C COMPLETE MULTIFLO CODE

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C	DRIVER
C
C	'DRIVER' IS A SIMPLE EXECUTIVE TO INVOKE THE 'MULTIFLO' COMMODITY
C	ROUTING PROGRAM.  'DRIVER' INVOKES SUBPROGRAM 'LOAD' TO READ
C	DATA INTO 'MULTIFLO' INPUT COMMON BLOCKS.  FILES READ BY
C	'LOAD' ARE CREATED BY A TERMINAL SESSION WITH THE USER FOR
C	NETWORK DEFINITION THROUGH THE USE OF PROGRAM 'SETUP'.
C
C	EXECUTION STEPS FOR PROGRAM 'DRIVER'
C
C		1) ASSIGN FORTRAN UNIT 01 AS CREATED BY PROGRAM 'LOAD'
C		2) ASSIGN FORTRAN UNIT 02 AS CREATED BY PROGRAM 'LOAD'
C		3) ASSIGN FORTRAN UNIT 06 AS A DESIGNATED OUTPUT FILE
C
C		E.G.:
C		   $ ASSIGN NETWORK.DAT FOR001
C		   $ ASSIGN TRAFFIC.DAT FOR002
C		   $ ASSIGN OUTPUT.DAT  FOR006
C	
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
	PROGRAM DRIVER
C
C	LOAD FORTRAN UNIT 01 AND FORTRAN UNIT 02 FROM DISK AS CREATED
C	FROM PROGRAM 'SETUP'
C
C      'INCLUDE' FILE PARAM.DIM
C
C       'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C	*******************  NETWORK PARAMETERS  **********************
C
      PARAMETER	(NNN=100)
C			MAXIMUM NUMBER OF NODES
      PARAMETER	(NNA=500)
C			MAXIMUM NUMBER OF ARCS
      PARAMETER	(NNUMOD=1000)
C			MAXIMUM NUMBER OF OD PAIRS
      PARAMETER	(NNUMPATH=10000)
C			MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER	(NMAXITER=50)
C			MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER	(NNORIG=100)
C			MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C



C      'INCLUDE' FILE PATHS.BLK
C
C      'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C      PATH FLOWS AND DESCRIPTION.
C
```

```
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &□□NUMPATH,NUMITER

      INTEGER*2□PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL□□FA(NNA)
C□□□THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER □PATHID(NNUMPATH)
C□□□THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER □NEXTPATH(NNUMPATH)
C             □THE NEXT PATH FOR THE SAME OD PAIR
      REAL□□FP(NNUMPATH)
C             □THE FLOW OF A PATH
      REAL □□DIST(NNN)
C             □SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL□□DTOT(NMAXITER)
C□□□THE TOTAL DELAY BY ITERATION
      INTEGER □NUMITER
C   CURRENT ITERATION NUMBER
      REAL□□CURERROR
C□□□CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C              □A SHORTEST PATH)
      INTEGER □NUMPATH
C□□□NUMBER OF GENERATED PATHS


C       'INCLUDE' FILE NETWRK.PRM
C
C□'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &□□NN,FRSTOU,LASTOU,
     &□□NA,STARTNODE,ENDNODE,BITRATE,
     &□□NUMCOMMOD,ORGID,STARTOD,
     &□□NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2□NN
C□□□NUMBER OF NODES IN THE NETWORK
      INTEGER*2 □FRSTOU(NNN)
C□□□THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2□LASTOU(NNN)
C□□□THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2□NA
C□□□NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2□STARTNODE(NNA)
C□□□THE START NODE FOR AN ARC
      INTEGER*2□ENDNODE(NNA)
C□□□THE END NODE FOR AN ARC
      REAL□□BITRATE(NNA)
C□□□THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2□NUMCOMMOD
C□□□THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2□ORGID(NNORIG)
C□□□THE NODE NUMBER OF THE ORIGIN
      INTEGER*2 □STARTOD(NNORIG)
C□□□THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2□NUMODPAIR
```

```
C□□□THE NUMBER OF OD PAIRS
      INTEGER*2□DEST(NNUMOD)
C□□□THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL□□INPUT_FLOW(NNUMOD)
C□□□THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C      'INCLUDE' FILE CONVRG.PRM
C
C□'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C□NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &□□MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C□□MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL□TOL
C□□TOLERANCE ON SOLUTION ACCURACY
      REAL□MAXUTI
C□□MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C□□OUTPUT CONTROL VARIABLE

      INTEGER COMMODITY,ORIGIN,DESTOD,OD,PATH
      CALL LOAD
C
C□EXECUTE THE 'MULTIFLO' NETWORK ALGORITHM.  'MULTIFLO' SCHEDULES
C□ITS OWN OUTPUTS TO FORTRAN UNIT 06 ON EACH ITERATION
C
C      INITIALIZE THE TIMER
C       CALL LIB$INIT_TIMER


     OPEN(6,FILE='FOR006.DAT',STATUS='NEW')
     REWIND(6)

      CALL MULTIFLO
C       RECORD THE COMPUTATION TIME
C        CALL LIB$SHOW_TIMER
C
C         PRINT MAX LINK UTILIZATION (RELEVANT FOR M/M/1 QUEUEING DELAY
C         OPTIMIZATION)
C
        UMAX=0.0
        DO 100 I=1,NA
          UMAX=MAX(UMAX,FA(I)/BITRATE(I))
100       CONTINUE
        WRITE(6,*)'MAXIMUM LINK UTILIZATION'
        WRITE(6,*)UMAX
C
C      PRINT FINAL PATH FLOW INFO
C
       WRITE(6,*)'ORIGIN / DESTINATION / PATH # / PATH FLOW'
       DO 1000 COMMODITY=1,NUMCOMMOD
         ORIGIN=ORGID(COMMODITY)
         DO 500 OD=STARTOD(COMMODITY),STARTOD(COMMODITY+1)-1
           DESTOD=DEST(OD)
```

```
          PATH=OD
          DO WHILE (PATH.GT.0)
            WRITE(6,*)ORIGIN,DESTOD,PATH,FP(PATH)
            PATH=NEXTPATH(PATH)
          END DO
500       CONTINUE
1000    CONTINUE

      ENDFILE(6)
      REWIND(6)

	STOP
	END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C	LOAD
C
C	'LOAD' READS IN DATA FROM DISK CREATED WITH PROGRAM 'SETUP' FOR
C	USE BY PROGRAM 'MULTIFLO'.  NETWORK SPECIFICATION DATA RESIDES
C	ON FORTRAN UNIT 01 AND NETWORK TRAFFIC SPECIFICATION DATA
C	RESIDES ON FORTRAN UNIT 02.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
	SUBROUTINE LOAD
C
C	*********************  INCLUDE COMMON BLOCKS  *********************
C
C       'INCLUDE' FILE PARAM.DIM
C
C        'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C	*******************  NETWORK PARAMETERS  **********************
C
       PARAMETER	(NNN=100)
C			MAXIMUM NUMBER OF NODES
       PARAMETER	(NNA=500)
C			MAXIMUM NUMBER OF ARCS
       PARAMETER	(NNUMOD=1000)
C			MAXIMUM NUMBER OF OD PAIRS
       PARAMETER	(NNUMPATH=10000)
C			MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
       PARAMETER	(NMAXITER=50)
C			MAXIMUM NUMBER OF ITERATIONS ALLOWED
       PARAMETER	(NNORIG=100)
C			MAXIMUM NUMBER OF COMMODITIES
       PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C


C       'INCLUDE' FILE NETWRK.PRM
C
C	'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
       COMMON /NETWORK/
     &		NN,FRSTOU,LASTOU,
```

```
     &  NA,STARTNODE,ENDNODE,BITRATE,
     &  NUMCOMMOD,ORGID,STARTOD,
     &  NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2  NN
C     NUMBER OF NODES IN THE NETWORK
      INTEGER*2  FRSTOU(NNN)
C     THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2 LASTOU(NNN)
C     THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2  NA
C     NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2 STARTNODE(NNA)
C     THE START NODE FOR AN ARC
      INTEGER*2 ENDNODE(NNA)
C     THE END NODE FOR AN ARC
      REAL  BITRATE(NNA)
C     THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2 NUMCOMMOD
C     THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2 ORGID(NNORIG)
C     THE NODE NUMBER OF THE ORIGIN
      INTEGER*2  STARTOD(NNORIG)
C     THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2 NUMODPAIR
C     THE NUMBER OF OD PAIRS
      INTEGER*2 DEST(NNUMOD)
C     THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL  INPUT_FLOW(NNUMOD)
C     THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C      'INCLUDE' FILE CONVRG.PRM
C
C 'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &  MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C  MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL TOL
C  TOLERANCE ON SOLUTION ACCURACY
      REAL MAXUTI
C  MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C  OUTPUT CONTROL VARIABLE

