



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

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

HARD RED WINTER WHEAT

SEPTEMBER 1990

IWR REPORT 90-R-5

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

HARD RED WINTER WHEAT

Prepared By:

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

**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-5

Content

	page
Introduction.....	i
Programs	
hrw1.f.....	1
hrw2.f.....	18
hrw3.f.....	29
Data files	
f1.hrw.....	56
f2.hrw.....	62
f3.hrw.....	72
f4.hrw.....	87
f8.hrw.....	92
f9.hrw.....	94
Output	
file06_2.hrw.....	98
final.out.....	99
Calibration.....	110

INTRODUCTION

This report documents the hard red winter wheat model. Additional reports document corn, soybeans, hard red spring 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 ---- *                HRW1.F                *
C ---- *      Documented on          Sept 1990      *
C ---- *      Run on                a 386 machine   *
C ---- *      Compiler used         NDP Fortran     *
C ---- *      Grain used             HARD RED WINTER (HRW) *
C ---- *      Data files used       F1,F2,F3,F4, and F8. *
C ---- *      Trace file            U6              *
C ---- *      Input file for HRW2.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.hrw.
C ---- F1.hrw contains information about the model.
C ---- U2 is the unit number for data file, F2.hrw.
C ---- F2.hrw contains information on TRUCK mileage.
C ---- U3 is the unit number for data file, F3.hrw.
C ---- F3.hrw contains information on RAIL costs.
C ---- U4 is the unit number for data file, F4.hrw.
C ---- F4.hrw contains information on BARGE costs.
C ---- U8 is the unit number for data file, F8.hrw.
C ---- F8.hrw contains information about SUPPLIES and DEMANDS.
C ---- U9 is for data files, F9.hrw.
C ---- F9.hrw 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, HRW2.

```

```

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, HRW1.F:

```


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, HRW2.F.

```
WRITE (U12,510) N, NODE, I, J, K, L, M, N
510 FORMAT ( 20I8 )
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 ( 20I4 )
520 FORMAT ( 10F8.3 )
600 FORMAT ( 1H1, 5X, 'NETWORK GENERATOR', /,
1          6X, 'FOR GRAIN SHIPMENT PROBLEM', // )
610 FORMAT ( 6X, 'SUPPLY', 115, 6X, 'DEMAND', 115, / )
620 FORMAT ( 6X, '????????????????????????????????????????????', /,
1          6X, 'INFEASIBLE NETWORK. DEMAND EXCEEDS SUPPLY', /,
```

```
2      6X, '????????????????????????????????????????????????????????', / )
650 FORMAT ( 3I7, 3I10 )
```

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

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

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

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

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

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

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

```
READ (U1,520) 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, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /A3/ SRND, REHD, DRND, PEND, FRND, DMND
COMMON /A4/ SINK, SRCE
COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
COMMON /B4/ PORT(20)
COMMON /C1/ SDTR(65,65) /C2/ SDRL(65,65)
COMMON /C3/ SRTR(65,45) /C4/ SRRL(65,45)
COMMON /C5/ SPTR(65,20) /C6/ SPRL(65,20)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE
COMMON /E1/ SPLY(65)
COMMON /F1/ STOR(65) /F2/ SCST(4)
COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION
COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
INTEGER    SRND, REHD, 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.HRW).

```

```

      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, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /A3/ SRND, REHD, 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 /GD/ 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; REHD, DRND, PEND, FRND, DMND
INTEGER DRGN, RIVR, PORT, UPPR
INTEGER U1,U2,U3,U4,U8,U9,U6,U12
LOGICAL ICED
500 FORMAT ( 20A4 )
510 FORMAT ( 20I4 )
520 FORMAT ( 10F8.3 )

```

```

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

```

```

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

```

```

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

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

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

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

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

      DO 1200 I = 1, MORE
      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, MORE
      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, MORE
      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, MORE
      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 ) + SRNO
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 ) + SRNO

```

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.HRW).

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

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

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

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

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

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

      SUBROUTINE PELVTR
      COMMON /A1/ NOSR, MODR, 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.HRW).

