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

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

HARD RED SPRING WHEAT

SEPTEMBER 1990

IWR REPORT 90-R-4

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

HARD RED SPRING 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-4**

Content

INTRODUCTION

This report documents the hard red spring wheat model. Additional reports document corn, soybeans, hard red winter wheat, soft wheat, durum 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 ----  *                      HRS1.F                                  *
C ----  *        Documented on                 Sept  1990              *
C ----  *        Run on                        a 386 machine           *
C ----  *        Compiler used                 NDP Fortran             *
C ----  *        Grain used                    HARD RED SPRING (HRS)   *
C ----  *        Data files used               F1,F2,F3,F4, and F8.    *
C ----  *        Trace file                    U6                      *
C ----  *        Input file for HRS2.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.hrs.
C ----  F1.hrs contains information about the model.
C ----  U2 is the unit number for data file, F2.hrs.
C ----  F2.hrs contains information on TRUCK mileage.
C ----  U3 is the unit number for data file, F3.hrs.
C ----  F3.hrs contains information on RAIL costs.
C ----  U4 is the unit number for data file,F4.hrs.
C ----  F4.hrs contains information on BARGE costs.
C ----  U8 is the unit number for data file, F8.hrs.
C ----  F8.hrs contains information about SUPPLIES and DEMANDS.
C ----  U9 is for data files, F9.hrs.
C ----  F9.hrs 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, HRS2.


       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, HRS1.F:
```

```
C ----
C ----                    U1,U2,U3,U4,U8,U9 (input files)
C ----                           |
C ----                           |
C ----                           V
C ----                       HRS1.F  (program)
C ----                           |
C ----                          /   \
C ----                         /     \
C ----                        V       V
C ----                    U6 (trace)   U12 (input to HRS2.F)

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

      OPEN (UNIT = U1, FILE = 'F1.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U2, FILE = 'F2.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U3, FILE = 'F3.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U4, FILE = 'F4.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U8, FILE = 'F8.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U6, FILE = 'FILE06_2.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U12, FILE = 'FILE12_2.HRS', 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, HRS2.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(LD) and unloading(RI) factors.
C ----  T stands for Truck, R for Rail, S for Ship and B for Barge.

       READ  (U1,520) 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, NORE, NOPE, NDFR
      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)
      COMMDN /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. 0 ) 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.HRS).

      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, NDPE, 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 ( 2014 )
  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.

      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.HRS).

      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          = NDTF * ( 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.

      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.HRS).

      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.HRS).

      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, NOSR, 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, NDAY(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 ----  *                          HRS2.F                             *
C ----  *          DOCUMENTED ON                 : SEPT 1990         *
C ----  *          RUN ON                        : A 386 machine     *
C ----  *          COMPILER USED                 : NDP Fortran       *
C ----  *          GRAIN USED                    : HARD RED SPRING   *
C ----  *          DATA FILES USED               : U12               *
C ----  *          TRACE FILE                    : U6                *
C ----  *          INPUT FILE FOR HRS3.F         : 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/ 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
      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, HRS1.
C ----  This data file contains the relevant information which
C ----  is necessary for HRS2 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, HRS3.

      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.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U6, FILE = 'FILE06_2.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U12, FILE = 'FILE12_2.HRS', STATUS = 'UNKNOWN',
```

```
     1       ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')


      NTIM       = 0
      FSBL       = .TRUE.
      MAXA       = 34000

C ----  Read in the data from the output file(generated by) HRS1.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(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
      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=MIDL(J)
      MIR(II)=LL
      MIDL(J)=II-1
 20   CONTINUE
C     ******************************************************************
C
C     GO - SUPERKILTER
C
```

```
      ND=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=MINO(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=NODE(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=MINO(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 NODE'/(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  LDST 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 ----  *                        HRS3.F                            *
C ----  *       Documented on              SEPT  1990              *
C ----  *       Run on                     a 386 machine           *
C ----  *       Compiler used              NDP Fortran             *
C ----  *       Grain used                 HARD RED SPRING         *
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 HRS2.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 /GO/ 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.hrs.
C ----  F1.hrs contains information about the model.
C ----  U2 is the unit number for data file, F2.hrs.
C ----  F2.hrs contains information on TRUCK mileage.
C ----  U3 is the unit number for data file, F3.hrs.
C ----  F3.hrs contains information on RAIL costs.
C ----  U4 is the unit number for data file,F4.hrs.
C ----  F4.hrs contains information on BARGE costs.
C ----  U8 is the unit number for data file, F8.hrs.
C ----  F8.hrs contains information about SUPPLIES and DEMANDS.
C ----  U9 is for the unit number for data files, F9.hrs.
C ----  F9.hrs contains the NAMES of all the regions.
C ----  UF1 is the unit number for the input file produced by HRS2.F.
C ----  It is the output file from HRS2.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 ----                            HRS3.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.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U2, FILE = 'F2.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U3, FILE = 'F3.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U4, FILE = 'F4.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U8, FILE = 'F8.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = U9, FILE = 'F9.HRS', STATUS = 'UNKNOWN',
     1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
      OPEN (UNIT = UF1, FILE = 'FLOW1_2.HRS', 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.HRS with unit number
C ----  UF1. The processed data is the output of HRS2.F.
C ----  N is the total number of data processed in HRS2.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 210D I = 1, NOSR
      READ  (U9,53D) ( NAM1(I,J), J = 1, 3 )
 2100 CONTINUE
      DO 22DD I = 1, NODR
      READ  (U9,53D) ( NAM2(I,J), J = 1, 3 )
 22D0 CONTINUE
      DO 23DD I = 1, NORE
      READ  (U9,530) ( NAM3(I,J), J = 1, 3 )
 23D0 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 /GD/ 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, 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, 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 i,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, NOTF ),
     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 ),
```

```
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 entered.
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.D66374 + 0.1D4892 * COST)
      ELSE IF ( COST .GT. 225  .and.  COST .LE. 245 ) THEN
      COST       = ( 23.67 + D.D * COST )
      ELSE IF ( COST .GT. 245 ) THEN
      COST       = ( D.68D37 + D.D93976 * COST )
      ELSE IF ( COST .GE. 9999. ) THEN
      GOTO 26DD
      ENDIF

      COST       = ( COST * TRUCK) / 1DD.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 D
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 foltowing totals: TGSO(surplus), TOPR(port).

      TGSD(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 /GO/ 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 220D
      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. 0.0 ) 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 ( 1DF8.3 )
  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 /GO/ 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.HRS

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

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

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

  HARD RED SPRING WHEAT SHIPMENT BAIE COMEAU ADDED, 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 42 then change 41
C       to 42.

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 = 25
C               Number of deficit  regions = 34
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.
```

```
  25  34  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       These lines show the codes for the 25 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.

 022 017 071 015 024 351 391 013 014 323 322 342 321 324 123 012 325 021 072 331
 023 311 361 016 011                                                    0

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

C       This line shows the codes for the 34 deficit regions  in this model.
C       For example, 085 is the code number for AMES IA,
C                    211 is the code number for ATLANTA GA

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.

 085 211 707 098 154 251 092 144 131 281 051 065 201 132 142 183 222 715 035 073
 096 053 113 713 381 074 291 043 066 401 313 045 371 241                0

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

C       These 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                644 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 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
999999.9999999.9999999.9999999.9999999.9                                        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
   5.654   0.191   5.654   0.191   6.486   0.208   6.339   0.205   5.654   0.191

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 SD is Linked to St Paul MN (601).
C       e.g, Ashley ND (the 2nd code number) is linked to barge loading region,
C            St Paul MN (601).

 601 601 644 601 609 638 999 644 601 638 638 638 638 638 604 644 638 609 601 639
 609 640 999 601 644                                                          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
   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202   6.508   0.208
   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202   6.508   0.208
   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202   6.508   0.208
   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202   6.508   0.208
   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202   6.508   0.208
   5.866   0.195   5.654   0.191   6.200   0.202   6.200   0.202               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, Ames IA (085) is linked to
C                    Clinton IA   (605) and
C                    the 2nd deficit region, Atlanta GA (211) is linked to
C                    Chattanoo TN (627).

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.