C
 IMPLICIT NONE

C  ********************  LOCAL VARIABLE DEFINITIONS  ******************
C
 INTEGER I
```

```
C□□DO LOOP INDEX
C
C□********************  EXECUTABLE CODE  ***************************
C      TERMINATION PARAMETERS. MAXITER GIVES THE MAX # OF ITERATIONS
C      TOL IS A SOLUTION ACCURACY TOLERANCE. RECOMMENDED VALUES
C      ARE 0.01 TO 0.0001. THE PROPER VALUE OF TOL IS LARGELY
C      INDEPENDENT OF THE PROBLEM SIZE.
       MAXITER=20
       TOL=0.01
C      THE FOLLOWING PARAMETER MAKES SENSE ONLY FOR ROUTING PROBLEMS
C      WHERE AN M/M/1 QUEUING FORMULA IS USED FOR DELAY.
C      IT GIVES THE THRESHOLD FRACTION OF CAPACITY BEYOND WHICH
C      THE DELAY FORMULA IS TAKEN TO BE QUADRATIC.
       MAXUTI=0.99
C
C□LOAD THE NETWORK CONFIGURATION FROM FORTRAN UNIT 01
C
C□NODE SPECIFICATIONS
C
     PRINT*,'************************************'
     PRINT *,'READING PROBLEM DATA'
     OPEN(1,FILE='FOR001.DAT',STATUS='OLD')
     REWIND(1)

□READ(1,*)NN
      DO 10 I=1,NN
□       READ(1,*)FRSTOU(I),LASTOU(I)
10     CONTINUE
C□LINK SPECIFICATIONS
C
□READ(1,*)NA
C
C      BITRATE(I) IS A PARAMETER ASSOCIATED WITH LINK I. IN THE
C      DATA NETWORK ROUTING CONTEXT IT HAS THE MEANING OF
C      TRANSMISSION CAPACITY OF LINK I.
C
□DO 20 I=1,NA
□    READ(1,*)STARTNODE(I),ENDNODE(I),BITRATE(I)
20     CONTINUE

     ENDFILE(1)
     REWIND(1)

C
C□INPUT COMMODITY DATA FROM FORTRAN UNIT 02
C
     OPEN(2,FILE='FOR002.DAT',STATUS='OLD')
     REWIND(2)

□READ(2,*)NUMCOMMOD
□DO 30 I=1,NUMCOMMOD
□    READ(2,*)ORGID(I),STARTOD(I)
30     CONTINUE
□READ(2,*)NUMODPAIR
□DO 40 I=1,NUMODPAIR
□    READ(2,*)DEST(I),INPUT_FLOW(I)
40     CONTINUE

     ENDFILE(2)
     REWIND(2)
```

```
 RETURN
 END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C MULTIFLO
C
C MULTICOMMODITY FLOW ALGORITHM BASED ON A PATH FLOW FORMULATION
C UPDATES THE PATH FLOWS OF OD PAIRS ONE AT A TIME ACCORDING TO
C AN ITERATION OF THE PROJECTION TYPE.
C
C DEVELOPED BY DIMITRI BERTSEKAS
C
C BASED ON THE PAPERS:
C
C  1)  BERTSEKAS,D.P., "A CLASS OF OPTIMAL ROUTING ALGORITHMS
C      FOR COMMUNICATION NETWORKS", PROC. OF 5TH ITERNATIONAL
C      CONFERENCE ON COMPUTER COMMUNICATION (ICCC-80),
C      ATLANTA, GA., OCT. 1980, PP.71-76.
C
C  2)  BERTSEKAS,D.P. AND GAFNI,E.M., "PROJECTION METHODS
C      FOR VARIATIONAL INEQUALITIES WITH APPLICATION TO
C      THE TRAFFIC ASSIGNMENT PROBLEM", MATH. PROGR. STUDY,17,
C      D.C.SORENSEN AND J.-B. WETS (EDS), NORTH-HOLLAND,
C      AMSTERDAM,1982, PP. 139-159.
C
C  3)  BERTSEKAS,D.P., "OPTIMAL ROUTING AND FLOW CONTROL
C      METHODS FOR COMMUNICATION NETWORKS", IN ANALYSIS AND
C      OPTIMIZATION OF SYSTEMS, (PROC. OF 5TH INTERNATIONAL
C      CONFERENCE ON ANALYSIS AND OPTIMIZATION, VERSAILLES,
C      FRANCE), A. BENSOUSSAN AND J.L. LIONS (EDS),
C      SPRINGER-VERLAG, BERLIN & NY,1982, PP. 615-643.
C
C  4)  BERTSEKAS,D.P. AND GAFNI, E.M., "PROJECTED NEWTON
C      METHODS AND OPTIMIZATION OF MULTICOMMODITY FLOWS",
C      IEEE TRANSACTIONS ON AUTOMATIC CONTROL, DEC. 1983.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE MULTIFLO
C
C *************** INCLUDE COMMON BLOCKS ****************************
C
C     'INCLUDE' FILE PARAM.DIM
C
C      'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C ******************* NETWORK PARAMETERS ***********************
C
      PARAMETER (NNN=100)
C   MAXIMUM NUMBER OF NODES
      PARAMETER (NNA=500)
C   MAXIMUM NUMBER OF ARCS
      PARAMETER (NNUMOD=1000)
C   MAXIMUM NUMBER OF OD PAIRS
      PARAMETER (NNUMPATH=10000)
C   MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER (NMAXITER=50)
```

```
C□□□MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER□(NNORIG=100)
C□□□MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C



C      'INCLUDE' FILE PATHS.BLK
C
C      'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C      PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &□□NUMPATH,NUMITER

      INTEGER*2□PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL□□FA(NNA)
C□□□THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER □PATHID(NNUMPATH)
C□□□THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER □NEXTPATH(NNUMPATH)
C             □THE NEXT PATH FOR THE SAME OD PAIR
      REAL□□FP(NNUMPATH)
C             □THE FLOW OF A PATH
      REAL □□DIST(NNN)
C             □SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL□□DTOT(NMAXITER)
C□□□THE TOTAL DELAY BY ITERATION
      INTEGER □NUMITER
C   CURRENT ITERATION NUMBER
      REAL□□CURERROR
C□□□CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C              □A SHORTEST PATH)
      INTEGER □NUMPATH
C□□□NUMBER OF GENERATED PATHS


C      'INCLUDE' FILE NETWRK.PRM
C
C□'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &□□NN,FRSTOU,LASTOU,
     &□□NA,STARTNODE,ENDNODE,BITRATE,
     &□□NUMCOMMOD,ORGID,STARTOD,
     &□□NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2□NN
C□□□NUMBER OF NODES IN THE NETWORK
      INTEGER*2 □FRSTOU(NNN)
C□□□THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2□LASTOU(NNN)
C□□□THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2□NA
C□□□NUMBER OF LINKS (ARCS) IN THE NETWORK
```

```
      INTEGER*2□STARTNODE(NNA)
C□□□THE START NODE FOR AN ARC
      INTEGER*2□ENDNODE(NNA)
C□□□THE END NODE FOR AN ARC
      REAL□□BITRATE(NNA)
C□□□THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2□NUMCOMMOD
C□□□THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2□ORGID(NNORIG)
C□□□THE NODE NUMBER OF THE ORIGIN
      INTEGER*2 □STARTOD(NNORIG)
C□□□THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2□NUMODPAIR
C□□□THE NUMBER OF OD PAIRS
      INTEGER*2□DEST(NNUMOD)
C□□□THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL□□INPUT_FLOW(NNUMOD)
C□□□THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C       'INCLUDE' FILE CONVRG.PRM
C
C□'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C□NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &□□MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C□□MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL□TOL
C□□TOLERANCE ON SOLUTION ACCURACY
      REAL□MAXUTI
C□□MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C□□OUTPUT CONTROL VARIABLE

C
C  □NODE ARRAYS (LENGTH NN):
C
C  □FRSTOU(NODE) - FIRST ARC OUT OF NODE
C  □LASTOU(NODE) - LAST ARC OUT OF NODE
C  □    NOTE: THE ARC LIST MUST BE ORDERED IN SEQUENCE SO
C                THAT ALL ARCS OUT OF ANY NODE ARE GROUPED TOGETHER
C
C  □ARC ARRAYS (LENGTH NA):
C
C  □FA(ARC) - THE TOTAL FLOW OF ARC
C  □STARTNODE(ARC) - THE HEAD NODE OF ARC
C  □ENDNODE(ARC) - THE TAIL NODE OF ARC
C
C  □COMMODITY LENGTH ARRAYS (LENGTH NUMCOMMOD):
C
C  □ORGID(COMMODITY) - THE NODE ID OF THE ORIGIN OF COMMODITY
C  □STARTOD(COMMODITY) - THE STARTING OD PAIR IN THE ODPAIR LIST
C      □□□CORRESPONDING TO THE ORIGIN IN POSITION RANK
```

