```
C
C
C
C     TRANSIENT-STATE SIMULATION PROGRAM FOR MINE VENTILATION
C                  UNDER THE INFLUENCE OF FIRES
C
C                  VERSION OF JUNE 1995 V 2.20 contract version 1.30
C                             (PART 1)
C
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      INCLUDE 'CMMN1.DAT'
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C     PARAMETER  (NMX=400, NMY=300, NMZ=150, IMX=15, IMY=10, IMZ=10,
C    .              LMX=10000, NXX=NMX*3)
C
C     COMMON  /CONTRL/ NB,NJ,NFNUM,NVPN,NETW,NTEMP,MADJ,ITN,DR,TR,TINC,
C    .                  SPAN,IOUT,TOUT,CONCT
C
C     NB: NUMBER OF AIRWAYS IN NETWORK.
C     NJ: NUMBER OF JUNCTIONS IN NETWORK.
C     NFNUM: NUMBER OF FAN CHARACTERISTICS.
C     NVPN: MARKER FOR PRESENCE OF JUNCTION CARDS, 0: PRESENT.
C     NETW: MARKER, 1: ONLY THE NETWORK PART WILL BE PERFORMED.
C     NTEMP: MARKER, 1: ONLY UP TO THE TEMP. PART WILL BE PERFORMED.
C     MADJ: MAX. NO. OF ITERATION IN TEMP. PART.
C     ITN: MAX. NO. OF ITERATION IN NON-STEADY STATE SIMULATION.
C     DR: REFERENCE DENSITY OF AIR, LBM/FT3.
C     TR: REFERENCE TEMPERATURE OF AIR, DEG. F.
C     TINC: TIME INCREMENT IN SIMULATION, SEC..
C     SPAN: TIME SPAN OF SIMULATION, MIN..
C     IOUT: OUTPUT. 0: BRIEF; 1: NORMAL; -1: DETAIL; -2: MORE DETAILED.
C     TOUT: TIME INTERVAL FOR OUTPUT, MIN.
C     CONCT: MARKER. 1: SIMULATION FOR TRANSITE CONCENTRATION SIMULA-
C                       TION ONLY
C
C     COMMON  /CTRL/ NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT
C
C     NAV: MARKER FOR PRESENCE OF AVE. VALUE CARDS.
C     MAXJ: HIGHEST JUNCTION NUMBER.
C     INFLOW: NUMBER OF CONTAMINATION CARDS.
C     CRITSM: ACCURACY OF FUME CALCULATION.
C     CRITGS: ACCURACY OF METHAME CALCULATION.
C     CRITHT: ACCURACY OF TEMPERATURE CALCULATION.
C
C     COMMON  /NTWK/ NO(NMX),JS(NMX),JF(NMX),NWTYP(NMX),R(NMX),Q(NMX),
C    .               P(NMX),KF(NMX),LA(NMX),A(NMX),O(NMX),RSTD(NMX),
C    .               DZRD(NMX),FRNVP(NMX),NREV(NMX),RDCH4(NMX),
C    .               RDPROP(NMX),TRD(NMX),TJS(NMX),RDOP(NMX),RCH4(NMX),
C    .               FFRNVP(NMX),RA(NMX),NNREV(NMX)
C
C     NO(NMX): AIRWAY NUMBER.
C     JS(NMX): JUNCTION NUMBER OF AIRWAY BEGINNING.
C     JF(NMX): JUNCTION NUMBER OF AIRWAY END.
C     NWTYP(NMX): AIRWAY TYPE.
C     R(NMX): RESISTANCE OF AIRWAY 1.E-10 IN. W.G./(CFM)**2.
C     Q(NMX): BALANCED AIRFLOW RATE, ASSUMING EQUILIBRIA, CFM
C     P(NMX): CALCULATED PRESSURE LOSS IN IN. W.G..
C     KF(NMX): FRICTION FACTOR.
C     LA(NMX): AIRWAY LENGTH IN FT.
C     A(NMX): AIRWAY CROSS SECTIONAL AREA IN FT**2.
C     O(NMX): AIRWAY PERIMETER IN FT.
C     RSTD(NMX): RESISTANCE FACTOR BASED ON TR.
C     DZRD(NMX): ELEVATION CHANGE IN AIRWAY IN FT.
C     FRNVP(NMX): PRODUCT TMRD*DZRD.
C     NREV(NMX): NUMBER OF AIRFLOW REVERSALS IN AN INTERVAL.
C     RDCH4(NMX): METHANE CONCENTRATION AT ROADWAY ENDS.
C     RDPROP(NMX): CONTAMINANT CONCENTRATION AT ROADWAY ENDS.
C     TRD(NMX): TEMPERATURE AT ROADWAY ENDS.
C     TJS(NMX): TEMPERATURE AT ROADWAY BEGINNING.
C     RDOP(NMX): BACKUP ARRAY FOR RDPROP.
C     RCH4(NMX): BACKUP ARRAY FOR RDCH4.
C     FFRNVP(NMX): DZRD/TMRD.
C     RA(NMX): BACKUP ARRAY FOR R.
C     NNREV(NMX): NUMBER OF AIRFLOW REVERSALS SINCE TIME ZERO..
C
C     COMMON  /FAN/ NOF(IMX),NFREG(IMX),RGRAD(IMX),NFCW(IMX),MPTS(IMX),
C    .              QF(IMX,IMY),PF(IMX,IMY),NSKP(IMX),NEGQ(IMX),JSB(NMX),
C    .              NSWT(IMX)
C     NOF(IMX): AIRWAY NUMBER OF FAN.
C     NFREG(IMX): LIST OF FANS WITH CHARACTERISTICS.
C     RGRAD(IMX): SLOPE OF FAN CHARACTERISTIC.
C     NFCW(IMX): LIST OF FANS WHOSE CHAR. EXCEEDED.
C     MPTS(IMX): NUMBER OF POINTS DEFINING FAN CHAR.
C     QF(IMX,IMY): AIRFLOW AT GIVEN POINT OF FAN CHAR.
C     PF(IMX,IMY): PRESSURE AT GIVEN POINT OF FAN CHAR.
C     NSKP(IMX): MARKER INDICATING PERFORMANCE OF SPLINE.
C     NEGQ(IMX): MARKER INDICATING AIRFLOW REVERSAL IN FAN BRANCH.
C     JSB(NMX): BACKUP ARRAY FOR JS.
C     NSWT(NMX): SWTCH TO SELECTED FAN CURVE FITTING METHOD
C                1 LEAST SQUARE 2.SPLINE 3.AUTOMATIC 4.AUTO+MANUAL
C
C      COMMON /FC/     FC1(IMY),FC2(IMY),FC3(IMY),DK(IMY),FK(IMY),
C                     FKQ(IMX,IMY,4)
C
C
C     COMMON  /MESH/ MNO,MEND(NMY),MSL(LMX),FNVP(NMY),RQ(NMX),INU(NMX),
C    .           KNO(NXX),KJS(NXX),KJF(NXX)
C
C     MNO: NUMBER OF MESHES IN NETWORK.
C     MEND(NMY): LIST OF MESH ENDS IN MSL-LIST.
C     MSL(LMX): LIST OF ALL INDEPENDENT MESHES.
C     FNVP(NMY): LIST OF NVP IN MESHES.
C     RQ(NMX): AUXILIARY LIST OF R*Q.
C     INU(NMX): AUXILIARY LIST FOR FORMING BASE SYST.
C     KNO(NXX): LIST OF AIRWAYS IN BASE SYSTEM.
C     KJS(NXX): LIST OF JUNCTIONS IN BASE SYSTEM.
C     KJF(NXX): LIST OF JUNCTIONS IN BASE SYSTEM.
C
C     COMMON  /JUNCT/ JNO(NMY),T(NMY),Z(NMY),CH4C(NMY),JNOL(NMY),
C    .                PROP(NMY),PRCH4(NMY),JLR(NMY)
C
C     JNO(NMY): JUNCTION NUMBER.
C     T(NMY): TEMPERATURE OF AIR IN JUNCTIONS.
C     Z(NMY): ELEVATION OF JUNCTIONS.
C     CH4C(NMY): METHANE CONCENTRATION IN JUNCTION.
C     JNOL(NMY): LIST OF JUNCTION NUMBERS IN INCREASING ORDER.
C     PROP(NMY): CONTAMINANT CONCENTRATION IN JUNCTION.
C     PRCH4(NMY): METHANE CONCENTRATION IN JUNCTION.
C     JLR(NMY): LIST RELATING JNO- AND JNOL-LISTS.
C
C     COMMON  /TEMP/ TROCK(NMX),TMRD(NMX),HA(NMX),HK(NMX)
C
C     TROCK(NMX): AVERAGE ROCK TEMPERATURE IN AIRWAY.
C     TMRD(NMX): MEAN TEMPERATURE OF AIR IN AIRWAY.
C     HA(NMX): THERMAL DIFFUSIVITY OF ROCK.
C     HK(NMX): THERMAL CONDUCTIVITY OF ROCK.
C
C     COMMON  /FUME/ NCENT(NMX),CH4V(NMX),CH4PA(NMX),CONT(IMX),CONC(IMX)
C    .                ,HEAT(IMX),O2MIN(IMX),SMPO2(IMX),HTPO2(IMX),
C    .                TFSI(IMX),O2BEH(IMX),TPR(IMX),HTAD(IMX),QCENT(IMX)
```

```
C
C      NCENT(NMX): NUMBER OF AIRWAYS INTO WHICH FUME ENTERS.
C      CH4V(NMX): METHANE EMISION RATE IN AIRWAYS.
C      CH4PA(NMX): METHANE EMISION RATE PER UNIT SURFACE AREA.
C      CONT(IMX): VOLUME FLOW RATE OF CONTAMINATED GAS INFLOW.
C      CONC(IMX): CONCENTRATION OF CONTAMINANT IN GAS INFLOW.
C      HEAT(IMX): HEAT ENTERING AIRWAY.
C      O2MIN(IMX): OXYGEN CONCENTRATION OF FUMES LEAVING FIRE ZONE
C      SMPO2(IMX): CONTAMINANT PRODUCTION PER CUBIC FT OF OXYGEN.
C      HTPO2(IMX): HEAT PRODUCTION PER CUBIC FT OF OXYGEN DELIVERY
C      TFSI(IMX): AIR TEMPERATURE BEHIND HEAT SOURCE.
C      O2BEH(IMX): OXYGEN CONTENT BEHIND FIRE SOURCES.
C      TPR(IMX): TRANSITION TIME OF FIRE IN MIN.
C      HTAD(IMX): REDUCED HEAT INPUT DUE TO HIGH AIR TEMPERATURE.
C      QCENT(IMX): "STD. Q' AT WHCIH FIRE TAKES THE INPUT PARAMETERS.
C
C      COMMON  /SCHEME/ NGIN(NMX),MMIN(NMX),NGOUT(NMX),LOUT(NMX)
C
C      NGIN(NMX): LIST OF AIRWAYS ENTERING JUNCTION.
C      MMIN(NMX): LIST OF LAST AIRWAY PER JUNCTION IN NGIN-LIST.
C      NGOUT(NMX): LIST OF AIRWAYS LEAVING JUNCTION.
C      LOUT(NMX): LIST OF LAST AIRWAY PER JUNCTION IN NGOUT-LIST.
C
C      COMMON  /EST/ MEMREC(NMX),NOREC(NMX),ESTPR(NMX),ESTCH4(NMX),
C    .               ESTTR(NMX)
C
C      MEMREC(NMX): TEMPORARY LIST OF AIRWAYS CARRYING RECIRC. AIR.
C      NOREC(NMX): PERMANENT LIST OF AIRWAYS.
C      ESTPR(NMX): ESTIMATED CONCENTRATION FOR RECIRCULATED AIR.
C      ESTCH4(NMX): ESTIMATED METHANE CONCENTRATION.
C      ESTTR(NMX): ESTIMATED TEMPERATURE.
C
C      COMMON  /FACTOR/ XNEW(NMX),DCOAGE(NMX)
C
C      XNEW(NMX): EXPONENT FOR TEMPERATURE CALCULATION.
C      DCOAGE(NMX): COEFFICIENT OF AGE.
C
C      COMMON  /TTJ/ TIME,TSTART,JSTART,TLEFT(NMX)
C
C      TIME: TIME ASSUMING QUASI-EQUILIBRIA.
C      TSTART: TEMPERATURE AT THE START JUNCTION.
C      JSTART: START JUNCTION (ATMOSPHERE).
C      TLEFT(NMX): TIME INCREMENT FOR MAKING UP A NORMAL INTERVAL.
C
C      COMMON  /SUM/ SUMQ(NMY),SUMC(NMY),SUMM(NMY),SMHEAT(NMY)
C
C      SUMQ(NMY): TOTAL AIRFLOW RATES ENTERING JUNCTION.
C      SUMC(NMY): TOTAL CONTAMINANT FLOW RATES ENTERING JUNCTION.
C      SUMM(NMY): TOTAL METHANE FLOW RATES ENTERING JUNCTION.
C      SMHEAT(NMY): TOTAL ENTHALPY/REFRENCE DENSITY ENTERING JUNCTI
C
C      COMMON  /AUX/ PROPS(NMX),PRCH4S(NMX),QQ(NMX),QTP(NMX),TTJS(NMX),
C    .               TTRD(NMX),TAS(NMX),BI(NMX)
C
C      PROPS(NMX): BACKUP ARRAY FOR PROP.
C      PRCH4S(NMX): BACKUP ARRAY FOR PRCH4.
C      QQ(NMX): DAMPED AND TIME-AVERAGED AIRFLOW RATE.
C      QTP(NMX): BACKUP ARRAY FOR Q, FIXED IN EACH INTERVAL.
C      TTJS(NMX): BACKUP ARRAY FOR TJS.
C      TTRD(NMX): BACKUP ARRAY FOR TRD.
C      TAS(NMX): BACKUP ARRAY OF T.
C      BI(NMX): BIOT NUMBER.
C
C      COMMON  /WRN/ WRNPR,WRNGS,WRNSM,WRNHT,IOMIT(IMX),ROMIT(IMX,3),IAC
C
C      WRNPR: PRESSURE DROP WARNING CRITERIA.
C      WRNGS: METHANE CONCENTRATION WARNING CRITERIA.
C      WRNSM: FUME CONCENTRATION WARNING CRITERIA.
C      WRNHT: HIGH TEMPERATURE WARNING CRITERIA.
C      IOMIT(IMX): AIRWAYS FOR WHICH DETAILED DATA RECORDS WERE OMITTED.
C      ROMIT(IMX,3): DATA AT STARTING J OF DATA-OMITTED AIRWAY.
C      IAC: COUNTER OF IOMIT AND ROMIT.
C
C      COMMON  /TRANS/ DELTAT,TACC,SUMFNV
C
C      DELTAT: SPAN OF TIME INCREMENT.
C      TACC: CUMULATED TIME AFTER EVENT IN SEC.
C      SUMFNV: SUM OF NVP CORRECTION IN SYSTEM.
C
C      COMMON  /RCD/ IBTN(NMX),DPPA(NMZ,10,5),NSAC(NMX,2),TMRDA(NMX),
C    .              DELTAQ(NMX),DPPB(NMZ,10,5),NSACB(NMX,2),GBTN(10,5),
C    .              JCH(NMY),TAUXA(NMZ,10),TAUXB(NMZ,10),TAUXC(NMZ),
C    .              TAUXD(NMZ),FMASS(NMZ)
C
C      IBTN(NMX): ARRAY HOLDING ADDRESS OF RECORD FOR AIRWAY (N
C      DPPA(NMZ,10,5): MASTER DATA STORAGE ARRAY IN DYNAMIC SIMULATI
C      NSAC(NMX,2): DATA RECORD STATUS OF AIRWAY (NMX).
C      TMRDA(NMX): BACKUP ARRAY FOR TMRD.
C      DELTAQ(NMX): VARIATION OF AIRFLOW IN AIRWAYS.
C      DPPB(NMZ,10,5): BACKUP ARRAY FOR DPPA.
C      NSACB(NMX,2): BACKUP ARRAY FOR NSAC.
C      GBTN(10,5): AUXILIARY ARRAY FOR DATA TRANSFER.
C      JCH(NMY): MARKER INDICATING JUNCTION COND. CHANGE.
C      TAUXA(NMZ,10): AIR TEMP. AT REAR ENDING OF SEGMENTS.
C      TAUXB(NMZ,10): BACKUP ARRAY FOR TAUXA.
C      TAUXC(NMZ): AIR TEMP. BEYOND THE FIRST SEGMENT.
C      TAUXD(NMZ): BACKUP ARRAY OF TAUXC.
C      FMASS(NMZ): MASS OF AIR BEYOND THE FIRST SEGMENT.
C
C      COMMON  /CCJ/ JDP, JDPP(IMX)
C
C      JDP: NUMBER OF JUNCTIONS INITIATING DATA RECORDS.
C      JDPP(IMX): JUNCTIONS WHICH INITIATE DATA RECORDS.
C
C      COMMON  /TREND/ QRCD(NMX,10),TMRCD(NMX,10),HRM(IMX)
C
C      QRCD(NMX,10): AIRFLOW RATES DURING ITERATIONS.
C      TMRCD(NMX,10): MEAN AIR TEMP. IN AIRWAYS DURING ITERATION.
C      HRM(IMX): VARIATION OF NVP PER MESH DURING ITERATION.
C
C      COMMON /CHRC/ ITITLE(50),NCOMT1,NCOMTS,NCOMT2,LLINE
C      COMMON /ERMSG/ ROW(80),TITLE(50,20)
C      COMMON /ATJUNT/JAN,JAJ,JA1(10),JATP(10),JSTAR(10),TSTAR(10),
C    .                MSTAR(10),ISTAR(10)
C
C      JAN:  TOTAL NUMBER OF ATMOSPHERE JUNCTIONS
C      JAJ:  NUMBER OF ATMOSPHERE JUNCTIONS EXCEPT STARTING JUNCTION
C      JA1(10):  JUNCTION ID NUMBER (IN ATMOSPHERE)
C      JATP(10): JUNCTION TYPE (IN ATMOSPHERE)
C      JSTAR(10): JUNCTIONC CONNECTED WITH ENTRY AIRWAYS EXCEPT STARING
C                 JUNCTION
C      TSTAR(10): TEMPERATURE IN THE JUNCTION AT ARRAY (JSTAR)
C      MSTAR(10): JUNCTION NUMBER CORRESPONDING TO ARRAY (JNOL)
C      ISTAR(10): JUNCTION NUMBER CORRESPONDING TO ARRAY (JNO)
C
C      COMMON /CONCN/  TTT(NMY),TJSS(NMX),TRDD(NMX),QQQ(NMX)
C      TTT: INTERMEDIATE ARRAY FOR TEMPERATURE WHEN SIMULATE FOR TRANSIENT
C           CONCENTRATION SIMULATION ONLY (CONCT=1).
C      TJSS:INTERMEDIATE ARRAY OFR TTJS WHEN CONCT=1.
C      TRDD: INTERMEDIATE ARRAY FOR TTRD WHEN CONCT=1.
```

```
C      QQQ:  INTERMEDIATE ARRAY FOR QQQ WHEN CONCT=1.
C
C      COMMON /FAN1/COF(IMX,IMX),ND(IMX),N1A,NSWT(IMX),NCOF(IMX),
C     .         INIFAN,NPLOT
C      COF: THE COEFFIENT OF FAN EQUATION FITTED BY LEAST SQUARES.
C      ND: THE DEGREE OF FAN EQUATION FITTED BY LEAST SQUARES.
C      N1A: THE NUMBER OF COEFFICIENTS OF FAN EQUATION.
C      NSWT: TO SELECT FAN CURVE FITTING METHOD
C                 1 LEAST SQUARE 2.SPLINE 3.AUTOMATIC 4.AUTO+MANUAL
C      NCOF: ARRAY FOR THE NUMBER OF COEFFICIENTS OF FAN EQUATION
C      INIFAN: THE FLAG SHOWS THE LARGE STATE CHANGE IN SYSTEM AND THE
C              MESHES REFORM DESIRED.
C      NPLOT: TREATMENT OF THE BOUNDARY RANGE OF FAN CURVE
C             1: EXTEND FAN CURVE BY FOLLOWING GRADIENTS OF TWO ENDS.
C             2: EXTEND FAN CURVE AS ABOVE AT LEFT BOUNDARY REGION AND
C                SEND GRADIENT OF RIGHT BOUNDARY REGION TO ZERO.
C             3. SEND GRADIENT OF BOTH SIDES OF BOUNDARY RANGE TO ZERO.
C
C
C      REAL LA
C      CHARACTER*4 TITLE
C      CHARACTER*1 ROW
C
C
C      MARKX: 0: THE PROG. HAS NOT REACHED THE DYNAMIC SIMULATION
C             PART YET.  1: THE DYNAMIC PART HAS BEEN REACHED.
C      MARKY: 0: BEFORE FINAL QUASI-EQUILIBRIUM SIMULATION.
C             1: IN THE FINAL QUASI-EQUIL. STAGE, READY FOR TERMINATION.
C      NSTOP: MARKER FOR INPUT DATA ERROR.
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
      CHARACTER*50 MD1,MD2,OPTION
C-----------------------------------------------------------------------
      OPEN (10,FILE='MFCTL1',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (10) NETWW
      IF (NETWW.EQ.1) GO TO 350
      OPTION='                                                  '
      MD1='                                                  '
      MD2='                                                  '
      OPEN (21,FILE='MFCTL2',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (21) OPTION
      READ (21) MD1,TACC
      IF (OPTION.EQ.'DONE') GO TO 350
      NSTOP=0
      MARKX=0
      IF (OPTION.EQ.'CONTINUE') THEN
         MARKY=1
         MD2='MFCTL5'
         WRITE (6,740)
      ELSE
         WRITE (6,910)
         PAUSE
         WRITE (6,900)
         PAUSE
         MARKY=0
         MD2='MFCTL6'
         WRITE (6,730)
         WRITE (6,680)
         READ (5,'(A)') MD1
      ENDIF
      IF (MARKY.EQ.0) GO TO 6
      OPEN (19,FILE='MFCTL7',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (19) NFNUM,NPLOT
      DO 5 I=1,NFNUM
      READ (19) NSWT(I)
5     CONTINUE
6     OPEN (9,FILE=MD1,STATUS='UNKNOWN')
      OPEN (8,FILE=MD2,STATUS='UNKNOWN')
      IF (MARKY.EQ.1) WRITE (8,760)
      MADJC=0
      LLINE=0
      MAXNO=0
      CALL INPUT (1,NSTOP,MARKY,MAXNO,KV)
      IF (NSTOP.GT.0) GO TO 300
      CALL CHECK1 (NSTOP,MAXNO,KV)
      IF (NSTOP.GT.0) GO TO 300
      IF (NETW.NE.1) THEN
         CALL INPUT (2,NSTOP,MARKY,MAXNO,KV)
         IF (NSTOP.GT.0) GO TO 300
      ENDIF
      IF (MARKY.EQ.1) GO TO 100
      NSTOP1=0
      CALL CHSFIT (NSTOP1)
100   ITCT=0
      IF (IOUT.LE.(-1).AND.MARKY.LE.0)
     .   CALL OUTPUT (1,MARKX,MARKY,MADJC)
      CALL ARR (MARKX)
      CALL BASE (NSTOP)
      IF (NSTOP.GE.1) GO TO 349
      CALL MSLIST (NSTOP)
      IF (NSTOP.GE.1) GO TO 349
      CALL MBLNC
      CALL NVP1
      CALL ITR (MARKX,NSFLOW,0,MADJC,ITCT)
      CALL RGLT (MARKY)
      IF (MARKY.EQ.0) WRITE (6,670)
      IF (NETW.EQ.1) THEN
         NETWW=1
         REWIND 10
         WRITE (10) NETWW
         WRITE (8,510)
         CALL OUTPUT (2,MARKX,MARKY,MADJC)
         KTL=0
         DO 9 I=1,NB
            IF (Q(I).LT.0.0) THEN
               IF (KTL.EQ.0) WRITE (8,470)
               WRITE (8,480) NO(I),IABS(JF(I)),IABS(JS(I))
               KTL=1
            ENDIF
9        CONTINUE
         IF (KTL.EQ.0) WRITE (8,482)
         GO TO 349
      ENDIF
C
C
      IF (MARKY.LE.0) THEN
         WRITE (6,855)
      ELSE
         WRITE (6,860)
      ENDIF
      WRITE (6,865)
      NSFLOW=0
      ITCT=0
      IF (MARKY.GT.0) THEN
         DO 91 I=1,NB
            TMRD(I)=TR
91       CONTINUE
      ENDIF
      CALL INPUT (3,NSTOP,MARKY,MAXNO,KV)
      IF (NSTOP.GT.0) GO TO 300
      CALL CCDATA (MARKY,NSTOP)
```

```
      IF (NSTOP.GT.0) GO TO 300
      IF (MARKY.EQ.1) THEN
         CALL CDCH (NSTOP)
         IF (NSTOP.GT.0) GO TO 300
      ENDIF
      CALL CH4EVA
      IF (IOUT.LE.(-1).AND.MARKY.GT.0)
     .   CALL OUTPUT (1,MARKX,MARKY,MADJC)
      IF (IOUT.LE.(-1).AND.MARKY.LE.0)
     .   CALL OUTPUT (2,MARKX,MARKY,MADJC)
      CALL OUTPUT (3,MARKX,MARKY,MADJC)
      JAJ=0
      IF (JAN.LE.1) GO TO 40
         DO 10 J=1,JAN
            IF (JATP(J).NE.0) THEN
               JAJ=JAJ+1
               JSTAR(JAJ)=JA1(J)
            ENDIF
10       CONTINUE
      DO 30 J=1,NJ
         DO 30 K=1,JAJ
            IF (JSTAR(K).EQ.JNO(J)) THEN
               TSTAR(K)=T(J)
            ENDIF
30    CONTINUE
40    NSFLOW=1
      CALL FWCT (NSFLOW,0)
C
C     DETERMINATION OF CONDITIONS IN THE STARTING JUNCTION.
