR '
>,'NO VIEWING, R TO RETREIVE TABLE)'/)
  READ(*,233,ERR=999) ST
233  FORMAT(A)
  IF (IPE.EQ.1) WRITE(6,237) ST
237  FORMAT(1X,A)
C
C DETERMINING WHETHER ST IS A NUMBER OR CHARACTER
C
  IF (INDEX(RES1,ST(1:1)).EQ.0) THEN
    IF (INDEX(RES2,ST(1:1)).EQ.0) THEN
      WRITE(6,235)
235  FORMAT(' INVALID ENTRY - TRY AGAIN')
      GOTO 227
    ELSE
      IF (ST(1:1).EQ.'N') THEN
C
C ASSIGNING TANK HEIGHTS BACK TO INITIAL TANK HEIGHTS
C
        DO 234 I1 = 1,ITN
          DO 234 J1 = 1, O(2)
            IF (INO(J1).EQ.NTANK(I1)) HE(J1) = TAH(I1,1)
234  CONTINUE
          RETURN
        ELSE
C
```

```
C      ST = 'R' SO DISPLAY TABLE AGAIN
C
C          GOTO 198
C          END IF
C          END IF
C          END IF
C
C      ST IS A NUMBER - CONVERTING STRING ST TO A NUMBER
C
C          NM1=INDEX(RES1,ST(1:1))-1
C          NM2=INDEX(RES1,ST(2:2))-1
C          NM3=INDEX(RES1,ST(3:3))-1
C          IF (NM2.EQ.-1.AND.NM3.EQ.-1) THEN
C              NUM = NM1
C          ELSEIF (NM3.EQ.-1.AND.NM2.NE.-1) THEN
C              NUM = NM1*10+NM2
C          ELSE
C              NUM = NM1*100+NM2*10+NM3
C          ENDIF
C
C      CHECKING IF NUMBER IS IN CORRECT RANGE
C
C          IF (NUM.LE.0.OR.NUM.GT.(2*ISTEP+1)) THEN
C              WRITE(6,235)
C              GOTO 227
C          END IF
C          ITC = ILC
C
C      ASSIGNING TIME STEP FOR TANKS, ITC, AND LOADS, ILF, BASED ON NUM
C
C          IF (MOD(NUM,2).EQ.0) THEN
C              ITC = NUM/2 +1
C              ILF = NUM/2
C          ELSE
C              ITC = (NUM-1)/2+1
C              ILF = ITC
C          END IF
C          IF (ILF.LE.0) ILF=1
C          IF (ITC.EQ.ILF) THEN
C              WRITE(6,299) (ITC-1)*TSLF,'+'
C          ELSE
C              WRITE(6,299) (ITC-1)*TSLF,'-'
C          END IF
299  FORMAT('// '   OUTPUT FOR TIME ',F6.2,A,' HOURS'/)
C
C      ...REASSIGNING CP,A WITH PIPEX
C
C          CALL PIPEX (O,A,CP,XL,HW,DI,IPEX,ICPE,IPI,-1.,TEB,TEE)
C
C      ...EXCLUDING SPECIFIED PIPES FOR INDICATED TIME STEP
C
C          DO 305 K=1,ICPE
C          DO 303 I1=1,O(1)
C          IF (CP(I1).GT.0.AND.XL(I1).LT.999999..AND.XL(I1).GT.0.AND.
>IPEX(K).EQ.IPI(I1)) THEN
C              DO 301 K1=1,7
C
C          IF (ITF(K).LT.ILC-1.AND.IDF(K).GE.ILC-1) THEN
C          IF (ITF(K).LE.ILC-1.AND.IDF(K).GT.ILC-1) THEN
C
```