```
C  □  NOTE: THIS SCHEME ASSUMES THAT OD PAIRS ARE LISTED IN SEQUENCE
C     □□I.E. THE OD PAIRS CORRESPONDING TO THE COMMODITY ONE
C     □□ARE LISTED FIRST. THEY ARE
C     □□FOLLOWED BY THE OD PAIRS OF THE COMMODITY TWO
C     □□AND SO ON.
C
C  □ODPAIR ARRAYS (LENGTH NUMOD):
C  □DEST(OD) - GIVES THE DESTINATION OF ODPAIR OD
C      INPUT_FLOW(OD) - GIVES THE INPUT TRAFFIC OF ODPAIR OD
C
C  □PATH ARRAYS (LENGTH DYNAMICALLY UPDATED):
C  □PATHID(PATH) - THE ITERATION # AT WHICH PATH WAS GENERATED
C  □NEXTPATH(PATH) - THE NEXT PATH FOR THE SAME OD PAIR FOLLOWING
C      □PATH. IT EQUALS 0 IF PATH IS THE LAST FOR THAT OD PAIR
C  □FP(PATH) - THE FLOW CARRIED BY PATH
C
C  □PATH DESCRIPTION LIST ARRAY (LENGTH MAXITER*NUMCOMD*NN)
C  □PRED(NODE,ITER,COMMODITY) - THIS TRIPLE INDEXED ARRAY SPECIFIES THE
C     □□SHORTEST PATH TREE GENERATED AT ITERATION ITER
C     □□& CORRESPONDING TO THE ORIGIN ASSOCIATED W/ COMMODITY
C□□IT GIVES THE LAST ARC ON THE SHORTEST PATH FROM ORIGIN TO NODE.
C
C□***************  LOCAL VARIABLE DEFINITIONS  ************************
C
□IMPLICIT NONE
C
      INTEGER*2□PRED(NNN,NMAXITER,NNORIG)
C             □PATH DESCRIPTION ARRAY - CONTAINS SHORTEST
C             □PATH TREES FOR ALL ITERATIONS
□LOGICAL□SPNEW
C□□LOGICAL INDICATING A NEW PATH FOUND
□LOGICAL SAME
C□□LOGICAL INDICATING A NEW SHORTEST PATH ALREADY EXISTING
□INTEGER NODE
C□□NODE IDENTIFIER
□INTEGER DESTOD
C□□THE DESTINATION NODE OF AN OD PAIR
□INTEGER ARC
C□□DO LOOP INDEX FOR ARCS
□INTEGER PATH
C□□A PATH INDEX
□INTEGER NUMLIST
C□□TOTAL NUMBER OF ACTIVE PATHS FOR OD PAIR UNDER CONSIDERATION
□INTEGER ITER
C□□SPECIFIC ITERATION
      INTEGER N1,N2
C             TEMPORARY VARIABLES
□REAL □MINFDER
C□□THE LENGTH FOR A SHORTEST PATH
□REAL□MINSDER
C□□THE SECOND DERIVATIVE LENGTH FOR THE SHORTEST PATH
      REAL    TMINSDER
C            TEMPORARY VALUE FOR SECOND DERIVATIVE LENGTH OF SHORTEST PATH
□REAL□INCR
C□□TOTAL SHIFT OF FLOW TO THE MINIMUM FIRST DERIVATIVE LENGTH PATH
□REAL□PATHINCR
C□□SHIFT OF FLOW FOR A GIVEN PATH
□REAL □FLOW
C□□FLOW FOR A PATH
□REAL□FDER
```

```
C□□THE ACCRUED LENGTH ALONG A PATH
□REAL□SDER
C□□THE ACCRUED SECOND DERIVATIVE LENGTH ALONG A PATH
□REAL□TEMPERROR
C□□TEMPORARY STORAGE FOR CONVERGENCE ERROR
□REAL □FDLENGTH(NMAXITER)
C□□ARRAY OF LENGTHS OF PATHS FOR AN OD PAIR
□REAL□SDLENGTH(NMAXITER)
C             ARRAY OF SECOND DERIVATIVE LENGTHS OF PATHS FOR AN OD PAIR
□INTEGER PATHLIST(NMAXITER)
C□□ARRAY OF ACTIVE PATHS FOR AN OD PAIR
□INTEGER COMMODITY
C□□DO LOOP INDEX FOR THE OD PAIR ORIGINS
□INTEGER ORIGIN
C□□SPECIFIC ORIGIN
□INTEGER I
C□□DO LOOP INDEX
□INTEGER OD
C□□OD DO LOOP INDEX
□INTEGER K
C□□DO LOOP INDEX
□INTEGER SHORTEST
C□□THE SHORTEST PATH
      LOGICAL MEMBER(NNA)
C             LOGICAL FOR AN ARC INCLUDED IN THE SHORTEST PATH
□REAL□DLENGTH
C□□DIFFERENCE IN PATH LENGTHS FOR THE TRAFFIC
□REAL□D1CAL
C□□ARC LENGTH
□REAL□D2CAL
C□□DERIVATIVE OF ARC LENGTH
C
C□********************  EXECUTABLE CODE  ****************************
C
C □*************************************
C     *   INITIALIZATION
C     *************************************
C
      DO 5 ARC=1,NA
       FA(ARC)=0.0
5     CONTINUE
C
      DO 7 I=1,NUMODPAIR
□       FP(I)=INPUT_FLOW(I)
7     CONTINUE
      STARTOD(NUMCOMMOD+1)=NUMODPAIR+1
      NUMPATH=0
      NUMITER=1
      DO 100 COMMODITY=1,NUMCOMMOD
         ORIGIN=ORGID(COMMODITY)
         CALL SP(ORIGIN,COMMODITY)
         DO 10 I=1,NN
            PRED(I,1,COMMODITY)=PA(I)
10       CONTINUE
C
C □   LOOP OVER OD PAIRS OF COMMODITY
C
        N1=STARTOD(COMMODITY)
        N2=STARTOD(COMMODITY+1)-1
         DO 50 OD=N1,N2
```

```
            NUMPATH=NUMPATH+1
            PATHID(NUMPATH)=1
            NEXTPATH(NUMPATH)=0
            FLOW=FP(NUMPATH)
            NODE=DEST(OD)
            DO WHILE (NODE.NE.ORIGIN)
               ARC=PA(NODE)
               FA(ARC)=FA(ARC)+FLOW
               NODE=STARTNODE(ARC)
            END DO
50         CONTINUE
100 CONTINUE
C
C      INITIALIZE THE MEMBER ARRAY
C
      DO 70 ARC=1,NA
         MEMBER(ARC)=.FALSE.
70     CONTINUE
C
C INITIALIZE THE TOTAL DELAY
C
      CALL DELAY(DTOT(NUMITER))
C
C OUTPUT THE CURRENT INFORMATION TO DISK
C
 CALL PRFLOW
C
C     ***********************************************
C     *  END OF INITIALIZATION
C     ***********************************************
C
C     ***** START NEW ITERATION *****
C
110    NUMITER=NUMITER+1
      CURERROR=0
C
C  **** LOOP OVER ALL COMMODITIES ****
C
      DO 1000 COMMODITY=1,NUMCOMMOD
         ORIGIN=ORGID(COMMODITY)
         CALL SP(ORIGIN,COMMODITY)
         DO 150 I=1,NN
            PRED(I,NUMITER,COMMODITY)=PA(I)
150         CONTINUE
C
C     **** LOOP OVER OD PAIRS OF COMMODITY
C
         N1=STARTOD(COMMODITY)
         N2=STARTOD(COMMODITY+1)-1
          DO 500 OD=N1,N2
C
C          CHECK IF THERE IS ONLY ONE ACTIVE PATH AND IF SO SKIP
C          THE ITERATION
C
           IF (NEXTPATH(OD).EQ.0) THEN
             NODE=DEST(OD)
             DO WHILE (NODE.NE.ORIGIN)
               ARC=PA(NODE)
               IF (ARC.NE.PRED(NODE,1,COMMODITY)) GO TO 180
               NODE=STARTNODE(ARC)
```