C
      DO 50 I=1,NJ
         PROP(I)=0.0
         PRCH4(I)=0.0
         IF (JNOL(I).EQ.JSTART) THEN
            MSTART=I
            ISTART=JLR(I)
            T(ISTART)=TSTART
         ENDIF
50    CONTINUE
70    MRC=0
      L=ISTART
      NSTART=MSTART
90    CALL CDENDS (MARKY,NSTART,L,NSTOP)
      IF (NSTOP.GE.1) GO TO 349
      JUMPBK=0
C      CALL CDJUNC (NSTART,JUMPBK,L,MOMIT,0)
      CALL CDJUNC (NSTART,JUMPBK,L,0)
      IF (JUMPBK.EQ.1) GO TO 90
      CALL RECIRC (MRC,JUMPBK,NSTART,L)
      IF (JUMPBK.EQ.1) GO TO 90
      LLST=0
      CALL PREP (MRC,LLST)
      IF (MRC.NE.0) THEN
C
C     ITCT: COUNTER OF ITERATIONS.
C
         ITCT=ITCT+1
         IF (LLST.NE.0) THEN
            IF (ITCT.LE.100) THEN
               L=ISTART
               NSTART=MSTART
               GO TO 90
            ENDIF
            WRITE (8,500) MADJC
         ENDIF
      ENDIF
      CALL TEVAL
C
C     NSFLOW: 0: DATA PREPARATION FOR NETWORK BALANCING IN TEMP. PART
C             HAS NOT BEEN COMPLETED. AIRFLOW REVERSAL EXISTS IN THE
C             SYSTEM.  1: DATA PREPARATION COMLETED.
C     SUMFNV: SUM OF ABSOLUTE DIFFERENCE OF NVP IN SUCCESSIVE INTERVALS.
C     MADJ: USER-SPECIFIED MAX. NUMBER OF ITERATIONS IN ONE PROG. RUN.
C     MADJC: COUNTER OF ITERATIONS IN ONE PROG. RUN.
C
C     ********************************
       IF (MADJC.LT.MADJ) THEN
         MADJC=MADJC+1
         ITCT=0
         DO 120 I=1,NB
            IF (JF(I).LT.0) JF(I)=-JF(I)
120      CONTINUE
        IF (NSFLOW.LE.0) THEN
            CALL ARR (MARKX)
            CALL BASE (NSTOP)
            CALL MSLIST (NSTOP)
            CALL MBLNC
         ENDIF
         CALL NVP2
C     **********************************
C      DO 120 I=1,NB
C         IF (JF(I).LT.0) JF(I)=-JF(I)
C120   CONTINUE
C     **************************
C      IF (NSFLOW.LE.0) THEN
C         CALL ARR (MARKX)
C         CALL BASE (NSTOP)
C         CALL MSLIST (NSTOP)
C         CALL MBLNC
C      ENDIF
C     ***************************
C      CALL NVP2
C      IF (MADJC.LT.MADJ) THEN
C         MADJC=MADJC+1
C         ITCT=0
         CALL ITR (MARKX,NSFLOW,1,MADJC,ITCT)
         WRITE (6,870) MADJC,SUMFNV/MNO
         IF (MADJC.LE.2.AND.MARKY.EQ.1) GO TO 124
         IF (MADJC.EQ.1) GO TO 124
         IF ((SUMFNV/MNO).LE.2.E-4) GO TO 130
124      IF (MADJC.GE.(MADJ-9)) THEN
            J=MADJC-MADJ+10
            HRM(J)=SUMFNV/MNO
            DO 125 I=1,NB
               QRCD(I,J)=ABS(Q(I))
               IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .            QRCD(I,J)=-ABS(Q(I))
               IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .            QRCD(I,J)=-ABS(Q(I))
               TMRCD(I,J)=TMRD(I)
125         CONTINUE
         ENDIF
         ITCT=0
         DO 127 I=1,NB
            IF (Q(I).LT.0.0) GO TO 40
127      CONTINUE
         NSFLOW=1
         GO TO 70
      ENDIF
130   CALL FWCT (NSFLOW,0)
```

```
      IF (MARKY.LE.0) WRITE (8,620)
      CALL OUTPUT (4,MARKX,MARKY,MADJC)
      CALL FWCT (NSFLOW,1)
      IF (IOUT.NE.0) CALL OUTPUT (5,MARKX,MARKY,MADJC)
      IF (NTEMP.EQ.1) WRITE (8,520)
      IF (MARKY.EQ.0) THEN
         WRITE (6,660)
      ELSE
         WRITE (6,750)
      ENDIF
C
C
      IF (NETW.GE.1.OR.NTEMP.GE.1) GO TO 300
      IF (OPTION.EQ.'CONTINUE') GO TO 350
      OPEN (20,FILE='MFCTL3',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN (18,FILE='MFCTL7',STATUS='UNKNOWN',FORM='UNFORMATTED')
      WRITE (20) NB,NJ,NFNUM,NETW,NTEMP,ITN,DR,TR,TINC,SPAN,IOUT,TOUT,
     .           CONCT,NSTOP
      WRITE (20) NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT,MNO
      WRITE (20) TIME,TSTART,JSTART,NSFLOW,WRNPR,WRNGS,WRNSM,WRNHT
      WRITE (20) MARKX,MARKY,NCOMT1,NCOMTS,NCOMT2,LLINE
      WRITE (20) JAN,JAJ
      DO 3200 I=1,NB
         WRITE (20) NO(I),JS(I),JF(I),NWTYP(I),R(I),Q(I),KF(I),LA(I)
         WRITE (20) A(I),O(I),RSTD(I),DZRD(I),FRNVP(I),NREV(I),RDCH4(I)
         WRITE (20) RDPROP(I),TRD(I),TJS(I),FFRNVP(I),NNREV(I),JSB(I)
         WRITE (20) TROCK(I),TMRD(I),HA(I),HK(I)
         WRITE (20) XNEW(I),DCOAGE(I),CH4V(I),CH4PA(I)
         WRITE (20) NGIN(I),MMIN(I),NGOUT(I),LOUT(I)
3200  CONTINUE
      DO 3202 I=1,IMX
         WRITE (20) NOF(I),NFREG(I),RGRAD(I),NFCW(I),MPTS(I),
     .              NSKP(I),NEGQ(I)
         WRITE (20) NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),SMPO2(I),
     .              HTPO2(I),TPR(I),QCENT(I)
         DO 3201 J=1,IMY
            WRITE (20) QF(I,J),PF(I,J)
            DO 3205 K=1,4
               WRITE (20) FKQ(I,J,K)
3205        CONTINUE
3201     CONTINUE
3202  CONTINUE
      write (18) nfnum,nplot
      do 3215 i=1,nfnum
         write (20) ncof(i),nd(i),nswt(i)
         write (18) nswt(i)
         ii=ncof(i)
         do 3215 j=1,ii
            write (20) cof(i,j)
3215  continue
      write (8,*) nfnum,nswt(1)
      WRITE (20) NPLOT
      DO 3203 I=1,NJ
         WRITE (20) JNO(I),T(I),Z(I),CH4C(I),JNOL(I),PROP(I),
     .              PRCH4(I),JLR(I)
3203  CONTINUE
      DO 3204 K=1,JAJ
         WRITE (20) JSTAR(K),TSTAR(K)
3204  CONTINUE
      DO 3206 I=1,MNO
         WRITE (20) MEND(I),FNVP(I)
3206  CONTINUE
      IYY=MEND(MNO)
      DO 3207 I=1,IYY
         WRITE (20) MSL(I)
```

```
3207  CONTINUE
      DO 3209 I=1,50
         WRITE (20) ITITLE(I)
         DO 3208 J=1,20
            WRITE (20) TITLE(I,J)
3208     CONTINUE
3209  CONTINUE
      DO 3210 I=1,NB
         WRITE (20) INU(I)
3210  CONTINUE
C
300   IF (MARKY.LE.0) THEN
      OPEN (21,FILE='MFCTL2',STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPTION='CONTINUE'
      IF (NSTOP.GE.1.OR.NETW.GE.1.OR.NTEMP.GE.1)
     .   OPTION='DONE'
      REWIND (21)
      WRITE (21) OPTION
      WRITE (21) MD1,TACC
      IF (NSTOP.GE.1.AND.NSTOP.LE.4) THEN
         NNB=NB
         IF (NSTOP.EQ.1) NNB=KV
         IF (NSTOP.NE.8) WRITE (8,800)
         WRITE (8,810) (NO(I),JS(I),JF(I),NWTYP(I),R(I),Q(I),
     .                  KF(I),LA(I),A(I),O(I),I=1,NNB)
      ENDIF
      IF (NVPN.GE.1) WRITE (8,880)
      IF (NVPN.LE.0.AND.NSTOP.GE.2.AND.NSTOP.LE.4) THEN
         NNJ=NJ
         IF (NSTOP.EQ.2) NNJ=KV
         WRITE (8,820)
         WRITE (8,825) (JNO(J),T(J),Z(J),CH4C(J),J=1,NNJ)
      ENDIF
      IF (NFNUM.GT.0.AND.NSTOP.GE.3.AND.NSTOP.LE.4) THEN
         NNFNUM=NFNUM
         IF (NSTOP.EQ.3) NNFNUM=KV
         WRITE (8,830) (NOF(K),K=1,NNFNUM)
         WRITE (8,840)
         DO 310 K=1,NNFNUM
            L=MPTS(K)
            IF (L.GT.IMY) L=IMY
            DO 305 I=1,L
               QF(K,I)=QF(K,I)/1000.0
305         CONTINUE
            WRITE (8,850) (QF(K,I),PF(K,I),I=1,L)
310      CONTINUE
      ENDIF
      IF (NSTOP.GT.0) WRITE (8,890)
      ENDIF
349   IF (NSTOP.GE.1) THEN
         OPEN (25,FILE='JUMP',STATUS='UNKNOWN',FORM='UNFORMATTED')
         WRITE (25) OPTION
      ENDIF
350   STOP
C
470   FORMAT (//,' REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING ',
     .        'PLACES:',/)
480   FORMAT (1X,'AIRWAY ',I5,' IS NOW CARRYING AIR FROM',I5,' TO',I5)
482   FORMAT (//,1X,'NO AIRFLOW REVERSAL WAS DETECTED IN THE SYSTEM.')
500   FORMAT (//,' * ATTENTION *  DATA PREPARATION DID NOT COMPLETE ',
     .        'WITHIN',/,' 100 ITERATIONS IN DATA PREPARATION CYCLE',I4)
510   FORMAT (//,1X,'*****  NO FURTHER CALCULATIONS BEYOND NETWORK ',
     .       'BALANCING WERE',/,7X,'DEMANDED DUE TO NETW=1, PROG. ',
     .       'RUN COMPLETED.',/)
520   FORMAT (//,1X,'***** NO FURTHER CALCULATIONS BEYOND TEMP. ',
```

```
     .       'CALCULATIONS WERE',/,7X,'DEMANDED DUE TO NTEMP=1, PROG.',
     .       ' RUN COMPLETED.',/)
 560  FORMAT (///,1X,71('*'),///)
 600  FORMAT (8X,F8.4,T31,I5,T47,F6.2)
 620  FORMAT (///,1X,71('*'),///,T13,'OUTPUT OF THE TEMPERATURE',
     .       ' PART OF THE PROGRAM',///,1X,71('*'))
 660  FORMAT (/,1X,'TEMPERATURE PART COMPLETED.',/)
 670  FORMAT (/,1X,'NETWORK CALCULATION PART COMPLETED.',/)
 680  FORMAT (/,1X,'NAME YOUR INPUT FILE BY "D:\PATH\FNAME1.EXT".',//)
 730  FORMAT (30(/),1X,'PROGRAM MFIRE ACTIVATED ...')
 740  FORMAT (//,1X,'QUASI-EQUILIBRIUM PART STARTED.',/)
 750  FORMAT (/,1X,'SIMULATION COMPLETED.',///)
 760  FORMAT (///,1X,71('*'),///,T15,'OUTPUT OF THE QUASI-EQUILIBRIUM',
     .       ' SIMULATION PART',///,1X,71('*'))
 800  FORMAT (////,1X,'THE FOLLOWING IS THE CRITICAL DATA READ IN ',
     .        'SO FAR:',///,1X,T4,'NO',T9,'JS',T14,'JF',T18,'NWTYP',
     .        T30,'R',T39,'Q',T46,'KF',T53,'LA',T61,'A',T69,'O',/)
 810  FORMAT (4I5,T23,F10.3,F10.0,T44,I4,T49,F7.0,T58,F6.1,T66,F5.1)
 820  FORMAT (///,T5,'JNO',T15,'T',T25,'Z',T35,'CH4C',/)
 825  FORMAT (T4,I4,T11,F6.1,T21,F7.1,T31,F8.2)
 830  FORMAT (//,' THESE CHARACTERISTICS WERE READ IN FOR FANS',5I5,/
     .        T45,5I5)
 840  FORMAT (//,5('  Q*1000   PF '),/)
 850  FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
 855  FORMAT (/,1X,'ENTERING TEMPERATURE PART OF THE PROGRAM.')
 860  FORMAT (/,1X,'ENTERING QUASI-EQUIL. PART OF THE PROGRAM.')
 865  FORMAT (1X,'PREFIXED ACCURACY CRITERION = 0.0002 IN.W.G. PER ',
     .        'MESH.',/)
 870  FORMAT (10X,'ITERATION ',I3,'  COMPLETED, CURRENT ERROR = ',
     .        F8.5)
 880  FORMAT (/,1X,'NO JUNCTION CARDS ARE EXPECTED TO READ IN DUE TO ',
     .        'NVPN>1 OR =1.',/,1X,'ONLY THE NETWORK PART WILL BE ',
     .        'PERFORMED.')
 890  FORMAT (//,1X,'PROGRAM RUN TERMINATED DUE TO INDICATED INPUT ',
     .        'ERROR.',/,' CORRECT THE ERROR THEN YOU MAY TRY AGAIN.',
     .        ///)
 900  FORMAT (30(/),T19,17('-'),' MFIRE ',18('-'),//,T20,'A program ',
     .        'for dynamic simulation of mine',/,T21,'ventilation ',
     .        'under thermal disturbances',///,T23,'Written by:',T39,
     .        'Xintan Chang, Ph.D',/T23,'Supervised by:',T39,'Dr. ',
     .        'R. E. Greuer',//,T25,'Oct. 1987',T38,'at MICHIGAN TECH',
     .        /,T19,42('-'),2(/),T28,'Version 2.20 JUNE 1995',//)
 910  FORMAT (30(/),T35,'DISCLAIMER',//,T10,'The Bureau of Mines express
     .ly declares that there are no',/,T10,'warranties express or implie
     .d which apply to the software',/,T10,'contained herein. By accepta
     .nce and use of said software',/,T10,'which is conveyed to the user
     . without consideration by the',/,T10,'Bureau of Mines, the user he
     .rof expressly waives any and',/,T10,'all claims for damage and/or
     .suits for or by reason of',/,T10,'personal injury, or property dam
     .age, including special',/,T10,'consequential or other similar dama
     .ges arising out of or',/,T10,'in any way connected with the use of
     . the software',/,T10,'contained herein.',///)
 975  FORMAT (//,2X,'SOME DATA POINTS OF FAN',I5,2X,'CURVE ARE TOO CLOSE
     .',/,'SO THE LEAST SQUARE METHOD IS DESIRED')
 980  FORMAT (//,2X,'THE DATA POINTS OF FAN CHARACTERISTIC ARE',I5,/,
     .'AT LEAST 3 DATA POINTS ARE DESIRED',/)
1000  FORMAT (//,2X,'RECOMMEND LEAST SQUARE METHOD FOR FAN',I5,/,
     .'SURVE FITTING',/)
1005  FORMAT (//,2X,'RECOMMEND SPLINE METHOD FOR FAN',I5,/,
     .'SURVE FITTING')
1010  FORMAT (//,'THESE CHARACTERISTICS WERE STORED FOR FAN IN AIRWAY'
     .,I5,/)
1020  FORMAT (//,5('  Q*1000   PF '),/)
1030  FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
1045  FORMAT (/,T5,'FAN',I5,/)
1085  FORMAT (/,2X,'INTERPOLATION:',/,5('  Q*1000   PF '),/)
1086  FORMAT (//,2X,'APPLYING SPLINE METHOD')
1087  FORMAT (//,2X,'APPLYING LEAST SQUARE METHOD')
1090  FORMAT (5(F8.1,F6.2))
1100  FORMAT (//,2X,'INPUT SELECT FAN CURVE FITTING METHOD',/,
     .'1-- FOR LEAST SQUARE METHOD, 2--FOR SPLINE METHOD')
C
      END
C
C
      BLOCK DATA
C
C     ---------------------------------------------------------------
C
C     SUBPROGRAM PURPOSE:
C     INITIATION OF THE VARIABLES AND ARRAYS IN COMMON BLOCKS.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
      PARAMETER ( K1=NMX*4,  K2=NMX*3,  K3=NMX*6,  K4=NMX*8,  K5=NMY*3,
     .            K6=IMX*2,  K7=IMX*IMY*2,    K8=LMX+NMY+1, K9=NMY+NMX,
     .            K10=NMY*2, K11=NMX*2+IMX*11, K16=NMZ*10*5, K22=NMX*2,
     .            K17=NMZ*10*5+NMX*2, K19=NMZ*22, K21=IMX*3, K23=NMY*4,
     .            K24=IMY*5+IMX*IMY*4, K25=NXX*3)
C
      DATA  NO,JS,JF,NWTYP,R,Q,P,KF,LA,A,O,RSTD,DZRD,FRNVP,NREV,RDCH4,
     .      RDPROP,TRD,TJS,RDOP,RCH4,FFRNVP,RA,NNREV
     .      /K1*0,K2*0.0,NMX*0,K3*0.0,NMX*0,K4*0.0,NMX*0/
      DATA  NOF,NFREG,RGRAD,NFCW,MPTS,QF,PF,NSKP,NEGQ,JSB
     .      /K6*0,IMX*0.0,K6*0,K7*0.0,IMX*0,IMX*0,NMX*0/
      DATA  FC1,FC2,FC3,DK,FK,FKQ
     .      /K24*0.0/
      DATA  MNO,MEND,MSL,FNVP,RQ,INU,KNO,KJS,KJF
     .      /K8*0,K9*0.0,NMX*0,K25*0/
      DATA  JNO,T,Z,CH4C,JNOL,PROP,PRCH4,JLR
     .      /NMY*0,K5*0.0,NMY*0,K10*0.0,NMY*0/
      DATA  TROCK,TMRD,HA,HK
     .      /K1*0.0/
      DATA  NCENT,CH4V,CH4PA,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,TFSI,
     .      O2BEH,TPR,HTAD,QCENT
     .      /NMX*0,K11*0.0/
      DATA  NGIN,MMIN,NGOUT,LOUT
     .      /K1*0/
      DATA  MEMREC,NOREC,ESTPR,ESTCH4,ESTTR
     .      /K22*0,K2*0.0/
      DATA  XNEW,DCOAGE
     .      /K22*0.0/
      DATA  TIME,TSTART,JSTART,TLEFT
     .      /0.0,0.0,0,NMX*0.0/
      DATA  SUMQ,SUMC,SUMM,SMHEAT
     .      /K23*0.0/
      DATA  PROPS,PRCH4S,QQ,QTP,TTJS,TTRD,TAS,BI
     .      /K4*0.0/
      DATA  DELTAT,TACC,SUMFNV
     .      /3*0.0/
      DATA  IBTN,DPPA,NSAC,TMRDA,DELTAQ,DPPB,NSACB,GBTN,JCH,TAUXA,
     .      TAUXB,TAUXC,TAUXD,FMASS
     .      /NMX*0,K16*0.0,K22*0,K17*0.0,K22*0,50*0.0,NMY*0,K19*0.0,
     .       NMZ*0.0/
      DATA  WRNPR,WRNGS,WRNSM,WRNHT,IOMIT,ROMIT,IAC
     .      /4*0.0,IMX*0,K21*0.0,0/
      DATA  ITITLE,NCOMT1,NCOMTS,NCOMT2,LLINE
     .      /54*0/
C
```

```
C
      END
C
C
      SUBROUTINE  INPUT (JUMP1,NSTOP,MARKY,MAXNO,K)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       INPUT OF DATA.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION DAL(15)
C
      IF (JUMP1.EQ.1) THEN
C
C
         NBR=0
         NCOMTS=0
         NCOMT1=0
10       CALL READIN (DAL,15,ISTOP,0)
         IF (ISTOP.EQ.1) THEN
            WRITE (8,520)
            WRITE (8,530) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         IF (DAL(15).LT.(-1.E20)) GO TO 10
         IF (NCOMTS.GT.NCOMT1) THEN
            NCOMT1=NCOMTS
            GO TO 10
         ENDIF
         NB=DAL(1)
         NFNUM=DAL(2)
         INFLOW=DAL(3)
         NVPN=DAL(4)
         NETW=DAL(5)
         TR=DAL(6)
         MADJ=DAL(7)
         ITN=DAL(8)
         NTEMP=DAL(9)
         TINC=DAL(10)
         SPAN=DAL(11)
         IOUT=DAL(12)
         TOUT=DAL(13)
         CONCT=DAL(14)
         DR=DAL(15)
         IF (MARKY.EQ.0) WRITE (6,720)
         IF (ABS(TR-60.).GT.50.) TR=75.0
         DRR=2112.6/(53.35*(TR+460.0))
         IF (DR.LT.0.001) THEN
            DR=DRR
         ELSE IF (ABS((DR-DRR)/DRR).GT.0.7) THEN
            WRITE (8,961) DR
            NSTOP=8
            RETURN
         ELSE IF (ABS((DR-DRR)/DRR).GT.0.5) THEN
            WRITE (8,960) DR
         ENDIF
         IF (MADJ.LT.5.OR.MADJ.GT.80) MADJ=10
         IF (ITN.LT.5.OR.ITN.GT.80) ITN=10
         IF (NB.GT.NMX) WRITE (8,210) NB,NMX
         IF (NB.GT.NMX) THEN
            WRITE (8,520)
            WRITE (8,530) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         IF (IOUT.GT.0) IOUT=1
         IF (NVPN.GE.1) NETW=1
         III=1
         KNT=NB
         IF (NB.LE.1) KNT=NMX
         NCOMT2=NCOMTS
         DO 30 K=1,KNT
20          CALL READIN (DAL,15,ISTOP,1)
            IF (ISTOP.EQ.1) THEN
               WRITE (8,540)
               WRITE (8,550) K,(ROW(IE),IE=1,80)
               NSTOP=1
               RETURN
            ENDIF
            IF (DAL(15).LT.(-1.E20)) GO TO 20
            IF (NCOMTS.GT.NCOMT2) THEN
               NCOMT2=NCOMTS
               IF (NCOMTS.LE.50) ITITLE(NCOMTS)=K
               GO TO 20
            ENDIF
            IF (DAL(1).LT.0.0.OR.DAL(2).LT.0.0.OR.DAL(3).LT.0.0.OR.
     .          DAL(7).LT.0.0.OR.DAL(8).LT.0.0.OR.DAL(9).LT.0.0.OR.
     .          DAL(10).LT.0.0) THEN
               WRITE (8,380) INT(DAL(1))
               WRITE (8,540)
               WRITE (8,550) K,(ROW(IE),IE=1,80)
               NSTOP=1
               RETURN
            ENDIF
            IF (DAL(2).GE.1000.0.OR.DAL(3).GE.1000.0) THEN
               WRITE (8,390) INT(DAL(1))
               WRITE (8,540)
               WRITE (8,550) K,(ROW(IE),IE=1,80)
               NSTOP=1
               RETURN
            ENDIF
            IF (ABS(DAL(4)).GT.1.1) THEN
               WRITE (8,400) INT(DAL(1))
               WRITE (8,540)
               WRITE (8,550) K,(ROW(IE),IE=1,80)
               NSTOP=1
               RETURN
            ENDIF
            NO(K)=DAL(1)
            JS(K)=DAL(2)
            JF(K)=DAL(3)
            NWTYP(K)=DAL(4)
            R(K)=DAL(5)
            Q(K)=DAL(6)
            KF(K)=DAL(7)
            LA(K)=DAL(8)
            A(K)=DAL(9)
            O(K)=DAL(10)
C
            IF ((ABS(A(K)).LE.0.01).AND.(ABS(O(K)).LE.0.01)) THEN
               A(K)=100.0
               O(K)=40.0
               WRITE (8,991) DAL(9),DAL(10),NO(K),A(K),O(K)
            ELSE IF ((ABS(A(K)).LE.0.01).AND.(ABS(O(K)).GT.0.01)) THEN
```

```
              A(K)=(0.25*O(K))*(0.25*O(K))
              WRITE (8,992) DAL(9),NO(K),DAL(10),A(K)
           ELSE IF ((ABS(A(K)).GT.0.01).AND.(ABS(O(K)).LE.0.01)) THEN
              O(K)=4.0*SQRT(A(K))
              WRITE (8,993) DAL(10),NO(K),DAL(9),O(K)
C          **********************************************************
C           ELSE IF ((ABS(A(K)).GT.0.01).AND.(ABS(O(K)).GT.0.01)) THEN
C              OLMT1=SQRT(A(K))
C              OLMT2=8.0*SQRT(A(K))
C              IF ((O(K).LT.OLMT1).OR.(O(K).GT.OLMT2)) THEN
C                 O(K)=4.0*SQRT(A(K))
C                 WRITE (8,994) DAL(10),NO(K),A(K),O(K)
C              ENDIF
C          **********************************************************
           ENDIF
C
           IF (DAL(11).GT.1.E-3) HA(K)=DAL(11)
           IF (DAL(12).GT.1.E-3) HK(K)=DAL(12)
           IF (DAL(13).GT.1.E-3) CH4V(K)=DAL(13)
           IF (DAL(14).GT.1.E-3) CH4PA(K)=DAL(14)
           IF (DAL(15).GT.1.E-3) TROCK(K)=DAL(15)
           IF (NWTYP(K).EQ.(-1).AND.Q(K).LT.1.0) THEN
              WRITE (8,510) NO(K),Q(K)
              WRITE (8,540)
              WRITE (8,550) K,(ROW(IE),IE=1,80)
              NSTOP=1
              RETURN
           ENDIF
           IF (NO(K).LT.9990) THEN
              NBR=NBR+1
              K1=JS(K)
              MSL(K1)=1
              K1=JF(K)
              MSL(K1)=1
           ELSE
              GO TO 40
           ENDIF
           NODE=0
           IF (Q(K).LT.1000.0) Q(K)=1000.0
           DO 30 I=1,999
              IF (MSL(I).EQ.1) NODE=NODE+1
              IF (K.EQ.NB.OR.NO(K).GT.9990) MSL(I)=0
30      CONTINUE
C
        IF ((NB.LE.1).AND.(NO(K).LE.9990)) THEN
           K=NMX
           WRITE (8,260) NMX,NO(K)
           NSTOP=1
           RETURN
        ENDIF
C
40      NCOMTS=100
        NCOMT2=NCOMTS
        IF (NBR.LE.1) THEN
           WRITE (8,360)
           NSTOP=1
           RETURN
        ENDIF
        IF (NB.LE.1.OR.NB.GT.NBR) NB=NBR
        IF (MARKY.EQ.0) WRITE (6,730) NB
        DO 50 I=1,NB
           JSB(I)=JS(I)
50      CONTINUE
C
C     NVPN: MARKER FOR PRESENCE OF JUNCTION CARDS.