```
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 + MODR
      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.HRW).

```

      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 ---- *                               HRW2.F                               *
C ---- *          DOCUMENTED ON          : SEPT 1990          *
C ---- *          RUN ON                  : A 386 machine      *
C ---- *          COMPILER USED           : NDP Fortran        *
C ---- *          GRAIN USED              : HARD RED WINTER    *
C ---- *          DATA FILES USED       : U12                 *
C ---- *          TRACE FILE              : U6                 *
C ---- *          INPUT FILE FOR HRW3.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, HRW1.
C ---- This data file contains the relevant information which
C ---- is necessary for HRW2 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, HRW3.

```

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

```

```

1:      ACCESS = 'SEQUENTIAL', FDRM = 'UNFORMATTED')

      NTIM      = 0
      FSBL      = .TRUE.
      MAXA      = 34000

C ---- Read in the data from the output file(generated by) HRW1.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/ NOOE(1500)
COMMON /B4/ MIOL(1500) /B5/ NSAVE(1500)
COMMON /C1/ ILO(34000) /C2/ ISAVE(34000) /C3/ JSAVE(34000)
COMMON /C4/ JWV(34000)
COMMON /O1/ KOS(68000) /O2/ MIR(68000) /O3/ NA(68000)
COMMON /O4/ NC(68000) /O5/ 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
5        NOOE(I)=0
          LABL(I)=0
          DO 10 M=1, NR
            I=NF(M)
            J=NA(M)
            IFLOW=NC(M)
            KOST=ISAVE(M)
            NOOE(I)=NOOE(I)+1
            NOOE(J)=NOOE(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)=NOOE(I)
          GO TO 1401
12         DO 13 I=1, NN1
13         NODE(I)=NSAVE(I)
          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-IL0(M)
14    CONTINUE
14D1  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 LABELING 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,T20,111
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=MINO(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,KDS(K) )
    GO TO 210
206 NONS=NONS-1
    JWV(NONS)=K
210 CONTINUE
211 CDNTINUE
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('OIS=',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)=NDDEB

```

```

      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/(20I6))
      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,15,5H ARC, 15, 16H LOST ON SHIFT ,14,4H LOC ,14
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 ---- *                HRW3.F                *
C ---- *      Documented on          SEPT 1990          *
C ---- *      Run on                  a 386 machine      *
C ---- *      Compiler used           NDP Fortran        *
C ---- *      Grain used               HARD RED WINTER    *
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 HRW2.F, it is used as an input for this program.
C ---- K is counter.
C ---- U1,U2,U3,U4,U8,U9,UF1,FD store unit numbers for the files.

```

```

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

```

```

C ---- Unit number for each file.
C ---- U1 is the unit number for data file, F1.hrw.
C ---- F1.hrw contains information about the model.
C ---- U2 is the unit number for data file, F2.hrw.
C ---- F2.hrw contains information on TRUCK mileage.
C ---- U3 is the unit number for data file, F3.hrw.
C ---- F3.hrw contains information on RAIL costs.
C ---- U4 is the unit number for data file,F4.hrw.
C ---- F4.hrw contains information on BARGE costs.
C ---- U8 is the unit number for data file, F8.hrw.
C ---- F8.hrw contains information about SUPPLIES and DEMANDS.
C ---- U9 is for the unit number for data files, F9.hrw.
C ---- F9.hrw contains the NAMES of all the regions.
C ---- UF1 is the unit number for the input file produced by HRW2.F.
C ---- It is the output file from HRW2.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 -----
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
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 Demanded 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 ---- RRIO Rail unloading (Deficit) cost.
 C ---- TRIR River location unloading TRuck cost.
 C ---- RRIR River location unloading Rail cost.
 C ---- BRIR River location unloading Barge cost.
 C ---- TRIP Port unloading Truck cost.
 C ---- RRIP Port unloading Rail cost.
 C ---- BRIP Port unloading Barge cost.