```
            END DO
            GO TO 500
          END IF
C
180         CONTINUE
C
C           MARK THE ARCS OF THE SHORTEST PATH
C
          DESTOD=DEST(OD)
          NODE=DESTOD
          DO WHILE (NODE.NE.ORIGIN)
            ARC=PA(NODE)
            MEMBER(ARC)=.TRUE.
            NODE=STARTNODE(ARC)
          END DO
C
C  GENERATE LIST OF ACTIVE PATHS FOR OD PAIR
C
        NUMLIST=1
        PATHLIST(1)=OD
        PATH=NEXTPATH(OD)
        DO WHILE (PATH.GT.0)
          NUMLIST=NUMLIST+1
          PATHLIST(NUMLIST)=PATH
          PATH=NEXTPATH(PATH)
        END DO
C
C  DETERMINE 1ST & 2ND DERIVATIVE LENGTH OF ACTIVE PATHS
C  ALSO DETERMINE WHETHER THE CALCULATED SHORTEST PATH
C  IS ALREADY IN THE LIST
C
        SPNEW=.TRUE.
        DO 200 K=1,NUMLIST
    SAME=.TRUE.
          FDER=0
            SDER=0
  TMINSDER=0
          PATH=PATHLIST(K)
          ITER=PATHID(PATH)
          NODE=DESTOD
          DO WHILE (NODE.NE.ORIGIN)
            ARC=PRED(NODE,ITER,COMMODITY)
   CALL DERIVS(COMMODITY,FA(ARC),ARC,D1CAL,D2CAL)
            FDER=FDER+D1CAL
            IF (.NOT.MEMBER(ARC)) THEN
              SDER=SDER+D2CAL
              SAME=.FALSE.
            ELSE
              SDER=SDER-D2CAL
                TMINSDER=TMINSDER+D2CAL
            END IF
            NODE=STARTNODE(ARC)
          END DO
          IF (SAME) THEN
            SPNEW=.FALSE.
            SHORTEST=PATH
            FDLENGTH(K)=FDER
                MINFDER=FDER
                MINSDER=TMINSDER
          ELSE
```

```
            FDLENGTH(K)=FDER
            SDLENGTH(K)=SDER
          END IF
200       CONTINUE
C
C       *** INSERT SHORTEST PATH IN PATH LIST IF IT IS NEW ***
C
         IF (SPNEW) THEN
           NUMPATH=NUMPATH+1
           SHORTEST=NUMPATH
           PATHID(NUMPATH)=NUMITER
           NEXTPATH(PATHLIST(NUMLIST))=NUMPATH
           NEXTPATH(NUMPATH)=0
             MINFDER=0
             MINSDER=0
             NODE=DESTOD
             DO WHILE (NODE.NE.ORIGIN)
               ARC=PA(NODE)
               CALL DERIVS(COMMODITY,FA(ARC),ARC,D1CAL,D2CAL)
               MINFDER=MINFDER+D1CAL
               MINSDER=MINSDER+D2CAL
               NODE=STARTNODE(ARC)
             END DO
         END IF
C
C      **** UPDATE PATH & LINK FLOWS ****
C
             INCR=0
           TEMPERROR=0
           DO 250 K=1,NUMLIST
        DLENGTH=FDLENGTH(K)-MINFDER
            IF (DLENGTH.GT.0) THEN
              PATH=PATHLIST(K)
                 FLOW=FP(PATH)
              IF ((FLOW.EQ.0.0).AND.(K.GT.1)) THEN
               NEXTPATH(PATHLIST(K-1))=NEXTPATH(PATH)
               GO TO 250
              END IF
              PATHINCR=DLENGTH/(SDLENGTH(K)+MINSDER)
              IF (FLOW.LE.PATHINCR) THEN
               FP(PATH)=0.0
               PATHINCR=FLOW
              ELSE
               FP(PATH)=FLOW-PATHINCR
              END IF
               INCR=INCR+PATHINCR
               TEMPERROR=TEMPERROR+FLOW*DLENGTH/FDLENGTH(K)
              ITER=PATHID(PATH)
              NODE=DESTOD
              DO WHILE (NODE.NE.ORIGIN)
               ARC=PRED(NODE,ITER,COMMODITY)
               FA(ARC)=FA(ARC)-PATHINCR
               NODE=STARTNODE(ARC)
              END DO
            END IF
250         CONTINUE
C
C
C      *** UPDATE THE ERROR CRITERION ***
C
```

```
     CURERROR=AMAX1(CURERROR,TEMPERROR/INPUT_FLOW(OD))
C
C   **** UPDATE FLOWS FOR SHORTEST PATH ****
C
        FP(SHORTEST)=FP(SHORTEST)+INCR
        NODE=DESTOD
        DO WHILE (NODE.NE.ORIGIN)
           ARC=PA(NODE)
           FA(ARC)=FA(ARC)+INCR
           MEMBER(ARC)=.FALSE.
           NODE=STARTNODE(ARC)
        END DO
C
500   CONTINUE
C
C     ***** END OF LOOP FOR OD PAIRS CORRESPONDING TO COMMODITY
C     ***** UPDATE TOTAL DELAY
C
     CALL DELAY(DTOT(NUMITER))
C
1000 CONTINUE
C
C      CHECK IF THE # OF ACTIVE PATHS EXCEED THE ALLOCATED NUMBER
C
      IF (NUMPATH.GT.NNUMPATH) THEN
       WRITE(6,*)'MAX # OF ALLOCATED PATHS EXCEEDED'
       STOP
      END IF
C
C OUTPUT THE CURRENT SOLUTION TO DISK
C
 CALL PRFLOW
C
C  ***** END OF ITERATION *****
C
C  *** IF THE ERROR IS SMALLER THAN TOL, OR THE LIMIT ON
C  THE NUMBER OF ITERATIONS IS REACHED RETURN
C  ELSE GO FOR ANOTHER ITERATION
C
      IF ((CURERROR.LT.TOL).OR.(NUMITER.EQ.MAXITER)) THEN
        RETURN
      ELSE
        GO TO 110
      END IF
C
      END
C  ************** END OF MULTIFLO ****************



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SHORTHEAP
C 'SHORTHEAP' SOLVES THE SHORTEST PATH PROBLEM BY
C DIJKSTRA'S ALGORITHM AND A HEAP DATA STRUCTURE.
C      THIS ALGORITHM SHOULD BE USED WHEN THE NUMBER OF
C      DESTINATIONS FOR EACH COMMODITY IS SMALL RELATIVE
C      TO THE TOTAL NUMBER OF NODES.
```

```
C
C  INPUT:
C  S - THE STARTING NODE
C      COMMODITY - THE CORRESPONDING COMMODITY
C
C  OUTPUT:
C  PA(I) - THE LAST ARC ON THE SHORTEST PATH ENDING AT NODE I
C  DIST(I) - THE SHORTEST DISTANCE TO NODE I
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SP(S,COMMODITY)
C
C ****************** INCLUDE COMMON BLOCKS **********************
C
C      'INCLUDE' FILE PARAM.DIM
C
C       'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C ******************* NETWORK PARAMETERS **********************
C
      PARAMETER (NNN=100)
C     MAXIMUM NUMBER OF NODES
      PARAMETER (NNA=500)
C     MAXIMUM NUMBER OF ARCS
      PARAMETER (NNUMOD=1000)
C     MAXIMUM NUMBER OF OD PAIRS
      PARAMETER (NNUMPATH=10000)
C     MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER (NMAXITER=50)
C     MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER (NNORIG=100)
C     MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C


C      'INCLUDE' FILE PATHS.BLK
C
C      'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C      PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &    NUMPATH,NUMITER

      INTEGER*2 PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL    FA(NNA)
C     THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER  PATHID(NNUMPATH)
C     THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER  NEXTPATH(NNUMPATH)
C            THE NEXT PATH FOR THE SAME OD PAIR
      REAL    FP(NNUMPATH)
C            THE FLOW OF A PATH
      REAL    DIST(NNN)
C            SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
```

```
      REAL  DTOT(NMAXITER)
C     THE TOTAL DELAY BY ITERATION
      INTEGER  NUMITER
C   CURRENT ITERATION NUMBER
      REAL  CURERROR
C     CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C              A SHORTEST PATH)
      INTEGER  NUMPATH
C     NUMBER OF GENERATED PATHS


C      'INCLUDE' FILE NETWRK.PRM
C
C 'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &  NN,FRSTOU,LASTOU,
     &  NA,STARTNODE,ENDNODE,BITRATE,
     &  NUMCOMMOD,ORGID,STARTOD,
     &  NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2 NN
C     NUMBER OF NODES IN THE NETWORK
      INTEGER*2  FRSTOU(NNN)
C     THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2 LASTOU(NNN)
C     THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2 NA
C     NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2 STARTNODE(NNA)
C     THE START NODE FOR AN ARC
      INTEGER*2 ENDNODE(NNA)
C     THE END NODE FOR AN ARC
      REAL  BITRATE(NNA)
C     THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2 NUMCOMMOD
C     THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2 ORGID(NNORIG)
C     THE NODE NUMBER OF THE ORIGIN
      INTEGER*2  STARTOD(NNORIG)
C     THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2 NUMODPAIR
C     THE NUMBER OF OD PAIRS
      INTEGER*2 DEST(NNUMOD)
C     THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL  INPUT_FLOW(NNUMOD)
C     THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C
C
 IMPLICIT NONE
C
C      *****************  LOCAL VARIABLE DEFINITIONS  *********************
C
      REAL  MIN
C     TEMPORARY MINIMUM VALUE
      REAL  D1,D2,DP
C             NODE DISTANCE
```