C
        IF (III.GT.50.AND.MARKY.LE.0) WRITE (8,255) III
        NJ=NODE
        IF (NJ.GT.NMY) THEN
           WRITE (8,220) NJ,NMY
           NSTOP=5
           RETURN
        ENDIF
        IF (NVPN.LE.0) THEN
           DO 70 K=1,NJ
60            CALL READIN (DAL,4,ISTOP,0)
              IF (ISTOP.EQ.1) THEN
                 WRITE (8,560)
                 WRITE (8,570) K,(ROW(IE),IE=1,80)
                 NSTOP=2
                 RETURN
              ENDIF
              IF (DAL(15).LT.(-1.E20)) GO TO 60
              IF (NCOMTS.GT.NCOMT2) THEN
                 NCOMT2=NCOMTS
                 GO TO 60
              ENDIF
              IF (DAL(1).LE.0.0.OR.DAL(1).GE.1000.0) THEN
                 WRITE (8,410) INT(DAL(1))
                 WRITE (8,560)
                 WRITE (8,570) K,(ROW(IE),IE=1,80)
                 NSTOP=2
                 RETURN
              ENDIF
              IF (DAL(4).LT.0.0.OR.DAL(4).GE.100.0) THEN
                 WRITE (8,830) INT(DAL(1))
                 WRITE (8,560)
                 WRITE (8,570) K,(ROW(IE),IE=1,80)
                 NSTOP=2
                 RETURN
              ELSE IF (DAL(4).GT.5.0.AND.MARKY.LE.0) THEN
                 WRITE (8,860) DAL(4),INT(DAL(1))
              ENDIF
              JNO(K)=DAL(1)
              IF (ABS(DAL(2)-75.).GT.50.0.AND.MARKY.LE.0)
     .           WRITE (8,420) DAL(2),JNO(K)
              T(K)=DAL(2)
              Z(K)=DAL(3)
              CH4C(K)=DAL(4)
70         CONTINUE
           IF (MARKY.EQ.0) WRITE (6,740) NJ
         ENDIF
72      CALL READIN (DAL,1,ISTOP,0)
        IF (ISTOP.EQ.1) THEN
           WRITE (8,575)
           WRITE (8,576) (ROW(IE),IE=1,80)
           NSTOP=2
           RETURN
        ENDIF
        IF (DAL(15).LT.(-1.E20)) GO TO 72
        IF (NCOMTS.GT.NCOMT2) THEN
           NCOMT2=NCOMTS
           GO TO 72
        ENDIF
        JAN=DAL(1)
        IF (JAN.GT.IMX.OR.JAN.LE.0) THEN
           WRITE (8,940) JAN,IMX
           NSTOP=2
           RETURN
        ENDIF
```

```
         IF (JAN.GT.1) THEN
            DO 75 I=1,JAN
73             CALL READIN (DAL,2,ISTOP,0)
               IF (ISTOP.EQ.1) THEN
                  WRITE (8,577)
                  WRITE (8,578) (ROW(IE),IE=1,80)
                  NSTOP=2
                  RETURN
               ENDIF
               IF (DAL(15).LT.(-1.E20)) GO TO 73
               IF (NCOMTS.GT.NCOMT2) THEN
                  NCOMT2=NCOMTS
                  GO TO 73
               ENDIF
               JA1(I)=DAL(1)
               JATP(I)=DAL(2)
75          CONTINUE
            WRITE (8,980)
            WRITE (8,990) (JA1(IU),IU=1,JAN)
         ENDIF
C
C     NFNUM: NUMBER OF FANS WITH THEIR CHARACTERISTICS.
C
         IF (NFNUM.GT.0) THEN
            IF (NFNUM.GT.IMX) THEN
               WRITE (8,230) NFNUM,IMX
               NSTOP=5
            ENDIF
            DO 110 K=1,NFNUM
80             CALL READIN (DAL,3,ISTOP,0)
               IF (ISTOP.EQ.1) THEN
                  WRITE (8,580)
                  WRITE (8,590) K,(ROW(IE),IE=1,80)
                  NSTOP=3
                  RETURN
               ENDIF
               IF (DAL(15).LT.(-1.E20)) GO TO 80
               IF (NCOMTS.GT.NCOMT2) THEN
                  NCOMT2=NCOMTS
                  GO TO 80
               ENDIF
               NOF(K)=DAL(1)
               DO 81 KP=1,NB
                  IF (NO(KP).EQ.NOF(K)) GO TO 82
81             CONTINUE
               WRITE (8,950) NOF(K)
82             MPTS(K)=DAL(2)
               INDEX=MPTS(K)
               IF (INDEX.LE.1.OR.INDEX.GT.IMY) THEN
                  WRITE (8,240) INDEX,IMY
                  WRITE (8,580)
                  WRITE (8,590) K,(ROW(IE),IE=1,80)
                  NSTOP=3
                  RETURN
               ENDIF
               IF (MARKY.EQ.1) GO TO 85
               NSWT(K)=DAL(3)
85             DO 100 JDEX=1,INDEX,5
                  I=10
                  IF ((JDEX+4).GT.INDEX) I=((INDEX+1)-JDEX)*2
90                CALL READIN(DAL,I,ISTOP,0)
                  IF (ISTOP.EQ.1) THEN
                     WRITE (8,600)
                     WRITE (8,610) K,(ROW(IE),IE=1,80)
                     NSTOP=3
                     RETURN
                  ENDIF
                  IF (DAL(15).LT.(-1.E20)) GO TO 90
                  IF (NCOMTS.GT.NCOMT2) THEN
                     NCOMT2=NCOMTS
                     GO TO 90
                  ENDIF
                  IF (INDEX.GE.(JDEX+4)) THEN
                     PF(K,JDEX+4)=DAL(10)
                     QF(K,JDEX+4)=DAL(9)
                  ENDIF
                  IF (INDEX.GE.(JDEX+3)) THEN
                     PF(K,JDEX+3)=DAL(8)
                     QF(K,JDEX+3)=DAL(7)
                  ENDIF
                  IF (INDEX.GE.(JDEX+2)) THEN
                     PF(K,JDEX+2)=DAL(6)
                     QF(K,JDEX+2)=DAL(5)
                  ENDIF
                  IF (INDEX.GE.(JDEX+1)) THEN
                     PF(K,JDEX+1)=DAL(4)
                     QF(K,JDEX+1)=DAL(3)
                  ENDIF
                  PF(K,JDEX)=DAL(2)
                  QF(K,JDEX)=DAL(1)
100            CONTINUE
               LSTOP=0
               DO 105 J=1,INDEX-1
                  IF (QF(K,J).GE.QF(K,J+1)) LSTOP=1
                  IF (QF(K,J).LT.0.0) THEN
                     WRITE (8,920) NOF(K)
                     NSTOP=3
                     RETURN
                  ENDIF
105            CONTINUE
               IF (LSTOP.EQ.1) THEN
                  WRITE (8,430) NOF(K)
                  NSTOP=3
                  RETURN
               ENDIF
110         CONTINUE
111      CALL READIN (DAL,1,ISTOP,0)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         IF (ISTOP.EQ.1) THEN
            WRITE (8,594)
            WRITE (8,595) (ROW(IE),IE=1,80)
            NSTOP=2
            RETURN
         ENDIF
         IF (DAL(15).LT.(-1.E20)) GO TO 111
         IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 111
         ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         NPLOT=DAL(1)
         ENDIF
         DO 120 I=1,NB
            IF (NWTYP(I).EQ.1) THEN
               IF (NFNUM.GT.0) THEN
                  DO 115 J=1,NFNUM
                     IF (NO(I).EQ.NOF(J)) GO TO 120
115               CONTINUE
               ENDIF
               IF (R(I).LT.0.0) THEN
```

```
                  WRITE (8,810) NO(I),R(I)
                  NSTOP=5
                  RETURN
               ELSE IF (R(I).GT.18.0.AND.MARKY.LE.0) THEN
                  WRITE (8,820) NO(I),R(I)
               ENDIF
            ENDIF
120      CONTINUE
         IF (MARKY.EQ.0) WRITE (6,750) NFNUM
         NCOMTT=NCOMT1
         IF (NCOMT1.GT.50) NCOMTT=50
         IF (NCOMT1.GT.0.AND.NSTOP.EQ.0.AND.NETW.GE.1.
     .      AND.MARKY.LE.0)THEN
            WRITE (8,201)
            WRITE (8,203) ((TITLE(I,J),J=1,18),I=1,NCOMTT)
            WRITE (8,201)
            IF (NVPN.GE.1) WRITE (8,270)
         ENDIF
      ELSE IF (JUMP1.EQ.2) THEN
C
C     INFLOW: NUMBER OF CONTAMINATION CARDS.
C
         IF (INFLOW.GT.0) THEN
            IF (INFLOW.GT.IMX) THEN
               WRITE (8,250) INFLOW,IMX
               NSTOP=5
               RETURN
            ENDIF
            IF ((NB+INFLOW).GT.NMX) THEN
               WRITE (8,340) NB,INFLOW,NMX
               NSTOP=5
               RETURN
            ELSE
               DO 160 I=1,INFLOW
140               CALL READIN (DAL,9,ISTOP,0)
                  IF (ISTOP.EQ.1) THEN
                     K=I
                     WRITE (8,660)
                     WRITE (8,670) I,(ROW(IE),IE=1,80)
                     NSTOP=4
                     RETURN
                  ENDIF
                  IF (DAL(15).LT.(-1.E20)) GO TO 140
                  IF (NCOMTS.GT.NCOMT2) THEN
                     NCOMT2=NCOMTS
                     GO TO 140
                  ENDIF
                  DO 142 J=1,NB
                     IF (INT(DAL(1)).EQ.NO(J)) GO TO 143
142               CONTINUE
                  IF (MARKY.LE.0) WRITE (8,470) INT(DAL(1))
                  GO TO 160
143               IF (DAL(2).LT.0.0.OR.DAL(3).LT.0.0.OR.DAL(5).LT.0.0.
     .                OR.DAL(6).LT.0.0.OR.DAL(7).LT.0.0) THEN
                     K=I
                     WRITE (8,840)
                     WRITE (8,660)
                     WRITE (8,670) I,(ROW(IE),IE=1,80)
                     NSTOP=4
                     RETURN
                  ENDIF
                  IF (DAL(3).GT.100.0) THEN
                     K=I
                     WRITE (8,880) DAL(3),INT(DAL(1))
                     WRITE (8,660)
                     WRITE (8,670) I,(ROW(IE),IE=1,80)
                     NSTOP=4
                     RETURN
                  ENDIF
                  NCENT(I)=DAL(1)
                  CONT(I)=DAL(2)
                  CONC(I)=DAL(3)
                  HEAT(I)=DAL(4)
                  O2MIN(I)=DAL(5)
                  SMPO2(I)=DAL(6)
                  HTPO2(I)=DAL(7)
                  QCENT(I)=DAL(8)
                  TPR(I)=DAL(9)
                  IF (TPR(I).LT.1.E-5) TPR(I)=1.E-5
                  IF (QCENT(I).LT.0.0) QCENT(I)=0.0
                  IF (QCENT(I).GT.10.0) THEN
                     VART=(9900.0+TR)**2+2.0*HEAT(I)/
     .                    (QCENT(I)*DR*2.4E-5)
                     IF (VART.LT.9.4E7) THEN
                        WRITE (8,930) NCENT(I)
                        NSTOP=1
                        RETURN
                     ENDIF
                     TESTI=-9900.0+SQRT(VART)
                     IF (O2MIN(I).LE.0.1.AND.SMPO2(I).LE.1.E-3.AND.
     .                  HTPO2(I).LT.1.0.AND.MARKY.LE.0) THEN
                        IF (TESTI.LT.(-70.0).OR.TESTI.GT.3000.0)
     .                     WRITE (8,870) NCENT(I)
                     ENDIF
                  ENDIF
                  DO 145 K=1,NJ
                     IF (JS(J).EQ.JNO(K)) GO TO 150
145               CONTINUE
                  WRITE (8,900) NCENT
                  NCENT(I)=0
                  GO TO 160
150               DO 155 I2=NB+1,J+1,-1
                     NO(I2)=NO(I2-1)
                     JS(I2)=JS(I2-1)
                     JF(I2)=JF(I2-1)
                     NWTYP(I2)=NWTYP(I2-1)
                     R(I2)=R(I2-1)
                     RSTD(I2)=RSTD(I2-1)
                     Q(I2)=Q(I2-1)
                     KF(I2)=KF(I2-1)
                     LA(I2)=LA(I2-1)
                     A(I2)=A(I2-1)
                     O(I2)=O(I2-1)
                     HA(I2)=HA(I2-1)
                     HK(I2)=HK(I2-1)
                     CH4V(I2)=CH4V(I2-1)
                     CH4PA(I2)=CH4PA(I2-1)
                     TROCK(I2)=TROCK(I2-1)
155               CONTINUE
                  NO(J+1)=1000+MAXNO+1
                  IF (NO(J+1).GT.9995) NO(J+1)=MAXNO+1
                  JS(J+1)=MAXJ+1
                  JF(J)=MAXJ+1
                  NWTYP(J)=10
                  RSTD(J+1)=R(J+1)
                  R(J)=R(J)*0.1/LA(J)
                  IF (R(J).GT.0.001) R(J)=0.001
                  RSTD(J)=R(J)
                  LA(J+1)=LA(J)
                  LA(J)=0.1
```

```
                  CH4V(J)=0.0
                  CH4PA(J)=0.0
                  JNO(NJ+1)=MAXJ+1
                  T(NJ+1)=T(K)
                  Z(NJ+1)=Z(K)
                  CH4C(NJ+1)=CH4C(K)
                  DO 159 I3=1,50
                     IF (ITITLE(I3).GT.J) ITITLE(J)=ITITLE(J)+1
159               CONTINUE
                  NB=NB+1
                  NJ=NJ+1
                  MAXNO=MAXNO+1
                  MAXJ=MAXJ+1
160            CONTINUE
               IF (MARKY.EQ.0) WRITE (6,760) INFLOW
               DO 165 I=1,NB
                  JSB(I)=JS(I)
165            CONTINUE
            ENDIF
         ENDIF
         NCOMTT=NCOMT1
         IF (NCOMT1.GT.50) NCOMTT=50
         IF (NSTOP.EQ.0) THEN
            IF (MARKY.LE.0) WRITE (8,910)
            IF (NCOMT1.GT.0.AND.MARKY.LE.0) THEN
               WRITE (8,201)
               WRITE (8,203) ((TITLE(I,J),J=1,18),I=1,NCOMTT)
               WRITE (8,201)
            ENDIF
            IF (MARKY.LE.0.OR.IOUT.LE.(-2)) WRITE (8,280) TR,DR
            IF (INFLOW.GT.0.AND.MARKY.EQ.0) THEN
               WRITE (8,350)
               DO 167 I=1,INFLOW
                  IF (HTPO2(I).GE.500.) WRITE (8,890) HTPO2(I),NCENT(I)
167            CONTINUE
               DO 168 I=1,INFLOW
                  IF (HEAT(I).LT.0.0) THEN
                     WRITE (8,970) NCENT(I)
                     O2MIN(I)=0.0
                     SMPO2(I)=0.0
                     HTPO2(I)=0.0
                  ENDIF
168            CONTINUE
            ENDIF
         ENDIF
C
C
      ELSE
170      CALL READIN (DAL,11,ISTOP,0)
         IF (ISTOP.EQ.1) THEN
            WRITE (8,640)
            WRITE (8,650) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         IF (DAL(15).LT.(-1.E20)) GO TO 170
         IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 170
         ENDIF

         NAV=DAL(1)
         JSTART=DAL(2)
         TSTART=DAL(3)
         TIME=DAL(4)
         CRITSM=DAL(5)
         CRITGS=DAL(6)
         CRITHT=DAL(7)
         WRNPR=DAL(8)
         WRNSM=DAL(9)
         WRNGS=DAL(10)
         WRNHT=DAL(11)
         IF (TIME.LT.1.E-3) TIME=10.0
         IF (CRITSM.LT.1.E-4.OR.CRITSM.GT.1.0) CRITSM=0.005
         IF (CRITGS.LT.1.E-4.OR.CRITGS.GT.1.0) CRITGS=0.01
         IF (CRITHT.LT.1.E-4.OR.CRITHT.GT.1.0) CRITHT=0.1
         IF (WRNPR.LE.1.E-10.OR.WRNPR.GT.0.5) WRNPR=0.01
         IF (WRNSM.LE.1.E-10.OR.WRNSM.GT.1.0) WRNSM=0.05
         IF (WRNGS.LE.1.E-10.OR.WRNGS.GT.10.0) WRNGS=1.0
         IF (WRNHT.LE.1.E-10.OR.WRNHT.GT.1000.0) WRNHT=100.0
         IF (MARKY.EQ.0) WRITE (6,780)
         J3=0
         DO 175 I=1,NJ
            IF (JNO(I).EQ.JSTART) J3=1
175      CONTINUE
         IF (J3.EQ.0) THEN
            WRITE (8,330) JSTART
            WRITE (8,640)
            WRITE (8,650) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         IF ((TSTART.LT.-40.0.OR.TSTART.GT.120.0).AND.MARKY.LE.0)
     .      WRITE (8,850) TSTART
         IF (TIME.LE.0.0) THEN
            WRITE (8,460) TIME
            WRITE (8,640)
            WRITE (8,650) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
      ENDIF
      RETURN
C
200   FORMAT (20A4)
201   FORMAT (///)
203   FORMAT (18A4)
210   FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' AIRWAYS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I5)
220   FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' JUNCTIONS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I5)
230   FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' FANS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
240   FORMAT (/,1X,'* ERROR *  FAN CURVE CONTAINS',I4,' POINTS',/,
     .        1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
250   FORMAT (/,1X,'* ERROR *  NETWORK CONTAINS',I4,' FIRE SOURCES'
     .        ,/,1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
255   FORMAT (/,1X,'* ATTENTION *  NUMBER OF TITLES:',I3,' EXCEEDS THE '
     .        ,'MAXIMUM (50).',/,' TITLE AFTER NO. 50 WILL BE IGNORED.')
260   FORMAT (/,1X,'* ERROR *  MAXIMUM ALLOWED BRANCHES (',I3,') WAS ',
     .        'READ IN WITHOUT FINDING',/,' ENDING AIRWAY (9999). LAST'
     .        ,' BRANCH READ IN WAS (',I5,')')
270   FORMAT (/,1X,'NO JUNCTION CARDS ARE EXPECTED TO READ IN DUE TO ',
     .        'NVPN>1 OR =1.',/,1X,'ONLY THE NETWORK PART WILL BE ',
     .        'PERFORMED.')
280   FORMAT (/////,' REF.TEMP.(TR): ',F6.1,' DEG.F',11X,'REF. DENSITY',
     .        ' (DR): ',F6.4,' LBM/FT3',///,' LIST OF UNITS USED IN ',
     .        'THIS OUTPUT FILE:',//,' LENGTH:  FT.;',T25,'AREA:  FT2;'
     .        ,T43,'VOLUME:   FT3;',T61,'MASS:   LBM;',/,' AIRFLOW: ',
     .        'FT3/MIN;',T25,'TEMP.  F;',T43,'RESISTANCE:',1X,'1.E-10',
```

```
     .        'IN.W.G./CFM2;',/,' FRICTION FACTOR:',1X,'1.E-10',
     .        'LBM*MIN2/FT4;',T43,'THERMAL DIFFUSIVITY:   FT2/HR;',/,1X,
     .        'THERMAL CONDUCTIVITY:  BTU/HR*FT*F;',T43,'CONCENTRATION:'
     .        ,5X,'PERCENTAGE;',/,' HEADLOSS:',14X,'IN.W.G.;',T43,
     .        'HEAT INPUT:',11X,'BTU/MIN;',/,' CH4 PRODUCTION:',8X,
     .        'FT3/MIN;',///)
 330  FORMAT (/,1X,'START JUNCTION (JSTART) ',I3,' NOT FOUND.')
 340  FORMAT (/,1X,'* ERROR *  CURRENT INPUT DATA:  NB ',I4,
     .        ' AND INFLOW ',I3,/,1X,'THE SUM OF NB AND INFLOW ',
     .        'MUST BE LESS THAN',I6)
 350  FORMAT (/,1X,'* ATTENTION *  A NEW BRANCH HAS BEEN SET UP FOR ',
     .        'EACH HEAT SOURCE.')
 360  FORMAT (/,1X,'* ERROR *  TOO FEW AIRWAY CARD INPUT. AIRWAY ID ',
     .        ' NUMBER > 9990',/,' MARKS THE ENDING OF AIRWAY DATA.')
 380  FORMAT (/,1X,'* ERROR *  ILLEGAL NEGATIVE INPUT DETECTED FOR ',
     .        'AIRWAY ',I4)
 390  FORMAT (/,1X,'* ERROR *  JUNCTION ID >= 1000 DETECTED FOR ',
     .        'AIRWAY ',I4)
 400  FORMAT (/,1X,'* ERROR *  ILLEGAL INPUT VALUE FOR NWTYP DETECTED',
     .        ' FOR AIRWAY ',I4)
 410  FORMAT (/,1X,'* ERROR *  JUNCTION ID >= 1000 OR <0 DETECTED IN ',
     .        'JUNCTION',/,1X,'CARDS AS ',I5)
 420  FORMAT (/,1X,'* ATTENTION *  CHECK UNLIKELY VALUE OF TEMPERATURE',
     .        F7.1,/,' FOR JUNCTION ',I4)
 430  FORMAT (/,1X,'* ERROR *  FAN CURVE INPUT MUST BE IN THE ORDER ',
     .        'FROM LOW-QF',/,' TO HIGH-QF. INVALID ORDER DETECTED ',
     .        'FOR FAN ',I4)
 460  FORMAT (/,1X,'* ERROR *  ILLEGAL ZERO OR NEGATIVE INPUT VALUE ',
     .        /,' OF TIME ',F7.1,' (HOURS) DETECTED.')
 470  FORMAT (/,1X,'* ATTENTION *  HEAT SOURCE ',I5,' DOES NOT MATCH ',
     .        'ANY AIRWAY',/,' IN THE NETWORK, DATA IGNORED.')
 510  FORMAT (//,' * ERROR *  THE SPECIFIED FIXED AIRFLOW FOR ',
     .        'AIRWAY ',I4,' WAS',/,' UNLIKELY SMALL OR NEGATIVE. ',
     .        ' INPUT OF Q WAS ',F12.1)
 520  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN')
 530  FORMAT (/,1X,'THE CURRENT INPUT WAS:',//,80A1)
 540  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'NO,JS,JF,NWTYP,R,Q,KF,LA,A,O,TROCK,HA,HK,',
     .        'CH4V,CH4PA')
 550  FORMAT (/,1X,'THE CURRENT INPUT FOR AIRWAY ',I4,' WAS:',//,80A1)
 560  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'JNO,T,Z,CH4C')
 570  FORMAT (/,1X,'THE CURRENT INPUT FOR JUNCTION ',I4,' WAS:',//,80A1)
 575  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'JAN')
 576  FORMAT (/,1X,'THE CURRENT INPUT FOR JUNCTION ',I4,' WAS:',//,80A1)
 577  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'JA,JATP')
 578  FORMAT (/,1X,'THE CURRENT INPUT FOR JUNCTION ',I4,' WAS:',//,80A1)
 580  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'NOF,MPTS,NSWT')
 590  FORMAT (/,' THE CURRENT INPUT FOR FAN ',I4,' WAS:',//,80A1)
 594  FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,1X,'NPLOT')
 595  FORMAT (/,' THE CURRENT INPUT FOR FAN ',I4,' WAS:',//,80A1)
 600  FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,' QF,PF, QF,PF, QF,PF, QF,PF, QF,PF')
 610  FORMAT (/,' THE CURRENT INPUT FOR FAN ',I4,' WAS:',//,80A1)
 640  FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,' NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,',
     .        'WRNPR,WRNSM,WRNGS,WRNHT')
 650  FORMAT (/,' THE CURRENT INPUT FOR THE SECOND CONTROL CARD WAS:',
     .        //,80A1)
 660  FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .        'IN:',//,' NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT')
 670  FORMAT (/,' THE CURRENT INPUT FOR HEAT SOURCE ',I4,' WAS:',//,
     .        80A1)
 720  FORMAT (/,5X,'CONTROL CARD 1 WAS READ IN.')
 730  FORMAT (5X,'TOTAL',I4,' AIRWAY CARDS WERE READ IN.')
 740  FORMAT (5X,'TOTAL',I4,' JUNCTION CARDS WERE READ IN.')
 750  FORMAT (5X,'TOTAL',I3,' FAN CURVE(S) WERE READ IN')
 760  FORMAT (5X,'TOTAL',I3,' HEAT SOURCE CARD(S) WERE READ IN.')
 780  FORMAT (5X,'CONTROL CARD 2 WAS READ IN.',/)
 810  FORMAT (/,1X,'* ERROR *  FIXED-PRESSURE FAN',I5,' HAS AN ILLEGAL'
     .        ,/,1X,'NEGATIVE PRESSURE VALUE ',F7.3)
 820  FORMAT (/,1X,'* WARNING *  THE INPUT VALUE FOR FIXED-PRESSURE ',
     .        'FAN',I5,' WAS',/,1X,'UNLIKELY TOO LARGE. THE INPUT ',
     .        'FAN PRESSURE WAS ',F9.3)
 830  FORMAT (/,1X,'* ERROR *  JUNCTION ',I4,' HAS AN INVALID CH4 ',
     .        'INPUT. IT MUST BE',/,' WITHIN 0.0 (NO CH4) AND 100.0',
     .        ' (PURE CH4).')