```

        IF (TEB(K,K1).LE.((ILF-1)*TSLF).AND.TEE(K,K1).GT.
>          ((ILF-1)*TSLF)) THEN
            CP(I1)=1E10
            A(I1)=1E-10
            WRITE(6,302) IPEX(K)
302        FORMAT(1X,'PIPE',I4,' CLOSED ')
            GOTO 305
        END IF
301    CONTINUE
        GOTO 305
    END IF
303 CONTINUE
305 CONTINUE
C
C....ASSIGNING LOADS AND ANY FIRE FLOWS FOR INDICATED TIME STEP
C
        CALL LOAD (NFIRE,ITF,O,INO,FF,DO,IPAT,XLOF,IDF,ILF,DOTMP,RATIO
> ,((ITC-1)*TSLF*60))
        DO 251 I1 = 1,20
            IF (NTANK(I1).EQ.0) GOTO 247
            DO 245 J1 = 1,O(2)
C
C....ASSIGNING TANK LEVELS FOR INDICATED TIME STEP AND REASSIGNING
C TANK NODE ELEVATIONS
        IF (NTANK(I1).EQ.INO(J1).AND.ART(I1).LT.1E10) THEN
            HE(J1) = TAH(I1,ITC)
            EL(J1) = ELTMP(I1)
C
C....CHECKING FOR DISCONNECTED TANKS
C
        IF (TAH(I1,ITC).GE.UPL(I1).OR.TAH(I1,ITC).LE.XLOL(I1))
>          THEN
            DO 255 K1 = 1, O(1)
C
C....ASSIGNING CP OF 1E20 TO DISCONNECT LINKS CONNECTED TO TANK
C
            IF (IBE(K1).EQ.INO(J1).OR.IEN(K1).EQ.INO(J1))
>          CP(K1)=1E20
255        CONTINUE
            END IF
        END IF
245 CONTINUE
247 IF (NPUMP(I1).EQ.0) GOTO 253
        DO 249 J1 = 1,O(1)
C
C....SEARCHING FOR PUMPS AND ASSIGNING PUMP STATUS FOR INDICATED TIME
C STEP
        IF (IPI(J1).EQ.NPUMP(I1)) THEN
            XL(J1) = PUST(I1,NUM+1)
            IF (INT(XL(J1)).EQ.-1) THEN
                A(J1) = 0
            ELSE
                A(J1) = 1.85/SQRT(DI(J1)**2-2*CP(J1)*HW(J1))
            END IF
        END IF
249 CONTINUE
251 CONTINUE
253 GOTO 50
999 CALL BLANK
        GOTO 227
```

```
END
C
SUBROUTINE TIMSTE (DO,DT,ITN,TOT,ILC)
INTEGER PNL,O
REAL*8 A,S,G,HE
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
COMMON /TOPOL/
IBE(PNL),IEN(PNL),IPI(PNL),INO(PNL),IBI(PNL),IEI(PN
>L)
COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XLOL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
DIMENSION DO(20),DDO(20),HDE(PNL),DTC(2)
IF (ILC*60*TSLF-TOT-.1.LT.1E-4) THEN
  DT=.1
  GOTO 150
END IF
ENDTM=AMOD(TOT+1E-5,60.*TSLF)
IF (ENDTM.LT.1E-4) THEN
  DT = 2.5
  GOTO 140
END IF
DTC(1) = 1000.
IF (TOT.GT.1E-4) THEN
C
C...SEARCHING FOR TANKS
C
  DO 200 I1=1,ITN
C
C...DO<0 MEANS TANK IS DRAINING - OUTFLOW
C  DO>0 MEANS TANK IS FILLING - INFLOW
C
  IF (DO(I1).LT.0.) THEN
    HC=XLOL(I1)
  ELSEIF (DO(I1).GT.0.) THEN
    HC=UPL(I1)
  ELSE
    GOTO 200
  END IF
  DO 195 I2 = 1,O(2)
  IF (INO(I2).NE.NTANK(I1)) GOTO 195
C
C....SEARCHING FOR DISCONNECTED TANKS
C
  IF ((HC-HE(I2))/DO(I1).LE.0) THEN
    DO 193 J1 = 1,O(1)
    IE = IEI(J1)
    IB = IBI(J1)
    IF (DO(IE).GT.1E19.AND.INO(IE).EQ.NTANK(I1)) THEN
      INB=IE
      INE=IB
    ELSEIF (DO(IB).GT.1E19.AND.INO(IB).EQ.NTANK(I1)) THEN
      INB=IB
      INE=IE
    ELSE
      GOTO 193
    