```
      REAL XLARGE
C     BIG X BY DEFAULT
      INTEGER S
C     INPUT NODE
      INTEGER COMMODITY
C           INPUT COMMODITY
      INTEGER P
C     NODE ALONG THE PATH OF S TO DESTINATIONS
      INTEGER I
C     DO LOOP INDEX
      INTEGER J
C     DO LOOP INDEX
      INTEGER ARC
C     DO LOOP INDEX
      INTEGER ND
C     A NODE INDEX
      INTEGER DNUMBER
C           # OF DESTINATIONS FOR COMMODITY
      INTEGER N1
C           TEMPORARY VARIABLE
      INTEGER N2
C           TEMPORARY VARIABLE
      INTEGER UPNODE,DOWNNODE,DOWNNODE1,LASTNODE
C           VARIABLES USED IN UPDATING THE HEAP ARRAY
      INTEGER CURRANK,NEWRANK
C           VARIABLES USED IN UPDATING THE HEAP ARRAY
      INTEGER ENDHEAP
C           MARKS THE LAST ELEMENT OF THE HEAP ARRAY
      INTEGER RANK(NNN)
C           RANK(NODE) GIVES THE RANK OF NODE IN THE HEAP
      INTEGER NRANK(NNN)
C           NRANK(I) GIVES THE NODE OF RANK I IN THE HEAP
      REAL D1CAL
C     FIRST DERIVATIVE OF DELAY WITH RESPECT TO LOAD
      LOGICAL FIRSTITER
C           TRUE IF THIS IS THE FIRST ITERATION
      LOGICAL SCAN(NNN)
C           LOGICAL INDICATING THAT A NODE HAS BEEN SCANNED
      LOGICAL DSTATUS(NNN)
C           LOGICAL SPECIFYING IF A NODE IS A DESTINATION
C
C     ******************  EXECUTABLE  CODE  ******************************
C
      XLARGE=1E15
      D1CAL=1.0
      P=S
      DO 10 I=1,NN
         DIST(I)=XLARGE
         SCAN(I)=.FALSE.
         DSTATUS(I)=.FALSE.
10    CONTINUE
      DIST(S)=0
      IF (NUMITER.EQ.1) THEN
        FIRSTITER=.TRUE.
      ELSE
        FIRSTITER=.FALSE.
      END IF
C
C      MARK THE DESTINATION NODES
C
```

```
      N1=STARTOD(COMMODITY)
      N2=STARTOD(COMMODITY+1)-1
      DNUMBER=N2-N1+1
      DO 15 I=N1,N2
        DSTATUS(DEST(I))=.TRUE.
15      CONTINUE
C
C     INITIALIZE THE HEAP FLOOR
C
     ENDHEAP=0
C
C    ***** SCAN NODE P *****
C
1000   CONTINUE
     SCAN(P)=.TRUE.
          IF (DSTATUS(P)) THEN
            IF (DNUMBER.EQ.1) RETURN
            DNUMBER=DNUMBER-1
          END IF
         IF (FRSTOU(P).NE.0) THEN
            DP=DIST(P)
        DO 20 ARC=FRSTOU(P),LASTOU(P)
       ND=ENDNODE(ARC)
         IF (.NOT.SCAN(ND)) THEN
                IF (.NOT.FIRSTITER) THEN
      CALL DERIV1(COMMODITY,FA(ARC),ARC,D1CAL)
                END IF
                D2=DIST(ND)
C       IF ND HAS NOT BEEN LABELLED INSERT IT IN THE HEAP
                IF (D2.EQ.XLARGE) THEN
                  ENDHEAP=ENDHEAP+1
                  RANK(ND)=ENDHEAP
                  NRANK(ENDHEAP)=ND
                END IF
           D1=DP+D1CAL
                IF (D1.LT.D2) THEN
             PA(ND)=ARC
             DIST(ND)=D1
                   CURRANK=RANK(ND)
50               NEWRANK=INT(CURRANK/2)
               IF (NEWRANK.GE.1) THEN
                 UPNODE=NRANK(NEWRANK)
                 IF (D1.LT.DIST(UPNODE)) THEN
                   NRANK(CURRANK)=UPNODE
                   RANK(UPNODE)=CURRANK
                   CURRANK=NEWRANK
                   GO TO 50
                 END IF
               END IF
               NRANK(CURRANK)=ND
               RANK(ND)=CURRANK
           END IF
        END IF
20            CONTINUE
         END IF
C
C     ******* FIND NEXT NODE TO SCAN *******
C
C        TEST FOR ERROR
         IF (ENDHEAP.EQ.0) THEN
```

```
          WRITE(6,*) 'ERROR IN THE SHORTEST PATH ROUTINE'
          STOP
        END IF
        P=NRANK(1)
C
C        RESTRUCTURE HEAP ARRAYS
C
        LASTNODE=NRANK(ENDHEAP)
        ENDHEAP=ENDHEAP-1
        D1=DIST(LASTNODE)
        CURRANK=1
100       NEWRANK=CURRANK+CURRANK
        IF (NEWRANK.LE.ENDHEAP) THEN
          DOWNNODE=NRANK(NEWRANK)
           IF (NEWRANK.EQ.ENDHEAP) THEN
             DOWNNODE1=DOWNNODE
           ELSE
             DOWNNODE1=NRANK(NEWRANK+1)
           END IF
          IF (DIST(DOWNNODE).LE.DIST(DOWNNODE1)) THEN
            IF (D1.GT.DIST(DOWNNODE)) THEN
              NRANK(CURRANK)=DOWNNODE
              RANK(DOWNNODE)=CURRANK
              CURRANK=NEWRANK
              GO TO 100
            END IF
          ELSE
            IF (D1.GT.DIST(DOWNNODE1)) THEN
              NRANK(CURRANK)=DOWNNODE1
              RANK(DOWNNODE1)=CURRANK
              CURRANK=NEWRANK+1
              GO TO 100
            END IF
          END IF
        END IF
        NRANK(CURRANK)=LASTNODE
        RANK(LASTNODE)=CURRANK
        GO TO 1000
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C DCAL
C
C 'DCAL' COMPUTES THE DELAY ACROSS A SPECIFIED ARC GIVEN THE FLOW.
C THE DELAY IS ASSUMED TO BE CONSISTENT WITH M/M/1 QUEUEING FOR
C FLOWS BELOW A MAXIMUM UTILIZATION AND QUADRATIC BEYOND WITH
C CONTINUITY IN THE DERIVATIVES AT THE MAXIMUM UTILIZATION.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
 REAL FUNCTION DCAL(X,ARC)

C
C ******************  INCLUDE COMMON BLOCKS  ***********************
C
C      'INCLUDE' FILE PARAM.DIM
```

```
C
C       'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C*******************  NETWORK PARAMETERS  ***********************
C
      PARAMETER (NNN=100)
C     MAXIMUM NUMBER OF NODES
      PARAMETER (NNA=500)
C     MAXIMUM NUMBER OF ARCS
      PARAMETER (NNUMOD=1000)
C     MAXIMUM NUMBER OF OD PAIRS
      PARAMETER (NNUMPATH=10000)
C     MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER (NMAXITER=50)
C     MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER (NNORIG=100)
C     MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C



C       'INCLUDE' FILE PATHS.BLK
C
C       'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C       PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &     NUMPATH,NUMITER

      INTEGER*2 PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL  FA(NNA)
C     THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER  PATHID(NNUMPATH)
C     THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER  NEXTPATH(NNUMPATH)
C             THE NEXT PATH FOR THE SAME OD PAIR
      REAL  FP(NNUMPATH)
C             THE FLOW OF A PATH
      REAL  DIST(NNN)
C             SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL  DTOT(NMAXITER)
C     THE TOTAL DELAY BY ITERATION
      INTEGER  NUMITER
C   CURRENT ITERATION NUMBER
      REAL  CURERROR
C     CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C              A SHORTEST PATH)
      INTEGER  NUMPATH
C     NUMBER OF GENERATED PATHS


C       'INCLUDE' FILE NETWRK.PRM
C
C 'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
```