 840  FORMAT (/,1X,'* ERROR *  ILLEGAL NEGATIVE INPUT DETECTED IN ',
     .        'READING HEAT SOURCE CARD.')
 850  FORMAT (/,1X,'* ATTENTION *  CHECK UNLIKELY VALUE OF AIR TEMP. ',
     .        'AT STARTING JUNCTION.',/,' THE CURRENT INPUT VALUE WAS',
     .        F9.2,' (F).')
 860  FORMAT (/,1X,'* WARNING *  AN UNLIKELY INPUT OF CH4 CONCENTRATION'
     .        ,F6.1,' %',/,' AT JUNCTION ',I3,' DETECTED.')
 870  FORMAT (/,1X,'* WARNING *  ABNORMAL TEMPERATURE MAY RESULT IN ',
     .        'BRANCH',I5,/,' ADJUST HEAT AND/OR QCENT.')
 880  FORMAT (/,' * ERROR *  FUME CONCENTRATION (CONC) HIGHER THAN ',
     .        '100% (CONC=',F5.0,')',/,' FOR FIRE SOURCE',I5)
 890  FORMAT (/,1X,'* WARNING *  (HTPO2=',F5.0,') FOR FIRE SOURCE',I5,
     .        ' IS UNLIKELY HIGH,',/,' ANOMALOUS CONSEQUENCE MAY ',
     .        'RESULT.')
 900  FORMAT (/,1X,' * WARNING *  AIRWAY OR STARTING JUNCTION NUMBER ',
     .        'OF HEAT',/,1X,'SOURCE ',I5,' IS NOT ON THE LIST. DATA ',
     .        'IGNORED.')
 910  FORMAT (//,1X,71('*'),///,22X,'OUTPUT OF PROGRAM MFIRE',///,1X,
     .        71('*'))
 920  FORMAT (/,1X,'* ERROR *  NEGATIVE QF INPUT DETECTED FOR FAN',I5)
 930  FORMAT (/,1X,'* ERROR *  ABNORMAL TEMPERATURE WAS RESULTED IN ',
     .        'AIRWAY ',I5,/,' ADJUST HEAT INPUT AND/OR QCENT.')
 940  FORMAT (//,1X,'* ERROR * CAPACITY OF SURFACE JUNCTION ARRAYS ',
     .        'EXCEEDED.',/,' CURRENT INPUT:  JAN=',I5,';  IMX=',I3)
 950  FORMAT (/,' * WARNING *  FAN ',I5,' IS ISOLATED FROM THE NETWORK')
 960  FORMAT (/,' * WARNING *  REF. DENSITY MUST MATCH REF. TEMPERATURE'
     .        ,' AND',/,' ATM. PRESSURE.    CURRENT INPUT = ',F8.4,
     .        ' LBM/FT3')
 961  FORMAT (/,' * ERROR *  REF. DENSITY MUST MATCH REF. TEMPERATURE'
     .        ,' AND',/,' ATM. PRESSURE.    CURRENT INPUT = ',F8.4,
     .        ' LBM/FT3')
 970  FORMAT (/,' * ATTENTION *  BRANCH ',I5,' IS DECLARED AS A FIXED-',
     .        'CAPACITY',/,' COOLING STATION.')
 980  FORMAT (/,' * ATTENTION *  THE FOLLOWING JUNCTIONS ARE IN THE ',
     .        'ATMOSPHERE.',/)
 990  FORMAT (3X,10I6)
 991  FORMAT (/,' * ATTENTION *  THE INPUT CROSS SECTIONAL AREA (',
     .       F4.1,' FT2)',/,'   AND PERIMETER (',F4.1, ' FT.) FOR ',
     .       'AIRWAY',I5,' ARE UNREASONABLE.',/,'   THE CROSS ',
     .       ' SECTIONAL AREA IS NOW MADE TO: ',F8.1,' FT2.',/,
     .       '   THE PERIMETER IS NOW MADE TO: ',F8.1,' FT.')
 992  FORMAT (/,' * ATTENTION *  THE INPUT CROSS SECTIONAL AREA (',
     .       F4.1,' FT2) FOR',/,'   AIRWAY',I5,' IS UNREASONABLE WITH',
     .       ' THE AIRWAY PERIMETER (',F8.1,' FT.).',/,'   THE CROSS',
     .       ' SECTIONAL AREA IS NOW MADE TO: ',F8.1,' FT2')
 993  FORMAT (/,' * ATTENTION *  THE INPUT PERIMETER (',F8.0,' FT.)',
     .     ' FOR AIRWAY',I5,' IS',/,'   UNREASONABLE WITH THE AIRWAY',
     .     ' CROSS SECTIONAL AREA (',F8.0,' FT2).',/,
```

```
     .     '   THE PERIMETER IS NOW MADE TO: ', F8.0,' FT.')
994   FORMAT (/,' * ATTENTION *  THE INPUT PERIMETER (',F8.0,'FT.) FOR',
     .     ' AIRWAY',I5,' IS',/,'   UNREASONABLE WITH THE AIRWAY CROSS',
     .     ' SECTIONAL AREA (',F8.0,'FT2).',/,'   THE PERIMETER IS',
     .     ' NOW MADE TO: ', F8.0,' FT.')
C
      END
C
C
C
      SUBROUTINE CHECK1 (NSTOP,MAXNO,KV)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) MAKE-UP OF THE MISSING DATA.
C       2) CHECK FOR CONSISTENCY OF INPUT DATA.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DO 10 I=1,NB
         IF (NO(I).GT.MAXNO) MAXNO=NO(I)
         IF (NWTYP(I).EQ.0) THEN
            IF (R(I).LE.0.0) THEN
C               IF (A(I).LE.0.0.OR.LA(I).LE.0.0.OR.KF(I).LE.0) THEN
              IF (A(I).LE.0.0.OR.LA(I).LE.0.0) THEN
C                 NCONT=0
C                 DO 5 J=1,JAN
C                    IF (JA1(J).EQ.JS(I).OR.JA1(J).EQ.JF(I)) THEN
C                       NCONT=NCONT+1
C                    ENDIF
C 5               CONTINUE
C                 IF (NCONT.EQ.2) THEN
C                    R(I)=0.000001
C                 ELSE
                   WRITE (8,130) NO(I)
                   KV=NB
C                    NSTOP=4
                   NSTOP=1
              ELSE
                 R(I)=KF(I)*LA(I)*O(I)/(5.2*A(I)**3)*DR/0.075
              ENDIF
            ENDIF
         ENDIF
         RSTD(I)=R(I)
10    CONTINUE
      LF1=0
      DO 15 I=1,NFNUM
         DO 15 J=1,NB
            IF (NWTYP(J).EQ.1.AND.NO(J).EQ.NOF(I)) LF1=1
15    CONTINUE
      IF (LF1.EQ.0) THEN
         DO 16 J=1,NB
            IF(NWTYP(J).EQ.1.AND.R(J).GT.0.0) LF1=1
16       CONTINUE
      ENDIF
      IF (LF1.EQ.0)  WRITE (8,*) "* WARNING * NO FAN IN THE NETWORK. "
C
C
      MAXJ=0
      DO 30 I=1,NB
         IF (JS(I).EQ.JF(I)) THEN
            WRITE (8,100) NO(I)
            KV=NB
            NSTOP=1
         ENDIF
         NREV(I)=0
         N1=0
         N2=0
         IF (NVPN.LE.0) THEN
            KV=NJ
            DO 20 J=1,NJ
               IF (JNO(J).GT.MAXJ.AND.I.EQ.1) MAXJ=JNO(J)
               IF (JS(I).EQ.JNO(J)) N1=N1+1
               IF (JF(I).EQ.JNO(J)) N2=N2+1
20          CONTINUE
            IF (N1.EQ.0) THEN
               WRITE (8,105) NO(I),JS(I)
               NSTOP=2
            ELSE IF (N1.GE.2) THEN
               WRITE (8,110) JS(I)
               NSTOP=2
            ENDIF
            IF (N2.EQ.0) THEN
               WRITE (8,105) NO(I),JF(I)
               NSTOP=2
            ELSE IF (N2.GE.2) THEN
               WRITE (8,110) JF(I)
               NSTOP=2
            ENDIF
         ENDIF
30    CONTINUE
      IF (NVPN.LE.0) THEN
         KV=NJ
         DO 50 J=1,NJ
            J1=0
            DO 40 I=1,NB
               IF (JS(I).EQ.JNO(J).OR.JF(I).EQ.JNO(J)) THEN
                  J1=J1+1
               ENDIF
40          CONTINUE
            IF (J1.EQ.0) THEN
               WRITE (8,115) JNO(J)
               NSTOP=2
            ELSE IF (J1.EQ.1) THEN
               WRITE (8,160) JNO(J)
               NSTOP=2
            ENDIF
50       CONTINUE
      ENDIF
      DO 70 I=1,NB-1
         J1=0
         DO 60 J=I+1,NB
            IF (NO(I).EQ.NO(J)) J1=J1+1
60       CONTINUE
         IF (J1.GT.0) THEN
            WRITE (8,140) NO(I)
            KV=NB
            NSTOP=1
         ENDIF
70    CONTINUE
      IF (NVPN.LE.0) THEN
         DO 90 I=1,NJ-1
            J2=0
            DO 80 J=I+1,NJ
               IF (JNO(I).EQ.JNO(J)) J2=J2+1
80          CONTINUE
            IF (J2.GE.1) THEN
```

```
             WRITE (8,150) JNO(I)
             KV=NJ
             NSTOP=2
          ENDIF
90      CONTINUE
      ENDIF
C
100   FORMAT (/,1X,'* ERROR *  AIRWAY',I5,' IS ISOLATED FROM ',
     .        'THE NETWORK.')
105   FORMAT (/,1X,'* ERROR *  AIRWAY',I5,' IS ISOLATED FROM ',
     .        'THE NETWORK ',/,' FOR THE UNSPECIFIED END',I5,' IN ',
     .        'JNO LIST.')
110   FORMAT (/,1X,'* ERROR *  DUPLICATED JUNCTION NUMBER',I5,' IN',
     .        ' JNO LIST.')
115   FORMAT (/,1X,'* ERROR *  JUNCTION',I4,' IS ISOLATED FROM ',
     .        'THE NETWORK.')
130   FORMAT (/,1X,'* ERROR *  NO RESISTANCE AND DIMENSION (OR KF) WERE'
     .       , ' STATED FOR AIRWAY',I5)
140   FORMAT (/,1X,'* ERROR *  DUPLICATED AIRWAY ID NUMBER',I5,
     .        ' IN THE AIRWAY CARDS.')
150   FORMAT (/,1X,'* ERROR *  DUPLICATED JUNCTION ID NUMBER',I5,
     .        ' IN THE JUNCTION CARDS.')
160   FORMAT (/,1X,'* ERROR *  JUNCTION ',I4,' IS A DEADEND.')
170   FORMAT (/,1X,'* ERROR *  AIRWAY',I5,' IS ISOLATED FROM THE ',
     .        'NETWORK',/,' DUE TO DEADEND',I6)
C
      RETURN
      END
C
      SUBROUTINE ARR (MARKX)
C
C     ------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) GROUPING AIRWAYS ACCORDING TO THEIR TYPES.
C       2) ARRANGING REGULAR AIRWAYS ACCORDING TO THE PRODUCT OF
C          Q AND R.
C
C     ------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C       ORDER OF AIRWAYS IN ARRAY INU: (1) FIXED-Q; (2) FAN; (3) HIGH-
C     RQ; (4) LOW-RQ; (5) FIXED-P.
C
      NBU=NB
      NBL=1
      DO 10 K=1,NB
         IF (NWTYP(K).LT.0) THEN
            INU(NBU)=K
            NBU=NBU-1
         ELSE IF (NWTYP(K).EQ.0.OR.NWTYP(K).EQ.10) THEN
            RQ(K)=ABS(R(K)*Q(K))
            NWTYP(K)=2
         ELSE
            IF (NFNUM.GT.0) THEN
               DO 5 J=1,NFNUM
                  IF (NOF(J).EQ.NO(K).AND.NWTYP(K).EQ.1) GO TO 10
5              CONTINUE
            ENDIF
            INU(NBL)=K
            NBL=NBL+1
         ENDIF
10    CONTINUE
      IF (NFNUM.GT.0) THEN
         DO 25 K=1,NB
            DO 20 J=1,NFNUM
               IF (NOF(J).EQ.NO(K).AND.NWTYP(K).EQ.1) THEN
                  RGRAD(J)=0.0
                  NFCW(J)=0
                  NFREG(J)=K
                  INU(NBU)=K
                  NBU=NBU-1
                  GO TO 25
               ENDIF
20          CONTINUE
25       CONTINUE
      ENDIF
C
C
      IF (NBU.GT.(NBL-1)) THEN
30       NRETU=0
         DO 40 K=1,NB
            IF (NWTYP(K).GE.2) THEN
               IF (NRETU.LE.0) THEN
                  NMIN=K
                  NRETU=1
               ENDIF
               IF (RQ(NMIN).GT.RQ(K)) NMIN=K
            ENDIF
40       CONTINUE
         INU(NBL)=NMIN
         NBL=NBL+1
         NWTYP(NMIN)=0
         IF ((NBU+1).GT.NBL) GO TO 30
      ENDIF
C
C       NWTYP(I)=10: AIRWAYS CONTAINING FIRE SOURCES.
C       MARKX=1: CALLED FROM DYNAMIC SIMULATION PART.
C
      IF (MARKX.GT.0) THEN
         DO 60 I=1,INFLOW
            DO 50 J=1,NB
               IF (NCENT(I).EQ.NO(J)) THEN
                  NWTYP(J)=10
                  GO TO 60
               ENDIF
50          CONTINUE
60       CONTINUE
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE BASE (NSTOP)
C
C
C     ---------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       FORMATION OF BASE SYSTEM.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C
      IND=INU(1)
      KJF(NJ)=JS(IND)
```

```
      KJS(NJ-1)=JS(IND)
      JS(IND)=-JS(IND)
      KJF(NJ-1)=JF(IND)
      KNO(NJ-1)=IND
      KNUM=NJ-1
10    DO 30 NUC=2,NB
        IND=INU(NUC)
        IF (JS(IND).LT.0) GO TO 30
        N1=0
        N0=0
        DO 20 K=KNUM,NJ
          IF (JS(IND).EQ.KJF(K)) N0=1
          IF (JF(IND).EQ.KJF(K)) N1=1
20      CONTINUE
        IF ((N1+N0).EQ.1) THEN
          IF (NWTYP(IND).LT.0) THEN
             WRITE (8,100) NO(IND)
             NWTYP(IND)=0
             IF (R(IND).LT.1.E-30) THEN
                WRITE (8,110)
                NSTOP=1
             ENDIF
          ENDIF
          KNUM=KNUM-1
          IF (N0.LE.0) THEN
            KJS(KNUM)=JF(IND)
            KJF(KNUM)=JS(IND)
            KNO(KNUM)=-IND
            JS(IND)=-JS(IND)
          ELSE
            KJS(KNUM)=JS(IND)
            KJF(KNUM)=JF(IND)
            KNO(KNUM)=IND
            JS(IND)=-JS(IND)
          ENDIF
          GO TO 10
        ELSE IF ((N1+N0).GT.1) THEN
          JS(IND)=-JS(IND)
          JF(IND)=-JF(IND)
        ENDIF
30    CONTINUE
      DO 40 K=1,NB
        JS(K)=IABS(JS(K))
40    CONTINUE
C
100   FORMAT (/,1X,'*ATTENTION*   REGULATOR ',I5,' HAS TO BE REGARDED ',
     .        'AS A NORMAL AIRWAY.')
110   FORMAT (1X,'* ERROR *  RESISTANCE OF AIRWAY ',I5,' MUST BE ',
     .         'SPECIFIED.')
C
      RETURN
      END
C
C
      SUBROUTINE MSLIST (NSTOP)
C
C
C     ---------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       SET-UP OF MESH LIST.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MESC=0
      MNO=0
      DO 60 K=1,NB
        IF (JF(K).LT.0) THEN
          JF(K)=-JF(K)
          MNO=MNO+1
          JBM=JS(K)
          JEM=JF(K)
          NK=K
10        MESC=MESC+1
          MEND(MNO)=MESC
          MSL(MESC)=NK
C
C
          DO 20 KCO=1,NJ
            IF (JBM.EQ.KJF(KCO)) THEN
              KB=KCO
              GO TO 30
            ENDIF
20        CONTINUE
30        DO 40 KCO=1,NJ
            IF (JEM.EQ.KJF(KCO)) THEN
              KE=KCO
              GO TO 50
            ENDIF
40        CONTINUE
50        IF (KB.LT.KE) THEN
            NK=KNO(KB)
            JBM=KJS(KB)
            GO TO 10
          ELSE IF (KB.GT.KE) THEN
            NK=-KNO(KE)
            JEM=KJS(KE)
            GO TO 10
          ENDIF
        ENDIF
60    CONTINUE
      IF (MEND(MNO).GT.LMX) THEN
         WRITE (8,100)
         NSTOP=1
      ENDIF
C
      RETURN
100   FORMAT (/,1X,'* ERROR *  NETWORK TOO LARGE, CAPACITY OF MSL ',
     .         'EXCEEDED.')
C
      END
C
C
      SUBROUTINE MBLNC
C
C
C     -----------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       REALIZATION OF BALANCE OF MASS FLOW RATES.
C
C     -----------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=2
      DO 20 K=1,MNO
         MENDW=MEND(K)
```

```
         DO 10 J=MBEGW,MENDW
            N=MSL(J)
            IF (N.LT.0) N=-N
            Q(N)=0.
10       CONTINUE
         MBEGW=MENDW+2
20    CONTINUE
C
C
      MBEGW=1
      DO 40 K=1,MNO
         MENDW=MEND(K)
         N=MSL(MBEGW)
         IF (N.LT.0) N=-N
         Q1=Q(N)
         M=MBEGW+1
         DO 30 J=M,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
            N=-N
            FACT=-1.
            ENDIF
            Q(N)=Q(N)+Q1*FACT
30       CONTINUE
         MBEGW=MENDW+1
40    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE NVP1
C
C     -------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF NVP THROUGH MANIPULATION OF INPUT JUNCTION DATA
C       IN TEMP. AND ELEVATION.
C
C     -------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=1
      DO 30 K=1,MNO
         IF (NVPN.GE.1) GO TO 30
         MENDW=MEND(K)
         TZ=0.0
         ZOT=0.0
C
         DO 20 J=MBEGW,MENDW
            N=MSL(J)
            NX=N
            IF (NX.LT.0) NX=-NX
            DO 10 L=1,NJ
               IF (JS(NX).EQ.JNO(L)) THEN
                  T0=T(L)+460.
                  Z0=Z(L)
               ELSE IF (JF(NX).EQ.JNO(L)) THEN
                  T1=T(L)+460.
                  Z1=Z(L)
               ENDIF
10          CONTINUE
            DZZ=(Z1-Z0)
            IF (N.LT.0) DZZ=-DZZ
            TZ=TZ+0.5*(T0+T1)*DZZ
            ZOT=ZOT+DZZ/(0.5*(T0+T1))
20       CONTINUE
C
         IF (ABS(ZOT).LE.0.000001) THEN
            TM=TR+460.0
         ELSE
            TM=ABS(SQRT(ABS(TZ/ZOT)))
         ENDIF
C
         IF (ABS(TM).LE.0.000001) TM=TR+460.0
         FNVP(K)=TZ*DR/(5.2*TM)
         MBEGW=MENDW+1
30    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE ITR (MARKX,NSFLOW,NSNVP,MADJC,ITCT)
C
C     -------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       APPLICATION OF HARDY CROSS METHOD TO NETWORK BALANCING.
C
C     -------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
5     IT=0
10    DQSUM=0.
      MBEGW=1
      DO 50 K=1,MNO
         MENDW=MEND(K)
         DPSUM=0.
         RQSUM=0.
         N=MSL(MBEGW)
         IF (N.LT.0) N=-N
         IF (NWTYP(N).EQ.(-1)) THEN
            MBEGW=MENDW+1
            GO TO 50
         ENDIF
         DO 30 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
            ENDIF
            IF(NWTYP(N).EQ.1) THEN
               IF (NFNUM.GT.0) THEN
                  DO 20 L=1,NFNUM
                     IF (NFREG(L).EQ.N) THEN
                        RQSUM=RQSUM-(RGRAD(L)*100000)
                        GO TO 25
                     ENDIF
20                CONTINUE
               ENDIF
25             IF (IABS(JS(N)).NE.JSB(N)) FACT=-FACT
               DPSUM=DPSUM-FACT*R(N)
            ELSE
               RQ2=R(N)*ABS(Q(N))*2.E-5
               RQSUM=RQSUM+RQ2
```

```
              DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
              DPSUM=DPSUM+FACT*DP
           ENDIF
30      CONTINUE
        IF (ABS(RQSUM).GT.1.E-10) THEN
           DQ=(DPSUM-FNVP(K))*1.E5/RQSUM
        ELSE
           DQ=0.0
        ENDIF
        DO 40 J=MBEGW,MENDW
           N=MSL(J)
           FACT=1.
           IF (N.LT.0) THEN
              N=-N
              FACT=-1.
           ENDIF
           Q(N)=Q(N)-(DQ*FACT)
40      CONTINUE
        DQSUM=DQSUM+ABS(DQ)
        MBEGW=MENDW+1
50    CONTINUE
C
      DO 100 KI=1,NB
        IF (NWTYP(KI).EQ.1.AND.NFNUM.GT.0) THEN
           DO 90 J=1,NFNUM
              IF (NOF(J).EQ.NO(KI)) THEN
                 NFCW(J)=0
                 NEGQ(J)=0
                 IF (Q(KI).LT.0.0.AND.IABS(JS(KI)).EQ.JSB(KI))
     .              NEGQ(J)=1
                 IF (Q(KI).GT.0.0.AND.IABS(JS(KI)).NE.JSB(KI))
     .              NEGQ(J)=1
                 IF (NVPN.GE.1.AND.MADJC.LE.0) THEN
                    TABF=TR
                 ELSE
                    NABF=JS(KI)
                    DO 70 L=1,NJ
                       IF (NABF.EQ.JNO(L)) THEN
                          TABF=T(L)
                          GO TO 80
                       ENDIF
70                  CONTINUE
                 ENDIF
80              IF (NEGQ(J).EQ.0) THEN
                    IF (NSWT(J).EQ.0) NSWT(J)=2
                    IF (NSKP(J).EQ.0) THEN
                       IF (NSWT(J).EQ.2) THEN
                          CALL SPLINE (J,Q(KI),R(KI),TABF,0)
                       ELSEIF (NSWT(J).EQ.1) THEN
                          CALL LSFAN (J,Q(KI),R(KI),TABF,0,0)
                       ENDIF
                       NSKP(J)=1
                    ELSE
                       IF (NSWT(J).EQ.2) THEN
                          CALL SPLINE (J,Q(KI),R(KI),TABF,1)
                       ELSEIF (NSWT(J).EQ.1) THEN
                          CALL LSFAN (J,Q(KI),R(KI),TABF,1,0)
                       ENDIF
                    ENDIF
                    GO TO 100
                 ELSE
                    R(KI)=PF(J,1)
                    R(KI)=R(KI)*(TR+460.)/(TABF+460.)
                    RGRAD(J)=0.0
                 ENDIF
                 ENDIF
90          CONTINUE
         ENDIF
100   CONTINUE
C
C
      IT=IT+1
         ITCT=ITCT+1
         IF (IT.LE.1) THEN
            GO TO 10
         ELSE IF ((DQSUM/FLOAT(MNO)).LT.10.) THEN
            RETURN
C         ELSE IF (ITCT.GT.80.AND.ITCT.LE.500) THEN
         ELSE IF (ITCT.GT.500) THEN
            NSFLOW=0
            WRITE (8,200) DQSUM/FLOAT(MNO)
            WRITE (8,205) MNO
            RETURN
         ELSE IF (IT.GT.50) THEN
C
            CALL ARR (MARKX)
            CALL BASE (NSTOP)
            CALL MSLIST (NSTOP)
            CALL MBLNC
C
C     NSNVP: 0: SUB. IS CALLED FROM NETWORK PART.  1: SUB. IS
C             CALLED FROM TEMP. OR DYNAMIC SIMULATION PART.
C
            IF (NSNVP.LE.0) THEN
               IF (NVPN.LE.0) CALL NVP1
            ELSE
               CALL NVP2
            ENDIF
            GO TO 5
         ENDIF
      GO TO 10
C
200   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 500'
     .        ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .        ' CURRENT ERROR:',F8.0)
205   FORMAT (//1X,'NUMBER OF MESHES=',I5)
210   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 100'
     .        ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .        ' CURRENT ERROR:',F8.0,/,' IT WAS HAPPENED IN DATA ',
     .        'PREPARATION CYCLE ',I5)
C
      END
C
C
      SUBROUTINE LSFAN (L,QKI,RKI,TABF,NTW,NEXP)
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       USING LEAST SQUARE METHOD FOR FAN CURVE FITTING.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
      DIMENSION ZZ(IMX),S(IMX),ALPHA(IMX),BETA(IMX),SIGMA2(IMX)
     .,PPP(IMX,IMX)
      DOUBLE PRECISION W(IMX)
C
```

```
      NFCW(L)=0
      NP=MPTS(L)
      IF (NTW.EQ.1) GO TO 145
      MARKXX=0
      DO 1 I=1,10
1     ALPHA(I)=0.0
      IF (NP.GT.2) GO TO 35
      ND(L)=1
      N1A=2
      NCOF(L)=N1A
      IF (NP.LT.1) THEN
         STOP 10
      ELSEIF (NP.EQ.1) THEN
         COF(L,1)=PF(L,1)
         COF(L,2)=0.
         GO TO 145
      ENDIF
      IF (QF(L,1).EQ.QF(L,2)) THEN
         STOP 25
      ELSE
         COF(L,1)=(PF(L,1)*QF(L,2)-PF(L,2)*QF(L,1))/(QF(L,2)-QF(L,1))
         COF(L,2)=(PF(L,2)-PF(L,1))/(QF(L,2)-QF(L,1))
         GO TO 145
      ENDIF
35    DSQ=0.0
      DO 40 J=1,NP
         DSQ=DSQ+PF(L,J)*PF(L,J)
         PPP(1,J)=0.