```

```
                ENDIF
C
C      HE = UPL MEANS TANK IS DISCONNECTED FULL
C
                IF (ABS(HE(INB)-UPL(I1)).LT.0.1) THEN
                    HC = UPL(I1)
                ELSE
                    HC = XLOL(I1)
                END IF
C
C      EXPECTED DTC (TIME STEP IN MINUTES) WHEN TANK WILL BE CONNECTED
C
                IF (HDE(INE).NE.HE(INE))
>                    DTC(2) = DT*(HE(INE)-HC)/(HDE(INE)-HE(INE))
C      PRINT*, 'DIS ',DTC(2),DT,HE(INE),HC,HDE(INE)
193      CONTINUE
                ELSE
C
C      EXPECTED DTC WHEN TANK WILL BE DISCONNECTED
C
                    DTC(2)=ART(I1)*(HC-HE(I2))/(D0(I1)*60.)
                END IF
195      CONTINUE
                IF (DTC(2).LT.DTC(1).AND.DTC(2).GT.0) DTC(1)=DTC(2)
C      PRINT*, 'T DTC ',HC,D0',NTANK(I1),DTC(2),HC,D0(I1)
197      FORMAT(F7.3)
200      CONTINUE
                IF (NPUMP(1).NE.0) THEN
C
C....CHECKING FOR PUMPS CONTROLLED BY TANK LEVEL
C
                DO 230 J = 1,20
                    IF (NOPU(J).EQ.0) GOTO 232
                    ITNK=0
                    DO 210 I = 1,ITN
                        IF (NOPU(J).EQ.NTANK(I)) THEN
                            DO 203 I1 = 1,O(1)
                                IF (IPI(I1).EQ.NPUMP(J)) THEN
                                    ITNK=1
                                    IF (D0(I).LT.0.AND.XL(I1).LT.0) THEN
                                        HC = HEON(J)
                                    ELSEIF (D0(I).GT.0.AND.XL(I1).GT.0) THEN
                                        HC = HEOF(J)
                                    ELSE
                                        GOTO 230
                                    END IF
                                END IF
                            GOTO 204
                        END IF
203      CONTINUE
204      DO 205 I2 = 1,O(2)
205      IF (INO(I2).EQ.NOPU(J).AND.HE(I2).NE.HC)
>                    DTC(2)=ART(I)*(HC-HE(I2))/(D0(I)*60.)
                    IF (DTC(2).LT.DTC(1).AND.DTC(2).GT.0) DTC(1)=DTC(2)
C      PRINT*, 'P DTC,HC,D0 ',NPUMP(J),DTC(2),HC,D0(I)
                    GOTO 230
                END IF
210      CONTINUE
                IF (ITNK.NE.1) THEN
C
C....CHECKING FOR PUMPS CONTROLLED BY NON TANK NODES
```

```
C
      DO 220 I2 = 1,0(2)
        IF (NOPU(J).EQ.INO(I2)) THEN
C
C     HDE IS HEAD AT PREVIOUS TIME STEP
C     HE < HDE MEANS HEAD IS DROPPING, XL=-1 MEANS PUMP OFF
C
          DO 223 I1 = 1,0(1)
            IF (IPI(I1).EQ.NPUMP(J)) THEN
              IF (HE(I2).LT.HDE(I2).AND.XL(I1).LT.0) THEN
                HC = HEON(J)
              ELSEIF (HE(I2).GT.HDE(I2).AND.XL(I1).GT.0) THEN
                HC = HEOF(J)
              ELSE
                GOTO 230
              END IF
            GOTO 225
          END IF
223      CONTINUE
225      DTC(2)=DT*(HE(I2)-HC)/(HDE(I2)-HE(I2))
          IF (DTC(2).LT.DTC(1).AND.DTC(2).GT.0) DTC(1)=DTC(2)
C          PRINT*, 'NT DTC,HC,HDE ',NPUMP(J),DTC(2),HC,HDE(I2)
          GOTO 230
        END IF
220      CONTINUE
      END IF
230      CONTINUE
      END IF
232      TB1=1E10
          TE1=1E10
          TPMP=1E10
C
C.....CHECKING FOR TIME CONTROLLED PUMPS
C
      DO 229 J=1,20
        IF (NPUMP(J).EQ.0) GOTO 235
        IF (TPE(J,1).LE.0) GOTO 229
        DO 226 I=1,0(1)
          IF (IPI(I).EQ.NPUMP(J)) THEN
            IF (XL(I).LT.0) THEN
C
C     FINDING SMALLEST ON TIME >TOT/60 FOR OFF PUMPS (XL=-1)
C
              DO 227 K=1,7
                IF (TPB(J,K).LT.0) GOTO 231
227          IF (TPB(J,K).LT.TB1.AND.TPB(J,K).GT.TOT/60.)
                >             TB1=TPB(J,K)
              ELSE
C
C     FINDING SMALLEST OFF TIME >TOT/60 FOR ON PUMPS
C
                DO 228 K=1,7
                  IF (TPE(J,K).LE.0) GOTO 231
228          IF (TPE(J,K).LT.TE1.AND.TPE(J,K).GT.TOT/60.)
                  >             TE1=TPE(J,K)
                END IF
                GOTO 231
              END IF
226      CONTINUE
C
```

