US Army Corps of Engineers
Water Resources Support Center
Institute for Water Resources

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

DURUM WHEAT

SEPTEMBER 1990

IWR REPORT 90-R-3

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

DURUM WHEAT

Prepared By:

Stephen W. Fuller
Warren R. Grant
Luis R. Fellin
Swee Hor Teh
Nicolas Gutieerez

Department of Agricultural Economics
Texas Agricultural Experiment Station
The Texas A&M University System
College Station, TX 77843

Commissioned By:

U.S. Army Corps of Engineers
Water Resources Support Center
Institute for Water Resources
Casey Building
Ft. Belvoir, VA 22060-5586

September 1990 IWR Report-90-R-3

Content

INTRODUCTION

This report documents the durum wheat model. Additional reports document corn, soybeans, hard red winter wheat, soft wheat, hard red spring wheat, and grain sorghum models A tutorial report and model and data requirements report are published separately.

In this report, the documented programs, data files and output listing are included For the compilation and execution of the network model, the three Fortran 77 programs and six data files are presented. The programs have to be compiled and run in a sequential order (program1 followed by program2, etc) The data must be entered into the corresponding data files

The documented FORTRAN 77 programs and grain related data files used in the model are provided. However, the documented programs and data files cannot be used to execute the model. In each program or data file, explanations are included to provide more detail to the user.

The intermediate output listing and final output listing are included in this report. The intermediate output listing is printed in order to explain the feasibility of the data provided. The final output listings show the optimal result of the transportation network model

The model was calibrated with Federal Grain Inspection Service, U.S Department of Agriculture, trade data. Special adjustments used in this calibration are noted in the calibration section. The calibrated model data and source code programs are included on the attached diskette.