 605 627 618 621 618 618 612 619 615 618 631 636 628 616 619 626 999 999 611 601
 607 631 642 643 999 601 645 613 636 999 639 613 999 999                      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                              F2.HRS

C       Stores all the truck mileages in this model.
C       F2.HRS is the 2nd file in this Hard Red Spring wheat 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 25 surplus regions and 34 deficit regions.

C       surplus region 1   ----------> deficit region 1
C                          ----------> deficit region 2
C                                      .
C                                      .
C                          ----------> deficit region 34


C       surplus region 2   ----------> deficit region 1
C                          ----------> deficit region 2
C                                      .
C                                      .
C                          ----------> deficit region 34
```

```
C       .
C       .
C       .

C       surplus region 25  ----------> deficit region 1
C                          ----------> deficit region 2
C                                      .
C                                      .
C                         ----------> deficit region 34

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 25  ----------> 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 25  ----------> 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 25 ----------> corresponding deficit region

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

C       This includes the truck mileage from each of the 25 surplus regions
C       to each of the 34 deficit regions.

C       A 25 by 34 matrix.
C       The first surplus region is Aberdeen SD (022).
C       The distances from Aberdeen to all deficit regions are given on the first
C       4 lines (see below):
C               Aberdeen SD (022) to Ames IA (085)     =  446 miles
C               Aberdeen SD (022) to Atlanta GA (211) = 1331 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 were not a surplus region, then the first 4 lines would
C       be removed. The file would start with 520.

 446.0001331.0001353.000 828.0001067.0001366.000 600.000 910.000 653.0001185.000
 716.000 967.0001260.000 639.000 866.0001212.0001617.0001649.000 406.000 279.000
 671.000 767.0001324.0001398.000 948.000 238.0001350.000 551.0001053.0001491.000
1219.000 608.0001304.0001367.000                                              0.000
 520.0001403.0001414.000 907.0001128.0001427.000 679.000 971.000 714.0001253.000
 787.0001038.0001332.000 711.000 927.0001273.0001689.0001639.000 485.000 351.000
 746.000 838.0001403.0001381.000 938.000 279.0001411.000 630.0001126.0001511.000
1202.000 679.0001294.0001428.000                                              0.000
 436.0001288.0001286.000 850.0001000.0001299.000 606.000 843.000 586.0001118.000
 897.0001147.0001227.000 596.000 799.0001145.0001574.0001952.000 532.000 272.000
 652.000 947.0001383.0001590.0001251.000 153.0001283.000 668.0001251.0001740.000
1411.000 788.0001539.0001300.000                                              0.000
 637.0001495.0001499.0001023.0001213.0001512.000 795.0001056.000 799.0001331.000
 884.0001135.0001434.000 803.0001012.0001358.0001781.0001617.000 592.000 460.000
```

855.000 935.0001514.0001265.000 916.000 364.0001496.000 746.0001223.0001506.000
1086.000 776.0001208.0001513.000 0.000
303.0001201.0001240.000 679.000 954.0001253.000 451.000 797.000 532.0001079.000
628.000 878.0001120.000 511.000 753.0001099.0001487.0001707.000 263.000 151.000
527.000 678.0001175.0001544.0001006.000 195.0001237.000 402.000 904.0001454.000
1365.000 519.0001362.0001254.000 0.000
714.0001574.0001697.0001028.0001410.0001709.000 801.0001236.000 983.0001529.000
797.0001024.0001482.000 934.0001192.0001504.0001859.0001104.000 577.000 709.000
883.000 861.0001449.0001074.000 403.000 773.0001694.000 732.0001145.0001059.000
985.000 731.000 759.0001710.000 0.000
684.0001398.0001621.000 835.0001307.0001597.000 637.0001099.000 943.0001486.000
556.000 757.0001306.000 857.0001058.0001328.0001666.0001059.000 486.000 765.000
719.000 609.0001195.0001238.000 504.000 856.0001651.000 535.000 878.000 835.000
1224.000 509.000 860.0001559.000 0.000
605.0001463.0001467.0001021.0001181.0001480.000 775.0001024.000 767.0001299.000
916.0001167.0001402.000 771.000 980.0001326.0001749.0001797.000 606.000 428.000
823.000 967.0001524.0001386.0001096.000 332.0001464.000 751.0001253.0001686.000
1207.000 808.0001357.0001481.000 0.000
733.0001591.0001595.0001112.0001309.0001608.000 884.0001152.000 895.0001427.000
947.0001206.0001530.000 899.0001108.0001454.0001877.0001529.000 675.000 556.000
951.0001018.0001597.0001166.000 828.000 460.0001592.000 835.0001306.0001424.000
987.000 859.0001109.0001609.000 0.000
979.0001841.0001845.0001354.0001559.0001858.0001126.0001402.0001145.0001677.000
1189.0001448.0001780.0001149.0001358.0001704.0002127.0001488.000 917.000 806.000
1201.0001260.0001839.000 986.000 775.000 710.0001842.0001077.0001548.0001539.000
807.0001101.0001016.0001859.000 0.000
1140.0002002.0002006.0001515.0001720.0002019.0001287.0001563.0001306.0001838.000
1350.0001605.0001941.0001310.0001519.0001865.0002288.0001392.0001078.000 967.000
1362.0001421.0002000.000 825.000 679.000 871.0002003.0001238.0001709.0001443.000
646.0001262.000 894.0002020.000 0.000
1466.0002360.0002423.0001827.0002137.0002436.0001599.0001980.0001722.0002510.000
1633.0001825.0002279.0001686.0001936.0002282.0002646.0001102.0001383.0001349.000
1681.0001688.0002274.000 337.000 614.0001297.0002420.0001544.0001953.0001341.000
221.0001564.000 515.0002437.000 0.000
1035.0001929.0001940.0001396.0001654.0001953.0001168.0001497.0001240.0001779.000
1208.0001435.0001848.0001243.0001453.0001799.0002215.0001284.000 952.000 883.000
1250.0001272.0001860.000 820.000 571.000 805.0001937.0001113.0001556.0001335.000
641.0001133.000 812.0001954.000 0.000
1025.0001919.0001996.0001386.0001710.0002009.0001158.0001547.0001281.0001828.000
1192.0001419.0001838.0001245.0001503.0001849.0002205.0001147.000 942.000 926.000
1240.0001256.0001844.000 772.000 434.000 888.0001993.0001103.0001540.0001198.000
593.0001123.000 675.0802010.000 0.000
270.000 812.000 808.000 525.000 522.000 821.000 404.000 365.000 105.000 640.000
782.000 992.000 751.000 193.000 321.000 667.0001098.0002012.000 475.000 288.000
389.000 828.000 984.0001950.0001348.000 335.000 805.000 540.000 963.0001668.000
1771.000 676.0001704.000 822.000 0.000
707.0001565.0001569.0001099.0001283.0001582.000 871.0001126.000 869.0001401.000
987.0001238.0001504.000 873.0001082.0001428.0001851.0001699.000 677.000 530.000
925.0001038.0001595.0001264.000 994.000 434.0001566.000 822.0001324.0001594.000
1085.000 879.0001235.0001583.000 0.000
821.0001715.0001743.0001182.0001457.0001756.000 954.0001300.0001043.0001575.000
1007.0001257.0001634.0001029.0001256.0001602.0002001.0001407.000 738.000 669.000
1036.0001077.0001657.0001008.000 694.000 628.0001740.000 899.0001366.0001350.000
829.000 919.000 935.0001757.000 0.000
546.0001431.0001453.000 921.0001167.0001466.000 693.0001010.000 753.0001285.000