```
C   DETERMINING SMALLEST OF OFF AND ON TIMES
C
231   IF (TB1.LT.TE1) THEN
        TPMP=TB1
    ELSE
        TPMP=TE1
    END IF
C
C   DETERMINING EARLIEST TIME STEP WHEN PUMP WILL CHANGE STATUS
C
        DTC(2)=TPMP*60.-TOT
        IF (DTC(2).LT.DTC(1)) DTC(1)=DTC(2)
229   CONTINUE
C
C.....CHECKING FOR TIME CONTROLLED PIPES
C
235   DO 239 J=1,ICPE
        DO 236 I=1,O(1)
            IF (IPI(I).EQ.IPEX(J)) THEN
                IF (CP(I).GE.1E10) THEN
C
C   FINDING SMALLEST OPEN TIME >TOT/60 FOR CLOSED PIPES (CP>1E10)
C
                    DO 237 K=1,7
                        IF (TEE(J,K).LE.0) GOTO 241
237   IF (TEE(J,K).LT.TE1.AND.TEE(J,K).GT.TOT/60.)
>       TE1=TEE(J,K)
                    ELSE
C
C   FINDING SMALLEST CLOSING TIME >TOT/60 FOR OPEN PIPES
C
                        DO 238 K=1,7
                            IF (TEB(J,K).LT.0) GOTO 241
238   IF (TEB(J,K).LT.TB1.AND.TEB(J,K).GT.TOT/60.)
>       TB1=TEB(J,K)
                            END IF
                                GOTO 241
                        END IF
236   CONTINUE
C
C   DETERMINING SMALLEST OF OFF AND ON TIMES
C
241   IF (TB1.LT.TE1) THEN
        TPMP=TB1
    ELSE
        TPMP=TE1
    END IF
C
C   DETERMINING EARLIEST TIME STEP WHEN PUMP WILL CHANGE STATUS
C
        DTC(2)=TPMP*60.-TOT
        IF (DTC(2).LT.DTC(1)) DTC(1)=DTC(2)
239   CONTINUE
        IF (IPRS.EQ.1) THEN
            DO 240 I1=1,O(1)
C
C.....SEARCHING FOR PRV'S , XL>999999
C
                IF (XL(I1).GT.999999.) THEN
                    IE = IEI(I1)
```



```

      HC = -EL(IE)+HW(I1)*2.308
      DTC(2)=DT*(HE(IE)-HC)/(HDE(IE)-HE(IE))
      IF (DTC(2).LT.DTC(1).AND.DTC(2).GT.0) DTC(1)=DTC(2)
C      PRINT*, 'PR DTC,HC,HDE ', IPI(I1), DTC(2), HC, HDE(IE)
      END IF
240  CONTINUE
      END IF
      END IF
      DO 100 I1=1, ITN
100  D0(I1)=ABS(D0(I1))
      DD=ABS(D0(1)-DD0(1))
      DO 110 I1=2, ITN
C
C      FINDING MAXIMUM ABSOLUTE CHANGE IN TANK FLOWS
C
      IF (ABS(D0(I1)-DD0(I1)).GT.DD) DD=ABS(D0(I1)-DD0(I1))
110  CONTINUE
      DO 120 I1=1, ITN
120  DD0(I1)=D0(I1)
      DM=D0(1)
      DO 130 I1=2, ITN
C
C      FINDING MAXIMUM TANK FLOW
C
      IF (D0(I1).GT.DM) DM=D0(I1)
130  CONTINUE
      IF (DD.GT..2*DM) THEN
        DT=2.5
        GOTO 140
      END IF
      IF (DD.NE.0..AND.DM.NE.0.) DT=INT(.1/(DD/DM)*DT)
C      PRINT*, 'DTG ', DT
      IF (DTC(1).LT.DT) DT=DTC(1)
      IF (DT.LT.2.5) DT=2.5
140  IF (TOT+DT.GT.ILC*60*TSLF-.1) DT=ILC*60*TSLF-TOT-.1
150  IF (NPUMP(1).NE.0) THEN
      DO 145 I = 1, 20
        IF (NPUMP(I).NE.0) THEN
          DO 143 I2 = 1, O(2)
143    IF (INO(I2).EQ.NOPU(I)) HDE(I2)=HE(I2)
          END IF
145  CONTINUE
        END IF
C
C      .....STORING HEAD OF THIS TIME STEP IN HDE
C
      DO 147 I1 = 1, O(1)
        IB = IBI(I1)
        IE = IEI(I1)
        IF (XL(I1).GT.999999.) THEN
          IPRS = 1
          HDE(IE)=HE(IE)
        END IF
        IF (DO(IB).GT.1E19.OR.DO(IE).GT.1E19) THEN
          HDE(IB)=HE(IB)
          HDE(IE)=HE(IE)
        END IF
147  CONTINUE
C      PRINT*, 'DTU ', DT
      RETURN
```