ACKNOWLEDGEMENTS

This research was supported through USDA Cooperative Agreement Number 58-3AEK-8-00120 with resources made available by the Water Resources Support Center - Institute of Water Resources, U.S. Army Corps of Engineers, Casey Building, Ft. Belvoir, Virginia. Special appreciation is expressed to Dr. Lloyd G. Antle, Chief, Navigation Division and Ms. Shilpa Patel, Computer Systems Analyst, both with the Water Resources Support Center - Institute for Water Resources. In addition, appreciation is expressed to personnel in District Offices, U.S. Army Corps of Engineers, who offered valuable comments during our research effort. Appreciation is also expressed to Dr. Lester H Myers, Branch Chief, USDA-CED-ERS-FMCE.

```
C ----  ***************************************************************
C ----  *                      DURUM1.F                                 *
C ----  *        Documented on                 Sept  1990              *
C ----  *        Run on                        a 386 machine           *
C ----  *        Compiler used                 NDP Fortran             *
C ----  *        Grain used                    DURUM   (DRM).          *
C ----  *        Data files used               F1,F2,F3,F4, and F8.    *
C ----  *        Trace file                    U6                      *
C ----  *        Input file for DURUM2.f       U12                     *
C ----  ***************************************************************


C ----  Declaration.
C ----  The array size used is 27500.


       COMMON /G0/ K, OPERATION
       COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
       COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
       INTEGER     UPPR
       INTEGER U1,U2,U3,U4,U8,U9,U6,U12


C ----  Unit number for each file.
C ----  U1 is the unit number for data file, F1.drm.
C ----  F1.drm contains information about the model.
C ----  U2 is the unit number for data file, F2.drm.
C ----  F2.drm contains information on TRUCK mileage.
C ----  U3 is the unit number for data file, F3.drm.
C ----  F3.drm contains information on RAIL costs.
C ----  U4 is the unit number for data file,F4.drm.
C ----  F4.drm contains information on BARGE costs.
C ----  U8 is the unit number for data file, F8.drm.
C ----  F8.drm contains information about SUPPLIES and DEMANDS.
C ----  U9 is for data files, F9.drm.
C ----  F9.drm contains the NAMES of all the regions.
C ----  U6 is the unit number for a temporary trace file.
C ----  The trace output file is used to check that the program is
C ----  working correctly.
C ----  U12 is the unit number for the output file.
C ----  The output produced is used as an input file for the next
C ----  program, DURUM2.


       U1 = 13
       U2 = 14
       U3 = 15
       U4 = 16
       U8 = 8
       U9 = 9
       U6 = 17
       U12 = 12
```

```
C ----  Diagram used to show the flow of this program, DURUM1.F:
C ----
C ----                    U1,U2,U3,U4,U8,U9 (input files)
C ----                             |
C ----                             |
C ----                             V
C ----                        DURUM2.F  (program)
C ----                             |
C ----                           /   \
C ----                          /     \
C ----                         V       V
C ----                    U6 (trace)   U12 (input to DURUM2.F)

C ----  To open files for reading and writing

      OPEN (UNIT = U1, FILE = 'F1.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U2, FILE = 'F2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U3, FILE = 'F3.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U4, FILE = 'F4.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U8, FILE = 'F8.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U6, FILE = 'FILE06_2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U12, FILE = 'FILE12_2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')


C ----  Initialisation and main program.
C ----  INFN is a large number used to initialise the amount produced
C ----  in each region.
C ----  LOWR is the amount of grain demanded in this model.
C ----  UPPR is the amount of grain supplied in this model.
C ----  KOST is the cost of transportation for each region.
C ----  DRIVER is a subroutine used to read in the data files and
C ----  invoke other subroutines, SURPLS, DEMAND, WRITER, PELVTR,
C ----  and RELVTR.

      INFN       = 99999999
      DO 1100 K  = 1, 27500
      LOWR(K)    = 0
      UPPR(K)    = INFN
      KOST(K)    = 0
 1100 CONTINUE
      K          = 0
      CALL DRIVER
      STOP
      END
```

```
C-----------------------------------------------------------------------
C                 =====    WRITER    ======
C ----  Write all nodes, iarcs, jarcs, lowr, uppr and cost
C ----  to the data file in the unit number U12.

      SUBROUTINE WRITER ( N, NODE, I, J, K, L, M )
      DIMENSION   I(N), J(N), K(N), L(N), M(N)
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12

C ----  Write N(= number of arcs), NODE(= number of nodes),
C ----  I(= IARC), J(= JARC), K=(= LOWR), L(= UPPER), M(=KOST),
C ----  N(=NODES) to the next program, DURUM2.F.

      WRITE (U12,510) N, NODE, I, J, K, L, M, N
  510 FORMAT ( 2018 )
      ENDFILE U12
      RETURN
      END


C-----------------------------------------------------------------------
C                 =====    DRIVER    ======
C ----  Read in data and invoke other subroutines.
C ----  Read in the number of surplus, deficit, river
C ----  and port locations and their code names.
C ----  Calculate the number of nodes connected.
C ----  Read in the loading and unloading costs.

      SUBROUTINE  DRIVER
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /A4/ SINK, SRCE
      COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20) /B5/ FRGN(25)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      DIMENSION   TITL(16)
      INTEGER     TITL
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     SINK, SRCE
      INTEGER     SRGN, DRGN, RIVR, PORT, FRGN
      INTEGER     UPPR, ARCS
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12

C ----  Outline the layout of the output.

  500 FORMAT ( 20A4 )
  510 FORMAT ( 2014 )
  520 FORMAT ( 10F8.3 )
  600 FORMAT ( 1H1, 5X, 'NETWORK GENERATOR', /,
     1          6X, 'FOR GRAIN SHIPMENT PROBLEM', // )
```

```
  610 FORMAT ( 6X, 'SUPPLY', I15, 6X, 'DEMAND', I15, / )
  620 FORMAT ( 6X, '??????????????????????????????????????????', /,
     1         6X, 'INFEASIBLE NETWORK. DEMAND EXCEEDS SUPPLY', /,
     2         6X, '??????????????????????????????????????????', / )
  650 FORMAT ( 3I7, 3I10 )

C ----  Write the heading and read and write the title of grain.

      WRITE (U6,600)
      READ  (U1,500) TITL
      WRITE (U6,500) TITL

C ----  Read in the number of surplus, deficit, river, port and
C ----  foreign regions; the time period and the number of days
C ----  in each time period.
C ----  Read in the code names of all the surplus regions.
C ----  Check that there is a region before reading the code name.

      READ  (U1,510) NOSR, NODR, NORE, NOPE, NOFR
      READ  (U1,510) NOTP, ( NDAY(I), I = 1, NOTP )
      READ  (U1,500) ( SRGN(I), I = 1, NOSR )
      IF ( NODR .GT. 0 ) READ  (U1,500) ( DRGN(I), I = 1, NODR )
      IF ( NORE .GT. 0 ) READ  (U1,500) ( RIVR(I), I = 1, NORE )
      IF ( NOPE .GT. 0 ) READ  (U1,500) ( PORT(I), I = 1, NOPE )
      IF ( NOFR .GT. 0 ) READ  (U1,500) ( FRGN(I), I = 1, NOFR )

C ----  Calculate the number of shipments by time.
C ----  Find all the connecting nodes in this model.

      NOTF      = NOTP + 1
      SRND      = NOTF * NOSR
      DRND      = NOTF * NODR + SRND
      REND      = NOTF * NORE + DRND
      PEND      = NOTF * NOPE + REND
      FRND      = NOTF * NOFR + PEND
      DMND      = NODR + NOFR + FRND
      SINK      = DMND + 1
      SRCE      = SINK + 1
      NODE      = SRCE

C ----  Read in the loading(LO) and unloading(RI) factors.
C ----  T stands for Truck, R for Rail, S for Ship and B for Barge.

      READ  (U1,52D) TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      READ  (U1,520) TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP

C ----  Initialisation and invoke other subroutines.
C ----  IPRD is the amount of grain produced.
C ----  IEXP is the amount of grained demanded.
C ----  Find the cost of transportation (RIVER) by calling RELVTR.
C ----  Find the cost of transportation (PORT) by calling PELVTR.

      IPRD      = 0
      IEXP      = 0
      K         = 0
```

```
      CALL SURPLS ( IPRD )
      IF ( NORE .GT. 0 ) CALL RELVTR
      IF ( NOPE .GT. 0 ) CALL PELVTR
      CALL DEMAND ( IEXP )

C ----  Invoke WRITER to write data to an output file.

      K          = K + 1
      IARC(K)    = SINK
      JARC(K)    = SRCE
      LOWR(K)    = IEXP
      UPPR(K)    = IPRD
      WRITE (U6,610) IPRD, IEXP
      IF ( IEXP .GT. IPRD ) WRITE (U6,620)
      ARCS       = K
      WRITE (U6,650) SINK, SRCE, NODE, ARCS
      CALL WRITER ( ARCS, NODE, IARC, JARC, LOWR, UPPR, KOST )
      RETURN
      END


C-----------------------------------------------------------------
C                    ======   SURPLS    =======
C ----  SURPLS is used to find out all the details relating to
C ----  the surplus regions such as the costs of transportation, and
C ----  storage from each surplus region to all the river regions,
C ----  port regions, barge locations and barge (river) loading points.
C ----  Invoke subroutine, GENARC to generate the appropriate arcs
C ----  connecting the nodes.

      SUBROUTINE  SURPLS ( IPRD )
      COMMON /A1/ NOSR, NODR, NDRE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /A4/ SINK, SRCE
      COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20)
      COMMON /C1/ SDTR(65,65) /C2/ SDRL(65,65)
      COMMON /C3/ SRTR(65,45) /C4/ SRRL(65,45)
      COMMON /C5/ SPTR(65,20) /C6/ SPRL(65,20)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE
      COMMON /E1/ SPLY(65)
      COMMON /F1/ STOR(65) /F2/ SCST(4)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     SINK, SRCE
      INTEGER     SRGN, DRGN, RIVR, PORT, FRGN
      INTEGER     UPPR
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12
```

```
  500 FORMAT ( 20A4 )
  520 FORMAT ( 10F8.3 )
  610 FORMAT (6X 'SUPPLY', F15.0, )
  820 FORMAT (6X, 'UNIT = 1 (THOUSAND BUSHEL)', /)
  830 FORMAT (6X, 'UNIT = 2 (THOUSAND SHORT TON )', /)
  840 FORMAT (6X, 'UNIT = 3 (THOUSAND METRIC TON)', /)

C ----  Read in the amount of grain produced by and the storage
C ----  capacity of each surplus region.

      READ  (U8,520) ( SPLY(I), I = 1, NOSR )
      READ  (U1,520) ( STOR(I), I = 1, NOSR )

C ----  Read in the cost, rail, truck, barge and ship factors; and
C ----  the period when the lakes are blocked.

      READ  (U1,520) COST, RAIL, TRUCK, BARGE, SHIP, PERIOD

C ----  Read in the conversion factor and the choice of measurement
C ----  used. Display the appropriate message.

      READ  (U1,520) OPERATION, CHOICE
      IF (CHOICE .EQ. 1.000) WRITE(17,820)
      IF (CHOICE .EQ. 2.000) WRITE(17,830)
      IF (CHOICE .EQ. 3.000) WRITE(17,840)

C ---- Calculate the storage cost for each period, SCST(N).
C ---- There are NOTP periods and the cost is COST.

      IF ( NOTP .LE. 0 ) GO TO 1200
      DO 1100 N = 1, NOTP
      SCST(N)   = COST * FLOAT(NDAY(N)) * 1000.0 / 365.0
 1100 CONTINUE
 1200 CONTINUE

C ----  Read in the mileage (by truck, TR) from each surplus region
C ----  to all the deficit regions .

      DO 1300 I = 1, NOSR
      READ  (U2,520) ( SDTR(I,J), J = 1, NODR )
 1300 CONTINUE

C ----  Read in the rail costs (RL) from each surplus region to all
C ----  the deficit regions.

      DO 1400 I = 1, NOSR
      READ  (U3,520) ( SDRL(I,J), J = 1, NODR )
 1400 CONTINUE

C ----  Read in the mileage (by truck, TR) from each selected (river)
C ----  barge loading location linked with all the surplus regions.

      READ  (U2,520) ( SRTR(I,1), I = 1, NOSR )

C ----  Read in the rail costs(RL) from each surplus region to all
```

```
C ----  the river regions.

       DO 1600 I = 1, NOSR
       READ  (U3,520) ( SRRL(I,J), J = 1, NORE )
 1600 CONTINUE

C ----  Read in the mileage (by truck, TR) from each surplus region
C ----  to all the port regions.

       DO 1700 I = 1, NOSR
       READ  (U2,520) ( SPTR(I,J), J = 1, NOPE )
 1700 CONTINUE
```

```
C ----  Read in the rail costs(RL) from each surplus region to
C ----  all the port regions.

      DO 1800 I = 1, NOSR
      READ  (U3,520) ( SPRL(I,J), J = 1, NOPE )
 1800 CONTINUE

C ----  Read in the alfas and betas of the surplus regions.
C ----  Currently not used.

      READ  (U1,520) ( ALFA(I), BETA(I), I = 1, NOSR )

C ----  Read in the selected barge (river) points which linked with
C ----  the surplus regions.

      READ  (U1,500) ( IDEN(I), I = 1, NOSR )

C ----  Find the amount produced, and storage capacity of each of
C ----  the surplus region.

      DO 4000 I = 1, NOSR
      ALF       = ALFA(I)
      BET       = BETA(I)
      NF        = I
      IA        = NOTF * ( NF - 1 )
      K         = K + 1
      IARC(K)   = SRCE
      JARC(K)   = IA + 1
      UPPR(K)   = SPLY(NF) * 1000.0
      IPRD      = IPRD + UPPR(K)
      ISTR      = STOR(NF)
      IF ( NOTP .LE. 0 ) GO TO 2200

C ----  Find the storage cost(SCST) and amount produced
C ----  per quarter for each surplus region.

      DO 2100 N = 1, NOTP
      K         = K + 1
      IARC(K)   = JARC(K-1)
      JARC(K)   = IARC(K) + 1
      UPPR(K)   = ISTR
      KOST(K)   = SCST(N)
 2100 CONTINUE
 2200 CONTINUE

C ----  Find the mileage(by truck) and calculate the cost by truck.

      DO 2400 M = 1, NODR
      NT        = M
      JA        = NOTF * ( NT - 1 ) + SRND
```

```
C ----  Check that the TRuck mileage from the surplus region
C ----  to the deficit region is feasible.
C ----  If the mileage is greater than 9999 then it is ignored else
C ----  the total cost includes the loading and unloading cost
C ----  is calculated.

      COST       = SDTR(NF,NT)
      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST       = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2300
      ENDIF

      COST       = ( COST * TRUCK + TLOS + TRID ) * 1000.0

      CALL GENARC ( IA, JA, COST )
 2300 CONTINUE

C ----  Check that the RaiL cost from the surplus region to
C ----  the deficit region is feasible.
C ----  If this cost is greater than 999 then it is ignored else
C ----  the cost (including loading and unloading) is found.

      COST       = SDRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 2400
      COST       = ( COST * RAIL + RLOS + RRID ) * 1000.0
      CALL GENARC ( IA, JA, COST )
 2400 CONTINUE
 2500 CONTINUE

C ----  Invoke the subroutine SERIAL which
C ----  check that all inputted data are corrected.

      IDNT       = IDEN(I)
      CALL SERIAL ( IDNT, NORE, RIVR, NT )
      IF ( NT .EQ. D ) GO TO 2600
      JA         = NOTF * ( NT - 1 ) + DRND

C ----  Find the mileage (by truck) from surplus region to the river
C ----  regions. If it is greater than 9999 then it is ignored
C ----  else the total cost by truck is found.

      COST       = SRTR(NF,1)
      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST       = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2600
```

```
      ENDIF

      COST       = ( COST * TRUCK + TLOS + TRID ) * 1000.0

      CALL GENARC ( IA, JA, COST )
 2600 CONTINUE

C ----  Find the rail cost from the surplus region to
C ----  the river regions. If it is greater than 999 then
C ----  it is ignored else the cost by rail is found.

      DO 2800 M = 1, NORE
      NT         = M
      JA         = NOTF * ( NT - 1 ) + DRND
      COST       = SRRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 2700
      COST       = ( COST * RAIL + RLOS + RRIR ) * 1000.0
      CALL GENARC ( IA, JA, COST )
 2700 CONTINUE
 2800 CONTINUE

C ----  Find the mileage(by truck) from the surplus region to
C ----  the port regions. If it is greater than 9999 then
C ----  it is ignored else the total cost by truck is found.

      DO 3000 M = 1, NOPE
      NT         = M
      JA         = NOTF * ( NT - 1 ) + REND
      COST       = SPTR(NF,NT)
      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST       = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2900
      ENDIF

      COST       = ( COST * TRUCK + TLOS + TRID ) * 1000.0

      CALL GENARC ( IA, JA, COST )
 2900 CONTINUE

C ----  Find the rail cost from the surplus region to
C ----  the port regions. If it is greater than 999 then
C ----  it is ignored else the cost by rail is found.

      COST       = SPRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 3000
      COST       = ( COST * RAIL + RLOS + RRIP ) * 1000.0
      CALL GENARC ( IA, JA, COST )
 3000 CONTINUE
 3900 CONTINUE
 4000 CONTINUE
```

```
C ----  Display the total amount of grain produced into an intermediate
C ----  file (FILE06_2.DRM).

      WRITE(17,610) (IPRD/OPERATION)
      RETURN
      END


C-----------------------------------------------------------------------
C               =====  RELVTR    =====
C ----  RELVTR deals with all the river regions.
C ----  RELVTR is used to find the transportation costs from river
C ----  regions.
C ----  It calculates the truck & rail cost and choose the minimum
C ----  cost and invokes subroutine GENARC to generate an arc with
C ----  this minimum cost.


      SUBROUTINE  RELVTR
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20)
      COMMON /C1/ RDTR(65,65) /C2/ RDRL(65,65)
      COMMON /C3/ RRBG(65,45) /C5/ RPBG(65,20)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
      DIMENSION   IDN1(10), IDN2(10), IDN3(25)
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     DRGN, RIVR, PORT, UPPR
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12
      LOGICAL     ICED
  500 FORMAT ( 20A4 )
  510 FORMAT ( 20I4 )
  520 FORMAT ( 10F8.3 )

C ----  Read in the alfas and betas of the deficit regions.
C ----  Read in the selected shipping(port) points which linked with
C ----  deficit regions.

      READ  (U1,520) ( ALFA(I), BETA(I), I = 1, NODR )
      READ  (U1,500) ( IDEN(I), I = 1, NODR )

C ----  Read in the total number of selected barge (river) points
C ----  and barge (port) points, and their respective codes.

      READ  (U1,510) NRES, NPES
      READ  (U1,500) ( IDN1(I), I = 1, NRES ), ( IDN2(I), I = 1, NPES )
```

```
C ----  Read in the mileage (by TRuck) of selected barge unloading
C ----  locations linked with each of the deficit region.

      READ  (U2,520) ( RDTR(1,J), J = 1, NODR )

C ----  Read in the rail (RL) costs of each river region
C ----  linked with all the deficit regions.

      DO 1200 I = 1, NORE
      READ  (U3,520) ( RDRL(I,J), J = 1, NODR )
 1200 CONTINUE

C ----  Read in the barge (BG) cost (per bushel) from each river
C ----  region linked with all of the selected barge (river)
C ----  shipping points.

      DO 1300 I = 1, NORE
      READ  (U4,520) ( RRBG(I,J), J = 1, NRES )
 1300 CONTINUE

C ----  Read in the barge (BG) cost (per bushel) from each river
C ----  region linked with all the selected barge (port) shipping
C ----  points.

      DO 1400 I = 1, NORE
      READ  (U4,520) ( RPBG(I,J), J = 1, NPES )
 1400 CONTINUE

C ----  Read in the number of river locations above the L&D 26
C ----  and their codes.
C ----  NOTE: This data is not being used in this model.

      READ  (U4,510) LAKE
      READ  (U4,500) ( IDN3(I), I = 1, LAKE )

C ----  Set all the variables to the appropriate values
C ----  and call SERIAL to check that all the required
C ----  information is correctly inputted.

      DO 3000 I = 1, NORE
      NF        = I
      IDNT      = RIVR(I)
      ICED      = .FALSE.
      CALL SERIAL ( IDNT, LAKE, IDN3, NT )
      IF ( NT .NE. 0 ) ICED = .TRUE.
      IA        = NOTF * ( NF - 1 ) + DRND

C ----  Find the truck (TR) cost linking each river region with
C ----  all the deficit regions.

      DO 2200 M = 1, NODR
      IDNT      = IDEN(M)
      CALL SERIAL ( IDNT, NORE, RIVR, NT )
      IF ( NF .NE. NT ) GO TO 2200
```

```
      NT          = M
      ALF         = ALFA(NT)
      BET         = BETA(NT)

C ----  If the truck cost is greater than 9999 then this cost
C ----  is ignored else the total truck cost is calculated
C ----  by taking into account of the respective alfa and beta,
C ----  and the loading (TLOR) and unloading (TRID) costs.

      COST        = RDTR(1,NT)
      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST        = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST        = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2200
      ENDIF

      JA          = NOTF * ( NT - 1 ) + SRND
      COST        = ( COST * TRUCK + TLOS + TRID ) * 1000.0

      CALL GENARC ( IA, JA, COST )
 2200 CONTINUE

C ----  Find the rail cost from each river region linked
C ----  with all the deficit regions.

      DO 2300 M = 1, NODR
      NT          = M
      JA          = NOTF * ( NT - 1 ) + SRND

C ----  If the rail cost is more than 999 than the cost is ignored
C ----  else the cost by rail is calculated.
C ----  RAIL is the rail factor, RLOR is the loading cost and
C ----  RRID is the unloading cost.

      COST        = RDRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 2300
      COST        = ( COST * RAIL + RLOR + RRID ) * 1000.0
      CALL GENARC ( IA, JA, COST )
 2300 CONTINUE

C ----  The barge cost(per bushel) is calculated for each of the chosen
C ----  barge(river) points.

      DO 2500 M = 1, NRES
      IDNT        = IDN1(M)
      CALL SERIAL ( IDNT, NORE, RIVR, NT )
      IF ( NT .EQ. NF ) GO TO 2500
      IF ( NT .EQ. 0 ) GO TO 2500
      JA          = NOTF * ( NT - 1 ) + DRND
```

```
C ---- If the cost by barge is greater than 999 then it is ignored.
C ----  If the river is iced then UPPR is set to 0, i.e, no passage
C ----  is possible thus no grain is transported.
C ----  BARGE is a barge ratio which can be altered (see data file F1.DRM).

      COST       = RRBG(NF,M) * 100.0
      IF ( COST .GE. 999. ) GO TO 2500
      COST       = ( COST * BARGE + BLOR + BRIR ) * 1000.0
      CALL GENARC ( IA, JA, COST )
      IF ( ICED ) UPPR(K-NOTP+(PERIOD - 1)) = 0
 2500 CONTINUE

C ----  Find the barge cost for each selected barge(port) point
C ----  with all the river points.

      DO 2800 M = 1, NPES
      IDNT       = IDN2(M)
      CALL SERIAL ( IDNT, NOPE, PORT, NT )
      IF ( NT .EQ. 0 ) GO TO 2800
      JA         = NOTF * ( NT - 1 ) + REND

C ----  If the barge cost is larger than 999 then it is ignored
C ----  else the total cost is calculated which includes the
C ----  loading(BLOR) and unloading (BRIP) costs.

      COST       = RPBG(NF,M) * 100.0
      IF ( COST .GE. 999. ) GO TO 2800
      COST       = ( COST * BARGE + BLOR + BRIP ) * 1000.0
      CALL GENARC ( IA, JA, COST )
      IF ( ICED ) UPPR(K-NOTP+(PERIOD - 1)) = 0
 2800 CONTINUE
 2900 CONTINUE
 3000 CONTINUE
      RETURN
      END

C-----------------------------------------------------------------------
C                    =====   PELVTR    =====
C ----  PELVTR deals with all the data relating to port regions.
C ----  PELVTR is used to find transportation cost from the port
C ----  regions. The truck and rail costs are calculated.
C ----  If the cost is not feasible, then it is ignored.
C ----  The variable, ICED is a boolean variable. It is used to
C ----  denote whether the river is iced or not. If it is iced,
C ----  then ICED is set to true which means that the river is
C ----  blocked and no passage is possible.


      SUBROUTINE  PELVTR
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /B4/ PORT(20) /B5/ FRGN(25)
      COMMON /C3/ PFSP(65,45)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
```

```
      : COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
        COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
        COMMON /G0/ K, OPERATION
        COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
        COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
        DIMENSION   IDN3(16)
        INTEGER     SRND, REND, DRND, PEND, FRND, DMND
        INTEGER     PORT, FRGN, UPPR
        INTEGER U1,U2,U3,U4,U8,U9,U6,U12
        LOGICAL     ICED
    500 FORMAT ( 20A4 )
    510 FORMAT ( 20I4 )
    520 FORMAT ( 10F8.3 )

C ----  Read in the shipping costs of each port location linked
C ----  with all the foreign regions.

        DO 1100 I = 1, NOPE
        READ  (U4,520) ( PFSP(I,J), J = 1, NOFR )
   1100 CONTINUE

C ----  Read in the number of lakes used for export and their
C ----  code names.
C ----  NOTE: This data might not be used in this model.

        READ  (U4,510) LAKE
        READ  (U4,500) ( IDN3(I), I = 1, LAKE )

C ----  Set ICED to false which means that the port is passable.
C ----  Call SERIAL to check that all the necessary information
C ----  is included.

        DO 2000 I = 1, NOPE
        NF        = I
        ICED      = .FALSE.
        IDNT      = PORT(I)
        CALL SERIAL ( IDNT, LAKE, IDN3, NT )
        IF ( NT .NE. 0 ) ICED = .TRUE.
        IA        = NOTF * ( NF - 1 ) + REND

C ----  Find the port (ship rates) cost of all the foreign regions
C ----  and generate the corresponding arcs.
C ----  Reset the value of UPPR if the port is ICED.

        DO 1300 M = 1, NOFR
        NT        = M
        JA        = NOTF * ( NT - 1 ) + PEND

C ----  If the ship cost is greater than 999, it is ignored,
C ----  else the total cost (including shipping rates) is found.
C ----  The condition of the port is checked, if it is iced then
C ----  no grain is transported, UPPR is set to 0.
C ----  SHIP is the ship ratio which can be altered (refer to F1.DRM).

        COST      = PFSP(NF,NT) * 100.0
```

```
      IF ( COST .GE. 999. ) GO TO 1300
      COST       = ( COST * SHIP + SLOP ) * 1000.0
      CALL GENARC ( IA, JA, COST )
      IF ( ICED ) UPPR(K-NOTP+(PERIOD - 1)) = 0
 1300 CONTINUE
 2000 CONTINUE
      RETURN
      END

C -----------------------------------------------------------------
C                          ====   DEMAND    =====
C ----  DEMAND deals with all the amount of grain produced
C ----  from surplus regions and the amount of grain demanded by
C ----  deficit regions.
C ----  Read in the demand required by the deficit and foreign regions.
C ----  Calculate the demand (per quarter) for the deficit and
C ----  foreign regions.
C ----  Find the total amount of grain supplied (UPPR)
C ----  and demanded (LOWR).

      SUBROUTINE  DEMAND ( IEXP )
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /A4/ SINK, SRCE
      COMMON /E2/ DDND(65) /E3/ FDND(25,4)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     SINK, SRCE
      INTEGER     UPPR
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12
  520 FORMAT ( 10F8.3 )
  610 FORMAT (6X, 'DEMAND', F15.0, /)

      IF ( NODR .LE. 0 ) GO TO 1400

C ----  Read in the amount of grain demanded by the deficit regions.

      READ  (U8,520) ( DDND(I), I = 1, NODR )

C ----  Find the amount demanded (in each quarter) in each of the
C ----  deficit region.

      DO 1300 I = 1, NODR
      IA        = NOTF * ( I - 1 ) + SRND
      JA        = FRND
      DO 1200 N = 1, NOTF
      K         = K + 1
      IARC(K)   = IA + N
      JARC(K)   = JA + I
      LOWR(K)   = DDND(I) * 250.0
```

```
      IEXP       = IEXP + LOWR(K)
 1200 CONTINUE
      K          = K + 1
      IARC(K)    = JA + I
      JARC(K)    = SINK
 1300 CONTINUE

C ----  Read in the amount of grain demanded by the foreign regions.

 1400 IF ( NOFR .LE. 0 ) RETURN
      DO 1500 I = 1, NOFR
      READ  (U8,520) ( FDND(I,N), N = 1, NOTF )
 1500 CONTINUE

C ----  Find the demand (in each quarter) by each foreign region.

      DO 1700 I = 1, NOFR
      IA         = NOTF * ( I - 1 ) + PEND
      JA         = FRND + NODR
      DO 1600 N = 1, NOTF
      K          = K + 1
      IARC(K)    = IA + N
      JARC(K)    = JA + I
      LOWR(K)    = FDND(I,N) * 1000.0
      IEXP       = IEXP + LOWR(K)
 1600 CONTINUE

      K          = K + 1
      IARC(K)    = JA + I
      JARC(K)    = SINK
 1700 CONTINUE

C ----  Display the total amount of grain demanded into an intermediate
C ----  file (FILE06_2.DRM).

      WRITE(17,610) (IEXP/OPERATION)
      RETURN
      END

C ------------------------------------------------------------------------
C                          ====   SERIAL     ======
C ----  Check that all data are inputted correctly. An error message
C ----  will be outputted if there is insufficient data.