```
 782.0001033.0001360.000 739.000 966.0001312.0001717.0001552.000 490.000 379.000
 769.000 833.0001412.0001298.000 851.000 338.0001450.000 644.0001121.0001441.000
1119.000 674.0001207.0001467.000                                              0.000
 339.0001222.0001244.000 752.000 958.0001257.000 507.000 801.000 544.0001076.000
 704.000 954.0001151.000 530.000 757.0001103.0001508.0001754.000 339.000 170.000
 565.000 754.0001248.0001507.0001053.000 129.0001241.000 475.000 977.0001532.000
1328.000 595.0001409.0001258.000                                              0.000
1577.0002390.0002556.0001843.0002261.0002551.0001617.0002060.0001846.0002392.000
1577.0001767.0002298.0001764.0002016.0002320.0002674.0001006.0001393.0001491.000
1699.0001630.0002216.000 258.000 556.0001439.0002557.0001540.0001895.0001283.000
 197.0001528.000 419.0002541.000                                              0.000
 593.0001487.0001564.000 954.0001278.0001577.000 726.0001115.000 849.0001396.000
 779.0001026.0001406.000 813.0001071.0001417.0001773.0001363.000 510.000 504.000
 808.000 846.0001429.0001204.000 662.000 538.0001561.000 671.0001138.0001229.000
1025.000 691.0001018.0001578.000                                              0.000
1446.0002340.0002403.0001807.0002117.0002416.0001579.0001960.0001702.0002235.000
1613.0001840.0002259.0001666.0001916.0002262.0002626.0001205.0001363.0001329.000
1661.0001677.0002265.000 348.000 712.0001277.0002400.0001524.0001961.0001451.000
 169.0001544.000 621.0002417.000                                              0.000
1777.0002442.0002743.0001958.0002448.0002738.0001804.0002247.0002036.0002579.000
1607.0001668.000 991.000 580.0002203.0002456.0002634.000 334.0001580.0001811.000
1886.0001603.0002117.000 630.000 699.0001875.0002744.0001706.0001796.000 834.000
 822.0001631.000 343.0002698.000                                              0.000
 502.0001360.0001364.000 916.0001078.0001377.000 672.000 921.000 664.0001196.000
 816.0001066.0001299.000 668.000 877.0001223.0001646.0001752.000 500.000 325.000
 720.000 866.0001412.0001396.0001051.000 229.0001361.000 639.0001144.0001617.000
1217.000 707.0001339.0001378.000                                              0.000
 836.0001694.0001698.0001228.0001412.0001711.0001000.0001255.000 998.0001530.000
1078.0001337.0001633.0001002.0001211.0001557.0001980.0001587.000 806.000 659.000
1054.0001149.0001724.0001135.000 875.000 563.0001695.000 951.0001437.0001523.000
 956.000 990.0001116.0001712.000                                              0.000

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

C       Explanation for the Second part

C       This links each surplus region (25) to its corresponding river
C       barge-loading locations. Conceptually, a (1 X25) 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 25  -------> corresponding river barge-loading location

C       e.g, 1st surplus region, Aberdeen SD (022) to its river barge region,
C            St. Paul MN (601) = 1706 miles
C       e.g, 2nd surplus region, Ashley ND (017) to its 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 were not a surplus region, then 1706 could be deleted.

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

1706.000 351.000 153.000 436.000 139.000 836.000 999.000 357.000 532.000 658.000
 497.000  25.000 492.000 444.000  97.000 479.000 680.000 381.000 181.000  65.000
 425.000 137.000 999.000 301.000 608.000                                   0.000


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

C       Explanation for the Third part

C        (3) Each surplus region(25) 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 25  ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                           .
C                                           .
C                          ----------> port region 20 for export

C       There are 25 surplus regions and 20 port locations thus a (25 by 20)
C       matrix.

C       e.g, 1st surplus region, Aberdeen SD (022) to 1st port region, Mobile
C            = 1394 miles
C       e.g, 1st surplus region, Aberdeen (022) to 2nd port region, New Orleans
C            = 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 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
1594.0001527.0001335.0001352.0001481.0001854.0001697.0001261.0001306.0001036.000
 912.0001094.0001074.0001155.0001104.0001153.0001261.0001306.0001036.000 912.000
1355.0001273.0001068.0001085.0001214.0001678.0001621.0001218.0001264.000 996.000
 994.0001307.0001238.0001235.0001089.0001108.0001218.0001264.000 996.000 994.000
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
 540.0001096.0001166.0001505.0001529.0001578.0001159.0001154.000 927.000 540.000
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
2393.0002352.0002101.0002079.0002197.0002632.0002423.0001987.0001991.0001755.000
1377.000 308.000 337.000 869.0001102.0001194.0001987.0001991.0001755.0001377.000
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
 945.0001006.0001183.0001351.0001486.0001017.000 808.000 372.000 417.000 140.000
 332.0001880.0001950.0002100.0002012.0002044.000 372.000 417.000 140.000 332.000
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
1748.0001735.0001568.0001589.0001718.0001952.0001743.0001307.0001327.0001075.000
 713.000 938.0001008.0001331.0001407.0001456.0001307.0001327.0001075.000 713.000
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
```

```
2376.0002294.0002043.0002021.0002139.0002670.0002556.0002124.0002133.0001897.000
1519.000 280.000 211.000 743.0001006.0001098.0002124.0002133.0001897.0001519.000
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
2373.0002343.0002151.0002168.0002295.0002612.0002403.0001967.0001971.0001735.000
1357.000 278.000 348.000 882.0001205.0001300.0001967.0001971.0001735.0001357.000
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
1465.0001476.0001364.0001451.0001586.0001573.0001364.000 928.000 923.000 696.000
 309.0001306.0001396.0001735.0001752.0001801.000 928.000 923.000 696.000 309.000
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 barge-unloading river location is linked to a
C           corresponding deficit region.
C       There are 34 deficit regions, thus a 1 by 34 matrix.

C       Diagrammatically,

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

C       barge-unloading location 2     ----------> corresponding deficit region
C       .
C       .
C       .
C       barge-unloading location 34    ----------> corresponding deficit region



C       e.g, barge-unloading river location, Clinton IA(605) to
C            deficit region Ames IA (085)    = 185 miles
C       e.g, barge-unloading river location, Chatanoo TN (627) to
C            deficit region Atlanta GA (211) = 114 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 Ames IA were not a deficit region, then 185 would be deleted.

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

 185.000 114.000 497.000 193.000 183.000 443.000  74.000  67.000  58.000 387.000
```

```
176.000 316.000  34.000  49.000 111.000   2.000 999.000 999.000  54.000  82.000
 66.000 141.000  80.000  47.000 999.000  73.000   1.000  65.000 198.000 999.000
132.000 197.000 999.000 999.000                                          0.000
```

```
C                              F3.hrs

C       This is the 3rd file for Hard Red Spring wheat and it deals with the
C       rail costs (cents/bushel).
C       Note: 999.999 indicates an infeasible 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             |                                      | (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 25 surplus regions and 34 deficit regions.
C       Each surplus region is linked to all deficit regions(34). Thus
C       resulting in a 25 by 34 matrix.

C       surplus region 1   ----------> deficit region 1
C                          ----------> deficit region 2
C                                            .
C                                            .
C                          ----------> deficit region 34


C       surplus region 2   ----------> deficit region 1
C                          ----------> deficit region 2
C                                            .
C                                            .
C                          ----------> deficit region 34

C       .
```

```
C       .
C       .

C       surplus region 25  ----------> deficit region 1
C                          ----------> deficit region 2
C                                      .
C                                      .
C                          ----------> deficit region 34


C       For example,
C        the rail cost from 1st surplus region Aberdeen SD (022) to
C        1st deficit region Ames IA (085)    = 41.400 cents/bushel
C        the rail cost from 1st surplus region Aberdeen SD (022) to
C        2nd deficit region Atlanta GA (211) = 89.400 cents/bushel.