```

      END
C
C   LOWP RECORDS THE MIN PRESS. AND NODE WHERE MIN PRESS. OCCURED
C
      SUBROUTINE LOWP (O,DO,HE,EL,XLP,ILP,ILC,TOT,TSLF)
      INTEGER PNL,O
      REAL*8 HE
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION O(15),DO(PNL),HE(PNL),EL(PNL),XLP(57),ILP(57)
      XLOW=1E10
C
C   ENUMERATING ALL NODES TO FIND LOWEST HEAD
C
      DO 100 I1=1,O(2)
      IF (DO(I1).GT.9E9) GOTO 100
      IF (HE(I1)-ABS(EL(I1)).LT.XLOW) THEN
        XLOW=HE(I1)-ABS(EL(I1))
        IP=I1
      END IF
100   CONTINUE
C
C   RECORDING MIN PRESS AS XLP AND NODE NUMBER AS ILP
C
      IF (TOT.LT.1E-4) THEN
        XLP(1)=XLOW/2.308
        ILP(1)=IP
      ELSE
        ISUB=2*ILC
        IF (MOD(TOT-2.49999,60*TSLF).LT.1E-4) ISUB=2*ILC-1
        XLP(ISUB)=XLOW/2.308
        ILP(ISUB)=IP
      END IF
      RETURN
      END
C
C   PIPEX EXCLUDES PIPES BY ASSIGNING CP = 1E10 AND A = 1E-10
C
      SUBROUTINE PIPEX (O,A,CP,XL,HW,DI,IPEX,ICPE,IPI,TOT,TEB,TEE)
      REAL*8 A
      INTEGER PNL,O
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      DIMENSION O(15),A(IA),CP(PNL),XL(PNL),HW(PNL),DI(PNL)
      DIMENSION IPEX(16),IPI(PNL),TEB(16,7),TEE(16,7)
C
      IF (TOT.LT.-.0001) THEN
        DO 90 I1=1,O(1)
        IF (CP(I1).LT.0.OR.XL(I1).GT.999999..OR.XL(I1).LT.0) GOTO 90
        CP(I1)=4.72*ABS(XL(I1))/(ABS(HW(I1))**1.85*DI(I1)**4.87)
        A(I1)=1/CP(I1)
90     CONTINUE
      END IF
C
C   SEARCHING FOR PIPES TO BE EXCLUDED
C
      DO 110 K=1,ICPE
      DO 95 I1=1,O(1)
      IF (CP(I1).GT.0.AND.XL(I1).LT.999999..AND.XL(I1).GT.0.AND.
      >IPEX(K).EQ.IPI(I1)) THEN
        DO 93 K1=1,7
          IF (ABS(TEB(K,K1)-TOT/60.).LE.0.04.AND.CP(I1).LT.1E10) THEN

```

```
C
C   EXCLUDE PIPE
C
      CP(I1)=1E10
      A(I1)=1E-10
      WRITE(6,97)TOT/60.,IPEX(K),': CLOSED'
97   FORMAT(1X,F6.2,6X,'PIPE ',I4,A)
      GOTO 110
      ELSEIF (ABS(TEE(K,K1)-TOT/60.).LE.0.04.AND.CP(I1).GE.1E10)
>     THEN
C
C   INCLUDE PIPE
C
      CP(I1)=4.72*ABS(XL(I1))/(ABS(HW(I1))*1.85*DI(I1)**4.87)
      A(I1)=1/CP(I1)
      WRITE(6,97) TOT/60.,IPEX(K),': OPEN'
      GOTO 110
      END IF
93   CONTINUE
      GOTO 110
      END IF
95   CONTINUE
110  CONTINUE
120  RETURN
      END