```
     &  NN,FRSTOU,LASTOU,
     &  NA,STARTNODE,ENDNODE,BITRATE,
     &  NUMCOMMOD,ORGID,STARTOD,
     &  NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2 NN
C   NUMBER OF NODES IN THE NETWORK
      INTEGER*2  FRSTOU(NNN)
C   THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2 LASTOU(NNN)
C   THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2 NA
C   NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2 STARTNODE(NNA)
C   THE START NODE FOR AN ARC
      INTEGER*2 ENDNODE(NNA)
C   THE END NODE FOR AN ARC
      REAL  BITRATE(NNA)
C   THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2 NUMCOMMOD
C   THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2 ORGID(NNORIG)
C   THE NODE NUMBER OF THE ORIGIN
      INTEGER*2  STARTOD(NNORIG)
C   THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2 NUMODPAIR
C   THE NUMBER OF OD PAIRS
      INTEGER*2 DEST(NNUMOD)
C   THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL  INPUT_FLOW(NNUMOD)
C   THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C      'INCLUDE' FILE CONVRG.PRM
C
C 'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &  MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C  MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL TOL
C  TOLERANCE ON SOLUTION ACCURACY
      REAL MAXUTI
C  MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C  OUTPUT CONTROL VARIABLE
C
C**********  ARGUMENT DEFINITIONS  **************************
C
 REAL  X
C  INPUT FLOW FOR THE ARC
 INTEGER ARC
```

```
C		INPUT ARC
C
C	****************  LOCAL VARIABLE DEFINITIONS  ********************
C
	REAL	RATE
C		MAXIMUM LINK CAPACITY
	REAL	Y
C		TEMPORARY VARIABLE
	REAL	Z
C		TEMPORARY VARIABLE
	REAL 	Q0
C		ZEROTH ORDER TERM IN THE QUADRATIC APPROXIMATION FOR
C		OVERLOADED LINKS
	REAL	Q1
C		FIRST ORDER TERM IN THE QUADRATIC APPROXIMATION
	REAL	Q2
C		SECOND ORDER TERM IN THE QUADRATIC APPROXIMATION
	REAL	EXCESS
C		FLOW BEYOND THE MAXIMUM ALLOWABLE UTILIZATION
C
C	**********************  EXECUTABLE CODE  ****************************
C
	RATE=BITRATE(ARC)
	Y=MAXUTI*RATE
C
C	M/M/1 DELAY
C
	IF(X.LT.Y) THEN
	   DCAL=X/(RATE-X)
	ELSE
C
C	   QUADRATIC APPROXIMATION TO AVOID OVERFLOWS
C
	   EXCESS=X-Y
	   Z=RATE-Y
	   Q0=Y/Z
	   Q1=Q0/(MAXUTI*Z)
	   Q2=Q1/Z
	   DCAL=Q0+Q1*EXCESS+Q2*EXCESS**2
	ENDIF
	RETURN
	END



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C	
C	DERIVS
C
C	'DERIVS' COMPUTES THE DERIVATIVES OF DELAY WITH RESPECT TO FLOW FOR
C	LINKS.  BELOW A MAXIMUM UTILIZATION, M/M/1 DELAY IS ASSUMED TO APPLY
C	WHEREAS A QUADRATIC APPROXIMATION IS ASSUMED FOR UTILIZATIONS BEYOND
C	THE MAXIMUM.  THE DERIVATIVES ARE CONTINUOUS AT THE MAXIMUM
C	UTILIZATION.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
	SUBROUTINE DERIVS(COMMODITY,X,ARC,D1CAL,D2CAL)

C
```

```
C□******************  INCLUDE COMMON BLOCKS  ************************
C
C      'INCLUDE' FILE PARAM.DIM
C
C       'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C□******************  NETWORK PARAMETERS  ***********************
C
      PARAMETER□(NNN=100)
C□□□MAXIMUM NUMBER OF NODES
      PARAMETER□(NNA=500)
C□□□MAXIMUM NUMBER OF ARCS
      PARAMETER□(NNUMOD=1000)
C□□□MAXIMUM NUMBER OF OD PAIRS
      PARAMETER□(NNUMPATH=10000)
C□□□MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER□(NMAXITER=50)
C□□□MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER□(NNORIG=100)
C□□□MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C



C      'INCLUDE' FILE PATHS.BLK
C
C      'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C      PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &□□NUMPATH,NUMITER

      INTEGER*2□PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL□□FA(NNA)
C□□□THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER □PATHID(NNUMPATH)
C□□□THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER □NEXTPATH(NNUMPATH)
C            □THE NEXT PATH FOR THE SAME OD PAIR
      REAL□□FP(NNUMPATH)
C            □THE FLOW OF A PATH
      REAL □□DIST(NNN)
C            □SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL□□DTOT(NMAXITER)
C□□□THE TOTAL DELAY BY ITERATION
      INTEGER □NUMITER
C   CURRENT ITERATION NUMBER
      REAL□□CURERROR
C□□□CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C             □A SHORTEST PATH)
      INTEGER □NUMPATH
C□□□NUMBER OF GENERATED PATHS


C      'INCLUDE' FILE NETWRK.PRM
C
```

```
C□'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &□□NN,FRSTOU,LASTOU,
     &□□NA,STARTNODE,ENDNODE,BITRATE,
     &□□NUMCOMMOD,ORGID,STARTOD,
     &□□NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2□NN
C□□□NUMBER OF NODES IN THE NETWORK
      INTEGER*2 □FRSTOU(NNN)
C□□□THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2□LASTOU(NNN)
C□□□THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2□NA
C□□□NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2□STARTNODE(NNA)
C□□□THE START NODE FOR AN ARC
      INTEGER*2□ENDNODE(NNA)
C□□□THE END NODE FOR AN ARC
      REAL□□BITRATE(NNA)
C□□□THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2□NUMCOMMOD
C□□□THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2□ORGID(NNORIG)
C□□□THE NODE NUMBER OF THE ORIGIN
      INTEGER*2 □STARTOD(NNORIG)
C□□□THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2□NUMODPAIR
C□□□THE NUMBER OF OD PAIRS
      INTEGER*2□DEST(NNUMOD)
C□□□THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL□□INPUT_FLOW(NNUMOD)
C□□□THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C      'INCLUDE' FILE CONVRG.PRM
C
C□'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C□NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &□□MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C□□MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL□TOL
C□□TOLERANCE ON SOLUTION ACCURACY
      REAL□MAXUTI
C□□MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C□□OUTPUT CONTROL VARIABLE


□IMPLICIT NONE
```

```
C
C ******************* ARGUMENT DEFINITIONS *************************
C
C ON INPUT:
      INTEGER COMMODITY
C           THE CORRESPONDING COMMODITY
C
 REAL  X
C   FLOW IN THE SPECIFIED LINK
 INTEGER  ARC
C    THE SPECIFIED LINK
C
C ON OUTPUT:
C
 REAL D1CAL
C   ARC LENGTH (1ST DERIVATIVE OF DELAY)
 REAL D2CAL
C   FIRST DERIVATIVE OF ARC LENGTH
C
C **************** LOCAL VARIABLE DEFINITIONS ***********************
C
 REAL MAXI
C   MAXIMUM ALLOWABLE FLOW FOR LINK FOR M/M/1 QUEUEING DELAY
 REAL RATE
C   THE MAXIMUM FLOW CAPACITY FOR THE LINK
 REAL EXCESS
C   FLOW BEYOND THE MAXIMUM ALLOWABLE FLOW
 REAL D1
C   TEMPORARY VARIABLE
      REAL  T
C           TEMPORARY VARIABLE
C
C ******************** EXECUTABLE CODE *******************************
C
 RATE=BITRATE(ARC)
 MAXI=MAXUTI*RATE
 EXCESS=X-MAXI
C
 IF(EXCESS.LE.0.0) THEN
C
C    DERIVATIVES OF M/M/1 QUEUEING DELAY
C
         T=RATE-X
     D1CAL=RATE/T**2
     D2CAL=2.0*D1CAL/T
 ELSE
C
C    DERIVATIVES OF THE QUADRATIC APPROXIMATION
C
         T=RATE-MAXI
     D1=RATE/T**2
     D2CAL=2.0*D1/T
     D1CAL=D1+D2CAL*EXCESS
 END IF
 RETURN
 END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C DERIV1
```

```
C
C 'DERIV1' COMPUTES THE FIRST DERIVATIVE OF DELAY WITH RESPECT
C TO FLOW FOR LINKS.  BELOW A MAXIMUM UTILIZATION, M/M/1 DELAY IS
C ASSUMED TO APPLY WHEREAS A QUADRATIC APPROXIMATION IS ASSUMED FOR
C UTILIZATIONS BEYOND THE MAXIMUM.  THE DERIVATIVES ARE CONTINUOUS
C AT THE MAXIMUM UTILIZATION.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
 SUBROUTINE DERIV1(COMMODITY,X,ARC,D1CAL)