  41.400  89.400 115.200  55.800  78.000 111.600  49.800  72.000  48.000  93.600
  63.000  75.000  94.200  48.600  66.000 100.800 118.800 999.999  29.400  31.200
  52.800  63.000  97.800 999.999 999.999  30.000 105.600  67.200  87.600 169.200
 999.999  64.800 999.999 117.600                                          0.000
  56.400 104.400 115.800  70.800 127.200 123.000  65.400  80.400  55.800  97.800
  78.600  90.600 109.800  57.600  72.000 115.800 133.800 135.000  44.400  44.400
  67.800  78.600 112.200 999.999 999.999  31.800 106.800  82.800 102.000 127.200
 103.800  79.800 999.999 133.200                                          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  32.400
 999.999 999.999 999.999 999.999 999.999  30.000 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                          0.000
  56.400 104.400 123.600  70.800  86.400 120.000  65.400  80.400  56.400 102.000
  78.000  90.000 109.200  57.000  74.400 115.800 133.800 124.800  44.400  39.000
  68.400  78.000 112.800 999.999  93.600  30.000 104.400  82.800 102.600 116.400
 999.999  79.800 999.999 124.200                                          0.000
  42.000  96.600 111.600  56.400  81.000 109.200  39.600  75.000  57.000  97.200
  69.600  84.600  90.600  55.800  62.400 100.200 118.200 148.200  36.600  17.400
  49.800  72.000 108.600 999.999 999.999  25.800  93.000  43.200  91.800 118.200
 106.800  61.800 999.999 115.800                                          0.000
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  64.800  72.000 999.999 999.999 999.999 999.999 999.999 102.000 999.999 999.999
  73.800  71.400 999.999 108.000  29.400 999.999 999.999  61.800  78.600  93.000
  73.800  61.200  74.400 999.999                                          0.000
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  57.000  58.200 999.999 999.999 999.999 999.999 999.999  95.400  37.200 999.999
  59.400  52.200 999.999 101.400  44.400 999.999 999.999  42.600  71.400  73.200
 106.800  42.000  67.200 999.999                                          0.000
  55.200 103.200 122.400  69.600 122.400 118.800  63.600  78.600  55.200 100.800
  76.800  88.800 108.000  55.800  73.200 114.600 132.600 133.200  43.200  37.800
  66.600  76.800 111.000 999.999  91.200  28.800 103.200  81.000 101.400 164.400
 999.999  78.600 999.999 131.400                                          0.000
  62.400 115.800 129.600  76.800  92.400 126.000  71.400  86.400  62.400 108.000
  84.000  96.000 120.600  63.000  80.400 121.800 139.800 118.800  50.400  45.000
  73.800  84.000 118.800 104.400  72.000  36.000 110.400  88.200 108.600 110.400
  88.200  85.800  91.800 138.600                                          0.000
  71.400 999.999 999.999 108.600 142.800 132.600  97.200 100.800  67.800 105.600
 114.600 128.400 999.999  66.000  90.000 120.000 999.999 110.400  82.800  57.000
  87.600 121.200 999.999  64.200  61.800  46.800 999.999  88.800 138.600 153.000
```

45.600 111.000 103.800 131.400 0.000
97.800 999.999 999.999 134.400 999.999 999.999 123.600 999.999 999.999 999.999
112.800 117.000 999.999 999.999 999.999 999.999 999.999 115.800 108.000 61.200
125.400 123.600 999.999 65.400 60.000 68.400 999.999 113.400 124.200 136.800
40.800 117.000 79.800 999.999 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 99.000 999.999 999.999
999.999 999.999 999.999 39.600 62.400 999.999 999.999 999.999 999.999 999.999
30.600 999.999 86.400 999.999 0.000
81.000 999.999 999.999 106.200 999.999 999.999 91.800 999.999 999.999 999.999
100.800 109.200 999.999 999.999 999.999 999.999 999.999 105.000 69.600 80.400
97.200 106.800 999.999 60.000 57.600 75.600 999.999 79.800 118.800 127.200
39.600 96.600 81.600 999.999 0.000
82.200 999.999 999.999 92.400 999.999 999.999 88.200 999.999 999.999 999.999
99.600 99.600 999.999 999.999 999.999 999.999 999.999 86.400 65.400 72.600
99.000 104.400 999.999 65.400 47.400 63.000 999.999 86.400 114.600 132.600
49.800 97.200 67.200 999.999 0.000
31.800 67.200 76.200 54.000 52.800 76.200 40.200 42.000 30.000 68.400
999.999 999.999 69.600 30.600 29.400 55.800 97.800 999.999 999.999 34.800
45.600 999.999 90.600 999.999 999.999 34.800 58.200 999.999 999.999 999.999
999.999 999.999 999.999 76.800 0.000
92.400 999.999 999.999 105.000 124.200 148.800 999.999 118.200 78.000 126.600
88.800 101.400 999.999 78.600 97.200 124.800 999.999 102.000 61.800 60.000
96.000 88.800 999.999 72.600 54.600 54.000 999.999 90.600 114.600 114.600
64.200 86.400 102.000 148.800 0.000
58.800 106.800 126.000 73.200 88.800 123.000 67.200 82.800 58.800 104.400
80.400 92.400 111.600 59.400 76.800 118.200 135.600 124.200 46.800 41.400
65.400 80.400 115.200 109.200 79.200 32.400 106.800 84.600 105.000 157.800
77.400 82.200 999.999 135.000 0.000
47.400 94.800 121.200 61.800 83.400 117.600 55.800 77.400 54.000 99.600
63.000 81.000 100.200 54.600 72.000 106.200 124.200 128.400 35.400 36.600
58.200 63.000 103.200 116.400 79.200 36.000 111.000 73.200 93.600 174.600
111.600 70.200 999.999 122.400 0.000
45.600 100.800 114.600 75.600 85.200 103.200 54.000 75.000 44.400 100.800
73.200 87.600 100.800 46.800 62.400 89.400 129.000 999.999 39.000 24.600
58.200 75.600 114.000 999.999 999.999 28.200 96.600 46.200 95.400 999.999
999.999 64.800 999.999 105.000 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 96.000 999.999 999.999
999.999 999.999 999.999 23.400 50.400 999.999 999.999 999.999 999.999 122.400
33.600 999.999 70.200 999.999 0.000
58.800 106.800 145.200 73.200 999.999 999.999 67.800 999.999 79.200 999.999
81.000 93.000 999.999 79.800 96.600 999.999 999.999 115.200 47.400 61.200
70.200 81.000 114.600 121.800 67.800 60.600 999.999 85.200 104.400 110.400
999.999 82.200 135.600 999.999 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 94.200 999.999 999.999
999.999 999.999 999.999 35.400 64.200 999.999 999.999 999.999 999.999 118.800
999.999 999.999 85.800 999.999 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 40.800 999.999 999.999
999.999 999.999 999.999 49.200 50.400 999.999 999.999 999.999 999.999 70.200
75.600 999.999 34.200 999.999 0.000
48.600 96.000 115.800 63.000 78.000 112.200 57.000 72.000 48.600 94.200
70.200 75.000 101.400 49.200 66.600 107.400 125.400 133.200 36.600 36.000

```
  59.400  69.600 104.400 118.800  85.800  21.600  96.600  74.400  94.800 175.800
 105.600  71.400 999.999 124.800                                          0.000
  66.000 116.400 133.200  80.400  95.400 129.600  74.400  89.400  66.000 111.600
  87.600  99.600 118.800  66.600  84.000 125.400 143.400 116.400  54.000  48.600
  72.000  87.600 121.800 101.400  71.400  39.600 114.000  91.800 112.200 150.000
  69.600  89.400 999.999 142.200                                          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 25  ----------> river barge-loading  region 1
C                          ----------> river barge-loading  region 2
C                                            .
C                                            .
C                         ----------> river barge-loading  region 43

C       A 25 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 (022)
C           to the 2nd river port region, Duluth RV (602) = 19.200 cents/bushel

  22.800  19.200 999.999 999.999 999.999 999.999 999.999 999.999  41.400 999.999
 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                                                  0.000
  37.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 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
 999.999 999.999 999.999                                                  0.000
```

```
 27.000  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
 33.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
999.999  31.200 999.999 999.999 999.999 999.999 999.999 999.999  21.600 999.999
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                                                   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  76.200 999.999
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
 30.700 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                   0.000
 40.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
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
999.999 999.999 999.999                                                   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  48.000 999.999
999.999 999.999 999.999                                                   0.000
999.999  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  38.400 999.999
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  12.600 999.999
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  45.600 999.999
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 47.400 999.999
999.999 999.999 999.999 0.000
999.999 999.999 999.999 17.400 999.999 999.999 999.999 999.999 999.999 999.999
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 0.000
36.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
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
999.999 999.999 999.999 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 63.600 999.999
999.999 999.999 999.999 0.000
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 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 0.000
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
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
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
12.600 999.999 999.999 0.000
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 34.200 999.999
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 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
24.000 999.999 999.999 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
27.000 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
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 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 25  ----------> 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       25 by 20 matrix.