      SUBROUTINE  SERIAL ( IDNT, NDSR, SRGN, NF )
      DIMENSION   SRGN(NOSR)
      INTEGER     SRGN
  600 FORMAT ( 5X, '?????  ERROR IN DATA. ', A4, 'IS MISSING' )
      NF         = 0
      DO 1100 I = 1, NOSR
      IF ( IDNT .EQ. SRGN(I) ) GO TO 1200
 1100 CONTINUE
      RETURN
 1200 NF         = I
      RETURN
```

```
      END

C -----------------------------------------------------------------------
C                        =====  GENARC   ======
C ----  Generate the arcs for each node.
C ----  Arcs are generated for each quarter.
C ----  The total cost of transportation is stored in KOST.

      SUBROUTINE GENARC ( IA, JA, COST )
      COMMON /A2/ NOTP, NOTF, NOAY(4)
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G5/ KOST(27500)
      DO 1100 N = 1, NOTF
      K          = K + 1
      IARC(K)    = IA + N
      JARC(K)    = JA + N
      KOST(K)    = COST
 1100 CONTINUE
      RETURN
      END


C -----------------------------------------------------------------------
```

```
C ----  ***********************************************************************
C ----  *                          DURUM2.F                                   *
C ----  *          DOCUMENTED ON                  : SEPT 1990                *
C ----  *          RUN ON                         : A 386 machine            *
C ----  *          COMPILER USED                  : NDP Fortran              *
C ----  *          GRAIN USED                     : DURUM                    *
C ----  *          DATA FILES USED                : U12                      *
C ----  *          TRACE FILE                     : U6                       *
C ----  *          INPUT FILE FOR DURUM3          : UF1                      *
C ----  ***********************************************************************


C ----  Declaration of all variables used.


      COMMON /AA/ NR, NN, FSBL, NTIM, TOTL, MAXA
      COMMON /B1/ IWV(1500) /B2/ LABL(1500) /B3/ NODE(1500)
      COMMON /B4/ MIDL(1500) /B5/ NSAVE(1500)
      COMMON /C1/ IL0(34000) /C2/ ISAVE(34000) /C3/ JSAVE(34000)
      COMMON /C4/ JWV(34000)
      COMMON /D1/ KOS(68000) /D2/ MIR(68000) /D3/ NA(68000)
      COMMON /D4/ NC(68000) /D5/ NF(68000)
      LOGICAL     FSBL
      INTEGER U6,U12,UF1
  610 FORMAT ( ' ******* OPTIMAL SOLUTION ******* ' )
  620 FORMAT ( ' ??????? INFEASIBLE SOLUTION ??????? ' )
  630 FORMAT ( ' TOTAL COST ', F15.0 )


C ----  Unit number for each file.
C ----  U6 is the unit number used for the temporary file
C ----  which is used to trace/check that the output from
C ----  this program.
C ----  U12 is the unit number for the output file created by
C ----  the first(previous) program, DURUM1.
C ----  This data file contains the relevant information which
C ----  is necessary for DURUM2 to run.
C ----  UF1 is the unit number for the output file of this program
C ----  This output file will be used in the third program, DURUM3.


      U6 = 17
      U12 = 12
      UF1 = 18
```

```
C ----  To open files FILE06_2, FLOW1_2, FILE12_2
C ----  for reading and writing.

      OPEN (UNIT = UF1, FILE = 'FLOW1_2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U6, FILE = 'FILE06_2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U12, FILE = 'FILE12_2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')


      NTIM       = 0
      FSBL       = .TRUE.
      MAXA       = 34000

C ----  Read in the data from the output file(generated by) DURUM1.F.
C ----  The data read in will be used in this current program.

      READ  (U12,100) NR,NN,( NF(K), K = 1, NR ),( NA(K), K = 1, NR ),
     1           ( ILO(K), K = 1, NR ), ( JSAVE(K), K = 1, NR ),
     2           ( ISAVE(K), K = 1, NR )
100   FORMAT(2018)

      DO 1100 K = 1, NR
      NC(K)      = 0
      KOS(K)     = 0
 1100 CONTINUE
      CALL SUPERK
      IF ( .NOT. FSBL ) WRITE (U6,620)
      IF ( FSBL ) WRITE (U6,610)
      CSUM       = 0
      DO 1500 K = 1, NR
      COST       = FLOAT ( ISAVE(K) )
      GRAN       = FLOAT ( NC(K) )
      CSUM       = CSUM + COST * GRAN
 1500 CONTINUE
      CSUM       = CSUM / 10.0

C ----  Write the optimal solution showing the total cost.
C ----  This information is only for checking purpose and
C ----  will not be used any further.

      WRITE (U6,630) CSUM

C ----  Write the solution to an output file, FILE12_2_grn
C ----  and this is used by the third and last program PROG03.
C ----  Note: Format used might not be right. Changes will have
C ----  to be made.

      WRITE  (UF1,2000)  NR, ( NC(K), K = 1, NR ), ( KOS(K), K = 1, NR )
2000  FORMAT (2018)
      ENDFILE UF1
      STOP
      END
```

```
C-----------------------------------------------------------------------
C                          =====  SUPERK     =====


      SUBROUTINE SUPERK
      COMMON /AA/ NR, NN, FSBL, NTIM, TOTL, MAXA
      COMMON /B1/ IWV(1500) /B2/ LABL(1500) /B3/ NODE(1500)
      COMMON /B4/ MIDL(1500) /B5/ NSAVE(1500)
      COMMON /C1/ ILO(34000) /C2/ ISAVE(34000) /C3/ JSAVE(34DD0)
      COMMON /C4/ JWV(34000)
      COMMON /D1/ KOS(68000) /D2/ MIR(68000) /D3/ NA(68000)
      COMMON /D4/ NC(68000) /D5/ NF(68000)
      LOGICAL     FSBL
 1234 CONTINUE
      INFIN=100000000
      IFLOW=0
      KLAB=0
      KPOT=0
      KBRK=0
      IP=0
      NUMS=0
      NONS=0
      IPL=0
      NR2=NR*2
      NN1=NN+1
      IF ( NTIM .GE. 1 ) GO TO 12
      DO 5 I=1,NN1
      NODE(I)=0
 5    LABL(I)=0
      DO 10 M=1,NR
      I=NF(M)
      J=NA(M)
      IFLOW=NC(M)
      KOST=ISAVE(M)
      NODE(I)=NODE(I)+1
      NODE(J)=NODE(J)+1
      N=M+NR
      NF(N)=J
      NA(N)=I
      KOS(M)=KOST
      KOS(N)=-KOST
      NC(M)=JSAVE(M)-IFLOW
      NC(N)=IFLOW-ILO(M)
 10   CONTINUE
      DO 11 I=1,NN1
 11   NSAVE(I)=NODE(I)
      GO TO 1401
 12   DO 13 I=1,NN1
      NODE(I)=NSAVE(I)
 13   LABL(I)=0
```

```
      DO 14 M=1,NR
      N=M+NR
      I = NF(M)
      J = NA(M)
      NF(N) = J
      NA(N) = I
      IFLOW=NC(M)
      KOST=ISAVE(M)+KOS(M)
      KOS(M)=KOST
      KOS(N)=-KOST
      NC(M)=JSAVE(M)-IFLOW
      NC(N)=IFLOW-ILO(M)
 14   CONTINUE
 1401 CONTINUE
C     ************************************************************************
C
C     SETUP SECTION
C
C     ************************************************************************
      KL=1
      DO 15 K=1,NN1
      JK=NODE(K)
      NODE(K)=KL
      JWV(K)=KL
      KL=JK+KL
 15   MIDL(K)=KL-1
      DO 20 L=1,NR
      LL=L+NR
      J=NA(L)
      I=NA(LL)
      KOST=KOS(L)
      K=NC(L)
      LO=-NC(LL)
C     RIGHT=2   LEFT=1
      MAIN=2
      MIRROR=2
      IF(KOST) 29,29,30
 29   IF(K)32,32,31
 30   IF(LO)35,36,31
 31   MAIN=1
 32   IF(KOST) 33,34,34
 33   IF(K) 35,36,36
 34   IF(LO) 35,36,36
 35   MIRROR=1
 36   GO TO(43,44),MAIN
 43   II=JWV(I)
      MIR(II)=L
      JWV(I)=II+1
      GO TO 45
 44   II=MIDL(I)
      MIR(II)=L
      MIDL(I)=II-1
 45   GO TO(46,47),MIRROR
 46   II=JWV(J)
      MIR(II)=LL
```

```
      JWV(J)=II+1
      GO TO 20
 47   II=MIOL(J)
      MIR(II)=LL
      MIOL(J)=II-1
 20   CONTINUE
C     ***********************************************************************
C
C     GO - SUPERKILTER
C
      NO=INFIN
C
C     MAIN LOOP (100)
C
      NR2=NR*2
      DO 1000 MAIN=1,NR
      MAINM=MAIN+NR
      DO 1000 MODE=1,2
      GO TO(52,53),MODE
 52   II=MAIN
      JZ=MAINM
      GO TO 54
 53   II=MAINM
      JZ=MAIN
 54   IF(NC(II)) 65,55,56
 55   IF(NC(JZ)) 63,990,990
 56   IF(KOS(II)) 63,55,55
C          IS,IT = START,END NODE NOS, JS,JT = ARC,MIRROR ARC NOS
C               FOR ARC NEEDING FLOW INCREASE
C          WANT TO INCREASE FLOW, START LABLEING AT JJ
 63   IS=NA(JZ)
      JS=II
      IT=NA(II)
      JT=JZ
      GO TO 70
C          WANT TO DECREASE FLOW, START LABELING AT II
 65   IT=NA(JZ)
      IS=NA(II)
      JS=JZ
      JT=II
C
C     LABELING PROCEDURE
C
C     ***********************************************************************
 70   IPL=1
      IPLL=1
      IPS=0
      NUMS=0
      LABL(IT)=JS
      IWV(IPL)=IT
 84   KLAB=KLAB+1
      GO TO 86
 85   IF(IPS-IPL)86,200,86
 86   IPS=IPS+1
      IA=IWV(IPS)
```

```
      IB=NODE(IA)
      IE=MIDL(IA)
      IF(IB-IE) 87,87,85
 87   DO 90JJ=IB,IE
      J=MIR(JJ)
      NUNODE=NA(J)
      IF(LABL(NUNODE)) 90,88,90
 88   LABL(NUNODE)=J
      IPL=IPL+1
      IWV(IPL)=NUNODE
      IF(NUNODE-IS) 90,96,90
 90   CONTINUE
      GO TO 85
C
C     BREAKTHROUGH      BREAKTHROUGH      BREAKTHROUGH
C
 96   KBRK=KBRK+1
 97   IALPHA=INFIN
C
C     FIRST RETRACE
C
C          IJ = PREDECESSOR ARC INDEX
C          JI = MIRROR ARC INDEX
C          K = JWV POINTER
C          NEXT = PREDECESSOR NODE
C
      K=0
      NOW=IS
 100  IJ=LABL(NOW)
      JI=IJ-NR
      IF(JI) 101,101,102
 101  JI=JI+NR2
 102  NEXT=NA(JI)
      K=K+1
      IF(KOS(IJ)) 105,105,104
 104  NET=-NC(JI)
      JWV(K)=NET
      GO TO 110
 105  NET=NC(IJ)
      JWV(K)=NET
 110  IALPHA=MIN0(IALPHA,NET)
      IF(NEXT-IS) 111,120,111
 111  NOW=NEXT
      GO TO 100
C     SECOND RETRACE
C
 120  K=0
      NOW=IS
 125  IJ=LABL(NOW)
      JI=IJ-NR
      IF(JI) 126,126,127
 126  JI=JI+NR2
 127  NEXT=NA(JI)
      K=K+1
      NC(IJ)=NC(IJ)-IALPHA
```

```
      NET=NC(JI)
      NETNU=NET+IALPHA
      NC(JI)=NETNU
      IF(KOS(JI)) 128,1271,128
 1271 IF(NET) 1272,1272,128
 1272 IF(NETNU) 128,128,1273
 1273 CALL LEFT(NOW,JI)
 128  IF(JWV(K)-IALPHA) 129,1281,129
 1281 CALL RIGHT(NEXT,IJ)
 129  IF(NEXT-IS) 130,150,130
 130  NOW=NEXT
      GO TO 125
C
C     ERASE LABELS AND GO FOR O-K CHECK
C
 150  DO 155 I=1,IPL
      J=IWV(I)
 155  LABL(J)=0
      GO TO 54
C
C     POTENTIAL CHANGE
C
200   KPOT=KPOT+1
 201  KSET=NUMS
      NEWLAB=0
      NUMS=0
      IMTHRU=0
      MIN=INFIN
      NEW=NONS
      NONS=MAXA+1
      IF(KSET) 204,204,202
 202  IF(NEW-MAXA) 295,295,312
C       NON-S (L,L-) SET RECYCLING FILTER
 295  MAXNEW=MAXA+NEW
      DO 310 L=NEW,MAXA
      K=MAXNEW-L
      KK=JWV(K)
      KKK=NA(KK)
      IF(LABL(KKK)) 310,300,310
 300  NONS=NONS-1
      JWV(NONS)=KK
 310  CONTINUE
C       S-SET RECYCLING FILTER
 312  DO 203 K=1,KSET
      KK=JWV(K)
      KKK=NA(KK)
      IF(LABL(KKK)) 203,2021,203
 2021 IF(KOS(KK)) 2023,2023,2022
 2022 NUMS=NUMS+1
      JWV(NUMS)=KK
      MIN=MIN0(MIN,KOS(KK))
      GO TO 203
 2023 NONS=NONS-1
      JWV(NONS)=KK
 203  CONTINUE
```

```
 204  CONTINUE
      IF(IPLL-IPL) 2039,2039,2111
C       FIND MIN(C-BAR) OVER SET S
 2039 DO 211 LL=IPLL,IPL
      L=IWV(LL)
      JMID=MIDL(L)+1
      JRT=NDDE(L+1)-1
      IF(JMID-JRT) 2045,2045,211
 2045 DO 210KK=JMID,JRT
      K=MIR(KK)
      I=NA(K)
      IF(LABL(I)) 210,2040,210
 2040 IF(NC(K)) 206,2041,2041
 2041 IF(KOS(K)) 206,206,205
 205  NUMS=NUMS+1
      JWV(NUMS)=K
      MIN=MIN0(MIN,KOS(K) )
      GO TO 210
 206  NONS=NONS-1
      JWV(NONS)=K
 210  CONTINUE
 211  CONTINUE
 2111 IPLL=IPL+1
      IF(NUMS) 212,212,215
  212 FSBL       = .FALSE.
      CALL DUMPO(NR,II)
      PRINT 2125,IS,IT,II
      IF ( .NOT. FSBL ) RETURN
      PRINT 2121,(I,LABL(I),I=1,NN)
      PRINT 2122,(I,IWV(I),I=1,IPL)
      PRINT 2123,(JWV(I),I=NEW,MAXA)
 2121 FORMAT(' LABELS, BY NDDE'/(5(I9,'=',I10)) )
 2122 FORMAT(' LABELED NODES (IWV)'/       (10I10))
 2123 FORMAT(' THE SET (L,L-), NON-S'/(10I10))
 2125 FORMAT('0IS=',I5,'    IT=',I5,10X,'INFEASIBLE ARC =',I5)
      RETURN
C     UPDATE RELATIVE COSTS
C
C       UPDATE COST FOR SET S
 215  DO 230 I=1,NUMS
      IJ=JWV(I)
      JI=IJ-NR
      IF(JI) 216,216,217
 216  JI=IJ+NR
 217  KOST=KOS(IJ)-MIN
      KOS(IJ)=KOST
      KOS(JI)=-KOST
      IF(KOST) 230,218,230
 218  IF(NC(IJ)) 230,230,220
 220  NODEB=NA(IJ)
      CALL LEFT(NA(JI),IJ)
      IF(LABL(NODEB)) 230,223,230
 223  LABL(NODEB) =IJ
      IPL=IPL+1
      IWV(IPL)=NODEB
```

```
      IF(NODEB-IS) 230,225,230
 225  IMTHRU=1
 230  CONTINUE
C       UPDATE COST FOR NON-S
      IF(NONS-MAXA) 240,240,345
 240  DO 270 I=NONS,MAXA
      IJ=JWV(I)
      JI=IJ-NR
      IF(JI) 242,242,244
 242  JI=IJ+NR
 244  KOSTA=KOS(IJ)
      KOSTB=KOSTA-MIN
      KOS(IJ)=KOSTB
      KOS(JI)=-KOSTB
C       CHECK FOR MIRROR LEAVING MU STATE
C       CHECK LATER FOR COMBINING IF-CHECKS HERE
 260  IF(KOSTA) 270,262,262
 262  IF(KOSTB) 264,270,270
 264  IF(NC(IJ)) 270,269,269
 269  IF(NC(JI)) 270,270,2691
 2691 CALL RIGHT(NA(IJ),JI)
 270  CONTINUE
C       OUT-OF-KILTER CHECK
 345  IF(NC(II)) 360,350,351
 350  IF(NC(JZ)) 360,980,980
 351  IF(KOS(II)) 360,350,350
C       BREAKTHROUGH CHECK
 360  IF(IMTHRU) 361,361,96
 361  IF(IPS-IPL) 84,200,84
 980  DO 981 I=1,IPL
      J=IWV(I)
 981  LABL(J)=0
 990  CONTINUE
 1000 CONTINUE
      TOTL = 0.0
      DO 1010 I=1,NR
      KOS(I)=KOS(I)-ISAVE(I)
      NC(I)=JSAVE(I)-NC(I)
      TOTL = TOTL + NC(I) * ISAVE(I)
 1010 CONTINUE
      RETURN
      END


C ---------------------------------------------------------------------
C                          =====   RIGHT   =====

      SUBROUTINE RIGHT(I,INDEX)
      COMMON /AA/ NR, NN, FSBL, NTIM, TOTL, MAXA
      COMMON /B1/ IWV(1500) /B2/ LABL(1500) /B3/ NODE(1500)
      COMMON /B4/ MIDL(1500) /B5/ NSAVE(1500)
      COMMON /C1/ ILO(34000) /C2/ ISAVE(34000) /C3/ JSAVE(34000)
      COMMON /C4/ JWV(34000)
      COMMON /D1/ KOS(68000) /D2/ MIR(68000) /D3/ NA(68000)
      COMMON /D4/ NC(68000) /D5/ NF(68000)
```

```
      LOGICAL      FSBL
1234  CONTINUE
      MID=MIDL(I)
      IA=NODE(I)
      DO 1 II=IA,MID
      IF(MIR(II)-INDEX) 1,3,1
1     CONTINUE
      KWAY=1
    2 PRINT 900, I,INDEX,KWAY
      IFROM=NODE(I)
      ITO=NODE(I+1)-1
      PRINT 910,IFROM,MIDL(I),ITO,(K,MIR(K),K=IFROM,ITO)
910   FORMAT(3I6/(2016))
      RETURN
3     ITEMP=MIR(MID)
      MIR(MID)=INDEX
      MIR(II)=ITEMP
      MIDL(I)=MID-1
      RETURN
      ENTRY LEFT(I,INDEX)
      MID=MIDL(I)+1
      IB=NODE(I+1)-1
      DO 10 II=MID,IB
      IF(MIR(II)-INDEX) 10,12,10
10    CONTINUE
      KWAY=2
      GO TO 2
12    ITEMP=MIR(MID)
      MIR(MID)=INDEX
      MIR(II)=ITEMP
      MIDL(I)=MID
      RETURN
900   FORMAT(5H NODE,I5,5H  ARC, I5, 16H  LOST ON SHIFT  ,I4,4H LOC ,I4
     1  )
      ENTRY DUMPO
      NLINES = 1
      ID = INDEX
      PRINT 1120,ID
      DO 1070 M=1,NLINES
      N=M+NR
      I=NA(N)
      J=NA(M)
      L=ILO(M)
      K=JSAVE(M)
      KOST=ISAVE(M)
      KBAR=KOS(M)
      IFLOW=K-NC(M)
      IF(IFLOW.LT.L .OR. IFLOW.GT.K) PRINT 1121
      IF(KBAR) 1065,1070,1067
1065  IF(IFLOW.LT.K) PRINT 1122
      GO TO 1070
1067  IF(IFLOW.GT.L) PRINT 1122
1070  PRINT 1125,M,I,J,L,K,IFLOW,KOST,KBAR
1125  FORMAT(3I5,3I10,5X,2I10)
1120  FORMAT('1 ARC    I    J          L          K       IFLOW
```

```
     *OST       KBAR' ,I15        /)
 1121 FORMAT(' THE FOLLOWING ARC IS PRIMAL INFEASIBLE')
 1122 FORMAT(' THE FOLLOWING ARC IS DUAL INFEASIBLE')
      RETURN
      END
C-----------------------------------------------------------------------
```

```
C ----  ************************************************************
C ----  *                        DURUM3.F                          *
C ----  *        Documented on              SEPT  1990             *
C ----  *        Run on                     a 386 machine          *
C ----  *        Compiler used              NDP Fortran            *
C ----  *        Grain used                 DURUM                  *
C ----  *        Data files used            F1,F2,F3,F4,F8,F9      *
C ----  *        Intermediate input file    UF1                    *
C ----  *        Final output file          FD19                   *
C ----  *                                                          *
C ----  ************************************************************


C ----  Declaration of 2 common blocks.
C ----  FLOW is the array that stores the output from the previous
C ----  program DURUM2.F, it is used as an input for this program.
C ----  K is counter.
C ----  U1,U2,U3,U4,U8,U9,UF1,FD store unit numbers for the files.


      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

C ----  Unit number for each file.
C ----  U1 is the unit number for data file, F1.drm.
C ----  F1.drm contains information about the model.
C ----  U2 is the unit number for data file, F2.drm.
C ----  F2.drm contains information on TRUCK mileage.
C ----  U3 is the unit number for data file, F3.drm.
C ----  F3.drm contains information on RAIL costs.
C ----  U4 is the unit number for data file,F4.drm.
C ----  F4.drm contains information on BARGE costs.
C ----  U8 is the unit number for data file, F8.drm.
C ----  F8.drm contains information about SUPPLIES and DEMANDS.
C ----  U9 is for the unit number for data files, F9.drm.
C ----  F9.drm contains the NAMES of all the regions.
C ----  UF1 is the unit number for the input file produced by DURUM2.F.
C ----  It is the output file from DURUM2.F and
C ----  is used as an input file for this program.
C ----  FD is the unit number for the final (output) data file.
C ----  This is the final output file which will show all the solutions
C ----  for this model.


      U1 = 13
      U2 = 14
      U3 = 15
      U4 = 16
      U8 = 8
      U9 = 9
      UF1 = 18
      FD = 19
```

```
C ---- The diagram below shows the files and program involved.

C ----                    U1,U2,U3,U4,U8,U9,UF1 (input files)
C ----                                  |
C ----                                  |
C ----                                  V
C ----                                  |
C ----                          DURUM3.F (program)
C ----                                  |
C ----                                  |
C ----                                  V
C ----                                  |
C ----                       FD (output/final file)


C ----  To open files for reading and writing

      OPEN (UNIT = U1, FILE = 'F1.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U2, FILE = 'F2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U3, FILE = 'F3.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U4, FILE = 'F4.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U8, FILE = 'F8.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U9, FILE = 'F9.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = UF1, FILE = 'FLOW1_2.DRM', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')
      OPEN (UNIT = FD, FILE = 'final.out', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')

C ----  The output data (from 2nd program) is required to run this
C ----  3rd and final program.
C ----  Read in the processed data in FLOW1_2.DRM with unit number
C ----  UF1. The processed data is the output of DURUM2.F.
C ----  N is the total number of data processed in DURUM2.F.
C ----  FLOW is an array which contains relevant data required to get
C ----  the final output of this model.
C ----  Invoke subroutine DRIVER which is responsible for the overall
C ----  supervision of this program.


      READ  (UF1,100) N, ( FLOW(I), I = 1, N )
100   FORMAT(2018)
      K          = 0
      CALL DRIVER
      ENDFILE UF1
      STOP
      END
```

```
C ------------------------------------------------------------------
C                        ===== DRIVER  ======
C
C ----  DRIVER acts as the "driver" of this program i.e it calls
C ----  the other subroutines, SURPLS, RELVTR, PELVTR, and DEMAND
C ----  and read in all the data from the respective data files.
C ----  Calculate the total cost of storage, cost of transportation
C ----  by truck, rail, barge, and ship and total handling cost.
C ----  Write all the results to the output file, FD.

C ------------------------------------------------------------------
C                    ====    SUBROUTINE  DRIVER    ====

C ----  Declaration of all variables.

      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20) /B5/ FRGN(25)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      COMMON /GA/ SUMG, SUMT, SUMR, SUMB, SUMS, SUMH
      COMMON /GB/ TOPR, TODD, TOFD
      COMMON /GC/ TGSO(65), TGRD(65), TGRF(65)
      COMMON /GD/ TGRR(65,4), TGRP(65,4)
      COMMON /R1/ NAM1(65,3), NAM2(65,3), NAM3(45,3)
      COMMON /R2/ NAM4(20,3), NAM5(25,3)
      DIMENSION   TITL(16)
      INTEGER     TITL
      INTEGER     FLOW
      INTEGER     SRGN, DRGN, RIVR, PORT, FRGN
      INTEGER U1,U2,U3,U4,U8,U9,UF1,FD


C ----  Layout of the output file.
C ----  The format used to display the solution of this model.


  500 FORMAT ( 20A4 )
  510 FORMAT ( 20I4 )
  520 FORMAT ( 10F8.3 )
  530 FORMAT ( 3A4 )
  600 FORMAT ( 1H1, 5X, 'Network Generator', /,
     1          6X, 'For Grain Shipment Problem', // )
  610 FORMAT ( 6X, 'SUPPLY', I15, 6X, 'DEMAND', I15, / )
  620 FORMAT ( 6X, '??????????????????????????????????????????', /,
     1          6X, 'INFEASIBLE NETWORK. DEMAND EXCEEDS SUPPLY', /,
     2          6X, '??????????????????????????????????????????', / )
  650 FORMAT ( 3I7, 3I10 )


C ----  Read in and write the title(TITL) of this model onto the final
C ----  report.
```

```
C ----  Read in the number of surplus(NOSR), deficit(NODR), river(NORE),
C ----  port(NOPE) and foreign(NOFR) regions.
C ----  Read in the number of time period(NOTP) and the number of days
C ----  (NDAY(I)) in each period.
C ----  Read in all the code numbers(SRGN(I)) of the surplus regions.


      READ  (U1,500) TITL
      WRITE (19,600)
      WRITE (19,500) TITL
      READ  (U1,510) NOSR, NODR, NORE, NOPE, NOFR
      READ  (U1,510) NOTP, ( NDAY(I), I = 1, NOTP )
      READ  (U1,500) ( SRGN(I), I = 1, NOSR )


C ----  Check that the number of regions is greater than 0.
C ----  If it is then read in all the code numbers for the regions
C ----  (deficit(DRGN), river(RIVR), port(PORT) and foreign(FRGN)).
C ----  The number of time factor (NOTF) is found to be the
C ----  number of time period(NOTP) plus 1. (i.e 3+1)

      IF ( NODR .GT. 0 ) READ  (U1,500) ( DRGN(I), I = 1, NODR )
      IF ( NORE .GT. 0 ) READ  (U1,500) ( RIVR(I), I = 1, NORE )
      IF ( NOPE .GT. 0 ) READ  (U1,500) ( PORT(I), I = 1, NOPE )
      IF ( NOFR .GT. 0 ) READ  (U1,500) ( FRGN(I), I = 1, NOFR )
      NOTF       = NOTP + 1


C ----  Initialise the following arrays:
C ----  TGSO stores Total amount of Grain shipped to a Surplus region.
C ----  TGRD stores Total amount of Grain shipped to a Deficit region.
C ----  TGRF stores Total amount of Grain shipped to a Foreign region.


      DO 1100 I = 1, 53
      TGSO(I)    = 0.0
      TGRD(I)    = 0.0
      TGRF(I)    = 0.0


C ----  Initialaise the arrays, TGRR and TGRP for each time factor(NOTF).
C ----  TGRR stores Total amount of Grain shipped to a River region.
C ----  TGRP stores Total amount of Grain shipped to a Port region.

      DO 1100 N = 1, NOTF
      TGRR(I,N) = 0.0
      TGRP(I,N) = 0.0
 1100 CONTINUE


C ----  Initialisation of all variables used.
C ----  TOPR TOtal grain PRoduced from surplus regions.
C ----  TODD TOtal grain Demamded by Deficit regions.
C ----  TOFD TOtal grain Demanded by Foreign regions.
C ----  SUMG SUM of storaGe cost.
```

```
C ----  SUMT SUM of Truck cost.
C ----  SUMR SUM of Rail cost.
C ----  SUMB SUM of Barge cost.
C ----  SUMS SUM of Shipping cost.
C ----  SUMH SUM of Handling cost.


      TOPR       = 0.0
      TODD       = 0.0
      TOFD       = 0.0
      SUMG       = 0.0
      SUMT       = 0.0
      SUMR       = 0.0
      SUMB       = 0.0
      SUMS       = 0.0
      SUMH       = 0.0


C ----  Read in the loading and unloading costs.
C ----  TLOS Truck Loading (country elevator) cOSt.
C ----  RLOS Railcar Loading (country elevator) cOSt.
C ----  TLOR Truck LOading (River location) cost.
C ----  RLOR Railcar LOading (River location) cost.
C ----  BLOR Barge LOading (River location) cost.
C ----  SLOP Ship LOading (Port loading ocean vessel) cost.
C ----  TRID TRuck unloading (Deficit) cost.
C ----  RRID Rail unloading (Deficit) cost.
C ----  TRIR River location unloading TRuck cost.
C ----  RRIR River location unloading Rail cost.
C ----  BRIR River location unloading Barge cost.
C ----  TRIP Port unloading Truck cost.
C ----  RRIP Port unloading Rail cost.
C ----  BRIP Port unloading Barge cost.

      READ  (U1,520) TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      READ  (U1,520) TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
```

```
C ----  Read in the actual (real) names of all the regions involved.
C ----  NAM1 stores the names of the surplus regions.
C ----  NAM2 stores the names of the deficit regions.
C ----  NAM3 stores the names of the river regions.
C ----  NAM4 stores the names of the port regions.
C ----  NAM5 stores the names of the foreign regions.

      DO 2100 I = 1, NOSR
      READ  (U9,530) ( NAM1(I,J), J = 1, 3 )
 2100 CONTINUE
      DO 2200 I = 1, NODR
      READ  (U9,530) ( NAM2(I,J), J = 1, 3 )
 2200 CONTINUE
      DO 2300 I = 1, NORE
      READ  (U9,530) ( NAM3(I,J), J = 1, 3 )
 2300 CONTINUE
      DO 2400 I = 1, NOPE
      READ  (U9,530) ( NAM4(I,J), J = 1, 3 )
 2400 CONTINUE
      DO 2500 I = 1, NOFR
      READ  (U9,530) ( NAM5(I,J), J = 1, 3 )
 2500 CONTINUE

C ----  IPRD stores the amount of grain PRoDuced and is set to 0.
C ----  IEXP stores the amount of grain EXPended is set to 0.
C ----  K is a counter.
C ----  Invoke subroutine SURPLS to deal with all the information
C ----  relating to the surplus regions.
C ----  Call subroutine RELVTR and PELVTR if there is more than
C ----  one river region and one port region involved.
C ----  RELVTR and PELVTR deal with the River rEgions and Port rEgions
C ----  respectively and their Truck and Rail costs.
C ----  Subroutine DEMAND is called to calculate all demand required
C ----  by each region.

      IPRD       = 0
      IEXP       = 0
      K          = 0
      CALL SURPLS ( IPRD )
      IF ( NORE .GT. 0 ) CALL RELVTR
      IF ( NOPE .GT. 0 ) CALL PELVTR
      CALL DEMAND ( IEXP )

C ----  Write the total cost of Storage, Truck, Rail, Barge
C ----  Ship and Handling to the output file, FD.
C ----  FD is the final output data file.
C ----  The format used is 740.

      WRITE (FD,740) SUMG, SUMT, SUMR, SUMB, SUMS, SUMH

C ----  The format used in displaying the above information
C ----  in the output file, FD.

  740 FORMAT ( 1H1, //, 5X, 'STORAGE COST', F12.0, /,
```

```
     1  5X, 'TRUCK COST  ', F12.0, /, 5X, 'RAIL COST    ', F12.0, /,
     2  5X, 'BARGE COST  ', F12.0, /, 5X, 'SHIP COST    ', F12.0, /,
     3  5X, 'HANDLING CST', F12.0 )
      TOPR = (TOPR / OPERATION)
      TODD = (TODD / OPERATION)
      TOFD = (TOFD / OPERATION)
      WRITE (FD,750) TOPR, TODD, TOFD
  750 FORMAT ( ////, 5X, 'GRAIN SHIPPED FROM SURPLUS REGIONS', F15.0,
     1            //, 5X, 'GRAIN SHIPPED TO DEFICIT REGIONS  ', F15.0,
     2            //, 5X, 'GRAIN SHIPPED TO FOREIGN REGIONS  ', F15.0,
     3          //// )
      RETURN
      END

C -----------------------------------------------------------------
C                        =====  SURPLS  ======
C ----  SURPLS is used to display the heading of the output file.
C ----  It reads the data concerning the storage, the amount of
C ----  grain produced by, truck(mileage) and rail cost
C ----  of all the surplus regions.

      SUBROUTINE  SURPLS ( IPRD )
      DIMENSION   T(4), R(4), STCT(65), S(65,4)
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20)
      COMMON /C1/ SDTR(65,65) /C2/ SDRL(65,65)
      COMMON /C3/ SRTR(65,45) /C4/ SRRL(65,45)
      COMMON /C5/ SPTR(65,16) /C6/ SPRL(65,16)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, CHOICE
      COMMON /E1/ SPLY(65)
      COMMON /F1/ STOR(65) /F2/ SCST(4)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      COMMON /GA/ SUMG, SUMT, SUMR, SUMB, SUMS, SUMH
      COMMON /GB/ TOPR, TODD, TOFD
      COMMON /GC/ TGSO(65), TGRD(65), TGRF(65)
      COMMON /GD/ TGRR(65,4), TGRP(65,4)
      COMMON /R1/ NAM1(65,3), NAM2(65,3), NAM3(45,3)
      COMMON /R2/ NAM4(20,3), NAM5(25,3)
      COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
      INTEGER     FLOW
      INTEGER     SRGN, DRGN, RIVR, PORT, FRGN
      INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

C ----  The format used to display the results on the output file.

  500 FORMAT ( 20A4 )
  520 FORMAT ( 10F8.3 )
  700 FORMAT ( 1H1, //, 9X, 18HORIGIN/DESTN  MODE, 6X, 6HSUPPLY, 14X,
     1          16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
     2          7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4I10, 4X,
     3          8HSHIPMENT, 5X, 28HCOST    HAULING COST    COSTS,// )
```

```
  710 FORMAT ( /, 5X, 1HS, A4, 1X, 3A4, 4X, F12.0, / )