C
C ******************** INCLUDE COMMON BLOCKS ***********************
C
C      'INCLUDE' FILE PARAM.DIM
C
C       'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C ******************* NETWORK PARAMETERS **********************
C
      PARAMETER (NNN=100)
C    MAXIMUM NUMBER OF NODES
      PARAMETER (NNA=500)
C    MAXIMUM NUMBER OF ARCS
      PARAMETER (NNUMOD=1000)
C    MAXIMUM NUMBER OF OD PAIRS
      PARAMETER (NNUMPATH=10000)
C    MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER (NMAXITER=50)
C    MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER (NNORIG=100)
C    MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C



C      'INCLUDE' FILE PATHS.BLK
C
C      'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C      PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &   NUMPATH,NUMITER

      INTEGER*2 PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL  FA(NNA)
C    THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER PATHID(NNUMPATH)
C    THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER NEXTPATH(NNUMPATH)
C           THE NEXT PATH FOR THE SAME OD PAIR
      REAL  FP(NNUMPATH)
C           THE FLOW OF A PATH
      REAL  DIST(NNN)
C           SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL  DTOT(NMAXITER)
```

```
C□□□THE TOTAL DELAY BY ITERATION
      INTEGER □NUMITER
C   CURRENT ITERATION NUMBER
      REAL□□CURERROR
C□□□CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C              □A SHORTEST PATH)
      INTEGER □NUMPATH
C□□□NUMBER OF GENERATED PATHS


C       'INCLUDE' FILE NETWRK.PRM
C
C□'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &□□NN,FRSTOU,LASTOU,
     &□□NA,STARTNODE,ENDNODE,BITRATE,
     &□□NUMCOMMOD,ORGID,STARTOD,
     &□□NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2□NN
C□□□NUMBER OF NODES IN THE NETWORK
      INTEGER*2 □FRSTOU(NNN)
C□□□THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2□LASTOU(NNN)
C□□□THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2□NA
C□□□NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2□STARTNODE(NNA)
C□□□THE START NODE FOR AN ARC
      INTEGER*2□ENDNODE(NNA)
C□□□THE END NODE FOR AN ARC
      REAL□□BITRATE(NNA)
C□□□THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2□NUMCOMMOD
C□□□THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2□ORGID(NNORIG)
C□□□THE NODE NUMBER OF THE ORIGIN
      INTEGER*2 □STARTOD(NNORIG)
C□□□THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2□NUMODPAIR
C□□□THE NUMBER OF OD PAIRS
      INTEGER*2□DEST(NNUMOD)
C□□□THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL□□INPUT_FLOW(NNUMOD)
C□□□THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C       'INCLUDE' FILE CONVRG.PRM
C
C□'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C□NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &□□MAXITER,TOL,MAXUTI,OUTPFL
```

```
C
      INTEGER MAXITER
C     MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL TOL
C     TOLERANCE ON SOLUTION ACCURACY
      REAL MAXUTI
C     MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C     OUTPUT CONTROL VARIABLE


 IMPLICIT NONE

C
C  ******************  ARGUMENT DEFINITIONS  *************************
C
C  ON INPUT:
C
       INTEGER COMMODITY
C             THE CORRESPONDING COMMODITY
C
 REAL  X
C     FLOW IN THE SPECIFIED LINK
       INTEGER ARC
C             THE SPECIFIED ARC
C
C  ON OUTPUT:
C
 REAL D1CAL
C     ARC LENGTH (1ST DERIVATIVE OF DELAY)
C
C  ****************  LOCAL VARIABLE DEFINITIONS  ***********************
C
 REAL MAXI
C     MAXIMUM ALLOWABLE FLOW FOR LINK FOR M/M/1 QUEUEING DELAY
 REAL RATE
C     THE MAXIMUM FLOW CAPACITY FOR THE LINK
 REAL EXCESS
C     FLOW BEYOND THE MAXIMUM ALLOWABLE FLOW
 REAL D1
C     TEMPORARY VARIABLE
       REAL    T
C             TEMPORARY VARIABLE
 REAL D2CAL
C             TEMPORARY VARIABLE
C
C  ********************  EXECUTABLE CODE  ********************************
C
 RATE=BITRATE(ARC)
 MAXI=MAXUTI*RATE
 EXCESS=X-MAXI
C
 IF(EXCESS.LE.0.0) THEN
C
C     DERIVATIVE OF M/M/1 QUEUEING DELAY
C
          T=RATE-X
     D1CAL=RATE/T**2
 ELSE
C
```

```
C    DERIVATIVE OF THE QUADRATIC APPROXIMATION
C
         T=RATE-MAXI
     D1=RATE/T**2
     D2CAL=2.0*D1/T
     D1CAL=D1+D2CAL*EXCESS
  END IF
  RETURN
  END



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  DELAY
C
C  DELAY COMPUTES THE TOTAL M/M/1 DELAY IN ROUTING COMMODITIES FROM
C  SOURCES TO SINKS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
  SUBROUTINE DELAY(DT)
C
C  *****************  INCLUDE COMMON BLOCKS  *************************
C
C       'INCLUDE' FILE PARAM.DIM
C
C        'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C  *******************  NETWORK PARAMETERS  **********************
C
      PARAMETER (NNN=100)
C      MAXIMUM NUMBER OF NODES
      PARAMETER (NNA=500)
C      MAXIMUM NUMBER OF ARCS
      PARAMETER (NNUMOD=1000)
C      MAXIMUM NUMBER OF OD PAIRS
      PARAMETER (NNUMPATH=10000)
C      MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER (NMAXITER=50)
C      MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER (NNORIG=100)
C      MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C



C       'INCLUDE' FILE PATHS.BLK
C
C       'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C       PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &   NUMPATH,NUMITER

      INTEGER*2 PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
```

```
      REAL  FA(NNA)
C     THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER  PATHID(NNUMPATH)
C     THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER  NEXTPATH(NNUMPATH)
C             THE NEXT PATH FOR THE SAME OD PAIR
      REAL  FP(NNUMPATH)
C             THE FLOW OF A PATH
      REAL  DIST(NNN)
C             SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL  DTOT(NMAXITER)
C     THE TOTAL DELAY BY ITERATION
      INTEGER  NUMITER
C   CURRENT ITERATION NUMBER
      REAL  CURERROR
C     CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C              A SHORTEST PATH)
      INTEGER  NUMPATH
C     NUMBER OF GENERATED PATHS


C      'INCLUDE' FILE NETWRK.PRM
C
C 'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &  NN,FRSTOU,LASTOU,
     &  NA,STARTNODE,ENDNODE,BITRATE,
     &  NUMCOMMOD,ORGID,STARTOD,
     &  NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2 NN
C     NUMBER OF NODES IN THE NETWORK
      INTEGER*2  FRSTOU(NNN)
C     THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2 LASTOU(NNN)
C     THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2 NA
C     NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2 STARTNODE(NNA)
C     THE START NODE FOR AN ARC
      INTEGER*2 ENDNODE(NNA)
C     THE END NODE FOR AN ARC
      REAL  BITRATE(NNA)
C     THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2 NUMCOMMOD
C     THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2 ORGID(NNORIG)
C     THE NODE NUMBER OF THE ORIGIN
      INTEGER*2  STARTOD(NNORIG)
C     THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2 NUMODPAIR
C     THE NUMBER OF OD PAIRS
      INTEGER*2 DEST(NNUMOD)
C     THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL  INPUT_FLOW(NNUMOD)
C     THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
```

```
C



C      'INCLUDE' FILE CONVRG.PRM
C
C 'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
    &  MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C  MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL TOL
C  TOLERANCE ON SOLUTION ACCURACY
      REAL MAXUTI
C  MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
      LOGICAL OUTPFL
C  OUTPUT CONTROL VARIABLE

C
 IMPLICIT NONE
C ********************  ARGUMENT DEFINITIONS  ************************
C
C ON OUTPUT:
C
 REAL DT
C  TOTAL SYSTEM DELAY
C
C ******************  EXTERNAL FUNCTIONS REFERENCED  ******************
C
 REAL  DCAL
C  DELAY AS A FUNCTION OF FLOW
C
C ******************  LOCAL VARIABLE DEFINITIONS  *********************
C
 INTEGER K
C  DO LOOP INDEX
C
C ***********************  EXECUTABLE CODE  ***************************
C
C      LOOP OVER ALL LINKS AND ACCRUE TOTAL DELAY
C
 DT=0.
 DO 50 K=1,NA
    DT=DT+DCAL(FA(K),K)
 50 CONTINUE
C
 RETURN
 END



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C PRFLOW
C
C 'PRFLOW' OUTPUTS INTERMEDIATE RESULTS IN THE MULTIFLO ALGORITHM.
C ITERATION #, DELAY, NUMBER OF ACTIVE PATHS GENERATED AND
```