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

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

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

```

      READ (U9,530) ( NAM4(I,J), J = 1, 3 )
2400 CONTINUE
      DO 2500 I = 1, NOFR
      READ (U9,530) ( NAM5(I,J), J = 1, 3 )
2500 CONTINUE

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

740 FORMAT ( 1H1, //, 5X, 'STORAGE COST', F12.0, /,
1 5X, 'TRUCK COST ', F12.0, /, 5X, 'RAIL COST ', F12.0, /,
2 5X, 'BARGE COST ', F12.0, /, 5X, 'SHIP COST ', F12.0, /,
3 5X, 'HANDLING CST', F12.0 )
      TOPR = (TOPR / OPERATION)
      TODD = (TODD / OPERATION)
      TOFD = (TOFD / OPERATION)
      WRITE (FD,750) TOPR, TODD, TOFD
750 FORMAT ( ////, 5X, 'GRAIN SHIPPED FROM SURPLUS REGIONS', F15.0,
1          //, 5X, 'GRAIN SHIPPED TO DEFICIT REGIONS ', F15.0,
2          //, 5X, 'GRAIN SHIPPED TO FOREIGN REGIONS ', F15.0,
3          //// )
      RETURN
      END

```

```

C -----
C          ===== SURPLS =====
C ---- SURPLS is used to display the heading of the output file.
C ---- It reads the data concerning the storage, the amount of

```

C ----: grain produced by, truck(mileage) and rail cost
C ---- of all the surplus regions.

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

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

```
500 FORMAT ( 20A4 )
520 FORMAT ( 10F8.3 )
700 FORMAT ( 1H1, //, 9X, 18HORIGIN/DESTN MODE, 6X, 6HSUPPLY, 14X,
1 16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
2 7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4I10, 4X,
3 8HSHIPMENT, 5X, 28HCOST HAULING COST COSTS, // )
710 FORMAT ( /, 5X, 1HS, A4, 1X, 3A4, 4X, F12.0, / )
720 FORMAT ( 5X, 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, NODR )  
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 alphas 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 ---- CDST (=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 GENFLD 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 ),

```

2 RT/OPERATION, COST*OPERATION, RCST, HCSR

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

TODD = TODD + RT
TGSO(I) = TGSO(I) + RT
TOPR = TOPR + RT
TGRD(M) = TGRD(M) + RT

2400 CONTINUE

2500 CONTINUE

C ---- SURPLUS region ---->TRUCK-----> RIVER region.
C ---- Subroutine SERIAL is used to check that all input data
C ---- correctly 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 = (0.066374 + 0.104892 * COST)
ELSE IF (COST .GT. 225 .and. COST .LE. 245) THEN
COST = (23.67 + 0.0 * COST)
ELSE IF (COST .GT. 245) THEN
COST = (0.68037 + 0.093976 * COST)
ELSE IF (COST .GE. 9999.) THEN
GOTO 2600
ENDIF

COST = (COST * TRUCK) / 100.0

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

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

```

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

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

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

C ---- Update 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 M = 1, NOTF
      TGRR(NT,M)= TGRR(NT,M) + R(M)
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(M)),
C ---- unit cost(COST), total hauling (truck)cost (TCST) and
C ---- handling cost (HCST) is printed.

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

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

      TGSO(I)   = TGSO(I) + TT

```

```

      TOPR      = TOPR + TT

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

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

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

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

C ----  If the amount of grain in total shipment (RT)
C ----  in this port region is less than 0 i.e no grain is
C ----  received then it is ignored, else output
C ----  the code number (PORT(M)) and the name (NAM4(M,L) of
C ----  the port region, the amount of grain in each shipment
C ----  by time (R(N)), the total shipment(RT), unit cost(COST),
C ----  total hauling cost(RCST) and handling cost(HCSR).

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

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

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

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

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

```

```

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

```

```

WRITE (FD,800)

```

```

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

```

```

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

```

```

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

```

```

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

```

```

SUBROUTINE RELVTR
DIMENSION T(4), R(4)
COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /B2/ DRGN(65) /B3/ RIVR(45)
COMMON /B4/ PORT(20)
COMMON /C1/ RDTR(65,65) /C2/ RDRL(65,65)
COMMON /C3/ RRBG(65,45) /C5/ RPBG(65,16)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
COMMON /F3/ TLDS, RLOS, TLOR, RLOR, BLOR, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, DPERATION /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(CDST) is greater than 9999 then it is
C ---- ignored else the cost by truck(COST) is calculated.
C ---- The COST includes BET and ALF.
C ---- Subroutine GENFLO is used to calculate the hauling
C ---- truck cost(TCST) and the handling cost (HCST).
C ---- T is the amount of grain in each shipment by time.
C ---- TT is the amount of grain in total shipment.