  720 FORMAT ( 5X, 1HO, A4, 1X, 3A4, 2X, 2HT , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  730 FORMAT ( 5X, 1HD, A4, 1X, 3A4, 2X, 2HR , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  740 FORMAT ( 5X, 1HR, A4, 1X, 3A4, 2X, 2HT , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  750 FORMAT ( 5X, 1HR, A4, 1X, 3A4, 2X, 2HR , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  760 FORMAT ( 5X, 1HP, A4, 1X, 3A4, 2X, 2HT , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  770 FORMAT ( 5X, 1HP, A4, 1X, 3A4, 2X, 2HR , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  800 FORMAT ( 1H1, //, 5X, 14HSURPLUS REGION, 10X, 6HSUPPLY,
     1          14X, 7HSTORAGE, 12X, 12HSTORAGE COST, / )
  810 FORMAT ( 3X, 1HS, A4, 1X, 3A4, 2X, F12.0, 3F10.0, F12.0 )

C ----  Read in the amount of grain produced (SPLY) and storage
C ----  available(STOR) in each surplus region; and
C ----  the cost(COST) and rail(RAIL) factors, the conversion
C ----  factor and the unit of measurement.

      READ  (U8,520) ( SPLY(I), I = 1, NOSR )
      READ  (U1,520) ( STOR(I), I = 1, NOSR )
      READ  (U1,520) COST, RAIL, TRUCK, BARGE, SHIP
      READ  (U1,520) OPERATION, CHOICE

C ----  Check that the number of periods(NOTP) is greater than 0.
C ----  If it is then find the storage cost for the leftover
C ----  (surplus) grain in storage for each time period.
C ----  SCST(N) is the cost of storage for each period.
C ----  NDAY(N) stores the number of days in each time period.
C ----  eg, NDAY(1) is 122 days.
C ----  eg, SCST(2) will be the cost of storing extra grain in
C ----  the second period.

      IF ( NOTP .LE. 0 ) GO TO 1200
      DO 1100 N = 1, NOTP
      SCST(N)    = COST * FLOAT(NDAY(N)) / 36.50
 1100 CONTINUE
 1200 CONTINUE

C ----  SURPLUS regions ----> by TRuck ----> DEFICIT regions.
C ----  Read in the Truck mileage from each surplus region
C ----  to all the deficit regions.

      DO 1300 I = 1, NOSR
      READ  (U2,520) ( SDTR(I,J), J = 1, NODR )
 1300 CONTINUE
```

```
C ----  SURPLUS regions ----> by RaiL ----> DEFICIT regions.
C ----  Read in the RaiL cost from each surplus region to all
C ----  the deficit regions.

      DO 1400 I = 1, NOSR
      READ  (U3,520) ( SDRL(I,J), J = 1, NODR )
 1400 CONTINUE

C ----  SURPLUS regions ----> by TRuck ----> Selected River points.
C ----  Read in the TRuck mileage from each surplus region to
C ----  the selected (river) points.

      READ  (U2,520) ( SRTR(I,1), I = 1, NOSR )

C ----  SURPLUS regions ----> by RaiL ----> RIVER regions.
C ----  Read in the RaiL cost from each surplus region to all
C ----  the river regions.

      DO 1600 I = 1, NOSR
      READ  (U3,520) ( SRRL(I,J), J = 1, NORE )
 1600 CONTINUE

C ----  SURPLUS regions ----> by TRuck ----> PORT regions.
C ----  Read in the TRuck mileage from each surplus region to
C ----  all the port regions.

      DO 1700 I = 1, NOSR
      READ  (U2,520) ( SPTR(I,J), J = 1, NOPE )
 1700 CONTINUE

C ----  SURPLUS regions ----> by RaiL ----> PORT regions.
C ----  Read in the RaiL cost from each surplus region to
C ----  all the port regions.

      DO 1800 I = 1, NOSR
      READ  (U3,520) ( SPRL(I,J), J = 1, NOPE )
 1800 CONTINUE

C ----  Read in alfas and betas of all the surplus regions.

      READ  (U1,520) (ALFA(I), BETA(I), I = 1, NOSR )

C ----  Read in selected (receiving) points (IDEN(I)) that are
C ----  linked with the surplus regions.

      READ  (U1,500) ( IDEN(I), I = 1, NOSR )

C ----  Print out the heading of the output file 1,e,
C ORIGIN/DESTN MODE SUPPLY SHIPMENT TOTAL   UNIT    TOTAL     HANDLING
C                                SHIPMENT COST HAULING COST  COSTS

      WRITE (FD,700) ( I, I = 1, 4 )
```

```
C ----  Find the storage cost of each of the surplus region.

      DO 4000 I = 1, NOSR
      ALF       = ALFA(I)
      BET       = BETA(I)
      NF        = I
      K         = K + 1
      ISTR      = STOR(NF)
      IF ( NOTP .LE. 0 ) GO TO 2200

C ----  For each surplus region, find
C ----  S(I,N) is the amount of grain in storage per period(FLOW),
C ----  STCT(I) is the total storage cost.
C ----  SCST(N) is the cost of storage for each period.
C ----  eg, SCST(2) will be the cost of storing extra grain in
C ----  the second period.

      STCT(I)   = 0.0
      DO 2100 N = 1, NOTP
      K         = K + 1
      S(I,N)    = FLOW(K)
      STCT(I)   = STCT(I) + FLOW(K) * SCST(N)
 2100 CONTINUE
 2200 CONTINUE

C ----  Calculate the actual supply(SPLY) and display the code
C ----  number (SRGN(I)) and the name (NAM1) of
C ----  and the supply (SPLY(I) from this surplus region.

      SPLY(I)   = SPLY(I) * 1000.0
      WRITE (FD,710) SRGN(I), ( NAM1(I,L), L = 1, 3 ),
     1               SPLY(I)/OPERATION

C ----  SURPLUS region ----> by TRUCK  ----> DEFICIT regions.
C ----  COST (=SDTR) is the truck mileage from each surplus to
C ----  all the deficit regions.
C ----  If the mileage is greater than 9999 then it is ignored
C ----  else the cost of handling and by truck is calculated.
C ----  Call the subroutine GENFLO to calculate the
C ----  handling cost (HCST) and truck cost(TCST) of the deficit
C ----  region.
C ----  Update the handling cost (SUMH) and the truck (hauling)
C ----  cost(SUMT).
C ----  TLOS and TRID is the loading and unloading costs.
C ----  T is the amount of grain per shipment (by time).
C ----  TT is the total shipment from this surplus region
C ----  to the selected deficit region.

      DO 2500 M = 1, NODR
      NT        = M
      COST      = SDTR(NF,NT)
      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST      = ( 23.67 + 0.0 * COST )
```

```
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2300
      ENDIF

      COST       = ( COST * TRUCK ) / 100.0

      CALL GENFLO ( COST, TLOS, TRID, T, TT, TCST, HCST )
      SUMH       = SUMH + HCST
      SUMT       = SUMT + TCST

C ----  If the total shipment (TT) to the selected deficit
C ----  region is less than 0 then it is ignored i.e no
C ----  grain is transported to this deficit region else
C ----  the name of the deficit region (DRGN), amount
C ----  of grain (T(N)) per shipment by time, total
C ----  shipment(TT), unit cost(COST), total (truck)
C ----  hauling cost (TCST) and total handling cost for
C ----  this deficit region are displayed on the output file, FD.


      IF ( TT  .LE. 0.0 ) GO TO 2300
      WRITE (FD,720) DRGN(M), ( NAM2(M,L), L = 1, 3 ),
     1             ( T(N)/OPERATION, N = 1, NDTF ),
     2             TT/OPERATION, COST*OPERATION, TCST, HCST

C ----  Update the following totals:
C ----  TODD is total amount grain demanded by the deficit regions.
C ----  TGSO is the total shipment by time for each surplus region.
C ----  TOPR is the amount of grain produced by the surplus region.
C ----  TGRD is the total shipment by time for each deficit region.


      TODD       = TODD + TT
      TGSO(I)    = TGSO(I) + TT
      TOPR       = TOPR + TT
      TGRD(M)    = TGRD(M)+ TT
 2300 CONTINUE

C ----  SURPLUS region ----->  by RAIL  -----> DEFICIT regions.
C ----  The RaiL cost (COST) from surplus region to this deficit
C ----  region is  tested. If it is greater than 999 then it is
C ----  ignored else the rail cost (COST) is calculated.
C ----  Subroutine GENFLO is called to find the total rail
C ----  (hauling) cost (RCST) and the total handling (HCSR) cost
C ----  by rail.
C ----  RLOS and RRID is the railcar loading and unloading costs.
C ----  R is the amount of grain per shipment (by time).
C ----  RT is the total amount shipment for this deficit region.
C ----  SUMR is the total rail cost.
C ----  SUMH is the total handling cost.

      COST       = SDRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 2400
```

```
      COST       = COST * RAIL / 100.0
      CALL GENFLO ( COST, RLOS, RRID, R, RT, RCST, HCSR )
      SUMR       = SUMR + RCST
      SUMH       = SUMH + HCSR

C ----  If the total shipment for this deficit region(RT) is
C ----  less than 0 i.e no grain is transported to this
C ----  deficit region then this region is ignored and no
C ----  output regarding this region is printed.

      IF ( RT .LE. 0.0 ) GO TO 2500

C ----  Print out the information about this deficit region
C ----  which receives grain.
C ----  DRGN is the name of the deficit region.
C ----  R(N) is the amount of shipment by time(per period).
C ----  RT is the total shipment received by this deficit
C ----  region.
C ----  COST is the unit cost.
C ----  RCST is the rail(hauling) cost.
C ----  HCSR is the handling cost.

      WRITE (FD,730) DRGN(M), ( NAM2(M,L), L = 1, 3 ),
     1            ( R(N)/OPERATION, N = 1, NOTF ),
     2            RT/OPERATION, COST*OPERATION, RCST, HCSR

C ----  Update the following totals:
C ----  TODD is the amount of grain received by the deficit regions.
C ----  TGSO is total shipment by time for each surplus region.
C ----  TOPR is the amount of grain produced by surplus regions.
C ----  TGRD is total shipment by time for each deficit region.

      TODD       = TODD + RT
      TGSO(I)    = TGSO(I) + RT
      TOPR       = TOPR + RT
      TGRD(M)    = TGRD(M) + RT
 2400 CONTINUE
 2500 CONTINUE

C ----  SURPLUS region ---->TRUCK-----> RIVER region.
C ----  Subroutine SERIAL is used to check that all input data
C ----  correctly entefed.
C ----  Check that the cost by truck (SRTR) is not greater than
C ----  999. If it is not, then the cost by truck from a surplus
C ----  region to this river region is found.
C ----  Subroutine GENFLO is called to calculate the cost of
C ----  hauling by truck (TCST) and the handling cost (HCST).
C ----  SUMH is the total handling cost.
C ----  SUMT is the total truck cost.

      IDNT       = IDEN(I)
      CALL SERIAL ( IDNT, NORE, RIVR, NT )
      IF ( NT .EQ. 0 ) GO TO 2600
      COST       = SRTR(NF,1)
```

```
      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST       = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2600
      ENDIF

      COST       = ( COST * TRUCK) / 100.0

      CALL GENFLO ( COST, TLOS, TRIR, T, TT, TCST, HCST )
      SUMH       = SUMH + HCST
      SUMT       = SUMT + TCST

C ----  If the amount of grain supplied (TT) is less than 0
C ----  no information on that region is outputted, else
C ----  the name (RIVR(NT)) of the river region, the
C ----  amount of grain on each shipment by time(T(N)),
C ----  unit cost(COST), total hauling (truck)cost and
C ----  handling cost (HCST) is printed on the output file, FD.

      IF ( TT .LE. 0.0 ) GO TO 2600
      WRITE (FD,740) RIVR(NT), ( NAM3(NT,L), L = 1, 3 ),
     1             ( T(N)/OPERATION, N = 1, NOTF ),
     2             TT/OPERATION, COST*OPERATION, TCST, HCST

C ----  Update TGSO, the total shipment by time
C ----  and TOPR, the amount of grain produced by the surplus
C ----  regions.

      TGSO(I)    = TGSO(I) + TT
      TOPR       = TOPR + TT

C ----  Updata TGRR, the total amount of grain shipped to a
C ----  river region.

      DO 2550 N = 1, NOTF
      TGRR(NT,N)= TGRR(NT,N) + T(N)
 2550 CONTINUE
 2600 CONTINUE

C ----  SURPLUS region -----> by RAIL ------> RIVER regions
C ----  Check that the cost by rail (COST) is less than 999.
C ----  If it is then find the cost by rail(COST) and invoke
C ----  subroutine GENFLO is calculate the hauling (rail)
C ----  cost(RCST) and the handling cost (HCSR).
C ----  R is the amount of grain in each shipment by time.
C ----  RT is the amount of grain in total shipment.

      DO 2800 M = 1, NORE
      NT         = M
      COST       = SRRL(NF,NT)
```

```
      IF ( COST .GE. 999. ) GO TO 2700
      COST       = COST * RAIL / 100.0
      CALL GENFLO ( COST, RLOS, RRIR, R, RT, RCST, HCSR )

C ----  If the amount of grain in total shipment in this river
C ----  region is less than 0 i.e no grain is received then
C ----  it is ignored, else update the total hauling(rail) cost
C ----  (SUMR) and the handling cost(SUMH) and output
C ----  the code number(RIVR(NT)) and the name (NAM3(NT,L) of
C ----  the river region, the amount of grain in each shipment
C ----  by time (R(N)), the total shipment(RT), unit cost(COST),
C ----  total hauling cost(RCST) and handling cost(HCSR).

      IF ( RT .LE. 0.0 ) GO TO 2700
      SUMH       = SUMH + HCSR
      SUMR       = SUMR + RCST
      WRITE (FD,750) RIVR(NT), ( NAM3(NT,L), L = 1, 3 ),
     1             ( R(N)/OPERATION, N = 1, NOTF ),
     2              RT/OPERATION, COST*OPERATION, RCST, HCSR

C ---- Update the following totals: TGSO(surplus),TOPR(port).

      TGSO(I)    = TGSO(I) + RT
      TOPR       = TOPR + RT

C ----  Update the total shipment (TGRR) to a river region.

      DO 2650 N = 1, NOTF
      TGRR(NT,N)= TGRR(NT,N) + R(N)
 2650 CONTINUE
 2700 CONTINUE
 2800 CONTINUE

C ----  SURPLUS REGION -----> by TRUCK -----> PORT regions
C ----  Check that the cost by truck (COST) is less than 9999.
C ----  If it is then find the cost by rail(COST) and invoke
C ----  subroutine GENFLO is calculate the hauling (truck)
C ----  cost(TCST) and the handling cost (HCST).
C ----  T is the amount of grain in each shipment by time.
C ----  TT is the amount of grain in total shipment.
C ----  Update the total hauling(truck) cost
C ----  (SUMT) and the handling cost(SUMH).

      DO 3000 M = 1, NOPE
      NT         = M
      COST       = SPTR(NF,NT)

      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST       = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2900
```

```
      ENDIF

      COST       = ( COST * TRUCK) / 100.0

      CALL GENFLO ( COST, TLOS, TRIP, T, TT, TCST, HCST )
      SUMT       = SUMT + TCST
      SUMH       = SUMH + HCST

C ----  If the amount of grain in total shipment in this port
C ----  region is less than 0 i.e no grain is received then
C ----  it is ignored, else
C ----  the name (PORT(M)) of the port region, the
C ----  amount of grain on each shipment by time(T(N)),
C ----  unit cost(COST), total hauling (truck)cost (TCST) and
C ----  handling cost (HCST) is printed.

      IF ( TT .LE. 0.0 ) GO TO 2900
      WRITE (FD,760) PORT(M), ( NAM4(M,L), L = 1, 3 ),
     1             ( T(N)/OPERATION, N = 1, NOTF ),
     2             TT/OPERATION, COST*OPERATION, TCST, HCST

C ----  Update the following totals: TGSO, and TOPR.

      TGSO(I)    = TGSO(I) + TT
      TOPR       = TOPR + TT

C ----  Update total shipment (TGRP) to a port region.

      DO 2850 N = 1, NOTF
      TGRP(M,N) = TGRP(M,N) + T(N)
 2850 CONTINUE
 2900 CONTINUE

C ----  SURPLUS region -----> by RAIL ------> PORT regions
C ----  SPRL stands for the Rail cost from a Surplus region
C ----  to a Port region.
C ----  Check that the cost by rail (COST) is less than 999.
C ----  If it is then find the cost by rail(COST) and invoke
C ----  subroutine GENFLO is calculate the hauling (rail)
C ----  cost(RCST) and the handling cost (HCSR).
C ----  R is the amount of grain in each shipment by time.
C ----  RT is the amount of grain in total shipment.
C ----  Update the total hauling(rail) cost
C ----  (SUMR) and the handling cost(SUMH)

      COST       = SPRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 3000
      COST       = COST * RAIL / 100.0
      CALL GENFLO ( COST, RLOS, RRIP, R, RT, RCST, HCSR )
      SUMR       = SUMR + RCST
      SUMH       = SUMH + HCSR

C ----  If the amount of grain in total shipment (RT)
C ----  in this port region is less than 0 i.e no grain is
C ----  received then it is ignored, else output
```

```
C ----  the code number (PORT(M)) and the name (NAM4(M,L) of
C ----  the port region, the amount of grain in each shipment
C ----  by time (R(N)), the total shipment(RT), unit cost(COST),
C ----  total hauling cost(RCST) and handling cost(HCSR).

       IF ( RT .LE. 0.0 ) GO TO 3000
       WRITE (FD,770) PORT(M), ( NAM4(M,L), L = 1, 3 ),
      1             ( R(N)/OPERATION, N = 1, NOTF ),
      2             RT/OPERATION, COST*OPERATION, RCST, HCSR

C ----  Update the following totals: TGSO(surplus), TOPR(port).

       TGSO(I)    = TGSO(I) + RT
       TOPR       = TOPR + RT

C ----  Update shipment by time to a port(TGRP).

       DO 2950 N = 1, NOTF
       TGRP(M,N) = TGRP(M,N) + R(N)
 2950 CONTINUE
 3000 CONTINUE
 3900 CONTINUE
 4000 CONTINUE

C ----  Write this heading:
C ----  SURPLUS REGION    SUPPLY  STORAGE    STORAGE COST
C ----  on the output file, FD.

       WRITE (FD,800)

C ----  Output the following data:
C ----  the code number (SRGN), name(NAM1(I,L),
C ----  the amount of grain produced by this surplus region (SPLY(I)),
C ----  the amount of grain in storage (S(I,N)) per
C ----  time period and storage cost (STCT) of each surplus region.
C ----  An example of the output:
C ----  SURPLUS REGION    SUPPLY       STORAGE          STORAGE COST
C ----  064 ABELINE, TX    5702     4424  2212    0         144642

C ----  Increment the total amount of storage.

       DO 5000 I = 1, NOSR
       WRITE (FD,810) SRGN(I), ( NAM1(I,L), L = 1, 3 ),
      1             SPLY(I)/OPERATION,
      2             ( S(I,N)/OPERATION, N = 1, NOTP ), STCT(I)
       SUMG       = SUMG + STCT(I)
 5000 CONTINUE
       RETURN
       END


C ---------------------------------------------------------------------
C                          =====  RELVTR  =====
C ----  Subroutine RELVTR deals with the river regions.
C ----  It finds the cost of transportation from the river
```

```
C ----  regions to selected barge (river) points and ports.
C ----  Truck, rail and barge costs are considered.

      SUBROUTINE  RELVTR
      DIMENSION   T(4), R(4)
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20)
      COMMON /C1/ RDTR(65,65) /C2/ RDRL(65,65)
      COMMON /C3/ RRBG(65,45) /C5/ RPBG(65,16)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      COMMON /GA/ SUMG, SUMT, SUMR, SUMB, SUMS, SUMH
      COMMON /GB/ TOPR, TODD, TOFD
      COMMON /GC/ TGSO(65), TGRD(65), TGRF(65)
      COMMON /GD/ TGRR(65,4), TGRP(65,4)
      COMMON /R1/ NAM1(65,3), NAM2(65,3), NAM3(45,3)
      COMMON /R2/ NAM4(20,3), NAM5(25,3)
      COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
      DIMENSION   IDN1(10), IDN2(10), IDN3(25)
      INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

C ----  The layout of the output.

  500 FORMAT ( 20A4 )
  510 FORMAT ( 20I4 )
  520 FORMAT ( 10F8.3 )
  700 FORMAT ( 1H1, //, 9X, 18HORIGIN/DESTN  MODE, 6X, 6HSUPPLY, 14X,
     1          16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
     2          7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4I10, 4X,
     3          8HSHIPMENT, 5X, 28HCOST    HAULING COST    COSTS,// )
  710 FORMAT ( /, 5X, 1HR, A4, 1X, 3A4, 16X, 4F10.0, / )
  720 FORMAT ( 5X, 1HD, A4, 1X, 3A4, 2X, 2HT , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  730 FORMAT ( 5X, 1HD, A4, 1X, 3A4, 2X, 2HR , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  740 FORMAT ( 5X, 1HR, A4, 1X, 3A4, 2X, 2HB , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )
  750 FORMAT ( 5X, 1HP; A4, 1X, 3A4, 2X, 2HB , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )

C ----  Read in the alfas and betas of the deficit regions.
C ----  Read in the selected (river) barge shipping
C ----  locations(IDEN(I)).

      READ  (U1,520) ( ALFA(I), BETA(I), I = 1, NODR )
      READ  (U1,500) ( IDEN(I), I = 1, NODR )

C ----  Read in the number of barge (river) points(NRES), and
C ----  barge (port) points(NPES); and their respective code
C ----  numbers.
C ----  IDN1(I) stores the code numbers of the barge (river)
```

```
C ----  points which are N,KV,Chat,Gun,Fl.
C ----  IDN2(I) stores the code numbers of the barge (port)
C ----  points which are NO,M,B,Ch,Port.

      READ  (U1,510) NRES, NPES
      READ  (U1,500) ( IDN1(I), I = 1, NRES ), ( IDN2(I), I = 1, NPES )

C ----  Read in the TRuck mileage from the selected barge
C ----  unloading (shipping)points to each of the deficit region.

      READ  (U2,520) ( RDTR(1,J), J = 1, NODR )

C ----  RIVER regions ----> by RAIL ----> DEFICIT regions
C ----  Read in the RaiL cost from each of the river
C ----  region to all the deficit regions.

      DO 1200 I = 1, NORE
      READ  (U3,520) ( RDRL(I,J), J = 1, NODR )
 1200 CONTINUE

C ----  RIVER regions ----> by BarGe ----> Barge (RIVER) points
C ----  Read in the barge cost(per bushel) from each of
C ----  the river region to all the barge(river) points.
C ----  NRES is the number of barge(river) points.
C ----  Barge (river) points are N,KV,Chat,Gun,Fl.

      DO 1300 I = 1, NORE
      READ  (U4,520) ( RRBG(I,J), J = 1, NRES )
 1300 CONTINUE

C ----  RIVER regions ----> by BarGe ----> Barge (PORT) points.
C ----  Read in the barge cost(per bushel) from each of
C ----  the river region to all the barge(port) points.
C ----  NPES is the number of barge(port) points which
C ---- are NO,M,B,Ch,Port.

      DO 1400 I = 1, NORE
      READ  (U4,520) ( RPBG(I,J), J = 1, NPES )
 1400 CONTINUE

C ----  Read in the number of ports (LAKE) above the L&D 26
C ----  their code numbers(IDN3(I)).
C ----  Note: might not be used in this model.

      READ  (U4,510) LAKE
      READ  (U4,500) ( IDN3(I), I = 1, LAKE )

C ----  Print out the heading of the output file i,e,
C ORIGIN/DESTN MODE SUPPLY SHIPMENT TOTAL   UNIT    TOTAL     HANDLING
C                                 SHIPMENT COST HAULING COST  COSTS

      WRITE (FD,700) ( I, I = 1, NOTF )

C ----  Write out the code number(RIVR(I)), name (NAM3(I,L)) of
C ----  the river region, and the amount of grain (TGRR) shipped by
```

```
C ----  time (per period) to each river region.
C ----  NOTF is the number of time factor.
C ----  FD is the output file, and format 700 is used.
C ----  eg  R 601 ST.PAUL,MN  0  0  0  0

      DO 3000 I = 1, NORE
      WRITE (FD,710) RIVR(I), ( NAM3(I,L), L = 1, 3 ),
     1             ( TGRR(I,N)/OPERATION, N = 1, NOTF )
      NF          = I

C ----  Selected BARGE (river)points-->by TRucK-->DEFICIT regions.
C ----  Read in the TRuck mileage from each of the barge
C ----  unloading point to all the deficit regions.
C ----  If the cost(COST) is greater than 9999 then it is
C ----  ignored else the cost by truck(COST) is calculated.
C ----  The COST includes BET and ALF.
C ----  Subroutine GENFLO is used to calculate the hauling
C ----  truck cost(TCST) and the handling cost (HCST).
C ----  T is the amount of grain in each shipment by time.
C ----  TT is the amount of grain in total shipment.

      DO 2200 M = 1, NODR
      IDNT        = IDEN(M)
      CALL SERIAL ( IDNT, NORE, RIVR, NT )
      IF ( NF .NE. NT ) GO TO 2200
      NT          = M
      COST        = RDTR(1,NT)
      IF ( COST-.GE. 9999. ) GO TO 2200
      ALF         = ALFA(NT)
      BET         = BETA(NT)

      IF ( COST .LE. 225 ) THEN
      COST = ( 0.066374 + 0.104892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST        = ( 23.67 + 0.0 * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST        = ( 0.68037 + 0.093976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 2200
      ENDIF

      COST        = ( COST * TRUCK ) / 100.0

      COST        = ( COST * BET + ALF ) / 100.0
      CALL GENFLO ( COST, TLOR, TRID, T, TT, TCST, HCST )

C ----  Check that the amount of grain shipped is more than 0.
C ----  If it is, update the total hauling (truck) cost
C ----  (SUMT) and the handling cost(SUMH).


      IF ( TT .LE. 0.0 ) GO TO 2200
      SUMT        = SUMT + TCST
      SUMH        = SUMH + HCST
```

```
C ----  Write all the information about the deficit region
C ----  which receives grain by truck.
C ----  DRGN(M) is the code number of the deficit region.
C ----  NAM2(M,L) is the name of the deficit region.
C ----  T(N) is the amount of grain received per shipment by
C ----  time. TT is the total shipment for this deficit region.
C ----  COST is the unit cost.
C ----  TCST is the total hauling(TRuck) cost.
C ----  HCST is the handling cost.

      WRITE (FD,720) DRGN(M), ( NAM2(M,L), L = 1, 3 ),
     1             ( T(N)/OPERATION, N = 1, NOTF ),
     2             TT/OPERATION, COST*OPERATION, TCST, HCST

C ----  Update the total amount of grain demanded(TODD) by
C ----  the deficit regions.
C ----  Update the total amount of grain shipped (TGRD(M))
C ----  to this deficit region.

      TODD       = TODD + TT
      TGRD(M)    = TGRD(M) + TT
 2200 CONTINUE

C ----Selected BARGE (river) points-->by RaiL-->DEFICIT regions.
C ----  Read in the RaiL cost from each of the barge
C ----  unloading point to all the deficit regions.
C ----  If the cost(COST) is greater than 999 then it is
C ----  ignored else the cost by truck(COST) is calculated.
C ----  The COST includes BET and ALF.
C ----  Subroutine GENFLO is used to calculate the hauling
C ----  rail cost(RCST) and the handling cost (HCSR).
C ----  R is the amount of grain in each shipment by time.
C ----  RT is the amount of grain in total shipment.

      DO 2300 M = 1, NODR
      NT         = M
      COST       = RDRL(NF,NT)
      IF ( COST .GE. 999. ) GO TO 2300
      COST       = COST * RAIL / 100.0
      CALL GENFLO ( COST, RLOR, RRID, R, RT, RCST, HCSR )

C ----  Check that the amount of grain shipped is more than 0.
C ----  If it is, update the total hauling (rail) cost
C ----  (SUMR) and the handling cost(SUMH).

      IF ( RT .LE. 0.0 ) GO TO 2300
      SUMR       = SUMR + RCST
      SUMH       = SUMH + HCSR

C ----  Write all the information about the deficit region
C ----  which received grain by rail.
C ----  DRGN(M) is the code number of the deficit region.
C ----  NAM2(M,L) is the name of the deficit region.
C ----  R(N) is the amount of grain received per shipment by
C ----  time. RT is the total shipment for this deficit region.
```

```
C ----  COST is the unit cost.
C ----  RCST is the total hauling(TRuck) cost.
C ----  HCSR is the handling cost.

      WRITE (FD,730) DRGN(M), ( NAM2(M,L), L = 1, 3 ),
     1             ( R(N)/OPERATION, N = 1, NOTF ),
     2             RT/OPERATION, COST*OPERATION, RCST, HCSR

C ----  Update the total amount of grain demanded(TODD) by
C ----  the deficit regions.
C ----  Update the total amount of grain shipped (TGRD(M))
C ----  to this deficit region.

      TODD       = TODD + RT
      TGRD(M)    = TGRD(M) + RT
 2300 CONTINUE

C ----  Selected BARGE (river)points-->by BarGE-->RIVER regions.
C ----  Read in the BarGe cost from each of the barge
C ----  unloading point to all the river regions.
C ----  If the cost(COST) is greater than 999 then it is
C ----  ignored else the cost by barge(COST) is calculated.
C ----  The COST includes BET and ALF.
C ----  Subroutine GENFLO is used to calculate the hauling
C ----  barge cost(BCST) and the handling cost (HCSB).
C ----  T is the amount of grain in each shipment by time.
C ----  TT is the amount of grain in total shipment.

      DO 2500 M = 1, NRES
      IDNT       = IDN1(M)
      CALL SERIAL ( IDNT, NORE, RIVR, NT )
      IF ( NT .EQ. NF ) GO TO 2500
      IF ( NT .EQ. 0 ) GO TO 2500
      COST       = RRBG(NF,M)
      IF ( COST .GE. 999. ) GO TO 2500
      COST       = COST * BARGE
      CALL GENFLO ( COST, BLOR, BRIR, T, TT, BCST, HCSB )

C ----  Check that the amount of grain shipped is more than 0.
C ----  If it is, update the total hauling (barge) cost
C ----  (SUMB) and the handling cost(SUMH).

      IF ( TT .LE. D.0 ) GO TO 2500
      SUMB       = SUMB + BCST
      SUMH       = SUMH + HCSB

C ----  Write all the information about the river region
C ----  which received grain by rail.
C ----  RIVR(NT) is the code number of the deficit region.
C ----  NAM3(NT,L) is the name of the deficit region.
C ----  T(N) is the amount of grain received per shipment by
C ----  time.
C ----  TT is the total shipment for this river region.
C ----  COST is the unit cost.
C ----  BCST is the total hauling(TRuck) cost.
```

```
C ----  HCSB is the handling cost.

       WRITE (FD,740) RIVR(NT), ( NAM3(NT,L), L = 1, 3 ),
      1             ( T(N)/OPERATION, N = 1, NOTF ),
      2             TT/OPERATION, COST*OPERATION, BCST, HCSB

C ----  Update the total amount of grain shipped(TGRR(NT,N)) to
C ----  a river region.

       DO 2450 N = 1, NOTF
       TGRR(NT,N)= TGRR(NT,N) + T(N)
 2450 CONTINUE
 2500 CONTINUE

C ----  Selected BARGE (port)points-->by BarGE-->RIVER regions.
C ----  Read in the BarGe cost from each of the barge
C ----  unloading point to all the river regions.
C ----  If the cost(COST) is greater than 999 then it is
C ----  ignored else the cost by barge(COST) is calculated.
C ----  Subroutine GENFLO is used to calculate the hauling
C ----  barge cost(BCST) and the handling cost (HCSB).
C ----  R is the amount of grain in each shipment by time.
C ----  RT is the amount of grain in total shipment.

       DO 2800 M = 1, NPES
       IDNT      = IDN2(M)
       CALL SERIAL ( IDNT, NOPE, PORT, NT )