40       PPP(2,J)=1.
      W(1)=NP
      BETA(1)=0.
      N1A=4
      IF (NP.LT.5) N1A=NP-1
      IF (NEXP.GT.0) N1A=NEXP
      DO 85 I=1,N1A
         K=I+1
         ZZ(I)=0.
         DO 60 J=1,NP
60          ZZ(I)=ZZ(I)+PF(L,J)*PPP(K,J)
         S(I)=ZZ(I)/W(I)
         DSQ=DSQ-S(I)*S(I)*W(I)
         SIGMA2(I)=DSQ/FLOAT(NP-I)
         IF(SIGMA2(I).LE.1.D-6) GO TO 105
         IF (I.GE.N1A) GO TO 90
            ALPHA(I)=0.
            DO 75 J=1,NP
75          ALPHA(I)=ALPHA(I)+QF(L,J)*PPP(K,J)*PPP(K,J)
            ALPHA(I)=ALPHA(I)/W(I)
            W(K)=0.
            DO 80 J=1,NP
               PPP(K+1,J)=(QF(L,J)-ALPHA(I))*PPP(K,J)-BETA(I)*PPP(I,J)
80             W(K)=W(K)+PPP(K+1,J)*PPP(K+1,J)
85    BETA(K)=W(K)/W(I)
90    SMALL=SIGMA2(1)
      I=1
      DO 100 J=2,N1A
         IF (SMALL.LE.SIGMA2(J)) GO TO 100
         I=J
         SMALL=SIGMA2(J)
100   CONTINUE
105   WRITE (8,195) SIGMA2(I)
      N1A=I
      NCOF(L)=N1A
      N=N1A-1
      IF (N.LE.0) THEN
         ND(L)=1
         COF(L,1)=S(1)
         COF(L,2)=0.
         GO TO 145
      ENDIF
      ND(L)=N
      DO 125 I=1,N1A
         DO 120 J=1,I
120         PPP(I,J)=0.0
125      PPP(I,I+1)=1.
      DO 130 J=1,N
130      PPP(1,J+2)=-PPP(1,J+1)*ALPHA(J)-PPP(1,J)*BETA(J)
      DO 135 I=2,N
         DO 135 J=I,N
135         PPP(I,J+2)=PPP(I-1,J+1)-PPP(I,J+1)*ALPHA(J)-PPP(I,J)*BETA(J)
      DO 140 I=1,N1A
         COF(L,I)=0.
         DO 140 J=1,N1A
140   COF(L,I)=COF(L,I)+PPP(I,J+1)*S(J)
      NSKP(L)=1
      write (8,*) 'ENTER FAN CURVE FITTING'
145   IF (MARKXX.EQ.0) WRITE (8,200) (COF(L,I),I=1,N1A)
      IF (ABS(QKI).LT.1.D-6) GO TO 160
      MARKXX=1
      IF (NPLOT.EQ.1) THEN
C         IF (QKI.LT.QF(L,1).OR.QKI.GT.QF(L,NP)) NFCW(L)=1
C         FANQ=QKI
C         RGRAD(L)=0.0
C         II=1
C         GO TO 165
         IF (QKI.LE.QF(L,1)) THEN
            DQKI=0
            DQKI1=QKI-QF(L,1)
            FANQ=QKI
            RGRAD(L)=COF(L,2)+2*COF(L,3)*DQKI+3*COF(L,4)*DQKI**2
            RKI=PF(L,1)-DQKI1*RGRAD(L)
            RKI=RKI*(TR+460.)/(TABF+460.)
            NFCW(L)=1
            RETURN
         ELSEIF (QKI.GT.QF(L,NP)) THEN
C             DQKI=QF(L,NP)-QF(L,NP-1)
            DQKI=QF(L,NP)
            DQKI1=QKI-QF(L,NP)
            FANQ=QKI
            RGRAD(L)=COF(L,2)+2*COF(L,3)*DQKI+3*COF(L,4)*DQKI**2
            RKI=PF(L,NP)+DQKI1*RGRAD(L)
            RKI=RKI*(TR+460.)/(TABF+460.)
            NFCW(L)=1
            RETURN
         ELSE
            FANQ=QKI
            RGRAD(L)=0.0
            II=1
            GO TO 165
         ENDIF
      ELSE IF (NPLOT.EQ.2) THEN
         IF (QKI.LT.QF(L,NP)) GO TO 146
         RKI=PF(L,NP)
         RGRAD(L)=0.0
         NFCW(L)=1
         RETURN
146      IF (QKI.GT.QF(L,1)) GO TO 147
            DQKI=0
            DQKI1=QKI-QF(L,1)
            FANQ=QKI
```

```
            RGRAD(L)=COF(L,2)+2*COF(L,3)*DQKI+3*COF(L,4)*DQKI**2
            RKI=PF(L,1)+DQKI1*RGRAD(L)
            RKI=RKI*(TR+460.)/(TABF+460.)
            NFCW(L)=1
            RETURN

147      FANQ=QKI
         RGRAD(L)=0.0
         II=1
         GO TO 165
      ELSE IF (NPLOT.EQ.3) THEN
         IF (QKI.LT.QF(L,NP)) GO TO 150
         RKI=PF(L,NP)
         RGRAD(L)=0.0
         NFCW(L)=1
         RETURN
150      IF (QKI.GT.QF(L,1)) GO TO 155
         RKI=PF(L,1)
         RGRAD(L)=0.0
         NFCW(L)=1
         RETURN
155      FANQ=QKI
         RGRAD(L)=0.0
         II=1
         GO TO 165
      ENDIF
      GO TO 165
160   QCS=QF(L,NP)-QF(L,1)
      QCSA=QCS/120.
      RGRAD(L)=0.0
      Q0=QF(L,1)-5.*QCSA
      Q2(1)=Q0
      FANQ=Q0
      II=130
165   NE=ND(L)
      DO 180 I=1,II
         H2(I)=COF(L,NE+1)*FANQ
         IF (NE.LE.1) GO TO 175
         KT=NE
         DO 170 J=2,NE
            H2(I)=(H2(I)+COF(L,KT))*FANQ
            IF (MARKXX.EQ.1) RGRAD(L)=(RGRAD(L)+KT*COF(L,KT+1))*FANQ
170         KT=KT-1
175      H2(I)=H2(I)+COF(L,1)
         RGRAD(L)=RGRAD(L)+COF(L,2)
         IF (MARKXX.EQ.1) RKI=H2(1)
         Q2(I+1)=FANQ+QCSA
         FANQ=Q2(I+1)
180   CONTINUE
      IF (MARKXX.EQ.1) RETURN
      WRITE (6,190) (Q2(I),H2(I),I=1,130)
190   FORMAT (1X,5(F8.5,F6.2))
195   FORMAT (/18X,8HSIGMA2= ,1PE14.7)
200   FORMAT (1H0,17X,12HCOEFFICIENTS,//8X,1P6E18.7)
      RETURN
      END

C
      SUBROUTINE SPLINE (J,QKII,RKI,TABF,NTW)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) FAN CURVE FITTING.
C       2) FAN PRESSURE INTERPOLATION.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C     FKQ(J,i,1-4): COEFFS. OF 3RD ORDER POLYNOMIAL FOR FAN NOF(J),
C                   SEGMENT I.
C     NSKP(J)=1: MARKER INDICATING THAT FAN CURVE FITTING FOR FAN
C                NOF(J) HAS BEEN PROCESSED.
C
      MF = MPTS(J)
      IF (NTW.EQ.0) THEN
C
         DO 5 I=1,MF-1
            DK(I)=QF(J,I+1)-QF(J,I)
5        CONTINUE
C
         DO 10 I=2,MF-1
            FC3(I)=3.0*(PF(J,I+1)*DK(I-1)-PF(J,I)*(QF(J,I+1)-QF(J,I-1))
     .             +PF(J,I-1)*DK(I))/(DK(I-1)*DK(I))
10       CONTINUE
C
         FK(1)=1.0
         FC1(1)=0.0
         FC2(1)=0.0
C
         DO 15 I=2,MF-1
            FK(I)=2.0*(QF(J,I+1)-QF(J,I-1))-DK(I-1)*FC1(I-1)
            FC1(I)=DK(I)/FK(I)
            FC2(I)=(FC3(I)-DK(I-1)*FC2(I-1))/FK(I)
15       CONTINUE
C
         FK(MF-1)=1.0
         FC3(MF-1)=0.0
         FKQ(J,MF-1,3)=0.0
C
         DO 20 I=MF-1,1,-1
            FKQ(J,I,3)=FC2(I)-FC1(I)*FKQ(J,I+1,3)
            FKQ(J,I,2)=(PF(J,I+1)-PF(J,I))/DK(I)-
     .                 DK(I)*(FKQ(J,I+1,3)+2.0*FKQ(J,I,3))/3.0
            FKQ(J,I,4)=(FKQ(J,I+1,3)-FKQ(J,I,3))/(3.0*DK(I))
20       CONTINUE
C
         DO 25 I=1,MF-1
            FKQ(J,I,1)=PF(J,I)
25       CONTINUE
         NSKP(J)=1
      ENDIF
C
C
      QKI=ABS(QKII)
      IF (NPLOT.EQ.1) THEN
         IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
            dqki=0.0
            dqki1=qki-qf(j,1)
            m=1
            RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .                 3*FKQ(J,M,4)*DQKI**2
            rki=pf(j,1)+dqki1*rgrad(j)
            rki=rki*(tr+460.)/(tabf+460.)
            NFCW(J) = 1
            RETURN
         ELSE IF (QKI.GE.QF(J,MF)) THEN
            dqki=qf(j,mf)-qf(j,mf-1)
            dqki1=qki-qf(j,mf)
```

```
            m=mf-1
            rgrad(j) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .                 3*FKQ(J,M,4)*DQKI**2
            rki=pf(j,mf)+dqki1*rgrad(j)
            rki=rki*(tr+460.)/(tabf+460.)
            NFCW(J) = 1
            RETURN
         ENDIF
      ELSEIF (NPLOT.EQ.2) THEN
         IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
            dqki=0.0
            dqki1=qki-qf(j,1)
            m=1
            RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .                 3*FKQ(J,M,4)*DQKI**2
            rki=pf(j,1)+dqki1*rgrad(j)
            rki=rki*(tr+460.)/(tabf+460.)
            NFCW(J) = 1
            RETURN
         ELSE IF (QKI.GE.QF(J,MF)) THEN
            RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
            RGRAD(J) = 0.0
            NFCW(J) = 1
            RETURN
         ENDIF
      ELSEIF (NPLOT.EQ.3) THEN
         IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
            RKI = PF(J,1)*(TR+460.)/(TABF+460.)
            RGRAD(J) = 0.0
            NFCW(J) = 1
            RETURN
         ELSE IF (QKI.GE.QF(J,MF)) THEN
            RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
            RGRAD(J) = 0.0
            NFCW(J) = 1
            RETURN
         ENDIF
      ENDIF
      IF (QF(J,1).GE.QKI) THEN
         M = 1
      ELSE
         DO 40 K = 2,MF
            IF (QF(J,K).GE.QKI) THEN
               M = K-1
               GO TO 45
            ENDIF
40       CONTINUE
      ENDIF
      M = MF-1
45    DQKI=QKI-QF(J,M)
      RKI = FKQ(J,M,1)+FKQ(J,M,2)*DQKI+FKQ(J,M,3)*DQKI**2
     .         +FKQ(J,M,4)*DQKI**3
      RKI=RKI*(TR+460.)/(TABF+460.)
      RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .           3*FKQ(J,M,4)*DQKI**2
C
      RETURN
      END
C
C
      SUBROUTINE CHSFIT (NSTOP1)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) SELECT SUITABLE FAN CURVE FITTING METHODS FOR EACH FAN.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION HFN(IMX), QFN(IMX),NAUTO(IMX)
C
      DO 75 L=1,NFNUM
      IF (NSWT(L).LT.3) GO TO 75
         IF (NSWT(L).EQ.3) NAUTO(L)=NSWT(L)
         DFQ1=ABS(QF(L,2)-QF(L,1))
         DFP1=ABS(PF(L,2)-PF(L,1))
         DO 55 I=2,MPTS(L)-1
            DFQ2=ABS(QF(L,I)-QF(L,I-1))
            DFP2=ABS(PF(L,I)-PF(L,I-1))
            IF (DFQ1.LT.DFQ2) THEN
               DFQ1=DFQ2
            ELSEIF (DFP1.LT.DFP2) THEN
               DFP1=DFP2
            ENDIF
55       CONTINUE
         DQMAX=DFQ1
         DPMAX=DFP1
         DO 60 I=2,MPTS(L)-1
            DFQ2=ABS(QF(L,I+1)-QF(L,I))
            DFP2=ABS(PF(L,I+1)-PF(L,I))
            IF (DFQ1.GT.DFQ2) THEN
               DFQ1=DFQ2
            ENDIF
            IF (DFP1.GT.DFP2) THEN
               DFP1=DFP2
            ENDIF
60       CONTINUE
         DQMIN=DFQ1
         DPMIN=DFP1
         DIVQ=DQMAX/DQMIN
         DIVP=DPMAX/DPMIN
         IF (DIVQ.GT.4.AND.DIVP.GT.4) THEN
            NSWT(L)=1
            WRITE (8,975) L
            GO TO 75
         ENDIF
         INDEX=MPTS(L)
         IF (INDEX.LT.3) THEN
            WRITE (*,980) INDEX
            NSTOP1=1
            RETURN
         ELSE
            THSUM1=0.
            THSUM2=0.
            DO 444 K3=1,INDEX-1
               DQZZ=(QF(L,K3+1)-QF(L,K3))
               QFM=QF(L,K3)+DQZZ/2
               CALL SPLINE (L,QFM,HFM,TR,0)
               DIFH1=HFM-PF(L,K3)
               DIFH2=HFM-PF(L,K3+1)
               IF (PF(L,K3).GT.PF(L,K3+1)) THEN
                  IF (DIFH1.GT.0) THEN
                     THSUM1=THSUM1+DIFH1
                  ELSEIF (DIFH2.LT.0) THEN
                     THSUM1=THSUM1+ABS(DIFH2)
                  ENDIF
               ELSE
                  IF (DIFH2.GT.0) THEN
```

```
                THSUM1=THSUM1+DIFH2
             ELSEIF (DIFH1.LT.0) THEN
                THSUM1=THSUM1+ABS(DIFH1)
             ENDIF
          ENDIF
          CALL LSFAN (L,QFM,HFM,TR,0,0)
          DIFH3=HFM-PF(L,K3)
          DIFH4=HFM-PF(L,K3+1)
          IF (PF(L,K3).GT.PF(L,K3+1)) THEN
             IF (DIFH3.GT.0) THEN
                THSUM2=THSUM2+DIFH3
             ELSEIF (DIFH4.LT.0) THEN
                THSUM2=THSUM2+ABS(DIFH4)
             ENDIF
          ELSE
             IF (DIFH4.GT.0) THEN
                THSUM2=THSUM2+DIFH4
             ELSEIF (DIFH3.LT.0) THEN
                THSUM2=THSUM2+ABS(DIFH3)
             ENDIF
          ENDIF
444    CONTINUE
       ENDIF
       IF (ABS(THSUM1).GT.1.2*ABS(THSUM2)) THEN
          NSWT(L)=1
          WRITE (*,1000) L
       ELSE
          NSWT(L)=2
          WRITE (*,1005) L
       ENDIF
       IF (NAUTO(L).EQ.3) GO TO 75
       WRITE (*,1010) NOF(L)
       WRITE (*,1020)
       LL=MPTS(L)
       WRITE (*,1030) (QF(L,I)/1000.0,PF(L,I),I=1,LL)
       WRITE (*,1045) NOF(L)
          WRITE (*,1086)
          WRITE (*,1085)
          DO 441 K1=1,INDEX-1
             QZZ=(QF(L,K1+1)-QF(L,K1))/5.0
             DO 42 K2=1,5
                QZZ1=QF(L,K1)+(K2-1)*QZZ
                NFCWZ=NFCW(L)
                CALL SPLINE (L,QZZ1,HZZ,TR,1)
                NFCW(L)=NFCWZ
                HFN(K2)=HZZ
                QFN(K2)=QZZ1
42           CONTINUE
             WRITE (*,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
441       CONTINUE
          WRITE (*,1087)
          WRITE (*,1085)
          DO 445 K1=1,INDEX-1
             QZZ=(QF(L,K1+1)-QF(L,K1))/5.0
             DO 443 K2=1,5
                QZZ1=QF(L,K1)+(K2-1)*QZZ
                NFCWZ=NFCW(L)
                CALL LSFAN (L,QZZ1,HZZ,TR,1,0)
                NFCW(L)=NFCWZ
                HFN(K2)=HZZ
                QFN(K2)=QZZ1
443          CONTINUE
             WRITE (*,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
445       CONTINUE
       WRITE (*,1100)
          READ (*,*) NSWT(L)
75       CONTINUE
C
        RETURN
C
975   FORMAT (//,2X,'SOME DATA POINTS OF FAN',I5,2X,'CURVE ARE TOO CLOSE
     .',/,'SO THE LEAST SQUARE METHOD IS DESIRED')
980   FORMAT (//,2X,'THE DATA POINTS OF FAN CHARACTERISTIC ARE',I5,/,
     .'AT LEAST 3 DATA POINTS ARE DESIRED',/)
1000  FORMAT (//,2X,'RECOMMEND LEAST SQUARE METHOD FOR FAN',I5,/,
     .'SURVE FITTING',/)
1005  FORMAT (//,2X,'RECOMMEND SPLINE METHOD FOR FAN',I5,/,
     .'SURVE FITTING')
1010  FORMAT (//,'THESE CHARACTERISTICS WERE STORED FOR FAN IN AIRWAY'
     .,I5,/)
1020  FORMAT (//,5('  Q*1000   PF '),/)
1030  FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
1045  FORMAT (/,T5,'FAN',I5,/)
1085  FORMAT (/,2X,'INTERPOLATION:',/,5('  Q*1000   PF '),/)
1086  FORMAT (//,2X,'APPLYING SPLINE METHOD')
1087  FORMAT (//,2X,'APPLYING LEAST SQUARE METHOD')
1090  FORMAT (5(F8.1,F6.2))
1100  FORMAT (//,2X,'INPUT SELECT FAN CURVE FITTING METHOD',/,
     .'1-- FOR LEAST SQUARE METHOD, 2--FOR SPLINE METHOD')
C
      END
C
C
C
      SUBROUTINE RGLT (MARKY)
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       CALCULATION OF VENTILATION RESISTANCES OF REGULATORS REQUIRED
C     BY THE USER-SPECIFIED AIRFLOW RATES IN THE AIRWAYS.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=1
      DO 20 K=1,MNO
         DPSUM=0.
         MENDW=MEND(K)
         NX=MSL(MBEGW)
         IF (NX.LT.0) NX=-NX
         IF (NWTYP(NX).LT.0) THEN
            M=MBEGW+1
            DO 10 J=M,MENDW
               IF (MSL(J).GT.0) THEN
                  N=MSL(J)
                  FACT=1.
               ELSE
                  N=-MSL(J)
                  FACT=-1.
               ENDIF
               IF (NWTYP(N).EQ.1) THEN
                  DP=-R(N)
               ELSE
                  DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
               ENDIF
               DPSUM=DPSUM+DP*FACT
10          CONTINUE
            IF (NVPN.LE.0) DPSUM=DPSUM-FNVP(K)
```

```
            RSTD(NX) =-DPSUM*1.E10/(Q(NX))**2
            R(NX)=RSTD(NX)
            IF (R(NX).LT.0.0) THEN
               R(NX)=DPSUM
               RSTD(NX)=R(NX)
               NWTYP(NX)=1
               IF (MARKY.LE.0) WRITE (8,30) R(NX),NO(NX)
            ELSE
               NWTYP(NX)=0
               IF (MARKY.LE.0) WRITE (8,40) R(NX),NO(NX)
            ENDIF
         ENDIF
         MBEGW=MENDW+1
20    CONTINUE
C
30    FORMAT (//,1X,'* ATTENTION *  A CONSTANT PRESSURE FAN OF ',F6.3,
     .        ' IN.W.G.',/,1X,'IS INSTALLED IN PREVIOUSLY FIXED-Q ',
     .        'AIRWAY ',I5)
40    FORMAT (//,1X,'* ATTENTION *  RESISTANCE OF ',E10.3,' IN.W.G./',
     .        'CFM2',/,1X,'IS PROVIDED IN PREVIOUSLY FIXED-Q ',
     .        'AIRWAY ',I5)
C
      RETURN
      END
C
C
      SUBROUTINE CCDATA (MARKY,NSTOP)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) COMPLETION OF INPUT DATA.
C       2) EVALUATION OF ROCK TEMP. AND AIRWAY WETNESS.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION DAL(15),NTT(NMX),LA1(NMX)
      DOUBLE PRECISION X,LA1
C
C
      HAAVR=-1.0
      HKAVR=-1.0
      KFAVR=-1
      LAAVR=-1
      AAVR=-1.0
      OAVR=-1.0
      IF (NAV.GT.0) THEN
10       CALL READIN (DAL,6,ISTOP,0)
         IF (ISTOP.EQ.1) THEN
            WRITE (8,240)
            WRITE (8,250) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         IF (MARKY.EQ.0) WRITE (6,260)
         IF (DAL(15).LT.(-1.E20)) GO TO 10
         IF (NCOMTS.GT.NCOMT2) THEN
            NCOMT2=NCOMTS
            GO TO 10
         ENDIF
         IF (DAL(1).LT.0.0.OR.DAL(2).LT.0.0.OR.DAL(3).LT.0.0.OR.
     .        DAL(4).LT.0.0.OR.DAL(5).LT.0.0.OR.DAL(6).LT.0.0) THEN
            WRITE (8,200)
            WRITE (8,240)
            WRITE (8,250) (ROW(IE),IE=1,80)
            NSTOP=5
            RETURN
         ENDIF
         HAAVR=DAL(1)
         HKAVR=DAL(2)
         KFAVR=DAL(3)
         LAAVR=DAL(4)
         AAVR=DAL(5)
         OAVR=DAL(6)
      ENDIF
      IF (HAAVR.LT.1.E-5.OR.HAAVR.GT.10.0) HAAVR=0.1
      IF (HKAVR.LT.1.E-5.OR.HKAVR.GT.50.0) HKAVR=3.0
      IF (KFAVR.LT.10.OR.KFAVR.GT.1000) KFAVR=100
      IF (LAAVR.LT.10.OR.LAAVR.GT.3000) LAAVR=500
      IF (AAVR.LT.0.1.OR.AAVR.GT.500.0) AAVR=100.0
      IF (OAVR.LT.0.1.OR.OAVR.GT.100.0) OAVR=40.0
      IF (MARKY.LE.0) THEN
         IF (NAV.LE.0.AND.IOUT.LT.0) WRITE (8,220)
         WRITE (8,230) HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR
      ENDIF
C
C
      DO 50 I=1,NB
         IF (HA(I).LE.1.E-20.OR.HA(I).GT.50.0) HA(I)=HAAVR
         IF (HK(I).LE.1.E-20.OR.HK(I).GT.100.0) HK(I)=HKAVR
         IF (KF(I).LE.0.OR.KF(I).GT.2000.0) KF(I)=KFAVR
         IF (LA(I).LE.1.E-5.OR.LA(I).GT.100000) LA(I)=FLOAT(LAAVR)
         IF (A(I).LE.1.E-3.OR.A(I).GT.100000.0) A(I)=AAVR
         IF (O(I).LE.1.E-3.OR.O(I).GT.10000.0) O(I)=OAVR
50    CONTINUE
C
C
      IIT=0
      DO 80 J=1,NB
         IF (NO(J).LE.0) NO(J)=-NO(J)
         L=0
         M=0
         DO 60 I=1,NJ
            IF (JS(J).EQ.JNO(I)) THEN
               ES=Z(I)
               TRS=T(I)
               L=1
            ELSE IF (JF(J).EQ.JNO(I)) THEN
               EF=Z(I)
               TRF=T(I)
               M=1
            ENDIF
            IF (Q(J).LT.0.0) THEN
               STORE=TRF
               TRF=TRS
               TRS=STORE
            ENDIF
            IF ((L+M).EQ.2) THEN
               DZRD(J)=EF-ES
               IF (LA(J).LT.ABS(DZRD(J))) THEN
                  LA(J)=ABS(DZRD(J))
                  IF (MARKY.LE.0) WRITE (8,270) NO(J),LA(J)
                  IF (LA(J).GT.5000..AND.MARKY.LE.0) WRITE(8,280) NO(J)
               ENDIF
               IF (JF(J).EQ.JSTART) TRF=TRS
               IF (TROCK(J).GT.150.0.AND.MARKY.LE.0)
     .            WRITE (8,300) TROCK(J),NO(J)
               IF (TROCK(J).LT.1.E-3.OR.TROCK(J).GT.150.0) THEN
```

```
                X=0.014*O(J)/(A(J)**0.8*(ABS(Q(J)))**0.2)
                LA1(J)=LA(J)
                XYZ=DEXP(-X*LA1(J))
                TROCK(J)=(TRF-TRS*XYZ+DZRD(J)*
     .             (1.+XYZ)/(2*187.))/(1.0-XYZ)
             ENDIF
             GO TO 70
          ENDIF
60       CONTINUE
70       IF (TROCK(J).GT.150.0.OR.TROCK(J).LT.0.0) THEN
            IIT=IIT+1
            NTT(IIT)=NO(J)
            TROCK(J)=TRF
         ENDIF
80    CONTINUE
      IF (IIT.GT.0.AND.(MARKY.LE.0.OR.IOUT.LE.(-2))) THEN
         WRITE (8,290) (NTT(JJ),JJ=1,IIT)
         WRITE (8,295)
      ENDIF
      RETURN
C
200   FORMAT (/,1X,'* ERROR *  ILLEGAL NEGATIVE VALUE IN AVE.-VALUE',
     .        ' CARD DETECTED.')