```
DO 2200 M = 1, NODR
  IDNT = IDEN(M)
  CALL SERIAL ( IDNT, NORE, RIVR, NT )
```

```

IF ( NF .NE. NT ) GO TO 2200
NT      = M
COST    = RDTR(1,NT)
IF ( COST .GE. 9999. ) GO TO 2200
ALF     = ALFA(NT)
BET     = BETA(NT)

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

COST = ( COST * TRUCK ) / 100.0

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

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

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

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

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

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

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

C ---- Selected BARGE (river) points-->by Rail-->DEFICIT regions.

```

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

```

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

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

```

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

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

```

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

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

```

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

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

C ---- barge cost(BCST) and the handling cost (HCSB).
 C ---- T is the amount of grain in each shipment by time.
 C ---- TT is the amount of grain in total shipment.

```

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

```

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

```

IF ( TT .LE. 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/ TRIO, 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, TOOO, TOFD
COMMON /GC/ TGSO(65), TGRD(65), TGRF(65)
COMMON /G0/ TGRR(65,4), TGRP(65,4)
COMMON /R1/ NAM1(65,3), NAM2(65,3), NAM3(45,3)
COMMON /R2/ NAM4(20,3), NAM5(25,3)
DIMENSION IDN3(16)
INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

```

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

```

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

```

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

```

READ (U1,520) ( ALPHA(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 CDNTINUE

```

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)   = FLDW(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, NDFR
READ (UB,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)   = FLDW(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, NOAY(4)
      COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
      INTEGER FLOW
      AT      = 0.0
      TCST    = 0.0
      HCST    = 0.0
      DO 1100 N = 1, NOTF
      K      = K + 1
      A(N)   = FLOW(K)
      AT     = AT + A(N)
1100 CONTINUE

```

```

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

```

```

      TCST      = AT * COST * 1000.0

```

```

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

```

```

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

```

C : F1.HRW

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.hrw.

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

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

HARD RED WINTER WHEAT, FULLER, GRANT, TEH, FELLIN

0
C

C This line shows the number of regions involved in this model.
C If there were any changes in the number of regions, the corresponding
C number must be changed.
C e.g. if the number of surplus regions were increased to 39 then change 31
C to 39.

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

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

C In this model, the

C Number of surplus regions = 38
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. **31 where * is a space.

38 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 38 surplus regions in this model.

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.

022 111 052 061 017 071 015 171 024 351 391 014 134 051 06H 323 312 322 034 638
321 044 324 062 033 012 325 021 053 072 023 411 713 042 031 041 016 011 0

C-----

C This line shows the codes for the 34 deficit regions in this model.

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.

085 211 098 154 251 092 144 131 013 281 362 065 201 132 142 183 222 715 035 073
096 702 331 381 06A 311 074 291 043 401 032 313 045 241 0

C-----

C These lines show the codes for the 43 river barge ports in this
C model.

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

C-----

C This line shows the codes for the 20 ports in this model.
 C If there were a change in the number of port regions, the code of
 C the new port would be added to the end of the list.
 C If there were a deletion, then the code involved would be deleted from
 C this list.

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

C-----

C These 2 lines codes for the 25 foreign regions in this model.
 C 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. ddddd.d where d is a digit.
 C e.g. 999999.9

999999.999999.999999.999999.999999.999999.999999.999999.999999.999999.9
 999999.999999.999999.999999.999999.999999.999999.999999.999999.999999.9
 999999.999999.999999.999999.999999.999999.999999.999999.999999.999999.9
 999999.999999.999999.999999.999999.999999.999999.999999.999999.9 0

C-----

C The grain storage cost, rail, truck, barge, and ship adjustment
 C factors 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 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
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
5.654	0.191	5.654	0.191	6.486	0.208				0

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

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

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

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

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

C e.g, the 1st entry is the surplus region, Abilene.

C Abilene is linked to the barge loading port, Muskogee (631).