C       For example,
C           from the 1st surplus region, Alexandria (111) to New Orleans (702)
C           = 94.200 cents/bushel

 999.999  94.200  87.600 999.999 999.999 999.999 120.000 999.999 999.999  61.200
  34.200 108.000 119.400 999.999 999.999 999.999 999.999 999.999  61.200  34.200
 999.999 112.800 108.600 999.999 999.999 999.999 999.999 999.999 999.999  54.600
  31.800 129.000 139.800 999.999 999.999 999.999 999.999 999.999  54.600  31.800
 999.999 107.400 106.800 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  19.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999  19.200
 999.999 105.600 103.200 999.999 999.999 999.999 999.999 999.999 999.999  54.600
  42.000  96.000 106.800 999.999 999.999 999.999 999.999 999.999  54.600  42.000
 999.999  91.200  84.600 999.999 999.999 999.999  97.200 999.999 999.999  49.800
  38.400 121.200 132.000 999.999 999.999 999.999 999.999 999.999  49.800 999.999
 999.999 999.999  96.000 118.800 999.999 999.999 999.999 999.999 999.999 999.999
 999.999  89.400 105.000  98.400  82.200  89.400 999.999 999.999 999.999 999.999
```

```
 999.999 999.999  54.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 104.400  93.000  97.800  96.600  96.600 999.999 999.999 999.999 999.999
 999.999 107.400 102.000 999.999 999.999 999.999 999.999 999.999 999.999  52.800
  60.000  94.700 105.600 999.999 999.999 999.999 999.999 999.999  52.800  60.000
 999.999 111.600 109.200 999.999 999.999 999.999 999.999 999.999 999.999  60.600
  48.000  89.400 100.800 999.999 999.999 999.999 999.999 999.999  60.600  48.000
 999.999 999.999 138.000 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  53.400  60.800  59.200 114.600 119.400 125.400 999.999 999.999 999.999 999.999
 999.999 999.999 138.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  58.200  55.800  53.400  95.400 108.600 120.600 999.999 999.999 999.999  58.200
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999  42.000  31.200  68.400  95.400 107.400 999.999 999.999 999.999 999.999
 999.999 999.999 121.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  79.200  57.800  56.200  62.400 110.400 118.200 999.999 999.999 999.999  79.200
 999.999 999.999 149.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  69.000  45.000  64.600  87.000  92.400 104.400 999.999 999.999 999.999  69.000
  75.000  75.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999  16.200
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999  16.200 999.999
 999.999 118.800 111.600 999.999 999.999 999.999 999.999 999.999 999.999  64.200
  38.400  87.000  98.400 124.200 127.800 999.999 999.999 999.999  64.200  38.400
 999.999 999.999 129.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  78.600  72.200  71.200  99.000  98.400 106.200 999.999 999.999 999.999  78.600
 999.999 100.800  94.200 999.999 999.999 999.999 999.999 999.999 999.999  61.800
  40.200 102.000 113.400 999.999 999.999 999.999 999.999 999.999  61.800  40.200
 999.999  92.400  92.400 999.999 999.999 999.999 100.800 999.999 999.999 115.800
 127.200 144.600 144.000 999.999 999.999 999.999 999.999 999.999 115.800 127.200
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999  31.200  20.400  66.000  96.000 999.999 999.999 999.999 999.999 999.999
 999.999 108.600 101.400 999.999 999.999 999.999 999.999 999.999 999.999  70.200
  57.000 112.200 123.600 999.999 999.999 999.999 999.999 999.999  70.200  57.000
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999  30.400  26.700 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999  19.200  42.000 999.999 999.999 999.999 999.999 999.999
 999.999 100.800 100.800 999.999 999.999 999.999 999.999 999.999 999.999  54.000
  27.600 104.400 115.200 999.999 999.999 999.999 999.999 999.999  54.000  27.600
 999.999 131.400 126.000 999.999 999.999 999.999 999.999 999.999 999.999  78.000
  45.600  65.000  64.000 130.800 123.600 129.600 999.999 999.999  78.000  45.600

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

C       Fourth part

C       (4) Each barge-unloading river location is linked to 34
C           deficit regions.

C       barge-unloading river location 1   ----------> deficit region 1
C                                          ----------> deficit region 2
C                                            .
C                                            .
C                                          ----------> deficit region 34


C       barge-unloading river location 2   ----------> deficit region 1
C                                          ----------> deficit region 2
```

```
C        .·                                            .
C                                                      .
C                                                     -----------> deficit region 34

C        .
C        .
C        .

C        barge-unloading river location 43   -----------> deficit region 1
C                                             -----------> deficit region 2
C                                                      .
C                                                      .
C                                             -----------> deficit region 34


C        The rail costs from each river port to the 34 deficit regions.
C        There are 43 river ports. Thus a 43 by 34 matrix.

C        For example,
C                from the 1st barge-unloading river location,
C                St. Paul, MN(601) to the 1st deficit region, Ames IA (085)
C                = 999.999 cents/bushel

C        For example,
C                from the 24th barge-unloading river location,
C                Memphis TN (624) to the 27th deficit region, Topeka KS (043)
C                = 25.200 cents/bushel

 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                               0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                               0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                               0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                               0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                               0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999                                               0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
```

999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999

999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	25.200	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	999.999
999.999	999.999	999.999	999.999	999.999	21.600	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						0
999.999	14.400	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						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	8.400	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						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999

999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	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
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	18.000	999.999	999.999	999.999
999.999	999.999	999.999	999.999						0

```
C                                F4.hrs

C       This is the 4th data file for Hard Red Spring Wheat dealing with the
C        barge costs and 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 (702) = 999.999 $/bushel
C                    St. Paul, MN (601) to Knoxville (701) =   0.327 $/bushel
C                    St. Paul, MN (601) to Chatanoo  (703) =   0.299 $/bushel
C                    St. Paul, MN (601) to Guntersv  (710) =   0.284 $/bushel
C                    St. Paul, MN (601) to Florence  (713) =   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
 999.000 999.000 999.000 999.000 999.000                                    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 (St. Paul)
C                 St. Paul MN (601) to New Orle, LA (702) = 0.181 $/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   $/bushel

   0.181   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.

 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 Scandinavia (801) = 0.475 $/bushel
C               Mobile, AL (701) to N. C. Europe(802) = 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    0.449    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.hrs

C       This is the 5th data file for Hard Red Spring Wheat which deals with
C        supplies and demands.

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 41.645 million/bushel.

  41.645  25.590  58.640  22.886  11.860   0.344   0.675  57.883  27.314  25.650
  27.654  16.645   2.115   0.628   0.385   8.884   3.144   3.489  27.032   0.353
   0.764   4.123   9.773  34.274  10.344                                       0

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

C       These lines show the amount of corn demanded by the 34 deficit regions.
C       For example, the 1st deficit region,
C                    Ames IA (085) requires  6.663 million/bushel.

   6.663   0.680   3.938   0.334   3.616   2.598   8.072   2.701   4.040  10.158
   0.735   0.583   0.787   6.769   1.190   5.658   9.404  36.171   7.896  37.408
   3.483   0.740   2.132   1.954   6.008  15.910  48.112   1.772   0.455   1.678
   0.879   0.665   1.337   8.950                                               0

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

C       The remaining lines show the amount demanded by 25 foreign regions
C       (demand per quarter).
C       For example, the 1st foreign region, Scandivania (801)
C       in the 1st quarter requires 0.000 million bushel
C       in the 2nd quarter requires 0.000 million bushel
C       in the 3rd quarter requires 0.000 million bushel
C       in the 4th quarter requires 0.496 million bushel

   0.000   0.000   0.000   0.496                                               0
   1.060   1.079   1.874   1.388                                               0
   0.477   2.355   3.630   5.318                                               0
   0.000   0.000   0.000   0.000                                               0
   0.000   0.000   0.000   0.000                                               0
   0.000   0.237   0.992   0.243                                               0
   1.994   0.645   0.190   0.667                                               0
   0.000   0.000   0.000   0.000                                               0
   1.896   1.544   1.354   1.430                                               0
   0.047   0.269   0.543   0.137                                               0
   0.003   0.018   0.151   0.149                                               0
   1.017   2.465   1.425   1.391                                               0
   0.000   0.000   0.000   0.000                                               0
   0.294   0.294   0.294   0.294                                               0
  10.016  10.016  10.016  10.016                                               0
   2.925   3.289   3.440   3.462                                               0
   2.010   2.557   2.928   2.405                                               0
   9.779   8.136   8.064   8.730                                               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
0.236	0.481	1.177	1.010	0
5.982	4.737	8.882	6.240	0
4.333	3.083	4.190	2.806	0

C F9.hrs

C This is the last and 6th data file with the location names and code numbers
C of all the regions. The grain used in this model is Hard Red Spring Wheat.