       IF ( NT .EQ. 0 ) GO TO 2800
       COST      = RPBG(NF,M)
       IF ( COST .GE. 999. ) GO TO 2800
       COST      = COST * BARGE
       CALL GENFLO ( COST, BLOR, BRIP, R, RT, BCST, HCSB )

C ----  Check that the amount of grain shipped(RT) is more than 0.
C ----  If it is, update the total hauling (barge) cost
C ----  (SUMB) and the handling cost(SUMH).

       IF ( RT. LE. 0.0 ) GO TO 2800
       SUMB      = SUMB + BCST
       SUMH      = SUMH + HCSB

C ----  Write all the information about the port region
C ----  which received grain by barge.
C ----  PORT(NT) is the code number of the port region.
C ----  NAM4(NT,L) is the name of the port region.
C ----  R(N) is the amount of grain received per shipment by
C ----  time.
C ----  RT is the total shipment for this port region.
C ----  COST is the unit cost.
C ----  BCST is the total hauling(BARGE) cost.
C ----  HCSB is the handling cost.

       WRITE (FD,750) PORT(NT), ( NAM4(NT,L), L = 1, 3 ),
      1             ( R(N)/OPERATION, N = 1, NOTF ),
      2             RT/OPERATION, COST*OPERATION, BCST, HCSB
```

```
C ----  Update the total amount of grain shipped(TGRP(NT,N)) to
C ----  a port region.

      DO 2750 N = 1, NOTF
      TGRP(NT,N)= TGRP(NT,N) + R(N)
 2750 CONTINUE

 2800 CONTINUE
 2900 CONTINUE
 3000 CONTINUE
      RETURN
      END

C ---------------------------------------------------------------------
C                          =====  PELVTR  =====
C ----  Subroutine PELVTR deals with the port regions.
C ----  It finds the cost of transportation from the port
C ----  regions to selected barge (port) points and river.
C ----  Truck, rail and barge costs are considered.


      SUBROUTINE  PELVTR
      DIMENSION   T(4)
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /B4/ PORT(20) /B5/ FRGN(25)
      COMMON /C3/ PFSP(65,45)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      COMMON /GA/ SUMG, SUMT, SUMR, SUMB, SUMS, SUMH
      COMMON /GB/ TOPR, TODD, TOFD
      COMMON /GC/ TGSO(65), TGRD(65), TGRF(65)
      COMMON /GD/ TGRR(65,4), TGRP(65,4)
      COMMON /R1/ NAM1(65,3), NAM2(65,3), NAM3(45,3)
      COMMON /R2/ NAM4(20,3), NAM5(25,3)
      DIMENSION   IDN3(16)
      INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

C ---- The layout of the output file, FD.

  500 FORMAT ( 20A4 )
  510 FORMAT ( 20I4 )
  520 FORMAT ( 10F8.3 )
  700 FORMAT ( 1H1, //, 9X, 18HORIGIN/DESTN  MODE, 6X, 6HSUPPLY, 14X,
     1          16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
     2          7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4I10, 4X,
     3          8HSHIPMENT, 5X, 28HCOST    HAULING COST     COSTS,// )
  710 FORMAT ( /, 5X, 1HP, A4, 1X, 3A4, 16X, 4F10.0, / )
  720 FORMAT ( 5X, 1HF, A4, 1X, 3A4, 2X, 2HS , 12X, 4F10.0,
     1          F12.0, F10.5, 2F12.0 )

C ----  Read in the alfas and betas of the deficit regions.
```

```
C ----  Read in the selected (port) barge shipping
C ----  locations(IDEN(I)).

      READ  (U1,520) ( ALFA(I), BETA(I), I = 1, NODR )
      READ  (U1,500) ( IDEN(I), I = 1, NODR )

C ----  PORT regions ----> by ShiP ----> FOREIGN regions
C ----  Read in the ship rate (PFSP(I,J)) from each port to
C ----  all the foreign regions.

      DO 1100 I = 1, NOPE
      READ  (U4,520) ( PFSP(I,J), J = 1, NOFR )
 1100 CONTINUE

C ----  Read in the number of ports (LAKE) in the Great Lakes
C ----  for export and their code numbers(IDN3(I)).

      READ  (U4,510) LAKE
      READ  (U4,500) ( IDN3(I), I = 1, LAKE )

C ----  Print out the heading of the output file i,e,
C ORIGIN/DESTN MODE SUPPLY SHIPMENT TOTAL   UNIT    TOTAL     HANDLING
C                                  SHIPMENT COST HAULING COST  COSTS

      WRITE (FD,700) ( I, I = 1, NOTF )

C ----  Print out the code number (PORT(I) and the name (NAM4(I,L))
C ----  of the port together with the total grain shipment by time
C ----  from each port region.

      DO 2000 I = 1, NOPE
      NF        = I
      WRITE (FD,710) PORT(I), ( NAM4(I,L), L = 1, 3 ),
     1            ( TGRP(I,N)/OPERATION, N = 1, NOTF )

C ----  PORT regions ----> by ShiP ----> FOREIGN regions.
C ----  Read in the ship rate from each of the port region
C ----  to all the foreign regions.
C ----  If the cost(COST) is greater than 999 then it is
C ----  ignored else the cost by truck(COST) is calculated.
C ----  Subroutine GENFLO is used to calculate the hauling
C ----  shipping cost(SCST) and the handling cost (HCSS).
C ----  T is the amount of grain in each shipment by time.
C ----  TT is the amount of grain in total shipment.
C ----  SLOP is the ship loading cost.
C ----  SHIP is the factor for ship rate and can be altered.

      DO 1300 M = 1, NOFR
      NT        = M
      COST      = PFSP(NF,NT)
      IF ( COST .GE. 999. ) GO TO 1300
      COST      = COST * SHIP
      CALL GENFLO ( COST, SLOP, 0.0, T, TT, SCST, HCSS )

C ----  Check that the amount of grain shipped(TT) is more than 0.
```

```
C ---- If it is, update the total hauling (ship) cost
C ----  (SUMB) and the handling cost(SUMH).

      IF ( TT .LE. 0.0 ) GO TO 1300
      SUMS       = SUMS + SCST
      SUMH       = SUMH + HCSS

C ----  Write all the information about the foreign region
C ----  which received grain by ship.
C ----  FRGN(NT)is the code number of the foreign region.
C ----  NAM5(NT,L) is the name of the foreign region.
C ----  T(N) is the amount of grain received per shipment by
C ----  time.
C ----  TT is the total shipment for this river region.
C ----  COST is the unit cost.
C ----  SCST is the total hauling (SHIP) cost.
C ----  HCSS is the handling cost.

      WRITE (FD,720) FRGN(NT), ( NAM5(NT,L), L = 1, 3 ),
     1             ( T(N)/OPERATION, N = 1, NOTF ),
     2             TT/OPERATION, COST*OPERATION, SCST, HCSS

C ----  Update the total amount of grain shipped (TOFD) to all
C ----  foreign regions and the amount shipped (TGRF(NT,N)) to
C ----  this particular foreign region.

      TOFD       = TOFD + TT
      TGRF(NT)   = TGRF(NT) + TT

 1300 CONTINUE
 2000 CONTINUE
      RETURN
      END

C ---------------------------------------------------------------------
C                          =====  DEMAND  =====
C

      SUBROUTINE  DEMAND ( IEXP )
      DIMENSION   A(5)
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20) /B5/ FRGN(25)
      COMMON /E2/ DDND(65) /E3/ FDND(25,4)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      COMMON /GA/ SUMG, SUMT, SUMR, SUMB, SUMS, SUMH
      COMMON /GB/ TOPR, TODD, TOFD
      COMMON /GC/ TGSO(65), TGRD(65), TGRF(65)
      COMMON /GD/ TGRR(65,4), TGRP(65,4)
      COMMON /R1/ NAM1(65,3), NAM2(65,3), NAM3(45,3)
      COMMON /R2/ NAM4(20,3), NAM5(25,3)
      INTEGER     FLOW
```

```
      INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

C ---- The layout of the output.

  520 FORMAT ( 10F8.3 )
  700 FORMAT ( 1H1, //, 52X, 14HDEMAND BY TIME, 17X, 5HTOTAL,
     1          /, 39X, 4I10, 2X, 8HSHIPMENT, / )
  720 FORMAT ( 5X, 1HD, A4, 1X, 3A4, 16X, 4F10.0, F12.0 )
  730 FORMAT ( 5X, 1HF, A4, 1X, 3A4, 16X, 4F10.0, F12.0 )

C ----  Check that the number of deficit regions are greater than 0.
C ----  If it is then read in the demand of each of the deficit
C ----  region (DDND(I)).

      IF ( NODR .LE. 0 ) GO TO 1400
      READ  (U8,520) ( DDND(I), I = 1, NODR )

C ----  Print out the heading : DEMAND BY TIME     TOTAL SHIPMENT

      WRITE (FD,700) ( I, I = 1, NOTF )

C ----  Find the demand by time of each deficit region A(N).

      DO 1300 I = 1, NODR
      DO 1200 N = 1, NOTF
      K         = K + 1
      A(N)      = FLOW(K)
 1200 CONTINUE

C ----  Write out the code number (DRGN(I)) and name (NAM2(I,L))
C ----  of each deficit region and their demand by time (A(N)).

      K         = K + 1
      A(5)      = FLOW(K)
      WRITE (FD,720) DRGN(I), ( NAM2(I,L), L = 1, 3 ),
     1                (A(N)/OPERATION, N = 1, 5 )
 1300 CONTINUE

C ----  Check that the number of foreign regions is greater than 0.
C ----  If it is then read in the demand by time of each foreign
C ----  region FDND(I,N)).


 1400 IF ( NOFR .LE. 0 ) RETURN
      DO 1500 I = 1, NOFR
      READ  (U8,520) ( FDND(I,N), N = 1, NOTF )
 1500 CONTINUE

C ----  Find the demand by time of each foreign region A(N).

      DO 1700 I = 1, NOFR
      DO 1600 N = 1, NOTF
      K         = K + 1
      A(N)      = FLOW(K)
 1600 CONTINUE
```

```
C ----  Write out the code number (FRGN(I)) and name (NAM5(I,L))
C ----  of each foreign region and their demand by time (A(N)).

      K          = K + 1
      A(5)       = FLOW(K)
      WRITE (FD,730) FRGN(I), ( NAM5(I,L), L = 1, 3 ),
     1               (A(N)/OPERATION, N = 1, 5 )
 1700 CONTINUE
      RETURN
      END

C ---------------------------------------------------------------------
C                        =====  SERIAL   =====
C ----  This subroutine is used to check that the data inputted
C ----  is correct.

      SUBROUTINE  SERIAL ( IDNT, NOSR, SRGN, NF )
      DIMENSION   SRGN(NOSR)
      INTEGER     SRGN
      DATA        NINE/' 999'/
  600 FORMAT ( 5X, '?????  ERROR IN DATA. ', A4, 'IS MISSING' )
      NF         = 0
      DO 1100 I = 1, NOSR
      IF ( IDNT .EQ. SRGN(I) ) GO TO 1200
 1100 CONTINUE
      IF ( IDNT .NE. NINE ) WRITE (FD,600) IDNT
      RETURN
 1200 NF         = I
      RETURN
      END

C---------------------------------------------------------------------
C                        =====  GENFLO  =====
C ----  Calculate the amount of grain flow per time period,
C ----  total hauling cost(TCST) and handling cost (HCST).

      SUBROUTINE GENFLO ( COST, COUT, CRIN, A, AT, TCST, HCST )
      DIMENSION   A(4)
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      INTEGER     FLOW
      AT         = 0.0
      TCST       = 0.0
      HCST       = 0.0
      DO 1100 N = 1, NOTF
      K          = K + 1
      A(N)       = FLOW(K)
      AT         = AT + A(N)
 1100 CONTINUE

C ----  To find the total hauling cost(TCST) e.g, for Abilene,
C ----  TCST =   5246*0.20052*1000

      TCST       = AT * COST * 1000.0
```

```
C ----  COUT and CRIN are the loading and unloading costs.
C ----  The handling cost, for Abilene,
C ----  HCST  = 5246 * (6.47 + 6.59) * 10

       HCST       = AT * ( COUT + CRIN ) * 10.00
       RETURN
       END
```

```
C                                F1.DRM

C       Note :
C       (1) Do not use this file to run the program.
C           This is a documented version of the actual file used.
C       (2) The letter C at the beginning of a sentence implies that it is a
C           comment or an explanation. Lines beginning with a C are not present
C           in the actual file used.
C       (3) Explanation appears before the actual code.
C       (4) Any new changes made must be in the same position as the old entry.
C       (5) If the entries on a line does not exceed the 80th column,
C           insert a digit zero, 0, on the 80th column.
C       (6) The format used for codes in the model is :
C           4 fields i.e. *ccc where c is a digit or a letter,
C           e.g. *601 where * is a space.

C-------------------------------------------------------------------------------

C       Explanation for f1.drm.
C       The explanation for each line of f1.drm is given below.

C-------------------------------------------------------------------------------

C       This line show the title heading used in the output file, file06_2.drm.
C       file06_2.drm is used to display the intermediate results.
C       If a new title is to be needed, change this sentence.

  DURUM WHEAT SHIPMENT, FULLER, GRANT, TEH, FELLIN                          0

C-------------------------------------------------------------------------------

C       This line shows the number of regions involved in this model.
C       If there were any changes in the number of regions, the corresponding
C       number must be changed.
C       e.g. if the number of surplus regions were increased to 39 then change 21
C       to 39.

C       There are limitations on the number of regions that are allowed in
C       this program :

C               Maximum number of surplus regions allowed = 70
C               Maximum number of deficit regions allowed = 70
C               Maximum number of river   regions allowed = 45
C               Maximum number of port    regions allowed = 20
C               Maximum number of foreign regions allowed = 25

C       In this model, the
C               Number of surplus  regions = 21
C               Number of deficit  regions = 09
C               Number of river    regions = 43
C               Number of port     regions = 20
C               Number of foreign  regions = 25
```

```
C       The format used is :
C       4 fields i.e. **dd where d is a digit.
C       e.g. **21 where * is a space.

  21    9  43  20  25                                                    0

C-------------------------------------------------------------------------

C       This line shows the number of periods and days per period in this
C       model.
C       There are 3 periods per year.
C       The number of days in each period :
C               number of days in the 1st period = 121
C               number of days in the 2nd period = 121
C               number of days in the 3rd period = 123
C       If the number of periods were different, alter 3 to the new number and
C       add the number of days in the new period.

C       The format used for the number of days is :
C       4 fields i.e. *ddd where d is a digit.
C       e.g. *121 where * is a space.

   3 121 121 123                                                          0

C-------------------------------------------------------------------------

C       The 4 lines show the codes for the 21 surplus regions in this model.

C       For example, 022 is the code number for Aberdeen SD, and
C                    017 is the code number for Ashley ND.

C       If there were any changes in the number of surplus regions, the code of
C       the new surplus region would be added to the end of the list.
C       If there were a deletion, then the code involved would be deleted from
C       this list.

 022 017 071 015 024 013 014 362 323 322 321 324 715 012 021 072 023 361 401 016
 011                                                                    0

C-------------------------------------------------------------------------

C       This line shows the codes for the 9 deficit regions in this model.
C       For example, 035 is the code number for Lincoln NE,
C                    123 is the code number for Madison, WI.

C       If there were any changes in the number of deficit regions, the code of
C       the new deficit region would be added to the end of the list.
C       If there were a deletion, then the code involved would be deleted from
C       this list.

 035 123 331 113 096 381 073 291 151                                    0

C-------------------------------------------------------------------------

C       These 3 lines show the codes for the 43 river barge ports in this
```

```
C       model.
C       For example, 601 is the code number for St Paul MN, and
C                    602 is the code number for Duluth, RV.

C       If there were any changes in the number of river barge ports, the code
C       of the new river port would be added to the end of the list.
C       If there were a deletion, then the code involved would be deleted from
C       this list.

 601 644 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620
 621 622 623 624 645 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640
 641 642 643                                                                   0


C-------------------------------------------------------------------------------

C       This line shows the codes for the 20 ports in this model.
C       For example, 701 is the code number for Mobile AL, and
C                    702 is the code number for New Orleans LA.

C       If there were a change in the number of port regions, the code of
C       the new port would be added to the end of the list.
C       If there were a deletion, then the code involved would be deleted from
C       this list.

 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720

C-------------------------------------------------------------------------------

C       These 2 lines codes for the 25 foreign regions in this model.
C       For example, 801 is the code number for Scandinavia and
C                    802 is the code number for NC Europe.

C       If there were any changes in the number of foreign regions, the code of
C       the new foreign region would be added to the end of the list. Make sure
C       that the number of regions does not exceed the maximum number allowed
C       in this region. The maximum number of foreign regions is 25.
C       If there were a deletion, then the code involved would be deleted from
C       this list.

 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820
 821 822 823 824 825                                                           0

C-------------------------------------------------------------------------------

C       This line shows the loading costs by truck, rail and
C       barge (cents/bushel) in this model.
C       If there were any changes in the loading costs, then alter the
C       corresponding loading cost.

C               Truck Load Country Elevation = 10.422
C               Rail Load Country Elevation  = 10.368
C               Truck Load River Location    =  8.940
C               Rail Load River Location     =  9.330
C               Barge Load River Location    =  7.758
```

```
C                Ship Load Port Location      =  5.660

C        The format used is :
C        8 fields i.e. dddd.ddd where d is a digit.
C        If there is no digit in front of the number, then add spaces.
C        e.g. 10.422 becomes **10.422 where * is a space.

  10.422  10.368   8.940   9.330   7.758   5.660                                  0

C-------------------------------------------------------------------------------

C        This line shows the unloading costs by truck, rail and barge
C        (cents/bushel) in this model.
C        If there were any changes in the unloading cost, then change the
C        affected cost.

C                River Unload Truck Surplus   =  8.520
C                River Unload Rail Surplus    =  6.774
C                River load Truck Deficit     =  6.450
C                River load Rail Deficit      =  6.318
C                River Unload Barge           = 11.076
C                Port Unload Truck            =  6.763
C                Port Unload Rail             =  4.082
C                Port Unload Barge            =  8.097

C        The format used is :
C        8 fields i.e. dddd.ddd where d is a digit.
C        If there is no digit in front of the number, then add spaces.
C        e.g. 8.520 becomes ***8.520 where * is a space.

   8.520   6.774   6.450   6.318  11.076   6.763   4.082   8.097                  0

C-------------------------------------------------------------------------------

C        The quantity of storage available in each surplus region.
C        If a new surplus region were added, then add one value to the end of
C        the list.
C        If there were deletion, then erase one from this list.

C        The format used is :
C        8 fields i.e. dddddd.d where d is a digit.
C        e.g. 999999.9

999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9
999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9
999999.9                                                                       0

C-------------------------------------------------------------------------------

C        The grain storage cost, rail, truck, barge, and ship adjustment
C        factors and the period chosen to be blocked are as follows:

C                storage cost factor (cents/bushel/year) = 11.566
C                rail  cost multiplier                   =  1.000
C                truck mileage multiplier                =  1.000
```

```
C              barge cost multiplier                    =  1.000
C              ship rate multiplier                     =  1.000
C              period blocked                           =  1st period.

C       If there is a need to test the effect of an increase or a decrease in
C       cost of a type of transport, then alter the value of the multiplier.

C       The format used is :
C       8 fields i.e. dddd.ddd where d is a digit.
C       If there is no digit in front of the number, then add spaces.
C       e.g. 1.000 becomes ***1.000 where * is a space.

  11.566   1.000   1.000   1.000   1.000   1.000                                  0

C-------------------------------------------------------------------------------

C     Conversion factor and choice of measurement
C       The conversion factor                    = 1.000
C       Choice of measurement                    = 1.000
C       Conversion factor for short ton is 2000/60 = 33.333
C       and for metric ton is 2204/60 = 36.733

C       Choice of measurement can be one of these:
C              1.000 is for bushel
C              2.000 is for short ton
C              3.000 is for metric ton

  1.000   1.000                                                                  0

C-------------------------------------------------------------------------------

C       Alphas and betas associated with each surplus region and represent
C       truck cost intercept and per unit costs. (currently not being used).

   5.654   0.191   5.654   0.191   6.486   0.208   6.339   0.205   5.654   0.191
   6.339   0.205   5.654   0.191   6.460   0.207   7.246   0.233   7.246   0.233
   6.339   0.205   6.508   0.208   7.055   0.224   6.508   0.208   6.339   0.205
   5.654   0.191   6.434   0.207   7.246   0.233   5.654   0.191   6.508   0.208
   6.508   0.208                                                              0

C-------------------------------------------------------------------------------

C       These lines show the codes for barge shipping ports, with each
C       code number (barge loading port) associated with a surplus region from
C       it receives grain.

C       If there were any changes in the number of surplus regions, then there
C       must be an associated addition or deletion of the code representing the
C       barge loading point for that surplus region.

C       Adding a surplus region means that the code of the barge loading port
C       linked to the surplus region must be included at the end of the list
```

```
C       A deletion means that the code of the barge loading port linked to the
C       surplus region must be removed.

C       If the surplus region is not linked with a barge loading point then its
C       code number is 999. 999 means that the route from the surplus region to
C       the barge loading port is not feasible. Thus there is no link between
C       the 2 regions.

C       e.g, the 1st entry is the surplus region, Aberdeen SD.
C            Aberdeen is linked to the barge loading port, Sioux City IA (609).
C       e.g, Ashley ND (the 2nd code number) is linked to barge loading region,
C            St. Paul MN (601).

 609 601 644 601 609 644 601 999 638 638 638 638 999 644 609 601 609 999 631 601
 644                                                                          0

C-------------------------------------------------------------------------------

C       Alphas and betas associated with each deficit region and truck cost
C       intercept and per unit cost. (currently not used).

   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202   6.508   0.208
   7.246   0.233   5.654   0.191   6.200   0.202   6.459   0.207               0

C-------------------------------------------------------------------------------

C       The line below shows the codes for barge shipping ports with each code
C       number (barge unloading port) associated with a deficit region to which
C       it sends grain.

C       For example, the 1st deficit region, Lincoln NE is linked to
C                    Nebraska City NE (611) and
C                    the 2nd deficit region, Madison WI is linked to
C                    Dubuque IA (604).

C       If there were any changes in the number of deficit regions, then there
C       must be associated addition or deletion of the code representing the
C       barge-unloading point for that deficit deficit region.
C       Adding a deficit region means that the code of the barge-unloading port
C       linked to the deficit region must be included at the end of the list.

C       A deletion means that the code of the barge-unloading port linked to
C       the deficit region must be removed.

 611 604 639 642 608 999 601 645 618                                          0

C-------------------------------------------------------------------------------

C       The number of shipping points in this model.
C               Number of barge receiving points = 5
C               Number of ports which receive from barge = 5

   5   5                                                                      0

C-------------------------------------------------------------------------------
```

```
C       The codes for the above shipping points in this model.

C       The 5 river shipping locations are :
C                Buffalo,   NY  (645)
C                Knoxville, TN  (626)
C                Chatanoo,  TN  (627)
C                Guntersv,  AL  (628)
C                Florence,  AL  (629)

C       The 5 river port shipping points are:
C                New Orle, LA (702)
C                Mobile,   AL (701)
C                Galvestn, TX (703)
C                Chicago,  IL (710)
C                Portland, OR (713)

 645 626 627 628 629 702 701 703 710 713                                      0
C-------------------------------------------------------------------------------
```

```
C       Stores all the truck mileages in this model.
C       F2.DRM is the 2nd file in this model.

C       In this model, four sets of truck mileages must be estimated.
C       This includes
C       (1) miles from each surplus region to each deficit region
C       (2) miles from each surplus region to a barge-loading river location
C       (3) miles from each surplus region to each port location
C       (4) miles from each deficit region to a barge-unloading location.

C       Diagrammatically,
C                          (3)
C       ----------------------------------------->port
C       .
C       |
C       |                  (2)
C       surplus regions ------> river barge
C       |                       loading regions
C       |                       _________________
C       |                       port barge            (4)
C       |                       unloading regions <------------ deficit regions
C       |
C       |
C       V                       (1)
C       ------------------------------------------------------> deficit regions

C       A zero '0' must be inserted on the 80th column on each line.
C       The letter 'C' at the start of a statement is an explanation or
C       a comment. It is not in the actual file used in the programs.

C       The format used is :
C       8 fields i.e. dddd.ddd where d is a digit.
C       If there is no digit in front of the number, then add spaces.
C       e.g, 100.443 becomes *100.443 where * is a space.

C-------------------------------------------------------------------------------

C       First part
C       (1) Each surplus region is linked to each of the deficit regions.
C       In this model there are 21 surplus regions and  9 deficit regions.

C       surplus region 1    ----------> deficit region 1
C                           ----------> deficit region 2
C                                       .
C                                       .
C                           ----------> deficit region 9


C       surplus region 2    ----------> deficit region 1
C                           ----------> deficit region 2
C                                       .
C                                       .
C                           ----------> deficit region 9
```

```
C       .
C       .
C       .

C       surplus region 21  ----------> deficit region 1
C                          ----------> deficit region 2
C                                      .
C                                      .
C                          ----------> deficit region 9