C
C PUMPCO CONTROLS PUMPS BY PRESSURE AND/OR TIME
C
      SUBROUTINE PUMPCO(O,HE,IPI,ILC,INO,TOT,PUBE,PUST)
      REAL*8 A,HE
      INTEGER PNL,O
      CHARACTER PUBE*3
      PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
      COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
      COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1     INTANK(20),UPL(20),XLOL(20),ART(20),HETI(20),
2     NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3     IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4     TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
      DIMENSION O(15),HE(PNL),PUBE(20),IPI(PNL),INO(PNL)
> ,PUST(20,114)
      DO 100 I1 = 1,20
      IF (NPUMP(I1).EQ.0) GOTO 200
      DO 10 J1 = 1,O(1)
10   IF (IPI(J1).EQ.NPUMP(I1)) GOTO 20
20   DO 30 J2 = 1, O(2)
30   IF (INO(J2).EQ.NOPU(I1)) GOTO 40
40   IF (ILC.EQ.0.AND.PUBE(I1).EQ.'OFF') THEN
      PUST(I1,1) = -1
      A(J1) = 0
      XL(J1) = -1
      ELSEIF (ILC.EQ.0) THEN
      PUST(I1,1) = 1
      XL(J1) = 1
      END IF
      IF (TPE(I1,1).LT.0..AND.TPB(I1,1).LT.0.) GOTO 50

C
C   TIME CONTROL
C
      DO 45 K1 = 1,7
```

```
IF (TPE(I1,K1).LE.0) GOTO 50
IF (ABS(TPB(I1,K1)-TOT/60.).LE.0.04.AND.XL(J1).LT.0.) THEN
C
C   TURN PUMP ON
C
A(J1) = 1.85/SQRT(DI(J1)**2-2*CP(J1)*HW(J1))
XL(J1) = 1
PRINT 111,TOT/60.,NPUMP(I1),'ON '
111  FORMAT (1X,F6.2,6X,'PUMP',I4,' : TIME CONTROL,',A3)
ELSE IF (ABS(TPE(I1,K1)-TOT/60.).LE.0.04.AND.XL(J1).GT.0.) THEN
C
C   TURN PUMP OFF
C
A(J1) = 0
XL(J1) = -1
PRINT 111,TOT/60.,NPUMP(I1),'OFF '
END IF
45  CONTINUE
C
C   PRESSURE CONTROL
C
50  IF (HEON(I1).LT.1E-4.AND.HEOF(I1).LT.1E-4) GOTO 100
IF (HEON(I1).GE.HE(J2).AND.XL(J1).LT.0) THEN
C
C   TURN PUMP ON
C
A(J1) = 1.85/SQRT(DI(J1)**2-2*CP(J1)*HW(J1))
XL(J1) = 1
PRINT 113,TOT/60.,NPUMP(I1),'ON '
113  FORMAT (1X,F6.2,6X,'PUMP',I4,' : LEVEL CONTROL,',A3)
END IF
C
C   TURN PUMP OFF
C
IF (HEOF(I1).LT.HE(J2).AND.XL(J1).GT.0) THEN
A(J1) = 0
XL(J1) = -1
PRINT 113,TOT/60.,NPUMP(I1),'OFF '
END IF
100 CONTINUE
200 RETURN
END
C
C   TSOUT DISPLAYS PRESSURES, OR TANK WATER LEVELS FOR NODES AND
C   FLOWS IN LINKS
C
SUBROUTINE TSOUT (ILC,TOT,ITFL)
C
INTEGER PNL,O
REAL*8 A,S,G,HE
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
CHARACTER NOLI*4,UNIT*3,PUBE*3
COMMON /TIME/TMAX,TSLF,NSTEP,RATIO,XLOF(57,5),
1NTANK(20),UPL(20),XL(20),ART(20),HETI(20),
2NFIRE(5),ITF(5),IDF(5),FF(5),IPEX(16),ICPE,TEB(16,7),TEE(16,7),
3IPAT(PNL),LPAT(5),NPUMP(21),HEON(21),HEOF(21),
4TPB(21,7),TPE(21,7),NOPU(21),NUSR(11),IDTL,IPR
COMMON /NODES/ O(15),HE(PNL),DO(PNL),EL(PNL),S(PNL),G(PNL)
COMMON /NAME/ NOLI(11), UNIT(11),PUBE(20)
COMMON /PIPES/ A(IA),CP(PNL),DI(PNL),XL(PNL),HW(PNL)
```