C----------

C 25 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
	351	Casper	WY	0
	391	Denver	CO	0
	013	Devils La	ND	0
	014	Dickinson	ND	0
	323	Glasgow	MT	0
	322	Havre	MT	0
	342	Lewiston	ID	0
	321	Lewistown	MT	0
	324	Livingston	MT	0
	123	Madison	WI	0
	012	Minot	ND	0
	325	Miles Cit	MT	0
	021	Mobridge	SD	0
	072	Ortonville	MN	0
	331	Pendleton	OR	0
	023	Rapid City	SD	0
	311	Spokane	WA	0
	361	Stockton	CA	0
	016	Valley Cit	ND	0
	011	Williston	ND	0

C----------

C 34 Deficit regions

C	Code number	Location name	State	
	085	Ames	IA	0
	211	Atlanta	GA	0
	707	Baltimore	MD	0
	098	Cabool	MO	0
	154	Cambridge	OH	0
	251	Charltv	VA	0
	092	Chillicoth	MO	0
	144	Columbus	IN	0
	131	De Kalb	IL	0
	281	Du Bois	PA	0
	051	Enid	OK	0
	065	Ft Worth	TX	0
	201	Gadsden	AL	0

132	Galesburg	IL	0
142	Indy	IN	0
183	Knoxville	TN	0
222	Lake city	FL	0
715	Los Ang	CA	0
035	Lincoln	NE	0
073	Mankato	MN	0
096	Moberley	MO	0
053	Ok City	OK	0
113	Prt Allen	LA	0
713	Salem	OR	0
381	Salt Lake	UT	0
074	St Cloud	MN	0
291	Buffalo	NY	0
043	Topeka	KS	0
06G	Tyler	TX	0
401	Tucson	AZ	0
313	Wenatch	Wa	0
045	Wichita	KS	0
371	Winnemu	NV	0
241	Winston	NC	0

C----------

C 43 River regions

C Code number	Location name	State	
601	ST. PAUL,	MN	0
644	DULUTH,	RV	0
603	MCGREGOR,	IA	0
604	DUBUQUE,	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
645	BUFFALO,	NY	0
626	KNOXVILL,	TN	0
627	CHATTANOO,	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 4342807552.

HARD RED SPRING WHEAT SHIPMENT BAIE COMEAU ADDED, FULLER, GRAN

UNIT = 1 (THOUSAND BUSHEL)

SUPPLY 422094.

DEMAND 422028.

648 649 649 11828

1 Network Generator For Grain Shipment Problem

HARD RED SPRING WHEAT SHIPMENT BAIE COMEAU ADDED, FULLER, GRAN

1

ORIGIN/DESTN		MODE	SUPPLY	SHIPMENT BY TIME 1	2	3	4	TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
S 022 Aberdeen	SD		41645.								
D 211 Atlanta	GA	R		170.	170.	170.	170.	680.	0.89400	607920.	116566.
D 098 Cabool	MO	R		83.	83.	83.	83.	332.	0.55800	185256.	56911.
D 092 Chillicoth	MO	R		00.	00.	940.	2017.	2957.	0.49800	1472586.	506889.
D 201 Gadsen	AL	R		196.	00.	00.	00.	196.	0.94200	184632.	33598.
D 222 Lake city	FL	R		2351.	2351.	2351.	2351.	9404.	1.18800	11171953.	1612034.
D 035 Lincoln	NE	R		1974.	1974.	1974.	1974.	7896.	0.29400	2321424.	1353532.
D 096 Moberley	MO	R		608.	181.	870.	870.	2529.	0.52800	1335312.	433521.
D 113 Prt Allen	LA	R		00.	00.	533.	533.	1066.	0.97800	1042548.	182734.
D 066 Tyler	TX	R		113.	113.	113.	113.	452.	0.87600	395952.	77482.
P 703 GALVESTN,	TX	R		16133.	00.	00.	00.	16133.	0.87600	14132508.	2331219.
S 017 Ashley	ND		25590.								
D 707 Baltimore	MD	R		984.	984.	984.	984.	3936.	1.15800	4557888.	674709.
D 281 Dubois	PA	R		2539.	2539.	2539.	2539.	10156.	0.97800	9932568.	1740942.
D 073 Mankato	MN	T		55.	514.	8527.	1738.	10834.	0.33666	3647369.	2052176.
D 045 Wichita	KS	T		166.	166.	166.	166.	664.	0.64490	428214.	125775.
S 071 Bemidgi	MN		58640.								
R 644 DULUTH,	RV	T		00.	12028.	12028.	12028.	36084.	0.16115	5814883.	6088093.
P 711 DULUTH,	MN	T		00.	5706.	7418.	7930.	21054.	0.16115	3392821.	3618130.
P 720 DULUTH,	BC	T		00.	645.	190.	667.	1502.	0.16115	242045.	258119.
S 015 Bismark	ND		22886.								
D 251 Charlty	VA	R		00.	600.	00.	600.	1200.	1.20000	1440000.	205704.
D 131 Dekalb	IL	R		00.	1010.	339.	1010.	2359.	0.56400	1330476.	404380.
D 132 Galesburg	IL	R		00.	1692.	1692.	1692.	5076.	0.57000	2893320.	870128.
D 142 Indy	IN	R		00.	297.	297.	297.	891.	0.74400	662904.	152735.
D 073 Mankato	MN	R		00.	858.	825.	571.	2254.	0.39000	879060.	386381.
D 291 Buffalo	NY	R		2158.	00.	00.	00.	2158.	1.04400	2252952.	369924.
D 241 Winston	NC	R		2237.	2237.	2237.	2237.	8948.	1.24200	11113416.	1533866.
S 024 Brookings	SD		11860.								
D 085 Ames	IA	T		1665.	1059.	588.	1665.	4977.	0.29155	1451049.	942743.
D 092 Chillicoth	MO	R		2017.	2017.	1077.	00.	5111.	0.39600	2023956.	876128.
D 043 Topeka	KS	T		443.	443.	443.	443.	1772.	0.38459	681489.	335652.
S 351 Casper	WY		344.								

D	381	Salt Lake	UT	R		344.	00.	00.	00.	344.	0.29400	101136.	58968.
S	391	Denver	CO		675.								
P	703	GALVESTN,	TX	R		675.	00.	00.	00.	675.	0.54600	368550.	97538.
S	013	Devils La	ND		57883.								
D	144	Columbus	IN	R		00.	675.	00.	675.	1350.	0.78600	1061100.	231417.
D	142	Indy	IN	R		297.	00.	00.	00.	297.	0.73200	217404.	50912.
D	291	Buffalo	NY	R		1819.	00.	00.	00.	1819.	1.03200	1877208.	311813.
R	601	ST. PAUL,	MN	R		00.	15755.	22144.	16518.	54417.	0.30700	16706020.	9080021.
S	014	Dickinson	ND		27314.								
D	715	Los Ang	CA	R		00.	8311.	9042.	6883.	24236.	1.18800	28792370.	4154535.
D	401	Tucson	AZ	R		419.	419.	419.	419.	1676.	1.10400	1850304.	287300.
D	371	Winnemu	NV	R		334.	334.	334.	334.	1336.	0.91800	1226448.	229017.
S	323	Glasgow	MT		25650.								
P	713	PORTLAND,	OR	R		12704.	7970.	1012.	3964.	25650.	0.59200	15184800.	3706425.
S	322	Havre	MT		27654.								
P	713	PORTLAND,	OR	R		00.	7294.	10016.	10344.	27654.	0.53400	14767237.	3996003.
S	342	Lewiston	ID		16645.								
D	713	Salem	OR	T		127.	00.	00.	00.	127.	0.32350	41085.	24056.
P	713	PORTLAND,	OR	R		9421.	856.	3298.	2943.	16518.	0.31200	5153616.	2386851.
S	321	Lewistown	MT		2115.								
D	313	Wenatch	Wa	R		219.	219.	219.	219.	876.	0.39600	346896.	150164.
P	713	PORTLAND,	OR	R		00.	00.	00.	1239.	1239.	0.56200	696318.	179036.
S	324	Livingston	MT		628.								
P	712	SEATTLE,	WA	R		236.	00.	392.	00.	628.	0.45000	282600.	90746.
S	123	Madison	WI		385.								
D	183	Knoxville	TN	R		385.	00.	00.	00.	385.	0.55800	214830.	65997.
S	012	Minot	ND		8884.								
D	715	Los Ang	CA	R		00.	00.	00.	2159.	2159.	1.02000	2202180.	370096.
D	713	Salem	OR	R		361.	488.	488.	488.	1825.	0.72600	1324950.	312842.
D	381	Salt Lake	UT	R		1158.	1025.	1336.	1381.	4900.	0.54600	2675400.	839958.
S	325	Miles Cit	MT		3144.								
P	713	PORTLAND,	OR	R		00.	00.	3144.	00.	3144.	0.71200	2238528.	454308.