C-------------------------------------------------------------------------

C       Second part

C       (2) Each surplus region is linked to a corresponding river barge-loading
C           location.

C       surplus region 1   ----------> corresponding river barge loading region
C       surplus region 2   ----------> corresponding river barge loading region
C                                      .
C                                      .
C       surplus region 21  ----------> corresponding river barge loading region

C-------------------------------------------------------------------------

C       Third part

C        (3) Each surplus region is linked to all the port locations for export.
C       There are 20 ports in this model.

C       surplus region 1   ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                      .
C                                      .
C                          ----------> port region 20 for export

C       surplus region 2   ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                      .
C                                      .
C                          ----------> port region 20 for export

C       .
C       .
C       .

C       surplus region 21  ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                      .
C                                      .
C                          ----------> port region 20 for export

C-------------------------------------------------------------------------
```

```
C       Fourth part

C       (4) Each barge-unloading river location is linked to a
C           corresponding deficit region.

C       barge-unloading location 1 ----------> corresponding deficit region

C       barge-unloading location 2 ----------> corresponding deficit region
C       .
C       .
C       .
C       barge-unloading location 21 ----------> corresponding deficit region



C-------------------------------------------------------------------------
C       Explanation for 1st part

C       This includes the truck mileage from each of the 21 surplus regions
C       to each of the 9 deficit regions. This resulted in a 21 by 9 matrix.

C       The 1st surplus region is Aberdeen SD(022).
C       The distances from Aberdeen SD to all deficit regions are given on the
C       first line:
C               Aberdeen SD (064) to Lincoln NE (035) = 406 miles
C               Aberdeen SD (064) to Madison WI (123) = 549 miles

C       The next matrix is for the next surplus region and its mileages to all
C       the deficit regions.

C       If a new surplus region were added to the end of the surplus region list,
C       then its truck mileages to all the deficit regions would be added to the
C       end of this 1st part.

C       If a new deficit region were added to position 10, then its truck mileage
C       to all the deficit regions would be positioned of the 10th location in
C       the matrix.

C       If a surplus region were deleted, then its truck mileages to all the
C       deficit regions would be removed from this 1st part.
C       e.g, if Aberdeen SD were not a surplus region, then the first line would
C       be removed. The file would start with 485.

 406.000 549.0001212.0001324.000 788.000 948.000 283.0001350.000 951.000   0.000
 485.000 613.0001195.0001403.000 867.000 938.000 351.0001411.0001012.000   0.000
 532.000 485.0001404.0001383.000 785.0001251.000 226.0001283.000 884.000   0.000
 592.000 698.0001079.0001514.000 983.000 916.000 460.0001496.0001097.000   0.000
 263.000 439.0001358.0001175.000 639.0001006.000 151.0001237.000 838.000   0.000
 606.0001069.0001200.0001524.000 956.0001096.000 428.0001464.0001065.000   0.000
 675.000 794.000 980.0001597.0001072.000 828.000 556.0001592.0001193.000   0.000
1473.0001933.0001150.0001638.0001609.000 742.0001773.0002562.0002144.000   0.000
 917.0001044.000 800.0001839.0001314.000 775.000 806.0001842.0001443.000   0.000
1078.0001205.000 639.0002000.0001475.000 679.000 967.0002003.0001604.000   0.000
 952.0001139.000 634.0001860.0001356.000 571.000 888.0001937.0001538.000   0.000
 942.0001202.000 586.0001844.0001346.000 434.000 926.0001993.0001594.000   0 000
```

```
1544.0002012.0001006.0001805.0001744.000 715.0001852.0002697.0002279.000   0.000
 677.0001171.0001078.0001595.0001058.000 994.000 530.0001566.0001167.000   0.000
 490.0001055.0001112.0001412.000 881.000 851.000 379.0001450.0001051.000   0.000
 339.000 846.0001321.0001248.000 698.0001053.000 181.0001241.000 842.000   0.000
 510.0001160.0001018.0001429.000 914.000 662.000 504.0001561.0001162.000   0.000
1580.0002292.000 737.0002117.0001977.000 699.0001861.0002744.0002338.000   0.000
1208.0001712.0001283.0001334.0001330.000 764.0001521.0002292.0001874.000   0.000
 500.000 966.0001210.0001412.000 853.0001051.000 325.0001361.000 962.000   0.000
 806.0001300.000 949.0001724.0001187.000 875.000 659.0001695.0001296.000   0.000


C-------------------------------------------------------------------------

C       Explanation for the Second part

C       This links each surplus region (21) to its corresponding river
C       barge-loading locations. Conceptually, a (1 X21) matrix.

C       Diagrammatically,

C       surplus region 1   --------> corresponding river barge-loading location
C       surplus region 2   --------> corresponding river barge-loading location
C                                          .
C                                          .
C       surplus region 21  --------> corresponding river barge-loading location

C       e.g, 1st surplus region, Aberdeen SD (022) to river barge region,
C            Sioux Ct IA (609) = 288 miles
C       e.g, 2nd surplus region, Ashley ND (017) to river barge region,
C            St. Paul MN (601) = 351 miles

C       If a new surplus region were added to the end of the surplus region list,
C       then its truck mileage to the corresponding river barge region would be
C       added to the end of this list.

C       If a new surplus region were added to position 10, then its truck mileage
C       to the corresponding river barge-loading location would be at position 10
C       in this list.

C       If a surplus region were deleted, then its truck mileage to the
C       river barge-loading location would be removed from this list.
C       e.g, if Aberdeen SD were not a surplus region, then 288 would be deleted.

C       If a river barge-loading location were deleted, make sure that no surplus
C       region is linked to it.

 288.000 351.000 153.000 436.000 139.000 357.000 532.000 999.000 658.000 497.000
 492.000 444.000 999.000 479.000 381.000 181.000 425.000 999.0001069.000 301.000
 608.000                                                                   0.000

C-------------------------------------------------------------------------

C       Explanation for the Third part

C        (3) Each surplus region(21) is linked to all 20 port locations for
```

```
C          export.

C        Diagrammatically,

C        surplus region 1    ----------> port region 1 for export
C                            ----------> port region 2 for export
C                                              .
C                                              .
C                            ----------> port region 20 for export

C        surplus region 2    ----------> port region 1 for export
C                            ----------> port region 2 for export
C                                              .
C                                              .
C                            ----------> port region 20 for export

C        .
C        .
C        .

C        surplus region 21   ----------> port region 1 for export
C                            ----------> port region 2 for export
C                                              .
C                                              .
C                            ----------> port region 20 for export

C        There are 21 surplus regions and 20 port locations thus a (21 by 20)
C        matrix.
C        e.g, 1st surplus region, Aberdeen SD(064) to 1st port region,
C             Mobile AL (701)      = 1394 miles
C        e.g, 1st surplus region, Aberdeen SD(D64) to 2nd port region,
C             New Orleans LA (702) = 1388 miles

C        The next matrix is for the next surplus region and its mileages to all
C        the port regions.

C        If a new surplus region were added to the end of the surplus region list,
C        then its truck mileages to all the port regions would be added to the list

C        If a new surplus region were added to position 10, then its truck mileage
C        to all the port regions would be positioned at the 10th position in the
C        matrix.

C        If a surplus region were deleted, then its truck mileages to all the
C        port regions would be removed.
C        e.g, if Aberdeen SD were removed as a surplus region, then the first 2 lines
C        (below) would be removed. The file would start with 1473.

C        If a port region were deleted, make sure that there is no surplus
C        region still linked to the deleted port region.

1394.0001388.0001265.0001352.0001487.0001562.0001353.000 917.000 962.000 685.000
 377.0001328.0001398.0001700.0001649.0001698.000 917.000 962.000 685.000 377.000
1473.0001467.0001336.0001423.0001558.0001623.0001414.000 978.0001004.000 746.000
 390.0001311.0001381.0001690.0001639.0001688.000 978.0001004.000 746.000 390.000
```

```
1393.0001434.0001417.0001532.0001667.0001495.0001286.000 850.000 767.000 618.000
 153.0001520.0001590.0001935.0001952.0002001.000 850.000 767.000 618.000 153.000
1589.0001583.0001433.0001520.0001655.0001708.0001499.0001063.0001058.000 831.000
 444.0001195.0001265.0001604.0001617.0001666.0001063.0001058.000 831.000 444.000
1245.0001239.0001151.0001263.0001398.0001449.0001240.000 804.000 849.000 572.000
 334.0001474.0001544.0001758.0001707.0001756.000 804.000 849.000 572.000 334.000
1568.0001588.0001465.0001460.0001687.0001676.0001467.0001031.000 971.000 799.000
 357.0001316.0001386.0001753.0001797.0001846.0001031.000 971.000 799.000 357.000
1678.0001672.0001516.0001554.0001683.0001804.0001595.0001159.0001154.000 927.000
2016.0001096.0001166.0001505.0001529.0001578.0001159.0001154.000 927.0002016.000
1810.0001716.0001414.0001324.0001428.0002292.0002501.0002141.0002208.0001953.000
1997.0001323.0001151.000 589.000 215.000 115.0002141.0002208.0001953.0001997.000
1920.0001914.0001758.0001783.0001912.0002054.0001845.0001409.0001371.0001177.000
 757.000 916.000 986.0001412.0001488.0001537.0001409.0001371.0001177.000 757.000
2081.0002075.0001916.0001933.0002062.0002215.0002006.0001570.0001532.0001338.000
 918.000 755.000 825.0001290.0001392.0001441.0001570.0001532.0001338.000 918.000
1962.0001938.0001746.0001763.0001892.0002149.0001940.0001504.0001499.0001272.000
 885.000 750.000 820.0001208.0001284.0001333.0001504.0001499.0001272.000 885.000
1952.0001922.0001730.0001747.0001876.0002199.0001996.0001560.0001587.0001328.000
 973.000 702.000 772.0001071.0001147.0001196.0001560.0001587.0001328.000 973.000
1977.0001883.0001581.0001491.0001595.0002459.0002636.0002276.0002322.0002054.000
1750.0001131.000 959.000 379.000  30.000 125.0002276.0002322.0002054.0001750.000
1665.0001659.0001536.0001623.0001758.0001778.0001569.0001133.0001093.000 901.000
 479.0001194.0001264.0001631.0001699.0001748.0001133.0001093.000 901.000 479.000
1487.0001481.0001331.0001418.0001553.0001662.0001453.0001017.0001062.000 785.000
 475.0001228.0001298.0001603.0001552.0001601.0001017.0001062.000 785.000 475.000
1316.0001312.0001224.0001339.0001474.0001453.0001244.000 808.000 853.000 576.000
 268.0001437.0001507.0001805.0001754.0001803.000 808.000 853.000 576.000 268.000
1520.0001507.0001337.0001364.0001493.0001767.0001564.0001128.0001173.000 896.000
 677.0001134.0001204.0001414.0001363.0001412.0001128.0001173.000 896.000 677.000
2289.0002195.0001910.0001820.0001924.0002731.0002743.0002311.0002357.0002089.000
1991.000 802.000 630.000  79.000 334.000 459.0002311.0002357.0002089.0001991.000
1506.0001412.0001110.0001020.0001124.0001988.0002231.0001876.0001943.0001688.000
1750.0001553.0001382.000 879.000 505.0009999.9991876.0001943.0001688.0001750.000
1465.0001476.0001364.0001451.0001586.0001573.0001364.000 928.000 923.000 696.000
 309.0001326.0001396.0001735.0001752.0001801.000 928.000 923.000 696.000 309.000
1794.0001788.0001647.0001685.0001814.0001907.0001698.0001262.0001222.0001030.000
 608.0001065.0001135.0001512.0001587.0001636.0001262.0001222.0001030.000 608.000

C-------------------------------------------------------------------------------

C       Explanation for the Fourth part

C       (4) Each deficit region is linked to its
C           corresponding barge-unloading river location.

C       There are 9 deficit regions, thus a 1 by 9 matrix.
C       Diagrammatically,

C       deficit region 1     ----------> corresponding barge-unloading location

C       deficit region 2     ----------> corresponding barge-unloading location
C       .
C       .
C       .
```

```
C       deficit region 9    ----------> corresponding barge-unloading location

C       e.g, the 1st deficit region Lincoln NE (035) to its
C            barge-unloading river location, Nebraska City NE (611) = 54 miles
C       e.g, the 2nd deficit region, Madison WI (123) to its
C            barge-unloading river location, Dubuque IA (604) = 97 miles

C       If a new deficit region were added to the end of the deficit region list,
C       then its truck mileage to the corresponding river barge-unloading region
C       would be added to the end of this list.

C       If a new deficit region were added to position 10, then its truck mileage
C       to the corresponding river barge-unloading region would be at position
C       10 in this list.

C       If a deficit region were deleted, then its truck mileage to the
C       river barge-unloading region would be removed from this list.
C       e.g, if Lincoln NE were not a deficit region, then 54 would be deleted.

C       If a barge-unloading river location were deleted, make sure that there
C       is no deficit region linked to it.

  54.000  97.000  65.000  80.000 104.000 999.000 114.000   1.000 171.000   0.000
```

```
C                                F3.drm

C       This is the 3rd file and it deals with the rail costs (cents/bushel).
C       Note: 999.999 indicates an unfeasible rail cost.

C       In this model, 4 sets of rail cost must be found.
C       This includes
C       (1) rail cost from each surplus region to each deficit region
C       (2) rail cost from each surplus region to each river
C           barge-loading region
C       (3) rail cost from each surplus region to each port region
C       (4) rail cost from each barge-unloading river location to
C           each deficit region

C       Diagrammatically,

C                            (1)
C              ---------------------------------> deficit regions
C              |                                        .
C              |                                        |
C              |                                        |
C              |                                        | (4)
C              |                                        |
C              |                                  river barge-unloading regions
C              |                  (2)             ------------------------------
C       surplus regions ---------------------> river barge-loading regions
C              |
C              |
C              |
C              |                  (3)
C              ---------------------------------> port regions


C-------------------------------------------------------------------------

C       First part
C       (1) Each surplus region is linked to each of the deficit regions.
C       In this model there are 21 surplus regions and 9 deficit regions.
C       Each surplus region is linked to all deficit regions(9). Thus
C       resulting in a 21 by 9 matrix.

C       surplus region 1   ----------> deficit region 1
C                          ----------> deficit region 2
C                                            .
C                                            .
C                          ----------> deficit region 9


C       surplus region 2   ----------> deficit region 1
C                          ----------> deficit region 2
C                                            .
C                                            .
C                          ----------> deficit region 9

C       .
```

```
C        .
C        .

C        surplus region 21  ----------> deficit region 1
C                           ----------> deficit region 2
C                                             .
C                                             .
C                          ----------> deficit region 9


C        For example,
C         the rail cost from Aberdeen SD (022) to 1st deficit region,
C         Lincoln NE (035) = 29.400 cents/bushel
C         the rail cost from Aberdeen SD (022) to 2nd deficit region,
C         Madison WI (123) = 46.800 cents/bushel

  29.400  46.800 100.200  97.800  60.000  85.200  26.400  97.200  76.800   0.000
  44.400  37.200 104.400 111.600  79.200  87.600  37.200 106.800  75.000   0.000
  49.200  48.000 999.999 999.999  84.000 109.800  27.000  82.200 108.600   0.000
  42.600  50.400  94.200 113.400  80.400  78.000  33.600 112.800  84.000   0.000
  25.200  37.200 113.400  92.400  55.200  78.000  31.200 105.600  77.400   0.000
  43.800  51.000  91.200 116.400  79.200  91.200  32.400  87.600 112.200   0.000
  48.600  56.400  88.200 120.000  86.400  71.400  39.600 118.800  90.600   0.000
 999.999 999.999 103.200 999.999 999.999  69.600 999.999 999.999 999.999   0.000
  96.000  65.400  55.800 999.999 999.999  76.200  47.400 999.999 999.999   0.000
  56.400  78.000  42.600 999.999 999.999  58.200  64.200 999.999 999.999   0.000
  67.200  75.600  43.800 999.999 999.999  60.000  69.000 999.999 999.999   0.000
  67.800  85.200  49.200 999.999 999.999  43.200  63.000 999.999 999.999   0.000
 999.999 999.999  90.000 999.999 999.999  56.400 999.999 999.999 999.999   0.000
  47.400  52.800  84.000 110.400  82.800  79.200  36.600  91.200 114.000   0.000
  35.400  52.800  94.200 103.200  66.000  79.200  31.800 111.000  82.800   0.000
  34.200  41.400 112.800  97.200  69.600  95.400  25.800  79.800 102.000   0.000
  38.400  58.800  99.600 109.800  72.600  67.800  48.600 127.200  99.000   0.000
 999.999 999.999  58.200 999.999 999.999 999.999 999.999 999.999 999.999   0.000
 999.999 999.999 999.999 999.999 999.999  91.800 999.999 999.999 999.999   0.000
  34.200  42.000 102.600 105.000  72.000  85.800  26.400  96.600  71.400   0.000
  55.200  64.800  76.800 123.000  90.000  71.400  44.400 123.600  94.800   0.000

C-------------------------------------------------------------------------

C        Second part

C        (2) Each surplus region is linked to 43 river barge-loading
C            locations.


C        surplus region 1   ----------> river barge-loading region 1
C                           ----------> river barge-loading region 2
C                                             .
C                                             .
C                          ----------> river barge-loading  region 43


C        surplus region 2   ----------> river barge-loading  region 1
C                           ----------> river barge-loading  region 2
```

```
C                                        .
C                                        .
C                          ----------> river barge-loading  region 43

C        .
C        .
C        .

C        surplus region 21  ----------> river barge-loading  region 1
C                           ----------> river barge-loading  region 2
C                                        .
C                                        .
C                          ----------> river barge-loading  region 43

C        A 21 by 43 matrix containing the rail costs from each surplus region
C        to 43 river ports.

C        For example,
C            the rail cost from 1st surplus region, Aberdeen SD (022)
C            to the 1st river port region, St. Paul MN (601)
C            = 22.800 cents/bushel, and
C            the rail cost from 2nd surplus region, Ashley ND (017)
C            to the 1st river port region, St. Paul MN (601)
C            = 21.600 cents/bushel

  22.800  34.200 999.999 999.999 999.999 999.999 999.999 999.999  21.000 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  21.600  31.800 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  26.400  19.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  33.600  42.000 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  31.200  38.400 999.999 999.999 999.999 999.999 999.999 999.999  19.800 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  32.400  31.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
```

```
 39.600  48.000 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999  51.000 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999  47.400 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999  57.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999  50.400 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999  68.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999  72.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999  58.800 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
 36.600  38.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
 31.800  40.200 999.999 999.999 999.999 999.999 999.999 999.999  31.800 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
 11.000  28.800 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
 48.600  57.000 999.999 999.999 999.999 999.999 999.999 999.999  36.500 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
```

```
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  26.400  27.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000
  44.400  45.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                   0.000

C-------------------------------------------------------------------------------

C       Third part

C        (3) Each surplus region is linked to all the port locations for export.
C       There are 20 ports in this model

C       surplus region 1   ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                            .
C                                            .
C                          ----------> port region 20 for export

C       surplus region 2   ------;---> port region 1 for export
C                          ----------> port region 2 for export
C                                            .
C                                            .
C                          ----------> port region 20 for export

C       .
C       .
C       .

C       surplus region 21  ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                            .
C                                            .
C                          ----------> port region 20 for export


C       The rail costs from each surplus region to the 20 ports. Thus, a
C       21 by 20 matrix.

C       For example,
```

```
C            from the 1st surplus region, Aberdeen (022) to
C            1st port region, Mobile AL (701)      = 999.999 cents/bushel
C            which means that it is not feasible.

C        Another example is
C            from the 1st surplus region, Aberdeen (022) to
C            2nd port region, New Orleans LA(702) = 94.200 cents/bushel

 999.999  94.200  60.690 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  34.200 108.000 119.400 130.200 133.800 999.999 999.999 999.999 999.999  34.200
 999.999 112.800  67.830 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  31.800 129.000 139.800 132.600 136.800 999.999 999.999 999.999 999.999  31.800
 999.999 107.400 106.800 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  19.200 103.200 114.000 159.000 158.400 999.999 999.999 999.999 999.999  19.200
 999.999 105.600  71.910 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  42.000  96.000 106.800 123.000 126.600 999.999 999.999 999.999 999.999  42.000
 999.999  91.200  84.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  38.400 121.200 132.000 127.200 130.800 999.999 999.999 999.999 999.999  38.400
 999.999 107.400  76.500 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  31.200  94.200 105.600 130.800 135.000 999.999 999.999 999.999 999.999  31.200
 999.999 111.600  76.500 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  48.000  89.400 100.800 128.400 120.600 999.999 999.999 999.999 999.999  48.000
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999  91.800  51.000  24.000 999.999 999.999 999.999 999.999 999.999
 999.999 135.000 134.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  50.400  66.000  65.000 111.600 138.600 999.999 999.999 999.999 999.999  50.400
 999.999 141.600 137.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  57.600  58.800  59.400 104.400 131.400 999.999 999.999 999.999 999.999  57.600
 999.999 146.400 126.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  68.400  53.400  55.800  91.200 118.200 999.999 999.999 999.999 999.999  68.400
 999.999 999.999 117.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  72.600  60.000  59.400  92.400 119.400 999.999 999.999 999.999 999.999  72.600
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999  82.800  37.800   9.600 999.999 999.999 999.999 999.999 999.999
 999.999 118.800  73.950 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  38.400  87.000  98.400 124.200 127.800 999.999 999.999 999.999 999.999  38.400
 999.999 100.800  93.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  40.200 102.000 113.400 124.200 127.800 999.999 999.999 999.999 999.999  40.200
 999.999  92.400  92.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  28.800 115.800 127.200 144.600 144.000 999.999 999.999 999.999 999.999  28.800
 999.999 108.600 101.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  57.000 112.200 123.600 112.800 117.000 999.999 999.999 999.999 999.999  57.000
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999  52.800  15.600  44.400 999.999 999.999 999.999 999.999 999.999
 999.999 999.999  60.900  88.800 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 100.800  75.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  27.600 104.400 115.200 142.800 135.000 999.999 999.999 999.999 999.999  27.600
 999.999 999.999  81.000 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  45.600  79.200  90.600 116.400 120.600 999.999 999.999 999.999 999.999  45.600
C--------------------------------------------------------------------------------

C        Fourth part

C        (4) Each barge-unloading river location is linked to 9
```

```
C            deficit regions.

C        barge-unloading river location 1   ----------> deficit region 1
C                                           ----------> deficit region 2
C                                             .
C                                             .
C                                           ----------> deficit region 9


C        barge-unloading river location 2   ----------> deficit region 1
C                                           ----------> deficit region 2
C                                             .
C                                             .
C                                           ----------> deficit region 9

C        .
C        .
C        .

C        barge-unloading river location 43  ----------> deficit region 1
C                                           ----------> deficit region 2
C                                             .
C                                             .
C                                           ----------> deficit region 9


C        The rail costs from each river port to the 9 deficit regions.
C        There are 43 river ports. Thus a 43 by 9 matrix.

C        For example,
C                from the 1st barge-unloading river location, St. Paul, MN(601)
C                to the 1st deficit region, Lincoln NE (035)
C                = 999.999 cents/bushel

C        Another example is
C                from the 8th barge-unloading river location, St. Louis MO (608)
C                to the 5th deficit region, Rolla MO (096) = 21.000 cents/bushel

 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999  21.000 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999       0
```