```
COMMON /TOPOL/  IBE(PNL) , IEN(PNL) , IPI(PNL) , INO(PNL) , IBI(PNL)
>, IEI(PNL)
CHARACTER BNS*80,ENS*80,TOS*80,PRVS*80,STRNG*3,TM1*4,TM3*3
DIMENSION XLIST(10)
IF (NUSR(1).EQ.0) GOTO 300
IF (TOT.LT.1E-4) THEN
3   DO 3 I = 1,80
      BNS(I:I)=' '
      ENS(1:80)=BNS(1:80)
      TOS(1:80)=BNS(1:80)
      NPRN =0
C
C   COUNTING NUMBER OF ENTRIES
C
      DO 4 I = 1,10
4     IF (NUSR(I).NE.0) NPRN = NPRN + 1
        K =1
C
C   PLACING PRV LINKS AT BEGINNING OF LIST
C
      DO 6 I = 1,NPRN
        IF (UNIT(I).EQ.' ') THEN
          TM1=NOLI(I)
          TM2=NUSR(I)
          TM3=UNIT(I)
          NOLI(I)=NOLI(K)
          NUSR(I)=NUSR(K)
          UNIT(I)=UNIT(K)
          NOLI(K)=TM1
          NUSR(K)=TM2
          UNIT(K)=TM3
          K=K+1
        END IF
6     CONTINUE
C
C   DISPLAYING TABLE HEADING
C
      WRITE(6,5) (NOLI(I),I=1,NPRN)
5     FORMAT(///,6X,10(3X,A4))
      WRITE(6,7) (NUSR(I),I=1,NPRN)
7     FORMAT(' TIME',3X,I4,9(3X,I4))
      WRITE(6,9) (UNIT(I),I=1,NPRN)
9     FORMAT(' HOURS',10(4X,A3))
      DO 93 I = 1,NPRN
        K=7*I+4
        IF (NOLI(I).EQ.'LINK') THEN
          DO 91 J = 1, O(1)
            IF (IPI(J).EQ.NUSR(I)) THEN
              CALL STRING(IBE(J),STRNG)
              BNS(K:K+3) = STRNG(1:3)
              TOS(K:K+3) = 'TO '
              CALL STRING(IEN(J),STRNG)
              ENS(K:K+3) = STRNG(1:3)
              GOTO 93
            END IF
91          CONTINUE
          END IF
93        CONTINUE
      WRITE(6,95) BNS(1:K+3)
      WRITE(6,95) TOS(1:K+3)
```

```
        WRITE(6,95) ENS(1:K+3)
95     FORMAT(A)
        ENDIF
        IPRV =1
        K =1
        DO 100 I1 = 1,NPRN
C
C     CHECKING FOR NODES
C
        IF (NOLI(I1).EQ.'NODE') THEN
            DO 10 J1 = 1,O(2)
10     IF (INO(J1).EQ.NUSR(I1)) GOTO 11
11     XLIST(I1) = HE(J1)
C
C     CHECKING FOR PRESSURES TO BE PRINTED AT NODES
C
        IF (UNIT(I1).EQ.'PSI') XLIST(I1) = (HE(J1)-ABS(EL(J1)))/2.308
        ELSE
C
C     SEARCHING FOR LINKS
        DO 20 J1 = 1 , O(1)
20     IF (IPI(J1).EQ.NUSR(I1)) GOTO 21
21     IB = IBI(J1)
        IE = IEI(J1)
        H1 = HE(IB) - HE(IE)
C
C     CHECKING FOR PUMPS
C
        IF (CP(J1).LT.0) THEN
            IF (XL(J1).LT.0) THEN
                XLIST(I1) = 0
            ELSE
                Y1 = CP(J1)
                Y2 = DI (J1)
                XLIST(I1) = ((-Y2-1.85/A(J1))/2/Y1)*448.831
            END IF
C
C     CHECKING FOR PRV'S
C
        ELSEIF (XL(J1).GE.999999.) THEN
            IF (A(J1).GE.9999) THEN
                PRVS(K:K+6)=' OPEN  '
            ELSEIF (ABS(HE(IE)+EL(IE)-HW(J1)*2.308).LT.0.1) THEN
                PRVS(K:K+6)=' ACTIVE'
            ELSE
                PRVS(K:K+6)=' CLOSED'
            ENDIF
            IPRV = IPRV +1
            K=K+7
        ELSE
            IF (XL(J1).LT.0.AND.H1.LT.0) THEN
                XLIST(I1) = 0
            ELSE
                XLIST(I1) = (ABS(H1)/CP(J1))**.54*448.831*SIGN(1.,H1)
            END IF
        END IF
        END IF
        END IF
100    CONTINUE
        IT=ILC
        IF (TOT.LT.1E-4) IT=0
```