S 021 Mobridge	SD		3489.								
D 051 Enid	OK	R		183.	183.	183.	183.	732.	0.63000	461160.	125479.
D 096 Moberley	MO	R		262.	689.	00.	00.	951.	0.58200	553482.	163020.
D 053 Ok City	OK	R		185.	185.	185.	185.	740.	0.63000	466200.	126851.
D 113 Prt Allen	LA	R		533.	533.	00.	00.	1066.	1.03200	1100112.	182734.
S 072 Ortonville	MN		27032.								
D 085 Ames	IA	T		00.	606.	1077.	00.	1683.	0.32538	547619.	318794.
D 183 Knoxville	TN	R		1029.	00.	00.	00.	1029.	0.89400	919926.	176391.
D 073 Mankato	MN	T		9297.	7980.	00.	7043.	24320.	0.17898	4352797.	4606695.
S 331 Pendleton	OR		353.								
P 713 PORTLAND,	OR	R		00.	00.	00.	353.	353.	0.20400	72012.	51009.
S 023 Rapid City	SD		764.								
D 381 Salt Lake	UT	T		00.	477.	166.	121.	764.	0.62892	480499.	144717.
S 311 Spokane	WA		4123.								
P 713 PORTLAND,	OR	R		1555.	2439.	129.	00.	4123.	0.26700	1100841.	595774.
S 361 Stockton	CA		9773.								
D 715 Los Ang	CA	T		9042.	731.	00.	00.	9773.	0.32068	3134040.	1851202.
S 016 Valley Cit	ND		34274.								
D 154 Cambrid	OH	R		904.	904.	904.	904.	3616.	0.78000	2820480.	619855.
D 251 Charltv	VA	R		649.	49.	649.	49.	1396.	1.12200	1566312.	239302.
D 144 Columbus	IN	R		675.	00.	675.	00.	1350.	0.72000	972000.	231417.
D 131 Dekalb	IL	R		1010.	00.	671.	00.	1681.	0.48600	816966.	288157.
D 065 Ft Worth	TX	R		145.	145.	145.	145.	580.	0.75000	435000.	99424.
D 132 Galesburg	IL	R		1692.	00.	00.	00.	1692.	0.49200	832464.	290043.
D 074 St Cloud	MN	R		3977.	3977.	3977.	3977.	15908.	0.21600	3436128.	2726950.
D 291 Buffalo	NY	R		8051.	00.	00.	00.	8051.	0.96600	7777266.	1380102.
S 011 Williston	ND		10344.								
P 713 PORTLAND,	OR	R		1344.	2149.	3017.	3834.	10344.	0.64000	6620160.	1494708.

1

SURPLUS REGION		SUPPLY		STORAGE		STORAGE COST
S 022 Aberdeen	SD	41645.	20017.	15145.	8111.	1664317.
S 017 Ashley	ND	25590.	21846.	17643.	5427.	1725612.
S 071 Bemidgi	MN	58640.	58640.	40261.	20625.	4595947.
S 015 Bismark	ND	22886.	18491.	11797.	6407.	1411023.
S 024 Brookings	SD	11860.	7735.	4216.	2108.	540387.
S 351 Casper	WY	344.	00.	00.	00.	00.
S 391 Denver	CO	675.	00.	00.	00.	00.
S 013 Devils La	ND	57883.	55767.	39337.	17193.	4316597.

S 014 Dickinson ND	27314.	26495.	17431.	7636.	1981834.
S 323 Glasgow MT	25650.	12946.	4976.	3964.	841667.
S 322 Havre MT	27654.	27654.	20360.	10344.	2244123.
S 342 Lewiston ID	16645.	7097.	6241.	2943.	626113.
S 321 Lewistown MT	2115.	1896.	1677.	1458.	193823.
S 324 LivingstonMT	628.	392.	392.	00.	30060.
S 123 Madison WI	385.	00.	00.	00.	00.
S 012 Minot ND	8884.	7365.	5852.	4028.	663762.
S 325 Miles Cit MT	3144.	3144.	3144.	00.	241095.
S 021 Mobridge SD	3489.	2326.	736.	368.	131747.
S 072 OrtonvilleMN	27032.	16706.	8120.	7043.	1226387.
S 331 Pendleton OR	353.	353.	353.	353.	40828.
S 023 Rapid CitySD	764.	764.	287.	121.	45014.
S 311 Spokane WA	4123.	2568.	129.	00.	103409.
S 361 Stockton CA	9773.	731.	00.	00.	28028.
S 016 Valley CitND	34274.	17171.	12096.	5075.	1319960.
S 011 Williston ND	10344.	9000.	6851.	3834.	757194.

1

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.	15755.	22144.	16518.				
R 626 KNOXVILL, TN	B		00.	1414.	1414.	1414.	4242.	0.32700	1387134.	798938.
R 628 GUNTERSV, AL	B		00.	196.	196.	196.	588.	0.28400	166992.	110744.
P 702 NEW ORLE, LA	B		00.	14145.	20534.	14908.	49587.	0.18100	8975247.	7862019.
R 644 DULUTH,	RV		00.	12028.	12028.	12028.				
R 645 BUFFALO, NY	B		00.	12028.	12028.	12028.	36084.	0.27000	9742680.	6796061.
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.	12028.	12028.	12028.				
D 291 Buffalo NY	T	00.	12028.	12028.	12028.	36084.	0.00171	61800.	6300267.
R 626 KNOXVILL, TN		00.	1414.	1414.	1414.				
D 183 Knoxville TN	T	00.	1414.	1414.	1414.	4242.	0.00276	11715.	740653.
R 627 CHATANOO, TN		00.	00.	00.	00.				

R 628 GUNTERSV, AL		00.	196.	196.	196.				
D 201 Gadsen AL T		00.	196.	196.	196.	588.	0.03633	21360.	102665.
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.				

1

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME				TOTAL	UNIT	TOTAL	HANDLING
			1	2	3	4	SHIPMENT	COST	HAULING COST	COSTS