999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	13.000	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	18.000	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	19.200	999.999	999.999	999.999	999.999	999.999	0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0

```
C                                   F4.drm

C       This is the 4th data file data dealing with the barge costs and
C       shiprates.

C       The barge costs from 43 river regions to 5 river shipping locations.
C       The 5 river shipping locations are :
C               Nashville, TN (625)
C               Knoxville, TN (626)
C               Chatanoo, TN  (627)
C               Guntersv, AL  (628)
C               Florence, AL  (629)

C       For example,
C       the barge cost from the 1st river barge region,
C                   St. Paul, MN (601) to Nashville TN (625) = 999.999 $/bushel
C                   St. Paul, MN (601) to Knoxville TN (626) =   0.327 $/bushel
C                   St. Paul, MN (601) to Chatanoo  TN (627) =   0.299 $/bushel
C                   St. Paul, MN (601) to Guntersv  AL (628) =   0.284 $/bushel
C                   St. Paul, MN (601) to Florence  AL (629) =   0.269 $/bushel

 999.999   0.327   0.299   0.284   0.269                                    0
   0.270 999.999 999.999 999.999 999.999                                    0
 999.999   0.273   0.246   0.230   0.215                                    0
 999.999   0.263   0.236   0.221   0.205                                    0
 999.999   0.254   0.227   0.211   0.197                                    0
 999.999   0.226   0.199   0.183   0.168                                    0
 999.999   0.209   0.181   0.166   0.151                                    0
 999.999   0.170   0.134   0.127   0.112                                    0
 999.999   0.378   0.350   0.335   0.320                                    0
 999.999   0.348   0.321   0.305   0.290                                    0
 999.999   0.458   0.421   0.400   0.337                                    0
 999.999   0.415   0.378   0.357   0.337                                    0
 999.999   0.389   0.352   0.331   0.311                                    0
 999.999   0.341   0.305   0.283   0.263                                    0
 999.999   0.324   0.287   0.266   0.245                                    0
 999.999   0.308   0.271   0.249   0.229                                    0
 999.999   0.290   0.253   0.232   0.211                                    0
 999.999   0.206   0.174   0.155   0.137                                    0
 999.999   0.191   0.158   0.139   0.122                                    0
 999.999   0.153   0.120   0.102   0.084                                    0
 999.999   0.136   0.104   0.085   0.067                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.999   0.247   0.215   0.196   0.178                                    0
 999.999   0.241   0.208   0.190   0.172                                    0
 999.999   0.199   0.167   0.148   0.130                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
```

```
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
 999.000 999.000 999.000 999.000 999.000                                    0
C-------------------------------------------------------------------------------

C        The barge costs from 43 river port shipping points to 5 port locations.
C        The 5 river port shipping points are:
C                 New Orle, LA (702)
C                 Mobile,   AL (701)
C                 Galvestn, TX (703)
C                 Chicago,  IL (710)
C                 Portland, OR (713)

C        For example, the 1st river region (NE. CITY)
C                 St. Paul MN (601) to New Orle, LA (702) =   0.283 $/bushel
C                 St. Paul MN (601) to Mobile,   AL (701) =   0.345 $/bushel
C                 St. Paul MN (601) to Galvestn, TX (703) =   0.398 $/bushel
C                 St. Paul MN (601) to Chicago,  IL (710) =   0.234 $/bushel
C                 St. Paul MN (601) to Portland, OR (713) = 999.999 $/bushel

   0.283   0.345   0.398   0.234 999.000                                    0
 999.999 999.999 999.999 999.999 999.999                                    0
   0.229   0.291   0.344   0.180 999.000                                    0
   0.219   0.281   0.334   0.170 999.000                                    0
   0.210   0.272   0.325   0.161 999.000                                    0
   0.182   0.244   0.297   0.133 999.000                                    0
   0.164   0.226   0.280   0.116 999.000                                    0
   0.125   0.187   0.240   0.109 999.000                                    0
   0.334   0.273   0.449   0.300 999.000                                    0
   0.304   0.366   0.419   0.270 999.000                                    0
   0.398   0.429   0.555   0.352 999.000                                    0
   0.355   0.398   0.512   0.309 999.000                                    0
   0.329   0.379   0.486   0.283 999.000                                    0
   0.281   0.344   0.438   0.238 999.000                                    0
   0.264   0.331   0.421   0.060 999.000                                    0
   0.248   0.319   0.404   0.079 999.000                                    0
   0.230   0.306   0.387   0.099 999.000                                    0
   0.158   0.309   0.430 999.000 999.000                                    0
   0.147   0.281   0.398 999.000 999.000                                    0
   0.120   0.214   0.319 999.000 999.000                                    0
   0.074   0.200   0.267 999.000 999.000                                    0
   0.069   0.187   0.253 999.000 999.000                                    0
   0.053   0.149   0.207 999.000 999.000                                    0
   0.057   0.157   0.216 999.000 999.000                                    0
 999.999 999.999 999.999 999.000 999.000                                    0
   0.123   0.216   0.327 999.000 999.000                                    0
   0.104   0.170   0.273 999.000 999.000                                    0
```

```
   0.094    0.144    0.242 999.000 999.000                                  0
   0.083    0.118    0.213 999.000 999.000                                  0
   0.098    0.229    0.287 999.000 999.000                                  0
   0.093    0.217    0.273 999.000 999.000                                  0
   0.063    0.144    0.188 999.000 999.000                                  0
 999.000  999.000  999.000 999.000 999.000                                  0
   0.042    0.122    0.176 999.000 999.000                                  0
   0.034    0.102    0.153 999.000 999.000                                  0
 999.000  999.000  999.000 999.000 999.000                                  0
 999.000  999.000  999.000 999.000   0.170                                  0
 999.000  999.000  999.000 999.000   0.145                                  0
 999.000  999.000  999.000 999.000   0.113                                  0
 999.000  999.000  999.000 999.000   0.063                                  0
 999.000  999.000  999.000 999.000   0.052                                  0
   0.000  999.000  999.000 999.000 999.000                                  0
 999.000  999.000  999.000 999.000   0.000                                  0


C-------------------------------------------------------------------------

C       The number of river shipping ports above L&D 26

  14                                                                        0

C-------------------------------------------------------------------------

C       The codes for the river shipping points, where
C       601 is the code number for St. Paul MN.

 601 644 603 604 605 606 607 609 610 611 612 613 614 645                    0

C-------------------------------------------------------------------------

C       Shiprates from 20 ports to 25 world demand locations

C       For example, the shiprate from the 1st port
C               Mobile, AL (701) to 1st world demand location, Scandinavia (801)
C               = 0.475 $/bushel
C               Mobile, AL (701) to 2nd world demand location, N.C. Europe(802)
C               = 0.374 $/bushel

   0.475    0.374    0.320    0.344    0.453    0.538    0.480    0.495    0.477    0.734
   1.024    0.418    0.511    0.519    0.828    0.461    0.632    0.434    0.707  999.999
   0.285    0.082    0.306    0.161    0.215                                      0.000
   0.475    0.374    0.320    0.344    0.453    0.538    0.480    0.495    0.477    0.734
   1.024    0.418    0.511    0.519    0.828    0.461    0.632    0.434    0.707  999.999
   0.285    0.082    0.306    0.161    0.215                                      0.000
   0.490    0.387    0.332    0.355    0.465    0.553    0.499    0.508    0.494    0.751
   1.045    0.435    0.520    0.525    0.840    0.461    0.621    0.417    0.696  999.999
   0.277    0.059    0.272    0.177    0.229                                      0.000
   0.490    0.387    0.332    0.355    0.465    0.553    0.499    0.508    0.494    0.750
   1.045    0.435    0.520    0.525    0.840    0.461    0.621    0.417    0.696  999.999
   0.277    0.059    0.272    0.177    0.229                                      0.000
   0.499    0.394    0.341    0.363    0.472    0.562    0.507    0.514    0.504    0.761
   1.056    0.445    0.525    0.525    0.856    0.358    0.855    0.417    0.967  999.999
```

```
 0.313   0.048   0.273   0.169   0.238                                   0.000
 0.434   0.374   0.332   0.348   0.505   0.462   0.535   0.552   0.399   0.645
 0.970   0.352   0.842   0.799   1.153   1.002   0.931   0.883   0.955 999.999
 0.339   0.166   0.301   0.148   0.148                                   0.000
 0.375   0.326   0.287   0.307   0.449   0.437   0.482   0.490   0.367   0.605
 0.973   0.330   0.577   0.547   0.902   0.947   0.887   0.838   0.982 999.999
 0.347   0.202   0.312   0.171   0.169                                   0.000
 0.740   0.740   0.624   0.636   0.934   0.839   0.999   0.974   0.711   1.157
 1.545   0.705   1.527   1.722   2.236   2.184   2.094   1.931   2.125 999.999
 0.932   0.660   0.872   0.637   0.541                                   0.000
 0.794   0.794   0.677   0.690   0.988   0.892   1.052   1.027   0.765   1.190
 1.599   0.759   1.580   1.775   2.290   2.237   2.147   1.984   2.178 999.999
 0.986   0.714   0.926   0.691   0.595                                   0.000
 0.853   0.762   0.736   0.749   1.047   0.951   1.111   1.091   0.824   1.250
 1.659   0.819   1.640   1.834   2.350   2.296   2.101   2.043   2.238 999.999
 0.995   0.775   0.978   0.750   0.654                                   0.000
 0.867   0.716   0.690   0.763   1.061   0.965   1.125   1.105   0.838   1.264
 1.671   0.831   1.654   1.848   2.362   2.310   2.218   2.057   2.192 999.999
 1.059   0.787   0.933   0.704   0.550                                   0.000
 0.832   0.753   0.775   0.760   0.881   0.901   0.689   0.652   0.835   1.211
 1.483   0.746   0.456   0.536   0.375   0.291   0.431   0.228   0.499 999.999
 0.317   0.570   0.257   0.447   0.630                                   0.000
 0.820   0.769   0.761   0.747   0.868   0.888   0.694   0.657   0.821   1.197
 1.413   0.727   0.461   0.541   0.380   0.296   0.430   0.233   0.252 999.999
 0.312   0.554   0.257   0.434   0.612                                   0.000
 0.764   0.716   0.707   0.692   0.815   0.831   0.882   0.877   0.761   1.138
 1.340   0.774   0.741   0.697   0.647   0.513   0.461   0.403   0.554 999.999
 0.246   0.492   0.382   0.376   0.549                                   0.000
 0.735   0.687   0.678   0.663   0.785   0.801   0.613   0.725   0.682   1.107
 1.517   0.822   0.489   0.576   0.416   0.289   0.427   0.225   0.492 999.999
 0.212   0.457   0.200   0.344   0.496                                   0.000
 0.794   0.746   0.736   0.720   0.844   0.853   0.613   0.914   0.737   1.257
 1.483   0.805   1.049   0.982   0.715   0.514   0.457   0.395   0.485 999.999
 0.216   0.451   0.361   0.334   0.487                                   0.000
 0.816   0.665   0.708   0.714   0.840   0.874   0.770   0.765   0.817   1.055
 1.468   0.818   0.846   1.173   1.148   1.132   1.503   1.055   1.612 999.999
 0.940   0.812   0.904   0.772   0.801                                   0.000
 0.859   0.708   0.741   0.757   0.883   0.917   0.813   0.817   0.860   1.098
 1.510   0.861   0.889   1.216   1.191   1.175   1.546   1.098   1.655 999.999
 0.983   0.855   0.947   0.815   0.844                                   0.000
 0.963   0.811   0.854   0.860   0.987   1.021   0.918   0.912   0.964   1.202
 1.615   0.965   0.993   1.320   1.295   1.279   1.650   1.202   1.759 999.999
 1.087   0.959   1.051   0.919   0.948                                   0.000
 0.871   0.733   0.763   0.647   0.895   0.925   0.826   0.820   0.872   1.110
 1.522   0.873   0.901   1.228   1.203   1.187   1.558   1.110   1.666 999.999
 0.995   0.867   0.959   0.827   0.856                                   0.000

C-------------------------------------------------------------------------------

C       Number of ports at Great Lakes for export
   8                                                                        0

C-------------------------------------------------------------------------------

C       Codes for the above ports
```

```
C               (708) Toledo,  OH
C               (709) Saginaw, MI
C               (710) Chicago, IL
C               (711) Duluth,  MN
C               (717) Toledo,  BC
C               (718) Saginaw, BC
C               (719) Chicago, BC
C               (720) Duluth,  BC

 708 709 710 711 717 718 719 720                                              0

C-------------------------------------------------------------------------------
```

```
C                                   F8.drm

C       This is the 5th data file for DURUM which deals with supplies and demands.

C-------------------------------------------------------------------------------

C       These lines show the amount of surplus produced by each surplus region.
C       For example, the 1st surplus region,
C                    Aberdeen SD (022) produces 3.143 million/bushel.

   3.143    5.250    1.433    8.695    0.681   14.873   13.869    0.000    2.355    1.081
   0.027    0.094    0.981   19.375    0.387    4.602    0.021    0.619    4.983    4.036
  27.848                                                                            0

C-------------------------------------------------------------------------------

C       These lines show the amount of corn demanded by the 9 deficit regions.
C       For example, the 1st deficit region,
C                    Lincoln NE (035) requires  3.941 million/bushel.

   3.941    9.573    1.489    0.741    3.317    1.863   17.329    3.000    3.302        0

C-------------------------------------------------------------------------------

C       The remaining lines show the amount demanded by 25 foreign regions
C       (demand per quarter).

C       For example, the 2nd foreign region, N.C Europe (802)
C         in the 1st quarter, it requires 0.217 million bushel,
C         in the 2nd quarter, it requires 0.337 million bushel,
C         in the 3rd quarter, it requires 0.693 million bushel,
C         in the 4th quarter, it requires 0.597 million bushel,

   0.000    0.000    0.000    0.000                                                  0
   0.217    0.337    0.693    0.597                                                  0
   0.213    0.557    0.858    1.157                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.307    1.453    0.000    0.060                                                  0
   0.000    0.000    0.000    0.000                                                  0
   2.779    0.926    0.418    0.734                                                  0
   0.003    0.003    0.003    0.003                                                  0
  17.145   11.253    9.748   10.538                                                  0
   0.011    0.062    0.124    0.031                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.222    0.185    0.191    0.198                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
   0.000    0.000    0.000    0.000                                                  0
```