C e.g, Alexandria (the 2nd code number) is not linked to any barge loading
 C region. So 999 is used to indicate that Alexandria is not linked to
 C to any barge loading port.

601 999 630 630 601 644 601 999 609 639 999 601 608 631 999 638 639 638 611 638
 638 630 638 630 611 644 638 609 631 601 609 999 643 613 611 613 601 644 0


```

:
C .
C .
C .

C surplus region 38 -----> deficit region 1
C -----> deficit region 2
C .
C .
C -----> deficit region 34

```

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 38 -----> corresponding river barge loading region

```

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

```

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 38 -----> corresponding deficit region

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

C This includes the truck mileage from each of the 38 surplus regions
C to each of the 34 deficit regions.

C A 38 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 (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 Abilene were not a surplus region, then the first 4 lines would
C be removed. The file would start with 857.

446.000	1331.000	828.000	1067.000	1366.000	600.000	910.000	653.000	200.000	1185.000
1676.000	967.000	1260.000	639.000	866.000	1212.000	1617.000	1649.000	406.000	279.000
671.000	1388.000	1212.000	948.000	1233.000	1050.000	238.000	1350.000	551.000	1491.000
287.000	1219.000	608.000	1367.000						0.000
857.000	562.000	478.000	987.000	1036.000	687.000	768.000	877.000	1431.000	1191.000
1527.000	338.000	471.000	791.000	774.000	666.000	661.000	1694.000	845.000	1030.000
676.000	189.000	2105.000	1549.000	429.000	2154.000	1140.000	1381.000	681.000	1223.000
1108.000	2302.000	627.000	869.000						0.000
710.000	972.000	507.000	1129.000	1356.000	572.000	903.000	918.000	1046.000	1330.000
1106.000	197.000	867.000	809.000	880.000	986.000	1152.000	1248.000	541.000	854.000
620.000	713.000	1601.000	1045.000	405.000	1663.000	949.000	1504.000	424.000	816.000
641.000	1798.000	285.000	1228.000						0.000
775.000	1097.000	618.000	1240.000	1481.000	637.000	1014.000	996.000	1069.000	1441.000
956.000	334.000	992.000	882.000	991.000	1111.000	1289.000	1091.000	586.000	899.000