220   FORMAT (//,1X,'AVE.-VALUE CARD WAS OMITTED BY USER (NAV=0)')
230   FORMAT (///,1X,'THE FOLLOWING AVERAGE VALUES WILL BE USED TO ',
     .       'FILL ANY MISSING DATA:',//,1X,'THERMAL DIFFUSIVITY OF',
     .       ' ROCK',T30,F9.3,T43,'FT2/HR',/,1X,'ROCK THERMAL ',
     .       'CONDUCTIVITY',T30,F9.3,T43,'BTU/(HR*FT*F)',/,1X,
     .       'FRICTION FACTOR',T30,I9,T43,'1.E-10 LBF*MIN2/FT4',/,1X,
     .       'AIRWAY LENGTH',T30,I9,T43,'FT',/,1X,'SECTIONAL AREA ',
     .       T30,F9.3,T43,'FT2',/,1X,'PERIMETER',T30,F9.3,T43,'FT',/)
240   FORMAT (/,' THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ ',
     .       'IN:',//,' HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR')
250   FORMAT (/,' THE CURRENT INPUT WAS:',//,80A1)
260   FORMAT (/,5X,'AVERAGE-VALUE CARD WAS READ IN.')
270   FORMAT (/,1X,'* ATTENTION *  THE LENGTH OF AIRWAY',I5,' WAS LESS'
     .       ,' THAN ITS ELEVATION',/,' DIFFERENCE. THEY ARE NOW ',
     .       'TAKEN EQUAL TO EACH OTHER AS: ',F9.1,' FT')
280   FORMAT (/,1X,'* WARNING *  THE ELEVATION DIFFERENCE OF AIRWAY',
     .       I5,' IS UNLIKELY LARGE.')
290   FORMAT (/,1X,'* ATTENTION *  UNLIKELY VALUES OF ROCK TEMPERATURE',
     .       ' WERE OBTAINED IN THE',/,' FOLLOWING  AIRWAY(S) BASED ON',
     .       ' USER SPECIFIED HEAT TRANSFER PARAMETERS.',/,10(12I6,/))
295   FORMAT (' THEY WERE MODEFIED AND SHOWN IN FOLLOWING TABLE(S).')
300   FORMAT (/,1X,'* ATTENTION *  THE UNLIKELY INPUT VALUE OF ROCK ',
     .       ' TEMPERATURE (',F6.0,' F)',/,' FOR AIRWAY',I5,' IS ',
     .       'IGNORED.')
C
      END
C
      SUBROUTINE  FWCT (NSFLOW,JRE)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) DATA REARRANGEMENT FOR AIRWAYS IN WHICH AIRFLOW REVERSAL
C          HAPPENED.
C       2) FORMATION OF AIRFLOW SCHEME IN THE NETWORK.
C       3) DETECTION OF POSSIBLE AIRFLOW RECIRCULATION PATHS.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
      DIMENSION LREC(NMX),IINU(NMX)
      DATA LREC,IINU / NMX*0,NMX*0 /
C
      IF (JRE.GE.1) GO TO 85
      DO 20 I=1,NB
         IF (Q(I).LT.0.0) THEN
            NSFLOW=0
            J=JS(I)
            JS(I)=JF(I)
            JF(I)=J
            Q(I)=-Q(I)
            DZRD(I)=-DZRD(I)
C
C     NNREV(K): NUMBER OF AIRFLOW DIRECTION CHANGES IN AIRWAY NO(K).
C
            IF (NWTYP(I).EQ.1) R(I)=-R(I)
            NNREV(I)=NNREV(I)+1
         ENDIF
20    CONTINUE
C
C

      L=0
      M=0
      N=1
      DO 60 I=1,MAXJ
         K=L
         DO 40 J=1,NB
            IF (JS(J).EQ.I) THEN
               L=L+1
               NGOUT(L)=J
            ENDIF
40       CONTINUE
         LOUT(N)=L
         MM=M
         DO 50 J=1,NB
            IF (JF(J).EQ.I) THEN
               M=M+1
               NGIN(M)=J
            ENDIF
50       CONTINUE
         MMIN(N)=M
         IF ((MM.NE.M).OR.(K.NE.L)) THEN
            JNOL(N)=I
            N=N+1
         ENDIF
60    CONTINUE
      DO 80 I=1,NJ
C
C     JNOL(J): LIST OF JUNCTION NUMBERS IN INCREASING ORDER.
C     JLR(J): BRIDGE ARRAY RELATING JNO AND JNOL LISTS.
C
         DO 70 J=1,NJ
            IF (JNOL(I).EQ.JNO(J)) THEN
               JLR(I)=J
               GO TO 80
            ENDIF
70       CONTINUE
80    CONTINUE
C
C
85    IF (JRE.EQ.1) THEN
         DO 86 I=1,NJ
            IF (JNOL(I).EQ.JSTART) JSTOUT=I
86       CONTINUE
         DO 90 I=1,NMX
            LREC(I)=0
90       CONTINUE
```

```
         DO 92 I=1,NXX
            KJF(I)=0
            KJS(I)=0
            KNO(I)=0
92       CONTINUE
         KREC=0
         N1=JSTOUT
         DO 95 I=1,NJ
            IF (JNO(I).EQ.JSTART) THEN
               JNO(I)=-JNO(I)
               GO TO 100
            ENDIF
95       CONTINUE
100      IF (N1.LE.1) THEN
            N2=1
         ELSE
            N2=LOUT(N1-1)+1
         ENDIF
         N3=LOUT(N1)
         DO 110 I1=N2,N3
            I2=NGOUT(I1)
            LREC(I2)=100
110      CONTINUE
115      JREC=0
         DO 130 I1=1,NJ
            IF (JNO(I1).GT.0) THEN
               DO 117 I9=1,NJ
                  IF (JNOL(I9).EQ.JNO(I1)) GO TO 118
117            CONTINUE
               WRITE (6,*) 'SHOULD NOT OCCUR (117)'
118            IF (I9.LE.1) THEN
                  L1=1
               ELSE
                  L1=MMIN(I9-1)+1
               ENDIF
               L2=MMIN(I9)
               KREADY=0
               DO 120 I3=L1,L2
                  I4=NGIN(I3)
                  IF (LREC(I4).LE.0) THEN
                     KREADY=1
                     JREC=I9
                  ENDIF
120            CONTINUE
               IF (KREADY.EQ.0) THEN
                  JNO(I1)=-JNO(I1)
                  N1=I9
                  GO TO 100
               ENDIF
            ENDIF
130      CONTINUE
         N1=JSTOUT
1000     IF (N1.LE.1) THEN
            N2=1
         ELSE
            N2=MMIN(N1-1)+1
         ENDIF
         N3=MMIN(N1)
         DO 1100 I1=N2,N3
            I2=NGIN(I1)
            LREC(I2)=100
1100     CONTINUE
1150     JREC=0
         DO 1300 I1=1,NJ
            IF (JNO(I1).GT.0) THEN
               DO 1170 I9=1,NJ
                  IF (JNOL(I9).EQ.JNO(I1)) GO TO 1180
1170           CONTINUE
               WRITE (6,*) 'SHOULD NOT OCCUR (1170)'
1180           IF (I9.LE.1) THEN
                  L1=1
               ELSE
                  L1=LOUT(I9-1)+1
               ENDIF
               L2=LOUT(I9)
               KREADY=0
               DO 1200 I3=L1,L2
                  I4=NGOUT(I3)
                  IF (LREC(I4).LE.0) THEN
                     KREADY=1
                     JREC=I9
                  ENDIF
1200           CONTINUE
               IF (KREADY.EQ.0) THEN
                  JNO(I1)=-JNO(I1)
                  N1=I9
                  GO TO 1000
               ENDIF
            ENDIF
1300     CONTINUE
C
C
         IF (JREC.GT.0) THEN
            KJSSX=0
            KJSSY=0
            KJSS=0
            N1=JREC
140         IF (N1.LE.1) THEN
               N2=1
            ELSE
               N2=LOUT(N1-1)+1
            ENDIF
            N3=LOUT(N1)
            NREADY=0
142         DO 170 I1=N2,N3
               I2=NGOUT(I1)
               IF (LREC(I2).EQ.NREADY) THEN
                  KJSS=KJSS+1
                  KJS(KJSS)=I2
                  IF (KJSS.GE.2) THEN
                  DO 160 I3=1,KJSS-1
                     J1=KJS(I3)
                     IF (JS(J1).EQ.JF(I2)) THEN
                        KREC=KREC+1
                        JUDGE=0
                        DO 143 I4=I3,KJSS
                           II1=KJS(I4)
                           IF (LREC(II1).LE.0) JUDGE=1
143                     CONTINUE
                        IF (JUDGE.EQ.0) KREC=KREC-1
                        DO 145 I4=I3,KJSS
                           IF (JUDGE.EQ.0) THEN
                              II1=KJS(I4)
                              LREC(II1)=100
                           ELSE
                              KJSSX=KJSSX+1
                              KJF(KJSSX)=KJS(I4)
                              II1=KJS(I4)
                              LREC(II1)=1
                           ENDIF
```

```
145                  CONTINUE
C
C
C
                     IF (JUDGE.GE.1) KNO(KREC)=KJSSX
                     DO 150 I4=1,KJSS
                        KJS(I4)=0
150                  CONTINUE
                     KJSS=0
                     DO 155 I4=1,NB
                        IF (KJSSY.GT.0) THEN
                           DO 154 II4=1,KJSSY
                              IF (I4.EQ.IINU(II4)) GO TO 155
154                        CONTINUE
                        ENDIF
                        IF (LREC(I4).LE.0) THEN
                           JJ1=JS(I4)
                           DO 153 I5=1,NJ
                           IF (JJ1.EQ.JNO(I5)) THEN
                              KJSSY=KJSSY+1
                              IINU(KJSSY)=I4
                              N1=I5
                              GO TO 140
                           ENDIF
153                        CONTINUE
                        ENDIF
155                  CONTINUE
                     GO TO 175
                  ENDIF
160            CONTINUE
               ENDIF
               DO 165 N1=1,NJ
                  IF (JNOL(N1).EQ.JF(I2)) GO TO 140
165            CONTINUE
               WRITE (6,*) 'SHOULD NOT OCCUR (165)'
            ENDIF
170      CONTINUE
         IF (NREADY.EQ.0) THEN
            NREADY=1
            GO TO 142
         ENDIF
      ENDIF
175   IF (KREC.GT.0) THEN
         WRITE (8,200)
         I2=1
         DO 190 I1=1,KREC
            I3=KNO(I1)
            K1=0
            DO 180 I4=I2,I3
               I5=KJF(I4)
               K1=K1+1
               IINU(K1)=NO(I5)
               K1=K1+1
               IINU(K1)=JF(I5)
180         CONTINUE
            WRITE (8,210) I1,(IINU(K2),IINU(K2+1),K2=1,K1,2)
            I2=I3+1
190      CONTINUE
      ELSE
         WRITE (8,220)
      ENDIF
      DO 195 I=1,NMX
         IINU(I)=0
         LREC(I)=0
195   CONTINUE
         DO 197 I=1,NXX
            KNO(I)=0
            KJS(I)=0
            KJF(I)=0
197      CONTINUE
         DO 198 I=1,NJ
            IF (JNO(I).LT.0) JNO(I)=-JNO(I)
198      CONTINUE
      ENDIF
C
200   FORMAT (//,1X,'THE FOLLOWING AIR RECIRCULATION PATH(S) WERE DETE',
     .       'CTED:',/)
210   FORMAT (/,1X,'RECIRCULATION PATH ',I3,'  "AIRWAY -> (JUNCTION)->"'
     .       ,50(/,1X,4(I5,'->(',I3,')->')))
220   FORMAT (//,1X,'NO AIR RECIRCULATION WAS DETECTED IN THE SYSTEM.')
      RETURN
C
      END
C
C
      SUBROUTINE  CH4EVA
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF METHANE EMISSION RATES IN AIRWAYS.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C     CH4C(J): CH4 CONCENTRATION IN JUNCTION JNO(J).
C     CH4V(I): CH4 EMISSION RATES IN AIRWAY NO(I).
C     CH4PA(I): CH4 EMISSION RATE PER UNIT SURFACE AREA.
C
      DO 40 I=1,NB
         IF (CH4V(I).GT.0.0) GO TO 40
         CH4V(I)=CH4PA(I)*LA(I)*O(I)
         IF (CH4V(I).GT.0.0) GO TO 40
         M=0
         N=0
         DO 30 L=1,NJ
            IF (JS(I).EQ.JNO(L)) THEN
               CH4S=CH4C(L)
               M=1
            ENDIF
            IF (JF(I).EQ.JNO(L)) THEN
               CH4F=CH4C(L)
               N=1
            ENDIF
            IF (M+N.GT.1) THEN
               IF (CH4F.GT.CH4S.AND.Q(I).GT.0.0) THEN
                  CH4V(I)=(CH4F-CH4S)*Q(I)/100.0
               ELSE IF (CH4F.LT.CH4S.AND.Q(I).LT.0.0) THEN
                  CH4V(I)=(CH4F-CH4S)*Q(I)/100.0
               ELSE
                  CH4V(I)=0.0
               ENDIF
               GO TO 40
            ENDIF
30       CONTINUE
40    CONTINUE
C
      RETURN
      END
```

```
C
C
      SUBROUTINE  CDENDS (MARKY,NSTART,L,NSTOP)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF AIR TEMP. AND CH4/FUME CONTENT AT AIRWAY ENDING.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      PROPJS=PROP(L)
      JX=L
      CH4JS=PRCH4(L)
      JNO(L)=-JNO(L)
      JY=1
      IF (NSTART.NE.1) JY=LOUT(NSTART-1)+1
      JZ=LOUT(NSTART)
      DO 60 K=JY,JZ
         ICFTM=0
         I=NGOUT(K)
         NM=0
         CONTAM=0.0
         CONTQ=0.0
         HEATAD=0.0
         TFS=0.0
C
         IF (MARKY.EQ.0.AND.NTEMP.NE.1) GO TO 30
            DO 20 N=1,INFLOW
               IF (NCENT(N).EQ.NO(I)) THEN
                  NM=1
                  FACT=1.0
                  IF (QCENT(N).GT.10.0) THEN
                     Q1=Q(I)
                     IF (Q1.LT.QCENT(N)) Q1=Q1+QCENT(N)
                     FACT=1.0+(Q(I)-QCENT(N))/Q1
                  ENDIF
                  CONTAM=CONT(N)*CONC(N)/100.
                  CONTQ=CONT(N)
                  O2BEH(N)=(0.21-PROPJS)*100.0
                  HEATAD=HEAT(N)*FACT
C
                  IF (O2MIN(N).GT.0.0.AND.HEAT(N).GE.0.0) THEN
                     O2SCL=O2MIN(N)+(21.0-O2MIN(N))*(FACT-1.0)
                     CONTAM=(0.21-PROPJS-O2SCL/100.)*Q(I)
                     O2SCLA=(0.21-PROPJS)*100.0
                     O2BEH(N)=MIN(O2SCL,O2SCLA)
                     IF (CONTAM.GT.0.0) THEN
                        HEATAD=CONTAM*437.
                     ELSE
                        HEATAD=0.0
                     ENDIF
                  ENDIF
                  IF (SMPO2(N).LE.0.0.OR.HEAT(N).LT.0.0) GO TO 30
C
                  CONTAM=(0.21-PROPJS)*Q(I)*SMPO2(N)
                  O2BEH(N)=0.0
                  IF (FACT.LE.1.0) THEN
                     HTSCL=HTPO2(N)
                  ELSE
                     HTSCL=HTPO2(N)/FACT
                  ENDIF
                  IF (CONTAM.GT.0.0) THEN
                     HEATAD=(0.21-PROPJS)*Q(I)*HTSCL
                  ELSE
                     HEATAD=0.0
                  ENDIF
                  GO TO 30
               ENDIF
20          CONTINUE
C
C
30       RDCH4(I)=(CH4JS*Q(I)+CH4V(I))/(Q(I)+CH4V(I))
C         CONTA=CONTAM
C         IF (CONTA.LT.0.0) CONTA=0.0
C         RDPROP(I)=(PROPJS*Q(I)+CONTA)/(Q(I)+CONTQ)
          RDPROP(I)=PROPJS*(Q(I)-CONTQ)/Q(I)+CONTAM/Q(I)
         TJS(I)=T(JX)
         IF (NM.GT.0) THEN
            HTAD(N)=HEATAD
            IF (HEATAD.NE.0.0) THEN
               VART=(9900.0+T(JX))**2+2.0*HEATAD/(Q(I)*DR*2.4E-5)
               IF (VART.LT.9.4E7) THEN
                  WRITE (8,110) NO(I),Q(I)
                  NSTOP=6
                  RETURN
               ENDIF
               TFS=-9900.0+SQRT(VART)
               TJS(I)=TFS
               TFSI(N)=TFS
            ELSE
               TFSI(N)=TJS(I)
            ENDIF
            IF (TJS(I).LT.(-200.0).OR.TJS(I).GT.5000.0) THEN
               WRITE (8,100) NO(I),Q(I),TJS(I)
               NSTOP=6
               RETURN
            ENDIF
         ENDIF
C
C         AN ITERATION PROCESS IS EMPLOYED FOR MEAN AIR TEMPERATURE.
C     TM: MEAN AIR TEMP. IN AIRWAY.
C
         IF (ICFTM.GT.0) GO TO 40
         TM=(TJS(I)+TROCK(I))/2.
         GO TO 50
40       ARGMT=(TJS(I)-TROCK(I))/(TRD(I)-TROCK(I))
         IF (ARGMT.LE.1.) THEN
            WRITE (8,*) 'ERROR IN SUB CDENDS'
            TM=(TJS(I)+TRD(I))/2.
         ELSE
            X=ALOG(ARGMT)/LA(I)
            TM=TROCK(I)+(TJS(I)-TROCK(I))*(1.0-EXP(-X*LA(I)))/(X*LA(I))
         ENDIF
50       CALL KALPHA (TM,FX,CP,I)
C
C     COAGE: COEFF. OF AGE.
C     XNEW(I): EXPONENTIAL FACTOR IN COMPUTING AIR TEMP. VARIATION
C              ALONG AIRWAY NO(I).
C
         COAGE=BI(I)-FX*BI(I)**2/(0.375+BI(I))
         DCOAGE(I)=COAGE
         XNEW(I)=HK(I)*O(I)**2.*COAGE/(120.*DR*Q(I)*A(I))
         G=XNEW(I)*LA(I)/CP
         IF (G.GE.25.0) THEN
            GX=0.0
         ELSE
            GX=EXP(-G)
```

```
         ENDIF
         IF (ICFTM.LE.0) THEN
            TOLD=TROCK(I)
         ELSE
            TOLD=TRD(I)
         ENDIF
         TRD(I)=TROCK(I)+(TJS(I)-TROCK(I))*GX-
     .           (DZRD(I)/(2.*778.26*CP))*(1.+GX)
         TDM=ABS(TOLD-TRD(I))
         IF (TDM.GT.50.) THEN
            ICFTM=1
            GO TO 40
         ENDIF
         IF (JF(I).GT.0) JF(I)=-JF(I)
60    CONTINUE
C
      RETURN
100   FORMAT (/,1X,'* ERROR *  ABNORMAL TEMPERATURE IN HEAT SOURCE ',
     .        I5,/,' CURRENT AIRFLOW: ',F8.0,' TEMPERATURE: ',
     .        F8.1,/,' HEAT INPUT AND/OR QCENT MUST BE ADJUSTED.')
110   FORMAT (/,1X,'* ERROR *  ABNORMAL TEMPERATURE IN HEAT SOURCE ',
     .        I5,/,' CURRENT AIRFLOW: ',F8.0,' TEMPERATURE: < -200.0',
     .        /,' HEAT INPUT AND/OR QCENT MUST BE ADJUSTED.')
      END
C
C
      SUBROUTINE  KALPHA (TM,FX,CP,I)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF "COEFF. OF AGE" ASSUMING DRY CASE.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      PI=3.1415936
      CP=0.24
      TMX=TM
      QX=Q(I)
      TX=TIME
C
C     WT: TEMP. CORRECTED AIR DENSITY, LBM/FT**3.
C     VISC: KINETIC VISCOSITY OF AIR, FT/HR.
C     HKA: THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F.
C     FO: FOURIER NUMBER.  BI: BIOT NUMBER.  RN: REYNOULD NUMBER.
C
      WT=DR*(TR+460.)/(TMX+460.)
      VISC=0.000145*((460.+TMX)/492.)**1.75
      HKA=0.014*((460.+TMX)/492.)**0.81
      RN=QX*DR/(15.*WT*O(I)*VISC)
      FR0=0.0032+0.221/RN**0.237
      POT=(100./RN)**0.125
      COR=((KF(I)*0.075)/(809*DR*FR0))**POT
      HC=0.005*HKA*O(I)*RN**0.8*COR/A(I)
      FO=TX*HA(I)*O(I)**2/(4.*A(I)**2)
      BI(I)=HC*2.*A(I)/(O(I)*HK(I))
      X=(0.375+BI(I))*SQRT(FO)
C
C
      IF (X.LT.2.5) THEN
         N=0
         SUMT=0.0
         ADDT=X
10       SUMT=SUMT+ADDT
         N=N+1
         ADDT=-ADDT*X**2*(2.*N-1)/(N*(2.*N+1))
         IF (ABS(ADDT).GE.0.00001) GO TO 10
         FX=1.0-(EXP(X**2))*(1.0-(2.0/SQRT(PI))*SUMT)
      ELSE
         N=0
         SUMT=0.0
         ADDT=1.0
20       SUMT=SUMT+ADDT
         N=N+1
         OLADDT=ADDT
         ADDT=-ADDT*(2*N-1)/(2.0*X**2)
         IF (ABS(ADDT).LT.0.00001) GO TO 30
         IF (ABS(OLADDT).GT.ABS(ADDT)) GO TO 20
30       FX=1.0-1.0*SUMT/(SQRT(PI)*X)
      ENDIF
C
      RETURN
      END
C
C
C
      SUBROUTINE  CDJUNC (NSTART,JUMPBK,LL,KS)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF AIR TEMP. AND CONTENT OF CONTAMINANTS IN JUNCTIONS
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C     KS=0: THE SUB. IS CALLED FOR FROM THE MAIN PROGRAM.
C     KS=1: THE SUB. IS CALLED FOR FROM SUB. RECIRC.
C
      DO 20 I=1,NJ
         L=JLR(I)
         IF (JNO(L).LT.0) GO TO 20
         IF (KS.EQ.1.AND.JNO(LL).NE.JNOL(I)) GO TO 20
         JY=1
         IF (I.GT.1) JY=MMIN(I-1)+1
         JZ=MMIN(I)
         SUMAIR=0.0
         SUMPR=0.0
         SUMCH4=0.0
         SUMHT=0.0
         DO 10 K=JY,JZ
            J=NGIN(K)
            IF (JF(J).GE.0.AND.KS.EQ.0) GO TO 20
            SUMAIR=SUMAIR+Q(J)
            SUMPR=SUMPR+RDPROP(J)*Q(J)
            SUMCH4=SUMCH4+RDCH4(J)*Q(J)
            SUMHT=SUMHT+TRD(J)*Q(J)*(0.2376+1.2E-5*TRD(J))
10       CONTINUE
C
C       IF DATA PREPARATION FOR ANY ONE OF THE JUNCTIONS IN THE
C     SYSTEM IS NOT COMPLETE, THE PROCESS OF DATA PREPARATION WILL
C     CONTINUE (JUMPBK=1).
C
         NSTART=I
         IF (SUMAIR.GT.1.E-5) THEN
            PROP(L)=SUMPR/SUMAIR
            PRCH4(L)=SUMCH4/SUMAIR
```

```
            RTCONT=9.801E7+2.0*SUMHT/(SUMAIR*2.4E-5)
            T(L)=-9900.+SQRT(RTCONT)
         ELSE
            WRITE (8,*) 'SUMQ AT JNUCTION',JNO(L),' DROPS TO ZERO.'
         ENDIF
         LL=L
         IF (JAN.LE.1) GO TO 15
         DO 12 II=1,JAJ
            IF (JNO(L).EQ.JSTAR(II) ) THEN
               T(L)=TSTAR(II)
               PROP(L)=0.0
               PRCH4(L)=0.0
            ENDIF
12       CONTINUE
15       JUMPBK=1
         RETURN
20    CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE  RECIRC (MRC,JUMPBK,NSTART,L)
C
C     ------------------------------------------------------------------

C       SUBROUTINE PURPOSE:
C       EVALUATION OF AIR TEMP., CONTAMINANT CONTENT AT AIRWAY ENDINGS
C     UNDER RECIRCULATION CONDITIONS.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      NREC=1
10    LJ=0
      DO 60 I=1,NJ
         IF (JNO(I).LE.0) THEN
C
C       JUNCTION JNO(I) IS COMPLETED WITH THE NECESSARY CALCULATIONS
C     IF JNO(I)<0.
C
            LJ=LJ+1
         ELSE
            N=0
            M=0
            SRPR=0.0
            SRCH4=0.0
            STRD=0.0
            QIN=0.0
            QREC=0.0
            DO 20 J=1,NB
               IF (JNO(I).EQ.(-JF(J))) THEN
                  N=N+1
                  SRPR=SRPR+RDPROP(J)*Q(J)
                  SRCH4=SRCH4+RDCH4(J)
                  STRD=STRD+TRD(J)
                  QIN=QIN+Q(J)
               ELSE IF (JNO(I).EQ.JF(J)) THEN
C
C       JF(J)>0 INDICATES THAT THE CALCULATION FOR AIRWAY NO(J) HAS
C     NOT BEEN COMPLETED.
C     MEMREC: TEMPORARY LIST OF AIRWAYS CARRYING RECIRC. AIR.
C
                  M=M+1
                  MEMREC(M)=J
                  QREC=QREC+Q(J)
               ENDIF
20          CONTINUE
            IF (N.LE.0) GO TO 60
            AVRPR=SRPR/QIN
            AVRCH4=SRCH4/N
            AVTRD=STRD/N
C
C       NREC SERVES AS FILTER TO FIND OUT THE JUNCTION WHICH HAS THE
C     LEAST NON-ZERO CIRCULATING TO UNCIRCULATING AIRFLOW RATIO.