```
C CONVERGENCE ARE THE PRIMARY OUTPUTS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
 SUBROUTINE PRFLOW
C **************** INCLUDE COMMON BLOCKS **************************
C
C      'INCLUDE' FILE PARAM.DIM
C
C       'PARAM.DIM' CONTAINS THE ARRAY DIMENSIONS
C
C ******************* NETWORK PARAMETERS ***********************
C
      PARAMETER (NNN=100)
C   MAXIMUM NUMBER OF NODES
      PARAMETER (NNA=500)
C   MAXIMUM NUMBER OF ARCS
      PARAMETER (NNUMOD=1000)
C   MAXIMUM NUMBER OF OD PAIRS
      PARAMETER (NNUMPATH=10000)
C   MAXIMUM NUMBER OF PATHS FOR CONSIDERATION
      PARAMETER (NMAXITER=50)
C   MAXIMUM NUMBER OF ITERATIONS ALLOWED
      PARAMETER (NNORIG=100)
C   MAXIMUM NUMBER OF COMMODITIES
      PARAMETER (NINDEX=100000)
C   MAXIMUM NUMBER OF ELEMENTS OF PATH
C   DESCRIPTION ARRAY (USED IN MULTIFLO1)
C


C      'INCLUDE' FILE PATHS.BLK
C
C      'PATHS.BLK' DEFINES THE ARRAYS NECESSARY TO MAINTAIN
C      PATH FLOWS AND DESCRIPTION.
C
      COMMON /PATHS/ PA,FA,PATHID,NEXTPATH,FP,DIST,DTOT,CURERROR,
     &  NUMPATH,NUMITER

      INTEGER*2 PA(NNN)
C   THE LAST ARC ON A SHORTEST PATH TO A NODE
      REAL  FA(NNA)
C   THE FLOW IN ANY GIVEN LINK (ARC)
      INTEGER PATHID(NNUMPATH)
C   THE PATH IDENTIFIER FOR ANY GIVEN PATH
      INTEGER NEXTPATH(NNUMPATH)
C           THE NEXT PATH FOR THE SAME OD PAIR
      REAL  FP(NNUMPATH)
C           THE FLOW OF A PATH
      REAL  DIST(NNN)
C           SHORTEST DISTANCE TO A NODE FROM THE ORIGIN
      REAL  DTOT(NMAXITER)
C   THE TOTAL DELAY BY ITERATION
      INTEGER NUMITER
C   CURRENT ITERATION NUMBER
      REAL  CURERROR
C   CONVERGENCE ERROR (NORMALISED % OF FLOW NOT ON
C            A SHORTEST PATH)
      INTEGER NUMPATH
```

```
C□□□NUMBER OF GENERATED PATHS


C      'INCLUDE' FILE NETWRK.PRM
C
C□'NETWRK.PRM' CONTAINS THE NETWORK SPECIFICATION PARAMETERS
C
      COMMON /NETWORK/
     &□□NN,FRSTOU,LASTOU,
     &□□NA,STARTNODE,ENDNODE,BITRATE,
     &□□NUMCOMMOD,ORGID,STARTOD,
     &□□NUMODPAIR,DEST,INPUT_FLOW
C
      INTEGER*2□NN
C□□□NUMBER OF NODES IN THE NETWORK
      INTEGER*2 □FRSTOU(NNN)
C□□□THE FIRST ARC EMANATING FROM A NODE
      INTEGER*2□LASTOU(NNN)
C□□□THE FINAL ARC EMANATING FROM A NODE
C
      INTEGER*2□NA
C□□□NUMBER OF LINKS (ARCS) IN THE NETWORK
      INTEGER*2□STARTNODE(NNA)
C□□□THE START NODE FOR AN ARC
      INTEGER*2□ENDNODE(NNA)
C□□□THE END NODE FOR AN ARC
      REAL□□BITRATE(NNA)
C□□□THE LINK CAPACITY IN BITS/SECOND
C
      INTEGER*2□NUMCOMMOD
C□□□THE NUMBER OF COMMODITIES IN THE NETWORK
      INTEGER*2□ORGID(NNORIG)
C□□□THE NODE NUMBER OF THE ORIGIN
      INTEGER*2 □STARTOD(NNORIG)
C□□□THE POINTER TO THE STARTING NODE IN AN OD PAIR
C
      INTEGER*2□NUMODPAIR
C□□□THE NUMBER OF OD PAIRS
      INTEGER*2□DEST(NNUMOD)
C□□□THE DESTINATION NODE OF TRAFFIC IN AN OD PAIR
      REAL□□INPUT_FLOW(NNUMOD)
C□□□THE INPUT TRAFFIC TO THE NODE IN BITS/SECOND
C



C      'INCLUDE' FILE CONVRG.PRM
C
C□'CONVRG.PRM' CONTAINS THE CONVERGENCE PARAMETERS FOR THE
C□NETWORK FLOW PROBLEM
C
      COMMON /CONVRG/
     &□□MAXITER,TOL,MAXUTI,OUTPFL
C
      INTEGER MAXITER
C□□MAXIMUM NUMBER OF ITERATIONS IN THE SOLUTION
      REAL□TOL
C□□TOLERANCE ON SOLUTION ACCURACY
      REAL□MAXUTI
C□□MAXIMUM UTILIZATION FOR M/M/1 QUEUE DELAY
```

```
      LOGICAL OUTPFL
C  OUTPUT CONTROL VARIABLE

C
C  **************  LOCAL VARIABLE DEFINITIONS  ***********************
C
 IMPLICIT NONE
C
 LOGICAL FIRFLG
C  FIRST PASS FLAG FOR OUTPUT CONTROL
 INTEGER I
C  DO LOOP INDEX
C
C  *****************  LOCAL DATA INITIALIZATION  *******************
C
 DATA FIRFLG/.TRUE./
C
C  ON THE VERY FIRST PASS, OUTPUT THE CONTENTS OF INPUT BLOCKS TO FILE
C
 IF(FIRFLG) THEN
    WRITE(6,*)'****************************************'
    WRITE(6,*)'*       MULTIFLO SUMMARY             *'
    WRITE(6,*)'****************************************'
    WRITE(6,*)' '
    WRITE(6,*)'****************************************'
    WRITE(6,*)'*       INITIALIZATION DATA          *'
    WRITE(6,*)'****************************************'
    WRITE(6,*)' '
    WRITE(6,*)'NETWORK SPECIFICATION DATA:'
    WRITE(6,*)' '
    WRITE(6,*)'NODE SPECIFICATIONS'
    WRITE(6,*)'NUMBER OF NODES:',NN
    WRITE(6,*)'NODE #         FRSTOU       LASTOU'
    DO 10 I=1,NN
       WRITE(6,*)I,FRSTOU(I),LASTOU(I)
10   CONTINUE
    WRITE(6,*)' '
    WRITE(6,*)'LINK SPECIFICATIONS:'
    WRITE(6,*)'NUMBER OF LINKS:',NA
    WRITE(6,*)'LINK #       STARTNODE     ENDNODE    BITRATE'
    DO 20 I=1,NA
       WRITE(6,*)I,STARTNODE(I),ENDNODE(I),BITRATE(I)
20   CONTINUE
    WRITE(6,*)' '
    WRITE(6,*)'COMMODITY SPECIFICATIONS'
    WRITE(6,*)'NUMBER OF COMMODITIES:',NUMCOMMOD
    WRITE(6,*)'COMMOD #       ORGID          STARTOD'
    DO 30 I=1,NUMCOMMOD
       WRITE(6,*)I,ORGID(I),STARTOD(I)
30   CONTINUE
    WRITE(6,*)' '
    WRITE(6,*)'OD PAIR SPECIFICATIONS'
    WRITE(6,*)'NUMBER OF OD PAIRS:   ',NUMODPAIR
    WRITE(6,*)'OD PAIR #      DEST          INPUT FLOW'
    DO 40 I=1,NUMODPAIR
       WRITE(6,*)I,DEST(I),INPUT_FLOW(I)
40   CONTINUE
    WRITE(6,*)' '
    WRITE(6,*)'****************************************'
    WRITE(6,*)'*       MULTIFLO DATA BY ITERATION    *'
```

```
    WRITE(6,*)'****************************************'
    WRITE(6,*)'ITERATION #  TOTAL DELAY  CONVERGENCE NUMBER OF'
    WRITE(6,*)'                          ERROR      ACTIVE'
    WRITE(6,*)'                                     PATHS'
    FIRFLG=.FALSE.
  END IF
  IF(NUMITER.GT.0) THEN
    WRITE(6,*)NUMITER,DTOT(NUMITER),CURERROR,NUMPATH
  END IF

  RETURN
  END
```