686.000 850.000 1444.000 888.000 516.000 1506.000 994.000 1615.000 489.000 677.000
601.000 1641.000 350.000 1353.000 0.000
520.000 1403.000 907.000 1128.000 1427.000 679.000 971.000 714.000 193.000 1253.000
1666.000 1038.000 1332.000 711.000 927.000 1273.000 1689.000 1639.000 485.000 351.000
746.000 1467.000 1195.000 938.000 1304.000 1033.000 279.000 1411.000 630.000 1511.000
294.000 1202.000 679.000 1428.000 0.000
436.000 1288.000 850.000 1000.000 1299.000 606.000 843.000 586.000 204.000 1118.000
1979.000 1147.000 1227.000 596.000 799.000 1145.000 1574.000 1952.000 532.000 272.000
652.000 1434.000 1404.000 1251.000 1413.000 1242.000 153.000 1283.000 668.000 1740.000
590.000 1411.000 788.000 1300.000 0.000
637.000 1495.000 1023.000 1213.000 1512.000 795.000 1056.000 799.000 180.000 1331.000
1644.000 1135.000 1434.000 803.000 1012.000 1358.000 1781.000 1617.000 592.000 460.000
855.000 1583.000 1079.000 916.000 1388.000 917.000 364.000 1496.000 746.000 1506.000
347.000 1086.000 776.000 1513.000 0.000
624.000 301.000 364.000 391.000 547.000 516.000 177.000 427.000 1175.000 595.000
1933.000 744.000 242.000 406.000 221.000 203.000 587.000 2073.000 736.000 764.000
432.000 578.000 2125.000 1613.000 953.000 2070.000 847.000 785.000 598.000 1629.000
1009.000 2239.000 691.000 437.000 0.000
303.000 1201.000 679.000 954.000 1253.000 451.000 797.000 532.000 355.000 1079.000
1700.000 878.000 1120.000 511.000 753.000 1099.000 1487.000 1707.000 263.000 151.000
527.000 1239.000 1358.000 1006.000 1144.000 1196.000 195.000 1237.000 402.000 1454.000
299.000 1365.000 519.000 1254.000 0.000
714.000 1574.000 1028.000 1410.000 1709.000 801.000 1236.000 983.000 693.000 1529.000
1131.000 1024.000 1482.000 934.000 1192.000 1504.000 1859.000 1104.000 577.000 709.000
883.000 1527.000 863.000 403.000 1206.000 816.000 773.000 1694.000 732.000 1059.000
322.000 985.000 731.000 1710.000 0.000
684.000 1398.000 835.000 1307.000 1597.000 637.000 1099.000 943.000 851.000 1486.000
1020.000 757.000 1306.000 857.000 1058.000 1328.000 1666.000 1059.000 486.000 765.000
719.000 1273.000 1027.000 504.000 939.000 1089.000 856.000 1651.000 535.000 835.000
394.000 1224.000 509.000 1559.000 0.000
733.000 1591.000 1112.000 1309.000 1608.000 884.000 1152.000 895.000 276.000 1427.000
1556.000 1206.000 1530.000 899.000 1108.000 1454.000 1877.000 1529.000 675.000 556.000
951.000 1672.000 980.000 828.000 1408.000 818.000 460.000 1592.000 835.000 1424.000
378.000 987.000 859.000 1609.000 0.000
404.000 508.000 266.000 387.000 667.000 310.000 161.000 216.000 955.000 588.000
1800.000 751.000 447.000 186.000 138.000 421.000 794.000 1935.000 530.000 544.000
246.000 707.000 1923.000 1411.000 989.000 1850.000 627.000 762.000 412.000 1530.000
789.000 2019.000 543.000 625.000 0.000
531.000 897.000 379.000 1001.000 1261.000 393.000 775.000 752.000 916.000 1202.000
1218.000 272.000 792.000 638.000 752.000 911.000 1152.000 1353.000 365.000 678.000
442.000 751.000 1577.000 1047.000 522.000 1613.000 773.000 1376.000 245.000 939.000
569.000 1774.000 109.000 1153.000 0.000
1099.000 1211.000 884.000 1506.000 1623.000 961.000 1280.000 1295.000 1410.000 1707.000
867.000 385.000 1120.000 1186.000 1257.000 1253.000 1332.000 1034.000 927.000 1240.000
997.000 854.000 1573.000 1017.000 329.000 1729.000 1335.000 1881.000 813.000 563.000
942.000 1770.000 681.000 1495.000 0.000
979.000 1841.000 1354.000 1559.000 1858.000 1126.000 1402.000 1145.000 400.000 1677.000
1515.000 1448.000 1780.000 1149.000 1358.000 1704.000 2127.000 1488.000 917.000 806.000
1201.000 1914.000 800.000 775.000 1637.000 638.000 710.000 1842.000 1077.000 1539.000
620.000 807.000 1101.000 1859.000 0.000
1531.000 2425.000 1892.000 2202.000 2501.000 1664.000 2045.000 1787.000 1123.000 2320.000
1357.000 1925.000 2344.000 1751.000 2001.000 2347.000 2711.000 1188.000 1448.000 1414.000
1746.000 2428.000 207.000 763.000 2082.000 85.000 1362.000 2485.000 1609.000 1490.000
1148.000 100.000 1629.000 2502.000 0.000
1140.000 2002.000 1515.000 1720.000 2019.000 1287.000 1563.000 1306.000 561.000 1838.000