P 701 MOBILE, AL		00.	00.	00.	00.				
P 702 NEW ORLE, LA		00.	14145.	20534.	14908.				
F 801 SCANDINAVIA	S	00.	00.	00.	496.	496.	0.47500	235600.	28074.
F 803 S.W. EUROPE	S	00.	2355.	3630.	3012.	8997.	0.32000	2879040.	509230.
F 806 USSR	S	00.	237.	992.	243.	1472.	0.53800	791936.	83315.
F 810 RED SEA	S	00.	269.	543.	137.	949.	0.73400	696566.	53713.
F 811 E AFRICA	S	00.	18.	151.	149.	318.	1.02400	325632.	17999.
F 812 W AFRICA	S	00.	2465.	1425.	1391.	5281.	0.41800	2207458.	298905.
F 814 W ASIA	S	00.	294.	294.	294.	882.	0.51900	457758.	49921.
F 816 TAIWAN	S	00.	3289.	3440.	1936.	8665.	0.46100	3994565.	490439.
F 823 WS AMERICA	S	00.	481.	1177.	1010.	2668.	0.30600	816408.	151009.
F 824 CENT AMERICA	S	00.	4737.	8882.	6240.	19859.	0.16100	3197299.	1124019.
P 703 GALVESTN, TX		16808.	00.	00.	00.				
F 802 N.C. EUROPE	S	1060.	00.	00.	00.	1060.	0.38700	410220.	59996.
F 803 S.W. EUROPE	S	477.	00.	00.	00.	477.	0.33200	158364.	26998.
F 807 E BLOCK EURO	S	1994.	00.	00.	00.	1994.	0.49900	995006.	112860.
F 809 N AFRICA	S	1896.	00.	00.	00.	1896.	0.49400	936624.	107314.
F 810 RED SEA	S	47.	00.	00.	00.	47.	0.75100	35297.	2660.
F 811 E AFRICA	S	3.	00.	00.	00.	3.	1.04500	3135.	170.
F 812 W AFRICA	S	1016.	00.	00.	00.	1016.	0.43500	441960.	57506.
F 824 CENT AMERICA	S	5982.	00.	00.	00.	5982.	0.17700	1058814.	338581.
F 825 CARIBBEAN	S	4333.	00.	00.	00.	4333.	0.22900	992257.	245248.
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.	5706.	7418.	7930.				
F 802 N.C. EUROPE	S	00.	1079.	1874.	1388.	4341.	0.71600	3108156.	245701.
F 803 S.W. EUROPE	S	00.	00.	00.	2306.	2306.	0.69000	1591140.	130520.
F 809 N AFRICA	S	00.	1544.	1354.	1430.	4328.	0.83800	3626864.	244965.

F 825 CARIBBEAN	S	00.	3083.	4190.	2806.	10079.	0.55000	5543450.	570471.
P 712 SEATTLE, WA		236.	00.	392.	00.				
F 818 JAPAN	S	00.	00.	392.	00.	392.	0.22800	89376.	22187.
F 823 WS AMERICA	S	236.	00.	00.	00.	236.	0.25700	60652.	13358.
P 713 PORTLAND, OR		25024.	20708.	20616.	22677.				
F 814 W ASIA	S	294.	00.	00.	00.	294.	0.54100	159054.	16640.
F 815 SE ASIA	S	10016.	10016.	10016.	10016.	40064.	0.38000	15224320.	2267622.
F 816 TAIWAN	S	2925.	00.	00.	1526.	4451.	0.29600	1317496.	251927.
F 817 KOREA	S	2010.	2557.	2928.	2405.	9900.	0.43000	4257000.	560340.
F 818 JAPAN	S	9779.	8135.	7672.	8730.	34316.	0.23300	7995628.	1942286.
P 714 SAN FRAN, CA		00.	00.	00.	00.				
P 715 LONG BEA, CA		00.	00.	00.	00.				
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.	645.	190.	667.				
F 807 E BLOCK EURO	S	00.	645.	190.	667.	1502.	0.82600	1240652.	85013.

1

	DEMAND BY TIME				TOTAL
	1	2	3	4	SHIPMENT
D 085 Ames IA	1665.	1665.	1665.	1665.	6660.
D 211 Atlanta GA	170.	170.	170.	170.	680.
D 707 Baltimore MD	984.	984.	984.	984.	3936.
D 098 Cabool MO	83.	83.	83.	83.	332.
D 154 Cambrid OH	904.	904.	904.	904.	3616.
D 251 Charlty VA	649.	649.	649.	649.	2596.
D 092 ChillicothMO	2017.	2017.	2017.	2017.	8068.
D 144 Columbus IN	675.	675.	675.	675.	2700.
D 131 Dekalb IL	1010.	1010.	1010.	1010.	4040.
D 281 Dubois PA	2539.	2539.	2539.	2539.	10156.
D 051 Enid OK	183.	183.	183.	183.	732.
D 065 Ft Worth TX	145.	145.	145.	145.	580.
D 201 Gadsen AL	196.	196.	196.	196.	784.
D 132 Galesburg IL	1692.	1692.	1692.	1692.	6768.

D 142	Indy	IN	297.	297.	297.	297.	1188.
D 183	Knoxville	TN	1414.	1414.	1414.	1414.	5656.
D 222	Lake city	FL	2351.	2351.	2351.	2351.	9404.
D 715	Los Ang	CA	9042.	9042.	9042.	9042.	36168.
D 035	Lincoln	NE	1974.	1974.	1974.	1974.	7896.
D 073	Mankato	MN	9352.	9352.	9352.	9352.	37408.
D 096	Moberley	MO	870.	870.	870.	870.	3480.
D 053	Ok City	OK	185.	185.	185.	185.	740.
D 113	Prt Allen	LA	533.	533.	533.	533.	2132.
D 713	Salem	OR	488.	488.	488.	488.	1952.
D 381	Salt Lake	UT	1502.	1502.	1502.	1502.	6008.
D 074	St Cloud	MN	3977.	3977.	3977.	3977.	15908.
D 291	Buffalo	NY	12028.	12028.	12028.	12028.	48112.
D 043	Topeka	KS	443.	443.	443.	443.	1772.
D 06G	Tyler	TX	113.	113.	113.	113.	452.
D 401	Tucson	AZ	419.	419.	419.	419.	1676.
D 313	Wenatch	Wa	219.	219.	219.	219.	876.
D 045	Wichita	KS	166.	166.	166.	166.	664.
D 371	Winnemu	NV	334.	334.	334.	334.	1336.
D 241	Winston	NC	2237.	2237.	2237.	2237.	8948.
F 801	SCANDINAVIA		00.	00.	00.	496.	496.
F 802	N.C. EUROPE		1060.	1079.	1874.	1388.	5401.
F 803	S.W. EUROPE		477.	2355.	3630.	5318.	11780.
F 804	ISLANDS		00.	00.	00.	00.	00.
F 805	ADRIATIC		00.	00.	00.	00.	00.
F 806	USSR		00.	237.	992.	243.	1472.
F 807	E BLOCK EURO		1994.	645.	190.	667.	3496.
F 808	E MEDITERRAN		00.	00.	00.	00.	00.
F 809	N AFRICA		1896.	1544.	1354.	1430.	6224.
F 810	RED SEA		47.	269.	543.	137.	996.
F 811	E AFRICA		3.	18.	151.	149.	321.
F 812	W AFRICA		1016.	2465.	1425.	1391.	6297.
F 813	PERSIAN GULF		00.	00.	00.	00.	00.
F 814	W ASIA		294.	294.	294.	294.	1176.
F 815	SE ASIA		10016.	10016.	10016.	10016.	40064.
F 816	TAIWAN		2925.	3289.	3440.	3462.	13116.
F 817	KOREA		2010.	2557.	2928.	2405.	9900.
F 818	JAPAN		9779.	8135.	8064.	8730.	34708.
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		236.	481.	1177.	1010.	2904.
F 824	CENT AMERICA		5982.	4737.	8882.	6240.	25841.
F 825	CARIBBEAN		4333.	3083.	4190.	2806.	14412.

1

STORAGE COST	24728926.
TRUCK COST	24308780.
RAIL COST	197175008.
BARGE COST	20272052.
SHIP COST	6484772O.
HANDLING CST	102951032.

GRAIN SHIPPED FROM SURPLUS REGIONS	422028.
GRAIN SHIPPED TO DEFICIT REGIONS	243424.
GRAIN SHIPPED TO FOREIGN REGIONS	178604.

Validation of Red Spring Wheat Model

Port	FGIS Recorded Exports	Model Solution
	Million Bu	
Mobile	0 4	0
New Orleans	49.5	49.6
Galveston	16.8	16.8
Corpus Christi	0	0
Brownsville	0	0
Charleston	0	0
Baltimore	0	0
Toledo	0	0
Saginaw	0	0
Chicago	0	0
Duluth	20 6	21.1
Seattle	0 6	0 6
Portland	88.9	89.0
California	0	0
San Francisco	-	-
Long Beach	-	-
San Diego	-	-
Baie Comeau	1.5	1.5
Toledo	-	-
Saginaw	-	-
Chicago	-	-
Duluth	-	1.5
Total	178 4	178 6

Special adjustments made in the calibration process:

1) Barge rate from St. Paul, Minnesota to New Orleans, Louisiana was lowered from $0.281 per bushel to $0.181 per bushel to force flows to New Orleans - otherwise they move through the Duluth port.

2) Short truck hauls are included at the river ports to move from barge to domestic location (when barge location and domestic demand location are the same).