C
            IF ((QREC*2.0/QIN).LE.NREC) THEN
               DO 30 L=1,M
                  MRC=MRC+1
                  K=MEMREC(L)
                  NOREC(MRC)=K
                  ESTPR(MRC)=AVRPR
                  RDPROP(K)=ESTPR(MRC)
                  ESTCH4(MRC)=AVRCH4
                  RDCH4(K)=ESTCH4(MRC)
                  ESTTR(MRC)=AVTRD
                  TRD(K)=ESTTR(MRC)
                  JF(K)=-JF(K)
30             CONTINUE
               L=I
               CALL CDJUNC (NSTART,JUMPBK,L,1)
C               CALL CDJUNC (NSTART,JUMPBK,L,I,1)
               RETURN
            ENDIF
         ENDIF
60    CONTINUE
      IF (LJ.LT.NJ) THEN
         NREC=NREC+1
         GO TO 10
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE  PREP (MRC,LLST)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       PREPARATION OF CALCULATED INTERMEDIATE RESULTS AS THE INITIAL
C     VALUES FOR THE NEXT ROUND OF ITERATIONS IN THE CASE WHEN
C     RECIRCULATION HAPPENS.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DO 10 I=1,NB
         JF(I)=IABS(JF(I))
10    CONTINUE
      DO 20 I=1,NJ
         JNO(I)=IABS(JNO(I))
20    CONTINUE
      IF (MRC.EQ.0) RETURN
      DO 30 I=1,MRC
         K=NOREC(I)
         DIFPR=ESTPR(I)-RDPROP(K)
```

```
          DIFCH4=ESTCH4(I)-RDCH4(K)
          DIFTRD=ESTTR(I)-TRD(K)
C
C         WHEN THE VARIATIONS IN TEMP. OR CONTAMINANT CONTENT ARE LARGER
C       THAN THE USER-SPECIFIED CRITERIA (LLST>0), MORE ITERATIONS WILL BE
C       PERFORMED.
C
          IF (ABS(DIFPR).GE.(CRITSM/100.)) LLST=LLST+1
          IF (ABS(DIFCH4).GE.(CRITGS/100.)) LLST=LLST+1
          IF (ABS(DIFTRD).GE.(CRITHT)) LLST=LLST+1
C
C         CALCULATED AIR TEMP., CH4 AND FUME CONTENT AT AIRWAY ENDING
C       ARE USED AS NEW ESTIMATIONS FOR THE NEXT ITERATION.
C
          ESTPR(I)=RDPROP(K)
          ESTCH4(I)=RDCH4(K)
          ESTTR(I)=TRD(K)
          JF(K)=-JF(K)
30    CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE  TEVAL
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) EVALUATION OF MEAN AND MEAN-SQUARE AIR TEMP.
C       2) CALCULATION OF THE INTEGRAL OF AIR DENSITY*ELEVATION
C          VARIATION FOR EACH AIRWAY.
C       3) EVALUATION OF VARIED AIRFLOW RESISTANCE DUE TO VARIED AIR
C          TEMP.
C       THIS SUBROUTINE IS CALLED FOR IN THE QUASI-EQUILIBRIUM
C     SIMULATION PART OF THE PROGRAM. ITS COUNTERPART IN THE DYNAMIC
C     SIMULATION PART IS PARAM.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DO 40 I=1,NB
         CP=0.2376+0.000012*(TRD(I)+TJS(I))
         E=DZRD(I)/(2.*778.26*CP)
         B=TJS(I)-TROCK(I)
         G=XNEW(I)*LA(I)/CP
         IF (G.GE.25.0) THEN
            GX=0.0
            GXX=0.0
         ELSE
            GX=EXP(-G)
            GXX=EXP(-2.*G)
         ENDIF
         F1=ABS(TJS(I)-TROCK(I))
         F2=ABS(TRD(I)-TROCK(I))
         FF=MAX(F1,F2)
         IF (FF.LT.0.1) THEN
            TMRD(I)=TJS(I)
            FRNVP(I)=2116.0*DZRD(I)/(53.352*(TJS(I)+460.0))
         ELSE
            TMRD(I)=TROCK(I)-B*(GX-1.0)/G-E*(0.5-(GX+(GX-1.0)/G)/G)
            TTROCK=TROCK(I)
            IF (ABS(TJS(I)-TTROCK).LT.0.01) TTROCK=TJS(I)-5.0
            TT=(TRD(I)-TTROCK)/(TJS(I)-TTROCK)
            IF (TT.LT.0.01) THEN
               TTROCK=AMIN1(TJS(I),TRD(I))-5.0
               TT=(TRD(I)-TTROCK)/(TJS(I)-TTROCK)
            ENDIF
            IF (ABS(TT-1.0).LT.1.E-3) THEN
               FRNVP(I)=2116.0*DZRD(I)/(53.352*(TJS(I)+460.0))
            ELSE
               YQ=ALOG((TRD(I)+460.0)/(TJS(I)+460.0))/ALOG(TT)
               FRNVP(I)=2116.0*DZRD(I)*(1.0-YQ)/(53.352*(TTROCK+460.))
            ENDIF
         ENDIF
         TRA1=TROCK(I)+460.0
         A1=2*E*E/(2*G)**3+2*E*B/(4*G*G)-B*B/(2*G)
         A2=4*E*E/G**3+(920.*E+2*E*TJS(I))/(G*G)-2*TRA1*B/G
         A3=2*E*E/G+4*E*E/(G*G)+(920.*E+2*E*TJS(I))/G
         A4=E*E/(2*G)+2*E*E/(4*G*G)+E*B/G
         TMSQR=TRA1*TRA1-TRA1*E-E*LA(I)/3.0+A1*(GXX-1.0)
     .             +A2*(GX-1.0)+A3*GX+A4*GXX
C
C         AIRFLOW RESISTANCE FOR A FAN BRANCH (NWTYP(I)=1) IS REGARDED
C       AS ZERO AND R FOR THAT BRANCH STANDS FOR FAN PRESSURE HEAD.
C
         IF (NWTYP(I).NE.1) R(I)=RSTD(I)*TMSQR/(460.+TR)**2
40    CONTINUE
C
      RETURN
      END

C
C
C
      SUBROUTINE  NVP2
C
C     ------------------------------------------------------------------
C
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF NVP ACCORDING TO CALCULATED FRNVP OF EACH AIRWAY
C     (REFER TO SUBS. TEVAL AND PARAM).
C
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=1
      SUMFNV=0.0
      DO 20 K=1,MNO
         MENDW=MEND(K)
C
C     FNVP(K): NVP IN MESH K. IN IN.W.G.
C     SUMFNV: SUM OF NVP VARIATION IN MESHES IN THE SUCCESSIVELY
C           PERFORMED ITERATIONS, SERVING AS A CRITERION TO JUDGE IF
C           THE CONVERGENT RESULTS ARE REACHED.
C
         ONVP=FNVP(K)
         FSM=0.0
         DO 10 J=MBEGW,MENDW
            N=MSL(J)
            IF (N.LT.0) THEN
               FACT=-1.0
               NX=-N
            ELSE
               FACT=1.0
               NX=N
            ENDIF
```

```
            FSM=FSM+FACT*FRNVP(NX)
10        CONTINUE
          FNVP(K)=-FSM/5.2
          SUMFNV=SUMFNV+ABS(ONVP-FNVP(K))
          MBEGW=MENDW+1
20      CONTINUE
C
        RETURN
        END
C
C
C
        SUBROUTINE CDCH (NSTOP)
C
C       -------------------------------------------------------------------
C
C         SUBROUTINE PURPOSE:
C         UPDATING CONDITION CHANGES ACCORDING TO USER-SPECIFIED TIME TABL
C
C       -------------------------------------------------------------------
C
        INCLUDE 'CMMN1.DAT'
C
        DIMENSION  QFX(20),PFX(20),DAL(15)
C
C
C
C        ICODE: CODE NUMBER. 1: CHANGE TO ORDINARY AIRWAY.   2: CHANGE
C               TO FAN BRANCH.  3: CHANGE TO FIRE SOURCE. 4: FIRE CEESED.
C               5: CHANGE SIMULATION STEPS. 6: CHANGE TIME INTERVAL FOR
C               OUTPUT. 7: JUNCTION WHOSE DOWN-STREAN SIDE AIRWAYS HAVE
C               DETAILED DATA RECORDS.
C        NBRR: AIRWAY CALLING NUMBER IN WHICH CONDITION CHANGE HAPPENS.
C
4       CALL READIN (DAL,14,ISTOP,0)
        IF (ISTOP.EQ.1) THEN
           ICODE=DAL(2)
           WRITE (8,300)
           IF (ICODE.EQ.1) WRITE (8,310)
           IF (ICODE.EQ.2) WRITE (8,320)
           IF (ICODE.EQ.3) WRITE (8,330)
           IF (ICODE.EQ.4) WRITE (8,340)
           IF (ICODE.EQ.5) WRITE (8,350)
           IF (ICODE.EQ.6) WRITE (8,360)
           IF (ICODE.EQ.7) WRITE (8,365)
           WRITE (8,370) (ROW(IE),IE=1,80)
           NSTOP=5
           RETURN
        ENDIF
        IF (DAL(15).LT.(-1.E20)) GO TO 4
        IF (NCOMTS.GT.NCOMT2) THEN
           NCOMT2=NCOMTS
           GO TO 4
        ENDIF
        TS=DAL(1)
        IF (TS*60.0.GT.TACC) GO TO 100
        ICODE=DAL(2)
        NBRR=DAL(3)
        DO 5 I=1,NB
           IF (NO(I).EQ.NBRR) THEN
              N=I
              GO TO 6
           ENDIF
5       CONTINUE
        GO TO 4
```

```
C
C         ICODE=1 INDICATES THAT AIRWAY NBRR BECOMES AN ORDINARY AIRWAY
C       WITH RESISTANCE EQUAL TO RCH AT TS MIN. AFTER EVENT.
C
6       IF (ICODE.EQ.1) THEN
           RSTD(N)=DAL(4)
           R(N)=RSTD(N)*(TMRD(N)+460.0)**2/(TR+460.0)**2
           NWTYP(N)=0
           IF (NFNUM.GT.0) THEN
              DO 8 J=1,NFNUM
                 IF (NOF(J).EQ.NBRR) THEN
                    IF (NFNUM.GT.J) THEN
                       DO 14 K=J,NFNUM-1
                          NOF(K)=NOF(K+1)
                          MPTS(K)=MPTS(K+1)
                          NSKP(K)=NSKP(K+1)
                          DO 7 KK=1,IMX
                             QF(K,KK)=QF(K+1,KK)
                             PF(K,KK)=PF(K+1,KK)
7                         CONTINUE
                          IF (NSWT(K).EQ.2) THEN
                             MF=MPTS(K+1)
                             DO 12 II=1,MF-1
                                FKQ(K,II,1)=FKQ(K+1,II,1)
                                FKQ(K,II,2)=FKQ(K+1,II,2)
                                FKQ(K,II,3)=FKQ(K+1,II,3)
                                FKQ(K,II,4)=FKQ(K+1,II,4)
12                           CONTINUE
                          ELSEIF (NSWT(K).EQ.1) THEN
                             NCOF(K)=NCOF(K+1)
                             N1A=NCOF(K)
                             DO 13 II=1,N1A
                                COF(K,II)=COF(K+1,II)
13                           CONTINUE
                          ENDIF
14                     CONTINUE
                    ENDIF
                    NOF(NFNUM)=0
                    MPTS(NFNUM)=0
                    NSKP(NFNUM)=0
                    NFNUM=NFNUM-1
                    GO TO 4
                 ENDIF
8             CONTINUE
           ENDIF
           GO TO 4
C
C         ICODE=2 INDICATES THAT AIRWAY NBRR BECOMES A FAN BRANCH WITH
C       CHARACTERISTICS.
C
        ELSE IF (ICODE.EQ.2) THEN
           INDEX=DAL(4)
           IF (INDEX.LE.0) THEN
              GO TO 4
           ELSE IF (INDEX.EQ.1) THEN
              RX=DAL(5)
              IF (NFNUM.GT.0) THEN
                 DO 50 J=1,NFNUM
                    IF (NOF(J).EQ.NBRR) THEN
                       DO 40 JJ=J,NFNUM-1
                          NOF(JJ)=NOF(JJ+1)
                          MPTS(JJ)=MPTS(JJ+1)
                          DO 40 KK=1,IMX
                             QF(JJ,KK)=QF(JJ+1,KK)
                             PF(JJ,KK)=PF(JJ+1,KK)
```

```
40                  CONTINUE
                    NFNUM=NFNUM-1
                 ENDIF
50            CONTINUE
           ENDIF
           NWTYP(N)=1
           R(N)=RX
           GO TO 4
        ELSE IF (INDEX.GT.5) THEN
           INDEX=5
        ENDIF
        DO 55 I=1,INDEX
           II=5+(I-1)*2
           QFX(I)=DAL(II)
           PFX(I)=DAL(II+1)
55      CONTINUE
        IF (NFNUM.GT.0) THEN
           DO 20 J=1,NFNUM
              IF (NOF(J).EQ.NO(N)) THEN
                 MPTS(J)=INDEX
                 DO 10 K=1,INDEX
                    QF(J,K)=QFX(K)
                    PF(J,K)=PFX(K)
10               CONTINUE
                 GO TO 4
              ENDIF
20         CONTINUE
        ENDIF
        NFNUM=NFNUM+1
        IF (NFNUM.GT.IMX) THEN
           NFNUM=NFNUM-1
           NWTYP(N)=1
           R(N)=PFX(1)
           GO TO 4
        ENDIF
        NWTYP(N)=1
        NOF(NFNUM)=NO(N)
        MPTS(NFNUM)=INDEX
        DO 30 K=1,INDEX
           QF(NFNUM,K)=QFX(K)
           PF(NFNUM,K)=PFX(K)
30      CONTINUE
        GO TO 4
C
C       ICODE=3 INDICATES THAT BRANCH NBRR BECOMES A FIRE SOURCE AT
C     TS MIN. AFTER EVENT.
C
      ELSE IF (ICODE.EQ.3) THEN
         CONTX=DAL(4)
         CONCX=DAL(5)
         HEATX=DAL(6)
         O2MINX=DAL(7)
         SMPO2X=DAL(8)
         HTPO2X=DAL(9)
         QCENTX=DAL(10)
         TPRX=DAL(11)
         IF (INFLOW.GT.0) THEN
            DO 60 J=1,INFLOW
               IF (NCENT(J).EQ.NBRR) THEN
                  CONT(J)=CONTX
                  CONC(J)=CONCX
                  HEAT(J)=HEATX
                  O2MIN(J)=O2MINX
                  SMPO2(J)=SMPO2X
                  HTPO2(J)=HTPO2X
                  QCENT(J)=QCENTX
                  TPR(J)=TPRX
                  GO TO 4
               ENDIF
60          CONTINUE
         ENDIF
         INFLOW=INFLOW+1
         IF (INFLOW.GT.IMX) GO TO 4
         NCENT(INFLOW)=NBRR
         CONT(INFLOW)=CONTX
         CONC(INFLOW)=CONCX
         HEAT(INFLOW)=HEATX
         O2MIN(INFLOW)=O2MINX
         SMPO2(INFLOW)=SMPO2X
         HTPO2(INFLOW)=HTPO2X
         QCENT(INFLOW)=QCENTX
         TPR(INFLOW)=TPRX
         GO TO 4
C
C       ICODE=4 INDICATES THAT BRANCH NBRR BECOMES AN ORDINARY AIRWAY
C     WITHOUT A FIRE SOURCE AT TS MIN. AFTER EVENT.
C
      ELSE IF (ICODE.EQ.4) THEN
         IF (INFLOW.GT.0) THEN
            DO 90 I=1,INFLOW
               IF (NCENT(I).EQ.NBRR) THEN
                  DO 80 J=I,INFLOW
                     NCENT(J)=NCENT(J+1)
                     CONT(J)=CONT(J+1)
                     CONC(J)=CONC(J+1)
                     HEAT(J)=HEAT(J+1)
                     O2MIN(J)=O2MIN(J+1)
                     SMPO2(J)=SMPO2(J+1)
                     HTPO2(J)=HTPO2(J+1)
                     QCENT(J)=QCENT(J+1)
                     TPR(J)=TPR(J+1)
80                CONTINUE
                  INFLOW=INFLOW-1
                  NWTYP(N)=0
                  GO TO 4
               ENDIF
90          CONTINUE
         ENDIF
      ENDIF
      GO TO 4
100   RETURN
C
C
300   FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ',
     .        ' IN:',//)
310   FORMAT (1X,'TS,1,NBRR,R')
320   FORMAT (1X,'TS,2,NBRR,INDEX,QF,PF,QF,PF,QF,PF,QF,PF,QF,PF')
330   FORMAT (1X,'TS,3,NBRR,CONT,CONC,HEAT,02MIN,SMPO2,HTPO2,QCENT')
340   FORMAT (1X,'TS,4,NBRR')
350   FORMAT (1X,'TS,5,DELTAT')
360   FORMAT (1X,'TS,6,TOUT')
365   FORMAT (1X,'TS,7,JDPP')
370   FORMAT (/,1X,'THE CURRENT INPUT WAS:',//,80A1)
C
      END
C
C
C
C
      SUBROUTINE READIN (DAL,MAX,ISTOP,ICT)
```

```
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       DATA INPUT.
C
C     ------------------------------------------------------------------
      INCLUDE 'CMMN1.DAT'
C
C     COMMON /CHRC/ ITITLE(50),NCOMT1,NCOMTS,NCOMT2,LLINE
C     COMMON /ERMSG/ ROW(80),TITLE(50,20)
C     CHARACTER*4 TITLE
C     CHARACTER*1 ROW,CHK(15),ER(80)
      CHARACTER*1 CHK(15),ER(80)
      DIMENSION DAL(15)
      DATA CHK/'0','1','2','3','4','5','6','7','8','9','.',',','+',
     .          '-',' '/
C
C
      KK=0
      KK1=0
      KK2=0
      NEG=0
      ISTOP=0
      DO 5 I=1,15
         DAL(I)=0.0
5     CONTINUE
      DO 7 I=1,80
         ER(I)=' '
7     CONTINUE
      READ (9,110,ERR=60) (ROW(I),I=1,80)
      LLINE=LLINE+1
      IDEC=0
      MLEV=1
      JJ=0
      I=15
      LEV=0
10    LEV=LEV+1
      II=I
      IF (LEV.GT.80) GO TO 50
      DO 20 I=1,15
         IF (ROW(LEV).EQ.CHK(I)) THEN
            IF (I.NE.15) KK=1
            GO TO 30
         ENDIF
20    CONTINUE
      KK2=KK2+1
      IF ((KK.EQ.0.AND.ROW(LEV).NE.'$').OR.KK2.GE.4) THEN
         ISTOP=0
         NCOMTS=NCOMTS+1
         IF (NCOMTS.LE.50) THEN
            IF (ICT.EQ.1) THEN
               KK3=0
               KK4=0
               DO 21 I=1,80
                  IF (ROW(I).NE.CHK(15)) GO TO 22
                  KK3=KK3+1
21             CONTINUE
22             DO 23 I=80,1,-1
                  IF (ROW(I).NE.CHK(15)) GO TO 24
                  KK4=KK4+1
23             CONTINUE
24             LR=(KK3+KK4-8)/2
               IF (LR.LT.0) LR=0
               KK5=80-KK4
               IF (LR.EQ.0) KK5=KK3+72
               DO 25 I=KK3+1,KK5
                  ER(I-KK3+LR)=ROW(I)
25             CONTINUE
               DO 26 I=1,80
                  IF (I.LE.LR) THEN
                     ROW(I)=' '
                  ELSE
                     ROW(I)=ER(I)
                  ENDIF
26             CONTINUE
            ENDIF
            DO 27 I=1,20
               J=(I-1)*4+1
               TITLE(NCOMTS,I)=ROW(J)//ROW(J+1)//ROW(J+2)//ROW(J+3)
27          CONTINUE
         ENDIF
         RETURN
      ELSE IF (KK.EQ.0.AND.ROW(LEV).EQ.'$') THEN
         KK1=1
         GO TO 50
      ELSE IF (KK1.NE.1) THEN
         ER(LEV)=ROW(LEV)
         ISTOP=1
         I=1
      ENDIF
30    IF (I.GT.10) GO TO 40
      IF (JJ.EQ.1) MLEV=MLEV+1
      JJ=0
      IF (IDEC.EQ.0) THEN
         DAL(MLEV)=(DAL(MLEV)*10.0)+((I-1)*1.0)
      ELSE IF (IDEC.GT.0) THEN
         DAL(MLEV)=DAL(MLEV)+((I-1)*1.0)/(10.0**IDEC)
      ENDIF
      IF (IDEC.NE.0) IDEC=IDEC+1
      GO TO 10
40    IF (I.EQ.11) IDEC=1
      IF (I.EQ.14) NEG=1
      IF (I.EQ.12.OR.I.EQ.15) THEN
         IF (NEG.EQ.1) DAL(MLEV)=0.0-DAL(MLEV)
         NEG=0
         IDEC=0
      ENDIF
      IF (JJ.EQ.1.AND.(I.EQ.11.OR.I.EQ.13.OR.I.EQ.14)) THEN
         MLEV=MLEV+1
         JJ=0
      ENDIF
      IF (I.EQ.12) THEN
         MLEV=MLEV+1
         JJ=0
      ENDIF
      IF (I.EQ.15.AND.II.EQ.12) JJ=5
      IF (I.EQ.15.AND.II.NE.15.AND.II.NE.12.AND.JJ.EQ.0) JJ=1
      IF (I.EQ.12) DAL(MLEV)=0.0
      IF (MLEV.LE.MAX) GO TO 10
50    IF (ISTOP.GT.0) WRITE (8,100) (ROW(I),I=1,80),(ER(I),I=1,80)
      IF (KK.EQ.0.OR.KK1.EQ.1) DAL(15)=-1.E25
      RETURN
60    WRITE (8,120)
      ISTOP=1
      RETURN
C
100   FORMAT (/,' * ERROR *  ILLEGAL CHARACTER DETECTED IN ',
     .        'THE LINE WHERE NUMERICAL INPUT',/,' WAS EXPECTED. ',
     .        'TWO LINES FOLLOW: 1 ORIGINAL; 2 ILLEGAL CHARACTERS.',
```

```
     .       //,80A1,/,80A1,/)
110   FORMAT (80A1)
120   FORMAT (//,'* ERROR *  INPUT ERROR, POSSIBLY END-OF-FILE.',//)
C
      END
C
C
C
      SUBROUTINE OUTPUT (IPT,MARKX,MARKY,MADJC)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       OUTPUT OF NORMAL CALCULATION RESULTS.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION HFN(5),QFN(5),LST(NMX),TQR(NMX)
C
C
      IF (IPT.NE.1) THEN
         PP=0.0
         DO 10 L=1,NB
            IF (NWTYP(L).LE.0.OR.NWTYP(L).EQ.10) THEN
               P(L)=R(L)*Q(L)*ABS(Q(L))*1.E-10
            ELSE
               P(L)=R(L)
            ENDIF
            PP=PP+P(L)
10       CONTINUE
         PCRIT=PP/FLOAT(NB)
C
         IF (MADJC.GE.MADJ.AND.IPT.EQ.4) THEN
            DO 1520 I=1,NB
               TTMN=0.0
               DO 1500 J=1,10
                  TTMN=TTMN+TMRCD(I,J)
1500           CONTINUE
               TTMN=TTMN/10.0
               TQR(I)=0.0
               DO 1510 J=1,10
                  TQR(I)=TQR(I)+(TMRCD(I,J)-TTMN)**2
1510           CONTINUE
1520        CONTINUE
            IZ=0
            LL=0
1530        II=0
            DO 1540 I=1,NB
               IF (NO(I).GT.0) THEN
                  IF (TQR(I).GT.TSR.OR.II.EQ.0) THEN
                     TSR=TQR(I)
                     II=I
                  ENDIF
               ENDIF
1540        CONTINUE
            IF (II.EQ.0) GO TO 1590
            IF (LL.GT.0) THEN
               KK=0
               DO 1550 J=1,LL
                  K=LST(J)
                  IF (TQR(II).GT.TQR(K)) KK=KK+1
1550           CONTINUE
               IF (KK.GE.(LL-2).AND.TQR(II).GT.20.0) GO TO 1560
               GO TO 1590
            ENDIF
1560        LL=LL+1
            LST(LL)=II
            NO(II)=-NO(II)
            JSS=JS(II)
            JFF=JF(II)
1570        DO 1580 I=1,NB
               IF (NO(I).GT.0) THEN
                  IF ((JS(I).EQ.JSS.OR.JF(I).EQ.JSS.OR.JS(I).EQ.JFF.OR.