0.000	0.000	0.000	0.000	
0.103	0.210	0.314	0.241	0
2.293	1.320	2.412	1.720	0
0.027	0.019	0.026	0.018	0
				0

C F9.drm

C This 6th and last data file consists of all the location names and
C code numbers of all the regions. The grain used in this model is DURUM.

C----------

C 21 Surplus regions

C Code number	Location name	State	
022	Aberdeen	SD	0
017	Ashley	ND	0
071	Bemidgi	MN	0
015	Bismarck	ND	0
024	Brookings	SD	0
013	Devils L	ND	0
014	Dickinson	ND	0
362	El Centro	CA	0
323	Glasgo	MT	0
322	Havre	MT	0
321	Lewistown	MT	0
324	Livingston	MT	0
715	Los Angels	CA	0
012	Minot	ND	0
021	Mobridge	SD	0
072	Ortonville	MN	0
023	Rapid City	SD	0
361	Stockton	CA	0
401	Tucson	AZ	0
016	Valley Cty	ND	0
011	Williston	ND	0

C----------

C 9 Deficit regions

C Code number	Location name	State	
035	Lincoln	NE	0
123	Madison	WI	0
331	Pendleton	OR	0
113	Prt Alan	LA	0
096	Rolla	MO	0
381	Salt Lk C	Ut	0
073	St Paul	MN	0
291	Buffalo	NY	0
151	Tiffin	OH	0

C----------

C 43 River regions

C Code number	Location name	State

601	ST. PAUL,	MN	0
602	DULUTH,	RV	0
603	MCGREGOR,	IA	0
604	OUBUQUE,	IA	0
605	CLINTON,	IA	0
606	BURLINGT,	IA	0
607	HANNIBAL,	MO	0
608	ST LOUIS,	MO	0
609	SIOUX CT,	IA	0
610	OMAHA,	NE	0
611	NE CITY,	NE	0
612	ST JOSEP,	MO	0
613	KANSAS C,	MO	0
614	GLASCO,	MO	0
615	OTTAWA,	IL	0
616	PEORIA,	IL	0
617	BEARDSTO,	IL	0
618	CINCINNA,	OH	0
619	LOUISVIL,	KY	0
620	EVANSVIL,	IN	0
621	CAIRO,	IL	0
622	HICKMAN,	KY	0
623	OSCEOLA,	TN	0
624	MEMPHIS,	TN	0
625	BUFFALO,	TN	0
626	KNOXVILL,	TN	0
627	CHATANOO,	TN	0
628	GUNTERSV,	AL	0
629	FLORENCE,	AL	0
630	CATOOSA,	OK	0
631	MUSKOGEE,	OK	0
632	PINEBLUF,	AR	0
633	DES ARC,	AR	0
634	GREENWOO,	MS	0
635	VICKSBUR,	MS	0
636	MONROE,	LA	0
637	LEWISTON,	ID	0
638	CENTRL F,	WA	0
639	PASCO,	WA	0
640	ROOSEVEL,	WA	0
641	THE DALL,	OR	0
642	NEW ORLE,	LA	0
643	PORTLAND,	OR	0

C----------

C 20 Port regions

C Code number Location name State

701	MOBILE,	AL	0
702	NEW ORLE,	LA	0
703	GALVESTN,	TX	0
704	CORPUS C,	TX	0
705	BROWNSVI,	TX	0

706	CHARLEST,	SC	0
707	BALTIMOR,	MD	0
708	TOLEDO,	OH	0
709	SAGINAW,	MI	0
710	CHICAGO,	IL	0
711	DULUTH,	MN	0
712	SEATTLE,	WA	0
713	PORTLAND,	OR	0
714	SAN FRAN,	CA	0
715	LONG BEA,	CA	0
716	SAN DIEG,	CA	0
717	TOLEDO,	BC	0
718	SAGINAW,	BC	0
719	CHICAGO,	BC	0
720	DULUTH,	BC	0

C----------

C 25 Foreign regions

C Code number Location name

801	SCANDINAVIA	0
802	N.C. EUROPE	0
803	S.W. EUROPE	0
804	ISLANDS	0
805	ADRIATIC	0
806	USSR	0
807	E BLOCK EURO	0
808	E MEDITERRAN	0
809	N AFRICA	0
810	RED SEA	0
811	E AFRICA	0
812	W AFRICA	0
813	PERSIAN GULF	0
814	W ASIA	0
815	SE ASIA	0
816	TAIWAN	0
817	KOREA	0
818	JAPAN	0
819	CHINA	0
820	CANADA	0
821	W MEXICO	0
822	E MEXICO	0
823	WS AMERICA	0
824	CENT AMERICA	0
825	CARIBBEAN	0

******* OPTIMAL SOLUTION *******
TOTAL COST 1299646976.
DURUM WHEAT SHIPMENT, FULLER, GRANT, TEH, FELLIN
UNIT = 1 (THOUSAND BUSHEL)

SUPPLY	114352.
DEMAND	114273.

507	508	508	6871

Network Generator for Grain Shipment Problem

DURUM WHEAT SHIPMENT, FULLER, GRANT, TEH, FELLIN

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME 1	2	3	4	TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
S 022 Aberdeen	SD	3143.								
D 035 Lincoln	NE R		156.	156.	156.	40.	508.	0.29400	149352.	87081.
D 096 Rolla	MO R		148.	829.	829.	829.	2635.	0.60000	1581000.	451692.
S 017 Ashley	ND	5250.								
D 123 Madison	WI R		2393.	2393.	464.	00.	5250.	0.37200	1953000.	899955.
S 071 Bemidgi	MN	1433.								
P 711 DULUTH,	MN T		00.	836.	597.	00.	1433.	0.16115	230926.	246261.
S 015 Bismarck	ND	8695.								
D 035 Lincoln	NE R		825.	825.	00.	639.	2289.	0.42600	975114.	392380.
D 073 St Paul	MN R		00.	4332.	1211.	863.	6406.	0.33600	2152416.	1098117.
S 024 Brookings	SD	681.								
D 096 Rolla	MO R		681.	00.	00.	00.	681.	0.55200	375912.	116737.
S 013 Devils L	ND	14873.								
P 711 DULUTH,	MN R		00.	10754.	1435.	597.	12786.	0.31200	3989232.	1847577.
P 720 DULUTH,	BC R		00.	929.	421.	737.	2087.	0.31200	651144.	301572.
S 014 Dickinson	ND	13869.								
P 703 GALVESTN,	TX R		13869.	00.	00.	00.	13869.	0.76500	10609785.	2004071.
S 362 El Centro	CA	00.								
S 323 Glasgo	MT	2355.								
D 331 Pendlton	OR R		361.	372.	372.	372.	1477.	0.55800	824166.	253187.
P 713 PORTLAND,	OR R		225.	185.	191.	198.	799.	0.65000	519350.	115456.
S 322 Havre	MT	1081.								
D 331 Pendlton	OR R		11.	00.	00.	00.	11.	0.42600	4686.	1886.
D 381 Salt Lk C	Ut R		27.	140.	438.	465.	1070.	0.58200	622740.	183419.
S 321 Lewistown	MT	27.								
D 381 Salt Lk C	Ut T		00.	00.	27.	00.	27.	0.54341	14672.	5114.
S 324 Livingston	MT	94.								
D 381 Salt Lk C	Ut R		94.	00.	00.	00.	94.	0.43200	40608.	16113.
S 715 Los Angels	CA	981.								
P 715 LONG BEA,	CA T		216.	210.	314.	241.	981.	0.03213	31521.	168585.
S 012 Minot	ND	19375.								
D 123 Madison	WI R		00.	00.	1929.	2393.	4322.	0.52800	2282016.	740877.
D 113 Prt Alan	LA R		185.	185.	185.	185.	740.	1.10400	816960.	126851.
D 073 St Paul	MN R		4332.	00.	3121.	3469.	10922.	0.36600	3997452.	1872249.
D 291 Buffalo	NY R		750.	750.	750.	750.	3000.	0.91200	2736000.	514260.
P 711 DULUTH,	MN R		00.	00.	391.	00.	391.	0.38400	150144.	56500.
S 021 Mobridge	SD	387.								
D 035 Lincoln	NE R		00.	00.	387.	00.	387.	0.35400	136998.	66340.
S 072 Ortonville	MN	4602.								

R 601 ST. PAUL, MN	R		00.	1382.	2536.	684.	4602.	0.11000	506220.	767890.
S 023 Rapid CitySD		21.								
D 035 Lincoln NE	R		00.	00.	21.	00.	21.	0.38400	8064.	3600.
S 361 Stockton CA		619.								
P 714 SAN FRAN, CA	T		619.	00.	00.	00.	619.	0.08353	51704.	106375.
S 401 Tucson AZ		4983.								
P 703 GALVESTN, TX	R		4983.	00.	00.	00.	4983.	0.60900	3034647.	720044.
S 016 Valley CtyND		4036.								
D 035 Lincoln NE	R		4.	4.	421.	306.	735.	0.34200	251370.	125994.
D 151 Tiffin OH	R		825.	825.	825.	825.	3300.	0.71400	2356200.	565686.
S 011 Williston ND		27848.								
D 381 Salt Lk C Ut	R		344.	325.	00.	00.	669.	0.71400	477666.	114680.
P 703 GALVESTN, TX	R		3408.	2010.	858.	2284.	8560.	0.81000	6933600.	1236920.
P 711 DULUTH, MN	R		00.	19.	8044.	10556.	18619.	0.45600	8490263.	2690446.

SURPLUS REGION	SUPPLY		STORAGE		STORAGE COST
S 022 Aberdeen SD	3143.	2839.	1854.	869.	213809.
S 017 Ashley ND	5250.	2857.	464.	00.	127334.
S 071 Bemidgi MN	1433.	1433.	597.	00.	77834.
S 015 Bismarck ND	8695.	7870.	2713.	1502.	464316.
S 024 Brookings SD	681.	00.	00.	00.	00.
S 013 Devils L ND	14873.	14873.	3190.	1334.	744567.
S 014 Dickinson ND	13869.	00.	00.	00.	00.
S 362 El Centro CA	00.	00.	00.	00.	00.
S 323 Glasgo MT	2355.	1690.	1133.	570.	130456.
S 322 Havre MT	1081.	1043.	903.	465.	92737.
S 321 Lewistown MT	27.	27.	27.	00.	2070.
S 324 LivingstonMT	94.	00.	00.	00.	00.
S 715 Los AngelsCA	981.	765.	555.	241.	60005.
S 012 Minot ND	19375.	14108.	13173.	6797.	1310929.
S 021 Mobridge SD	387.	387.	387.	00.	29677.
S 072 OrtonvilleMN	4602.	4602.	3220.	684.	326571.
S 023 Rapid CitySD	21.	21.	21.	00.	1610.
S 361 Stockton CA	619.	00.	00.	00.	00.
S 401 Tucson AZ	4983.	00.	00.	00.	00.
S 016 Valley CtyND	4036.	3206.	2377.	1131.	258146.
S 011 Williston ND	27848.	24096.	21742.	12840.	2257974.

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME 1	2	3	4	TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
R 601 ST. PAUL, MN			00.	1382.	2536.	684.				
P 702 NEW ORLE, LA	B		00.	1382.	2536.	684.	4602.	0.28300	1302366.	729647.
R 644 DULUTH, RV			00.	00.	00.	00.				
R 603 MCGREGOR, IA			00.	00.	00.	00.				
R 604 DUBUQUE, IA			00.	00.	00.	00.				
R 605 CLINTON, IA			00.	00.	00.	00.				
R 606 BURLINGT, IA			00.	00.	00.	00.				
R 607 HANNIBAL, MO			00.	00.	00.	00.				
R 608 ST LOUIS, MO			00.	00.	00.	00.				

R 609 SIOUX CT, IA	00.	00.	00.	00.
R 610 OMAHA, NE	00.	00.	00.	00.
R 611 NE CITY, NE	00.	00.	00.	00.
R 612 ST JOSEP, MO	00.	00.	00.	00.
R 613 KANSAS C, MO	00.	00.	00.	00.
R 614 GLASCO, MO	00.	00.	00.	00.
R 615 OTTAWA, IL	00.	00.	00.	00.
R 616 PEORIA, IL	00.	00.	00.	00.
R 617 BEARDSTO, IL	00.	00.	00.	00.
R 618 CINCINNA, OH	00.	00.	00.	00.
R 619 LOUISVIL, KY	00.	00.	00.	00.
R 620 EVANSVIL, IN	00.	00.	00.	00.
R 621 CAIRO, IL	00.	00.	00.	00.
R 622 HICKMAN, KY	00.	00.	00.	00.
R 623 OSCEOLA, AR	00.	00.	00.	00.
R 624 MEMPHIS, TN	00.	00.	00.	00.
R 645 BUFFALO, NY	00.	00.	00.	00.
R 626 KNOXVILL, TN	00.	00.	00.	00.
R 627 CHATANOO, TN	00.	00.	00.	00.
R 628 GUNTERSV, AL	00.	00.	00.	00.
R 629 FLORENCE, AL	00.	00.	00.	00.
R 630 CATOOSA, OK	00.	00.	00.	00.
R 631 MUSKOGEE, OK	00.	00.	00.	00.
R 632 PINEBLUF, AR	00.	00.	00.	00.
R 633 DES ARC, AR	00.	00.	00.	00.
R 634 GREENWOO, MS	00.	00.	00.	00.

R 635 VICKSBUR, MS			00.	00.	00.	00.				
R 636 MONROE, LA			00.	00.	00.	00.				
R 637 LEWISTON, ID			00.	00.	00.	00.				
R 638 CENTRL F, WA			00.	00.	00.	00.				
R 639 PASCO, WA			00.	00.	00.	00.				
R 640 ROOSEVEL, WA			00.	00.	00.	00.				
R 641 THE DALL, OR			00.	00.	00.	00.				
R 642 NEW ORLE, LA			00.	00.	00.	00.				
R 643 PORTLAND, OR			00.	00.	00.	00.				

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME 1	2	3	4	TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
P 701 MOBILE, AL			00.	00.	00.	00.				
P 702 NEW ORLE, LA			00.	1382.	2536.	684.				
F 810 RED SEA	S		00.	62.	124.	31.	217.	0.73400	159278.	12282.
F 824 CENT AMERICA	S		00.	1320.	2412.	653.	4385.	0.16100	705985.	248191.
P 703 GALVESTN, TX			22260.	2010.	858.	2284.				
F 802 N.C. EUROPE	S		217.	00.	00.	00.	217.	0.38700	83979.	12282.
F 803 S.W. EUROPE	S		213.	557.	858.	1157.	2785.	0.33200	924620.	157631.
F 805 ADRIATIC	S		307.	1453.	00.	60.	1820.	0.46500	846300.	103012.
F 807 E BLOCK EURO	S		2666.	00.	00.	00.	2666.	0.49900	1330334.	150896.
F 809 N AFRICA	S		17145.	00.	00.	00.	17145.	0.49400	8469630.	970407.
F 810 RED SEA	S		11.	00.	00.	00.	11.	0.75100	8261.	623.
F 824 CENT AMERICA	S		1674.	00.	00.	1067.	2741.	0.17700	485157.	155141.
F 825 CARIBBEAN	S		27.	00.	00.	00.	27.	0.22900	6183.	1528.
P 704 CORPUS C, TX			00.	00.	00.	00.				
P 705 BROWNSVI, TX			00.	00.	00.	00.				
P 706 CHARLEST, SC			00.	00.	00.	00.				
P 707 BALTIMOR, MD			00.	00.	00.	00.				
P 708 TOLEDO, OH			00.	00.	00.	00.				
P 709 SAGINAW, MI			00.	00.	00.	00.				
P 710 CHICAGO, IL			00.	00.	00.	00.				
P 711 DULUTH, MN			00.	11609.	10467.	11153.				
F 802 N.C. EUROPE	S		00.	337.	693.	597.	1627.	0.71600	1164932.	92088.
F 809 N AFRICA	S		00.	11253.	9748.	10538.	31539.	0.83800	26429682.	1785107.
F 825 CARIBBEAN	S		00.	19.	26.	18.	63.	0.55000	34650.	3566.
P 712 SEATTLE, WA			00.	00.	00.	00.				

P 713	PORTLAND, OR		225.	185.	191.	198.				
F 808	E MEDITERRAN	S	3.	00.	00.	00.	3.	0.65700	1971.	170.
F 818	JAPAN	S	222.	185.	191.	198.	796.	0.23300	185468.	45054.
P 714	SAN FRAN, CA		619.	00.	00.	00.				
F 824	CENT AMERICA	S	619.	00.	00.	00.	619.	0.37600	232744.	35035.
P 715	LONG BEA, CA		216.	210.	314.	241.				
F 807	E BLOCK EURO	S	113.	00.	00.	00.	113.	0.61300	69269.	6396.
F 823	WS AMERICA	S	103.	210.	314.	241.	868.	0.20000	173600.	49129.
P 716	SAN DIEG, CA		00.	00.	00.	00.				
P 717	TOLEDO, BC		00.	00.	00.	00.				
P 718	SAGINAW, BC		00.	00.	00.	00.				
P 719	CHICAGO, BC		00.	00.	00.	00.				
P 720	DULUTH, BC		00.	929.	421.	737.				
F 807	E BLOCK EURO	S	00.	926.	418.	734.	2078.	0.82600	1716428.	117615.
F 808	E MEDITERRAN	S	00.	3.	3.	3.	9.	0.82000	7380.	509.

		DEMAND BY TIME				TOTAL
		1	2	3	4	SHIPMENT
D 035	Lincoln NE	985.	985.	985.	985.	3940.
D 123	Madison WI	2393.	2393.	2393.	2393.	9572.
D 331	Pendlton OR	372.	372.	372.	372.	1488.
D 113	Prt Alan LA	185.	185.	185.	185.	740.
D 096	Rolla MO	829.	829.	829.	829.	3316.
D 381	Salt Lk C Ut	465.	465.	465.	465.	1860.
D 073	St Paul MN	4332.	4332.	4332.	4332.	17328.
D 291	Buffalo NY	750.	750.	750.	750.	3000.
D 151	Tiffin OH	825.	825.	825.	825.	3300.
F 801	SCANDINAVIA	00.	00.	00.	00.	00.
F 802	N.C. EUROPE	217.	337.	693.	597.	1844.
F 803	S.W. EUROPE	213.	557.	858.	1157.	2785.
F 804	ISLANDS	00.	00.	00.	00.	00.
F 805	ADRIATIC	307.	1453.	00.	60.	1820.
F 806	USSR	00.	00.	00.	00.	00.
F 807	E BLOCK EURO	2779.	926.	418.	734.	4857.
F 808	E MEDITERRAN	3.	3.	3.	3.	12.
F 809	N AFRICA	17145.	11253.	9748.	10538.	48684.
F 810	RED SEA	11.	62.	124.	31.	228.
F 811	E AFRICA	00.	00.	00.	00.	00.
F 812	W AFRICA	00.	00.	00.	00.	00.
F 813	PERSIAN GULF	00.	00.	00.	00.	00.
F 814	W ASIA	00.	00.	00.	00.	00.
F 815	SE ASIA	00.	00.	00.	00.	00.
F 816	TAIWAN	00.	00.	00.	00.	00.
F 817	KOREA	00.	00.	00.	00.	00.
F 818	JAPAN	222.	185.	191.	198.	796.
F 819	CHINA	00.	00.	00.	00.	00.
F 820	CANADA	00.	00.	00.	00.	00.
F 821	W MEXICO	00.	00.	00.	00.	00.
F 822	E MEXICO	00.	00.	00.	00.	00.
F 823	WS AMERICA	103.	210.	314.	241.	868.
F 824	CENT AMERICA	2293.	1320.	2412.	1720.	7745.
F 825	CARIBBEAN	27.	19.	26.	18.	90.

STORAGE COST	6098036.
TRUCK COST	328823.
RAIL COST	56626104.
BARGE COST	1302366.
SHIP COST	43035852.
HANDLING CST	22574224.

GRAIN SHIPPED FROM SURPLUS REGIONS	114273.
GRAIN SHIPPED TO DEFICIT REGIONS	44544.
GRAIN SHIPPED TO FOREIGN REGIONS	69729.

Validation of Durum Wheat Model

Port	FGIS Recorded Exports	Model Solution
	Million Bu.	
Mobile	0	0
New Orleans	4.6	4.6
Galveston	27 5	27.4
Corpus Christi	0	0
Brownsville	0	0
Charleston	0	0
Baltimore	0	0
Toledo	0	0
Saginaw	0	0
Chicago	0	0
Duluth	33.1	33.2
Seattle	0	0
Portland	0.8	0 8
California	1 6	1 6
San Francisco	-	0.6
Long Beach	-	1.0
San Diego	-	-
Baie Comeau	2.1	2.1
Toledo	-	-
Saginaw	-	-
Chicago	-	-
Duluth	-	2.1
Total	69 7	69 7