```
200 IF (IDTL.EQ.1) THEN
    IF (IPRV.GT.1) THEN
        WRITE(6,103) TOT/60.,PRVS(1:K-1),(XLIST(J1),J1=IPRV,NPRN)
    ELSE
        WRITE(6,101) TOT/60., (XLIST(J1),J1 = 1,NPRN)
    END IF
ELSE
    IF (IPRV.GT.1) THEN
        WRITE(6,103) IT*TSLF,PRVS(1:K-1),(XLIST(J1),J1=IPRV,NPRN)
    ELSE
        WRITE(6,101) IT*TSLF, (XLIST(J1),J1 = 1,NPRN)
    END IF
ENDIF
101 FORMAT(1X,F6.2,10F7.0)
103 FORMAT(1X,F6.2,A,9F7.0)
    IF (ITFL.EQ.0.AND.TOT.GT.1E-4) PRINT 105
105 FORMAT (60X,'ALL TANKS ARE EMPTY. ')
300 RETURN
END
```

C

C...STRING IS CALLED BY TSOUT TO PLACE BEGINNING AND ENDING NODE
C NUMBERS OF PRINTED LINKS IN A CHARACTER ARRAY

C

```
SUBROUTINE STRING(NDE,STRNG)
CHARACTER STRNG*3,NUMB*10
NUMB = '0123456789'
STRNG = ' '
IF (NDE.LT.10) THEN
    STRNG(1:1)=NUMB(NDE+1:NDE+1)
ELSEIF (NDE.GE.10.AND.NDE.LT.100) THEN
    I2 = MOD(NDE,10)+1
    I1 = INT(NDE/10)+1
    STRNG(1:1) = NUMB(I1:I1)
    STRNG(2:2) = NUMB(I2:I2)
ELSE
    I3 = MOD(NDE,10)+1
    I2 = MOD(INT(NDE/10),10)+1
    I1 = INT(NDE/100)+1
    STRNG(1:1) = NUMB(I1:I1)
    STRNG(2:2) = NUMB(I2:I2)
    STRNG(3:3) = NUMB(I3:I3)
END IF
RETURN
END
```

C

C...LOAD MULTIPLIES LOADING FACTORS TO WATER USES AT WATER USE NODES
C AND CHECKS FOR FIRE FLOWS AT EACH USER DEFINED TIME STEP

C

```
SUBROUTINE LOAD (NFIRE,ITF,O,INO,FF,DO,IPAT,XLOF,IDF,ILC,DOTMP
>,RATIO,TOT)
INTEGER PNL,O
PARAMETER (PNL=800,LNL=750,MNL=40,IA=7500)
```

C

```
DIMENSION NFIRE(5),ITF(5),O(15),INO(PNL),FF(5),DO(PNL),IPAT(PNL)
DIMENSION XLOF(57,5),IDF(5),DOTMP(PNL)
DO 40 I1=1,O(2)
```

C

C...SKIPPING TANK NODES

C

```
IF (DO(I1).GT.9E9) GOTO 40
```

```
      J1=IPAT(I1)
      DO 20 K=1,5
C
C...SEARCHING FOR FIRE FLOW NODES
C
      IF (INO(I1).EQ.NFIRE(K)) GOTO 30
20    CONTINUE
      K=0
C
C...ASSIGNING LOADS AND FIRE FLOWS FOR TIME STEP 1
C
30    IF (ILC.EQ.1) THEN
      IF (ABS(DOTMP(I1)).GT.1E-4) DO(I1)=DOTMP(I1)*RATIO*XLOF(1,J1)
      IF (K.EQ.0) GOTO 40
      IF (ITF(K).EQ.0) THEN
        DO(I1)=DO(I1)+FF(K)
        WRITE(6,33) TOT/60,NFIRE(K),'ON '
33      FORMAT(1X,F6.2,6X,'FIRE AT NODE',I4,1X,A3)
      END IF
    ELSE
C
C...ASSIGNING LOADS AND FIRE FLOWS FOR OTHER TIME STEPS
C
      IF (K.EQ.0) GOTO 35
      IF (ITF(K).LT.ILC-1.AND.IDF(K).GE.ILC-1) THEN
        DO(I1)=DO(I1)-FF(K)
        WRITE(6,33) TOT/60.,NFIRE(K),'OFF '
      END IF
35    IF (ABS(DOTMP(I1)).GT.1E-4) DO(I1)=DOTMP(I1)*RATIO*XLOF(ILC,J1)
      IF (K.EQ.0) GOTO 40
      IF (ITF(K).LE.ILC-1.AND.IDF(K).GT.ILC-1) THEN
        DO(I1)=DO(I1)+FF(K)
        WRITE(6,33) TOT/60.,NFIRE(K),'ON '
      END IF
    END IF
40    CONTINUE
      RETURN
      END
C *****
```