     .               JF(I).EQ.JFF).AND.(TQR(I).GT.20.0)) THEN
                     LL=LL+1
                     LST(LL)=I
                     NO(I)=-NO(I)
                     IF (JS(I).EQ.JSS) THEN
                        JSS=JF(I)
                     ELSE IF (JF(I).EQ.JSS) THEN
                        JSS=JS(I)
                     ELSE IF (JS(I).EQ.JFF) THEN
                        JFF=JF(I)
                     ELSE
                        JFF=JS(I)
                     ENDIF
                     GO TO 1570
                  ENDIF
               ENDIF
1580        CONTINUE
            IZ=IZ+1
            IF (IZ.LE.3) GO TO 1530
1590        CRT1=0.0
            CRT2=0.0
            CRT3=0.0
            DO 1630 I=1,LL
               K=LST(I)
               NO(K)=IABS(NO(K))
               CRT=0.0
               DO 1600 J=10,3,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-2))**2
1600           CONTINUE
               CRT1=CRT1+CRT/8.0
               CRT=0.0
               DO 1610 J=10,4,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-3))**2
1610           CONTINUE
               CRT2=CRT2+CRT/7.0
               CRT=0.0
               DO 1620 J=10,5,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J)-TMRCD(K,J-4))**2
1620           CONTINUE
               CRT3=CRT3+CRT/6.0
1630        CONTINUE
            CRT1=CRT1/LL
            CRT2=CRT2/LL
            CRT3=CRT3/LL
            LSG=LL
            IF (MIN(CRT1,CRT2,CRT3)*10.0.LE.MAX(CRT1,CRT2,CRT3)) THEN
               ICYCLE=2
               IF (CRT2.LT.0.95*CRT1) ICYCLE=3
               IF (CRT3.LT.0.9*CRT1.AND.CRT3.LT.CRT2) ICYCLE=4
            ELSE
               ICYCLE=0
            ENDIF
         ENDIF
      ENDIF
C
```

```
      IF (IPT.EQ.1) THEN
C
C
         WRITE (8,490)
         LINE=NCOMT1+1
         DO 15 K=1,NB
13          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
            IF (K.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 13
            ENDIF
            WRITE (8,500) NO(K),JS(K),JF(K),Q(K),NWTYP(K),LA(K),A(K),
     .                    R(K),KF(K),O(K)
15       CONTINUE
         IF (NVPN.LE.0) THEN
            WRITE (8,980)
            WRITE (8,990) (JNO(K),T(K),Z(K),CH4C(K),K=1,NJ)
         ENDIF
C
      ELSE IF (IPT.EQ.2) THEN
C
C
         WRITE (8,110)
         WRITE (8,120)
         L=0
         LINE=NCOMT1+1
         DO 20 K=1,NB
18          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
            IF (K.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 18
            ENDIF
            IF (NWTYP(K).GT.0) THEN
               L=L+1
            ELSE
               WRITE (8,130) NO(K),JS(K),JF(K),NWTYP(K),R(K),Q(K),P(K)
            ENDIF
20       CONTINUE
         IF (L.LE.0) THEN
            WRITE (8,140)
         ELSE
            WRITE (8,150)
            DO 30 K=1,NB
               IF (NWTYP(K).LE.0) GO TO 30
               WRITE (8,160) NO(K),JS(K),JF(K),Q(K),P(K)
30          CONTINUE
            IF (NFNUM.GT.0) THEN
               WRITE (8,170) (NOF(K),K=1,NFNUM)
               WRITE (8,175)
               DO 40 K=1,NFNUM
                  L=MPTS(K)
                  WRITE (8,180) (QF(K,I)/1000.0,PF(K,I),I=1,L)
40             CONTINUE
               IF (IOUT.LE.(-2).AND.MARKY.LE.0) THEN
                  WRITE (8,1040)
                  DO 45 K=1,NFNUM
                     INDEX=MPTS(K)
                     N1A=NCOF(K)
                     IF (INDEX.GT.1.OR.N1A.GT.1) THEN
                        IF (NSWT(K).EQ.1) THEN
                           WRITE (8,1044) NOF(K)
C                           DO 42 K1=1,N1A
                              WRITE (8,1046) (COF(K,II),II=1,N1A)
                              WRITE (8,*) (COF(K,II),II=1,N1A)
C42                         CONTINUE
                        ELSEIF (NSWT(K).EQ.2) THEN
                           WRITE (8,1045) NOF(K)
                           DO 43 K1=1,INDEX-1
                              WRITE (8,1050) (FKQ(K,K1,K2),K2=1,4),
     .                                       QF(K,K1),QF(K,K1+1)
43                         CONTINUE
                        ENDIF
                        WRITE (8,1085)
                        DO 441 K1=1,INDEX-1
                           QZZ=(QF(K,K1+1)-QF(K,K1))/5.0
                           DO 44 K2=1,5
                              QZZ1=QF(K,K1)+(K2-1)*QZZ
                              NFCWZ=NFCW(K)
                              IF (NSWT(K).EQ.2) THEN
                                 CALL SPLINE (K,QZZ1,HZZ,TR,1)
                              ELSEIF (NSWT(K).EQ.1) THEN
                                 CALL LSFAN (K,QZZ1,HZZ,TR,1,0)
                              ENDIF
                              NFCW(K)=NFCWZ
                              HFN(K2)=HZZ
                              QFN(K2)=QZZ1
44                         CONTINUE
                           WRITE (8,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
441                     CONTINUE
                     ENDIF
45                CONTINUE
               ENDIF
            ENDIF
         ENDIF
         KTL=0
         DO 48 K=1,NB
            IF (Q(K).LT.0.0) THEN
               IF (KTL.EQ.0) WRITE (8,470)
               WRITE (8,480) NO(K),IABS(JF(K)),IABS(JS(K))
               KTL=1
            ENDIF
48       CONTINUE
```

```
         IF (KTL.EQ.0) WRITE (8,482)
         IF (NFNUM.GT.0) THEN
            DO 62 K=1,NFNUM
               DO 50 KK=1,NB
                  IF (NO(KK).EQ.NOF(K).AND.NWTYP(KK).EQ.1) GO TO 60
50             CONTINUE
               GO TO 62
60             IF (NFCW(K).GT.0) WRITE (8,200) NOF(K)
               IF ((Q(KK).LT.0.0.AND.IABS(JS(KK)).EQ.JSB(KK)).OR.
     .            (Q(KK).GT.0.0.AND.IABS(JS(KK)).NE.JSB(KK)))
     .            WRITE (8,840) NOF(K)
62          CONTINUE
         ENDIF
         WRITE (8,190) NB,NJ
      ELSE IF (IPT.EQ.3) THEN
C
C
       IF (IOUT.GT.0.OR.(IOUT.EQ.(-1).AND.MARKY.GE.1)) THEN
         WRITE (8,1060)
         LINE=NCOMT1+1
         DO 66 K=1,NB
64          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
            IF (K.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 64
            ENDIF
            WRITE (8,1070) NO(K),JS(K),JF(K),NWTYP(K),TROCK(K),LA(K),
     .                     DZRD(K),RSTD(K),HK(K),HA(K),CH4V(K)
66       CONTINUE
         IF (MARKY.LE.0) WRITE (8,1080)
       ENDIF
       IF (IOUT.LE.(-2).OR.(IOUT.EQ.(-1).AND.MARKY.LE.0)) THEN
         WRITE (8,800)
         LINE=NCOMT1+1
         DO 74 I=1,NB
72          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
            IF (I.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 72
            ENDIF
            WRITE (8,810) NO(I),JS(I),JF(I),DZRD(I),TROCK(I),
     .                    CH4V(I),HK(I),HA(I)
74       CONTINUE
       ENDIF
       IF (MARKY.LE.0.OR.IOUT.NE.0) THEN
         WRITE (8,240) TSTART,JSTART
         IF (INFLOW.GT.0) THEN
            WRITE (8,250)
            WRITE (8,260) (NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),
     .         SMPO2(I),HTPO2(I),TPR(I),INT(QCENT(I)),I=1,INFLOW)
         ELSE
            WRITE (8,270)
         ENDIF
       ENDIF
      ELSE IF (IPT.EQ.4) THEN
C
C
         ITL=MADJC
         JTL=MADJ
         IF (MARKX.GE.1.AND.MARKY.LE.0) THEN
            WRITE (8,320) TACC
         ELSE IF (MARKY.GE.1) THEN
            WRITE (8,970) TIME
         ENDIF
         IF (ITL.GE.JTL) THEN
            WRITE (8,690) SUMFNV/MNO
            KTL=0
            DO 78 I=1,NB
               NREVA=NREV(I)
               NREVA=NNREV(I)
               IF (NREVA.GE.MADJ/3) THEN
                  IF (KTL.EQ.0) THEN
                     WRITE (8,700)
                     KTL=1
                  ENDIF
                  IF (NWTYP(I).EQ.1) THEN
                     PP=R(I)
                  ELSE
                     PP=R(I)*Q(I)*ABS(Q(I))*1.E-10
                  ENDIF
                  WRITE (8,710) NO(I),Q(I),PP,NREVA,ITL
               ENDIF
78          CONTINUE
            IZ=MADJ-9
            IF (ICYCLE.EQ.0) THEN
               WRITE (8,890) (IJ,IJ=IZ,IZ+9)
            ELSE
               WRITE (8,900) ICYCLE,(IJ,IJ=IZ,IZ+9)
            ENDIF
            WRITE (8,905) (HRM(IJ),IJ=1,10)
            DO 80 I=1,LSG
               K=LST(I)
               IF (JSB(K).EQ.JS(K)) THEN
                  JSS=JS(K)
                  JFF=JF(K)
               ELSE
                  JSS=JF(K)
                  JFF=JS(K)
               ENDIF
               WRITE (8,910) NO(K),JSS,JFF,(TMRCD(K,J),J=1,10)
               WRITE (8,920) (QRCD(K,J)/1000.0,J=1,10)
80          CONTINUE
```

```
      ELSE
         IF (SUMFNV/MNO.LE.2.E-4) WRITE (8,770) SUMFNV/MNO,ITL
         IF (SUMFNV/MNO.GT.2.E-4) WRITE (8,780) SUMFNV/MNO,ITL
      ENDIF
      LINE=NCOMT1+1
      WRITE (8,300)
      DO 84 I=1,NB
82       IJ=0
         IF (LINE.LE.50) IJ=ITITLE(LINE)
         IF (I.EQ.IJ) THEN
            IF (LINE.EQ.1) I1=0
            IF (LINE.NE.1) I1=ITITLE(LINE-1)
            IF (LINE.EQ.50) I2=0
            IF (LINE.NE.50) I2=ITITLE(LINE+1)
            IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
               WRITE (8,930) (TITLE(LINE,J),J=1,18)
            ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
               WRITE (8,960) (TITLE(LINE,J),J=1,18)
            ELSE IF (I1.EQ.ITITLE(LINE)) THEN
               WRITE (8,940) (TITLE(LINE,J),J=1,18)
            ELSE
               WRITE (8,950) (TITLE(LINE,J),J=1,18)
            ENDIF
            LINE=LINE+1
            GO TO 82
         ENDIF
         WRITE (8,310) NO(I),IABS(JS(I)),IABS(JF(I)),Q(I),TMRD(I),
     .                 TRD(I),RDPROP(I),RDCH4(I),P(I)
84    CONTINUE
      IF (INFLOW.GT.0) THEN
         WRITE (8,820)
         DO 89 L=1,INFLOW
            DO 88 J=1,NB
               IF (NCENT(L).EQ.NO(J)) THEN
                  WRITE (8,830) NO(J),Q(J),TJS(J),RDPROP(J),
     .                  O2BEH(L),HTAD(L)
                  IF (TJS(J).GT.3500.0) WRITE (8,995) NO(J)
                  GO TO 89
               ENDIF
88          CONTINUE
89       CONTINUE
      ENDIF
      IF (IOUT.NE.0) THEN
         WRITE (8,650)
         WRITE (8,660) (JNO(I),T(I),PROP(I),PRCH4(I),I=1,NJ)
      ENDIF
      KTL=0
      DO 90 I=1,NB
         NNREVA=NNREV(I)
         IF ((NNREVA/2)*2.NE.NNREVA) THEN
            IF (KTL.EQ.0) THEN
               WRITE (8,470)
               KTL=1
            ENDIF
            WRITE (8,480) NO(I),IABS(JS(I)),IABS(JF(I))
         ENDIF
90    CONTINUE
      IF (KTL.EQ.0) WRITE (8,482)
   ELSE IF (IPT.EQ.5) THEN
      WRNSUM=WRNPR+WRNGS+WRNSM+WRNHT
      IF (WRNSUM.LE.0.) THEN
         WRITE (8,410)
      ELSE
         J=0
         WRNP=WRNPR
         IF (PCRIT*0.05.LE.0.5*WRNPR) WRNP=PCRIT*0.05
         DO 95 I=1,NB
            K=0
            L=0
            M=0
            N=0
            IF ((100.*RDCH4(I)).GE.WRNGS) K=K+1
            IF ((100.*RDPROP(I)).GE.WRNSM) L=L+1
            IF (TRD(I).GE.WRNHT) M=M+1
            IF (P(I).LT.WRNP) N=N+1
            IF ((K+L+M+N).GT.0) THEN
               IF (J.EQ.0) THEN
                  WRITE (8,420) WRNGS,WRNSM,WRNHT,WRNP
                  J=J+1
               ENDIF
               WRITE (8,430) NO(I),IABS(JS(I)),IABS(JF(I)),RDCH4(I),
     .                       RDPROP(I),TRD(I),P(I)
            ENDIF
95       CONTINUE
         IF (J.LE.0) WRITE (8,440)
         J=0
         KM=0
         DO 100 I=1,NJ
            K=0
            L=0
            M=0
            IF ((100.*PRCH4(I)).GE.WRNGS) K=K+1
            IF ((100.*PROP(I)).GE.WRNSM) L=L+1
            IF (T(I).GE.WRNHT) M=M+1
            IF ((K+L+M).GT.0) THEN
               IF (J.EQ.0) THEN
                  WRITE (8,450) WRNGS,WRNSM,WRNHT,WRNGS,WRNSM,WRNHT
               ENDIF
               J=J+1
               IF (2*(J/2).NE.J) THEN
                  KM=I
               ELSE
                  WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM),
     .                          JNO(I),PRCH4(I),PROP(I),T(I)
                  KM=0
               ENDIF
            ENDIF
100      CONTINUE
         IF (KM.GT.0) WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM)
         IF (J.LE.0) WRITE (8,445)
      ENDIF
   ENDIF
C
   RETURN
C
110   FORMAT (///,22X,'NETWORK CALCULATION RESULTS ',/)
120   FORMAT (1X,'AIRWAY',2X,'FROM',4X,'TO',3X,'TYPE',8X,'RESISTANCE',
     .        7X,'AIRFLOW',6X,'HEADLOSS',/)
130   FORMAT (I5,I7,I6,I7,8X,E9.4,6X,F10.0,3X,F10.3)
140   FORMAT (///,1X,'NETWORK CONTAINS NO FANS')
150   FORMAT (///,T20,'FANS'//' AIRWAY  FROM    TO      AIRFLOW    FAN',
     .        ' PRESSURE',/)
160   FORMAT (I5,I7,I7,F13.0,F13.3)
170   FORMAT (/////,' THESE CHARACTERISTICS WERE STORED FOR FANS',5I5,/
     .        T45,5I5)
175   FORMAT (//,5('  Q*1000   PF '),/)
180   FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
190   FORMAT (///,1X,'THE STATED NUMBER OF AIRWAYS WAS',I9,/,1X,'THE ',
     .        'STATED NUMBER OF JUNCTIONS WAS',I7)
200   FORMAT (///,1X,'FAN CHARACTERISTICS ARE EXCEEDED FOR FAN NO.',I5)
```

```
  240   FORMAT (///,1X,’A TEMPERATURE OF’,F6.1,’ WAS ASSIGNED TO ’,
     .         ’JUNCTION NO.’,I7)
  250   FORMAT (////,T18,’THE FOLLOWING HEAT/CONTAMI. WAS ASSUMED’,//,
     .         12X,’FIXED INFLUX’,T31,’O2-RICH’,T41,’FUEL-RICH’,T55,
     .         ’LEADING T’,T67,’REF. Q’,/,’ AIRWAY  FLUX  %CONC’,
     .         3X,’HEAT’,5X,’%O2’,4X,’FUMES’,2X,’HEAT’,/)
  260   FORMAT (I4,F8.1,1X,F6.2,1X,E9.3,F7.2,F7.2,F7.1,F10.2,I12)
  270   FORMAT (///,1X,’ NO HEAT/CONTAMINATION WAS SPECIFIED’)
  300   FORMAT (///,T8,’TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN’
     .         ,’ AIRWAYS’,//,’ AIRWAY’,T9,’FROM’,T15,’TO’,T20,’AIRFLOW’,
     .         T31,’AVE. T’,T40,’T AT END’,T50,’FUMES’,T58,’METHANE’,T68,
     .         ’HEADLOSS’,/)
  310   FORMAT (I5,I6,I5,T18,0PF9.0,T30,0PF7.2,T40,0PF7.2,T49,2PF8.5,
     .         T58,2PF6.2,T68,0PF7.4)
  320   FORMAT (///,20X,’TIME AT ’,F6.0,’ SEC. AFTER EVENT’,//)
  410   FORMAT (///,1X,’NO THRESHOLD FOR CRITICAL STATES ’,
     .         ’WERE SPECIFIED.’)
  420   FORMAT (///,9X,’IN THE FOLLOWING AIRWAYS EXIST CRITICAL ’,
     .         ’CONDITIONS’,//,1X,’AIRWAY’,T10,’FROM’,T16,’TO’,T22,’CH4 %’
     .         ,T33,’FUMES %’,T45,’TEMPERATURE’,T61,’HEADLOSS’,/,
     .         T22,’>’,F7.5,T33,’>’,F8.5,T46,’>’,F5.0,’ F’,T59,
     .         ’<’,F8.5,’ IN.WG.’,/)
  430   FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.5,T45,0PF8.1,T62,0PF8.5)
  440   FORMAT (///,1X,’NO CRITICAL CONDITIONS AT AIRWAY ENDS ’,
     .         ’WERE DETECTED’)
  445   FORMAT (///,1X,’NO CRITICAL CONDITIONS IN JUNCTIONS WERE ’,
     .         ’DETECTED.’)
  450   FORMAT (///,10X,’IN THE FOLLOWING JUNCTIONS EXIST CRITICAL ’,
     .         ’CONDITIONS’,//,’ JUNCTION’,T11,’CH4  %’,T19,’FUMES %’,T28
     .         ,’TEMP. F’,T40,’JUNCTION’,T49,’CH4  %’,T57,’FUMES %’,T66,
     .         ’TEMP. F’,/,T11,’>’,F5.2,T19,’>’,F7.5,T29,’>’,F5.0,T49,
     .         ’>’,F5.2,T57,’>’,F7.5,T67,’>’,F5.0,/)
  460   FORMAT (I6,T11,2PF5.2,T17,2PF8.5,T27,0PF7.1,T39,I6,T49,2PF5.2,
     .         T55,2PF8.5,T65,0PF7.1)
  470   FORMAT (///,1X,’REVERSAL OF AIRFLOW HAS OCCURRED IN THE ’,
     .         ’FOLLOWING PLACES’,/)
  480   FORMAT (1X,’AIRWAY’,I6,’ IS NOW CARRYING AIR FROM’,I5,’ TO’,I5)
  482   FORMAT (//,1X,’NO AIRFLOW REVERSAL WAS DETECTED IN THE SYSTEM.’)
  490   FORMAT (///,16X,’BASIC DATA FOR AIRWAYS IN THE NETWORK’,
     .         //,’ AIRWAY FROM’,T15,’TO’,T22,’AIRFLOW’,T30,’TYPE’,T37,
     .         ’LENGTH’,T47,’AREA’,T57,’R’,T64,’K’,T68,’PERI.’,/)
  500   FORMAT (I5,I6,T13,I4,T19,F10.0,T30,I3,T36,F7.1,T44,F7.1,T53,E8.3,
     .         T62,F7.1,F7.1)
  650   FORMAT (///,19X,’PARAMETERS OF AIR IN JUNCTIONS’,//,1X,’JUNCTION’
     .         ,T13,’TEMP.’,T20,’FUMES’,T29,’METHANE’,T44,’JUNCTION’,T55,
     .         ’TEMP.’,T60,’FUMES’,T68,’METHANE’,/)
  660   FORMAT (I5,T6,0PF7.2,T18,2PF8.5,T28,2PF7.2,T43,I5,T51,0PF7.2,T60,
     .         2PF8.5,T68,2PF7.2)
  690   FORMAT (///,1X,’THE CALCULATION WAS NOT COMPLETED SINCE THE ’,
     .         ’NUMBER OF EXCHANGES BETWEEN’,/,’ NETWORK AND TEMPERATURE’
     .         ,’ PARTS OF THE PROGRAM BECAME EXCESSIVE.’,/,’ THRESHOLD’,
     .         ’ IN NVP IS 0.0002 IN.W.G. PER MESH. THE NVP CORRECTION ’,
     .         ’PER’/,’ MESH NOW EQUALS ’,F7.5,’ IN.W.G.’)
  700   FORMAT (///,1X,’THE FOLLOWING UNSTABLE AIRWAYS MAY HAVE’,1X,
     .         ’CAUSED THE INCOMPLETION’,/,’ OF NETWOK BALANCING:’,/)
  710   FORMAT (1X,’AIRWAY’,I5,’ OF AIRFLOW RATE’,F7.0,’ (CFM) AND ’,
     .         ’HEADLOSS’,F7.3,’ IN.W.G.’,/,’ HAD ITS AIRFLOW DIRECTION’
     .         ,’ REVERSED’,I3,’ TIMES IN’,I3,’ ITERATIONS’)
  770   FORMAT (/////,’ THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS ’,
     .         ’PER MESH < 2.E-4 IN.W.G.)’,/,’ SATISFIED. CURRENT ’,
     .         ’SUMFNV PER MESH’,F9.6,’ IN.W.G.,   ITERATIONS ’,I5)
  780   FORMAT (/////,1X,’THRESHOLD IN ACCURACY (SUM OF NVP’,1X,
     .         ’CORRECTIONS PER MESH<2.E-4 IN. W.G.)’,/,’ NOT ’,
     .         ’SATISFIED. CURRENT SUMFNV:’,F9.6,’ IN.W.G., ITER.’,I5)
  800   FORMAT (/////,12X,’DATA FOR CONCENTRATION AND TEMPERATURE’,
     .         ’ CALCULATION’,//,1X,’AIRWAY  FROM’,T16,’TO’,T21,
     .         ’ELEV. DIFF.’,T34,’ROCK T’,T43,’CH4 PROD.’,T54,’COND.’,
     .         T61,’DIFFUSIVITY’,/)
  810   FORMAT (I5,I7,I5,T21,F9.2,T33,F7.2,T43,F7.1,T52,F7.2,T61,F8.3)
  820   FORMAT (///,23X,’DATA FOR HEAT SOURCES’,//,1X,’AIRWAY’,T11,
     .         ’AIRFLOW’,T21,’TEMPERATURE’,T37,’FUME’,T51,’O2 LEFT’,T61,
     .         ’HEAT INPUT’,/)
  830   FORMAT (I5,T8,0PF10.0,T22,0PF8.2,T35,2PF8.5,T50,0PF7.2,T60,E10.3)
  840   FORMAT (/,1X,’AIRFLOW REVERSED IN FAN BRANCH NO. ’,I5)
  890   FORMAT (///,1X,’THE FAILURE IN CONVERGING WAS NOT CAUSED BY A’,
     .         ’ CYCLIC PATTERN. DATA OF’,/,’ SIGNIFICANT CONDITION ’,
     .         ’CHANGE DURING LAST 10 ITERATIONS FOLLOW. FURTHER’,/,
     .         ’ ANALYSES ARE NEEDED. ("ITR" STANDS FOR ITERATION)’,//,
     .         ’ UNITS: Q: AIRFLOW RATE IN 1000FT3/MIN;’,17X,’T: TEMP. ’,
     .         ’IN F;’,/,8X,’H: NVP CORRECTION PER MESH IN IN.W.G.’,//,
     .         ’ ITR’,T5,I5,9I7,/)
  900   FORMAT (///,1X,’THE FAILURE IN CONVERGING WAS CAUSED BY A CYCLIC’,
     .         ’ PATTERN IN EVERY’,I3,/,’ ITERATIONS AS SHOWN BELOW:’,
     .         ’ ("ITR" STANDS FOR ITERATION)’,//,’ UNITS: Q: AIRFLOW ’,
     .         ’RATE IN 1000FT3/MIN;’,17X,’T: TEMP. IN F;’,/,8X,’H: NVP ’,
     .         ’CORRECTION PER MESH IN IN.W.G.’,//,’ ITR’,T5,I5,9I7,/)
  905   FORMAT (1X,’H’,T3,10(1X,F6.4),/)
  910   FORMAT (1X,’AIRWAY: ’,I4,10X,’JS:’,I4,6X,’JF:’,I4,/,1X,
     .         ’T’,T3,10F7.1)
  920   FORMAT (1X,’Q’,T3,10F7.1,/)
  930   FORMAT (/,18A4,/)
  940   FORMAT (18A4,/)
  950   FORMAT (/,18A4)
  960   FORMAT (18A4)
  970   FORMAT (/////,20X,’TIME AT ’,F5.1,’ HOUR(S) AFTER EVENT’,//)
  980   FORMAT (///,T2,’JUNCTION’,T12,’TEMP.’,T20,’ELEVATION’,T32,’CH4’,
     .         T40,’JUNCTION’,T50,’TEMP.’,T58,’ELEVATION’,T70,’CH4’,/)
  990   FORMAT (T2,I5,T10,F7.1,T19,F8.1,T30,F5.2,T40,I5,T48,F7.1,T57,F8.1
     .         ,T68,F5.2)
  995   FORMAT (/,’* WARNING *  AIR TEMP. IN AIRWAY’,I5,’ IS ABNORMAL ’,
     .         ’UNDER MINING CONDITIONS.’,/,’ THE POSSIBLE CAUSES ARE:’,/,
     .         ’ 1. "QCENT" WAS NOT SPECIFIED;’,/,’ 2. "QCENT" DOES NOT’,
     .         ’ MATCH "HEAT";’,/,’ 3. "HTPO2" WAS TOO LARGE.’)
 1040   FORMAT (///,15X,’COEFF. OF POLYNOMIAL OF FAN CURVES’,/,20X,
     .         ’PF=A0+A1*Q+A2*Q**2+A3*Q**3’)
 1044   FORMAT (/,T5,’FAN’,I5,//,T7,’A0’,T19,’A1’,T31,’A2’,T43,’A3’,/)
 1045   FORMAT (/,T5,’FAN’,I5,//,T7,’A0’,T19,’A1’,T31,’A2’,T43,’A3’,T52,
     .         ’FROM QF’,T65,’TO QF’,/)
 1046   FORMAT (T3,E9.3,T15,E9.3,T27,E9.3,T39,E9.3)
 1050   FORMAT (T3,E9.3,T15,E9.3,T27,E9.3,T39,E9.3,T51,E9.3,T63,E9.3)
 1060   FORMAT (///,7X,’BASIC DATA TO DEFINE THE NETWORK AND THE ’,
     .         ’SIMULATION PROBLEM’,//,’ AIRWAY FROM’,T14,’TO’,T17,’TY’,
     .         T21,’ROCK T’,T28,’LENGTH’,T35,’ELEV. D’,T48,’R’,T54,’COND’,
     .         T61,’DIFF’,T67,’CH4 PR’,/)
 1070   FORMAT (I5,I6,T12,I4,I3,T20,F6.1,T27,F7.1,T35,F7.1,T44,E8.3,T53,
     .         F5.2,F7.3,F8.1)
 1080   FORMAT (/,1X,’* NOTE *  FOR DETAILED DATA LIST, INPUT "IOUT" ’,
     .         ’AS -1 OR LESS.’)
 1085   FORMAT (/,2X,’INTERPOLATION:’,/,5(’  Q*1000   PF ’),/)
 1090   FORMAT (5(F8.1,F6.2))
C
        END
```