1419.0001605.0001941.0001310.0001519.0001865.0002288.0001392.0001078.000 967.000
1362.0002075.000 639.000 679.0001787.000 477.000 871.0002003.0001238.0001443.000
781.000 646.0001262.0002020.000 0.000
310.0001085.000 539.000 968.0001258.000 312.000 767.000 564.000 571.0001107.000
1380.000 612.000 993.000 475.000 723.0001015.0001370.0001447.000 104.000 396.000
394.0001069.0001316.000 804.000 866.0001297.000 491.0001272.000 243.0001136.000
234.0001466.000 253.0001245.000 0.000
1466.0002360.0001827.0002137.0002436.0001599.0001980.0001722.0001058.0002510.000
1246.0001825.0002279.0001686.0001936.0002282.0002646.0001102.0001383.0001349.000
1681.0002352.000 142.000 614.0001933.000 110.0001297.0002420.0001544.0001341.000
1083.000 221.0001564.0002437.000 0.000
1035.0001929.0001396.0001654.0001953.0001168.0001497.0001240.000 573.0001779.000
1311.0001435.0001848.0001243.0001453.0001799.0002215.0001284.000 952.000 883.000
1250.0001938.000 634.000 571.0001617.000 472.000 805.0001937.0001113.0001335.000
652.000 641.0001133.0001954.000 0.000
631.0001093.000 538.0001136.0001403.000 493.000 910.000 852.000 905.0001337.000
1051.000 421.000 988.000 738.000 887.0001082.0001344.0001186.000 422.000 735.000
542.000 920.0001384.000 854.000 604.0001445.000 830.0001511.000 345.000 786.000
437.0001581.000 214.0001324.000 0.000
1025.0001919.0001386.0001710.0002009.0001158.0001547.0001281.000 682.0001828.000
1174.0001419.0001838.0001245.0001503.0001849.0002205.0001147.000 942.000 926.000
1240.0001922.000 586.000 434.0001601.000 424.000 888.0001993.0001103.0001198.000
642.000 593.0001123.0002010.000 0.000
895.0001104.000 709.0001331.0001508.000 757.0001105.0001116.0001189.0001532.000
931.000 290.0001013.0001002.0001082.0001138.0001256.0001098.000 706.0001019.000
806.000 817.0001477.000 921.000 398.0001626.0001114.0001706.000 609.000 627.000
721.0001674.000 470.0001380.000 0.000
430.0001183.000 624.0001089.0001372.000 412.000 874.000 693.000 677.0001241.000
1267.000 632.0001091.000 599.000 833.0001114.0001456.0001319.000 228.000 517.000
494.0001129.0001255.000 724.000 837.0001225.000 611.0001406.000 326.0001023.000
200.0001394.000 323.0001355.000 0.000
707.0001565.0001099.0001283.0001582.000 871.0001126.000 869.000 122.0001401.000
1726.0001238.0001504.000 873.0001082.0001428.0001851.0001699.000 677.000 530.000
925.0001659.0001078.000 994.0001493.000 916.000 434.0001566.000 822.0001594.000
452.0001085.000 879.0001583.000 0.000
821.0001715.0001182.0001457.0001756.000 954.0001300.0001043.000 422.0001575.000
1434.0001257.0001634.0001029.0001256.0001602.0002001.0001407.000 738.000 669.000
1036.0001735.000 822.000 694.0001443.000 660.000 628.0001740.000 899.0001350.000
438.000 829.000 919.0001757.000 0.000
546.0001431.000 921.0001167.0001466.000 693.0001010.000 753.000 268.0001285.000
1579.0001033.0001360.000 739.000 966.0001312.0001717.0001552.000 490.000 379.000
769.0001481.0001112.000 851.0001286.000 950.000 338.0001450.000 644.0001441.000
245.0001119.000 674.0001467.000 0.000
577.000 839.000 362.000 984.0001223.000 439.000 758.000 773.000 967.0001185.000
1214.000 200.000 734.000 664.000 735.000 853.0001082.0001349.000 415.000 728.000
475.000 668.0001630.0001100.000 466.0001677.000 823.0001359.000 291.000 935.000
640.0001827.000 159.0001095.000 0.000
339.0001222.000 752.000 958.0001257.000 507.000 801.000 544.000 281.0001076.000
1778.000 954.0001151.000 530.000 757.0001103.0001508.0001754.000 339.000 170.000
565.0001312.0001321.0001053.0001220.0001159.000 129.0001241.000 475.0001532.000
374.0001328.000 595.0001258.000 0.000
593.0001487.000 954.0001278.0001577.000 726.0001115.000 849.000 489.0001396.000
1390.0001026.0001406.000 813.0001071.0001417.0001773.0001363.000 510.000 504.000
808.0001507.0001018.000 662.0001218.000 856.000 538.0001561.000 671.0001229.000
210.0001025.000 691.0001578.000 0.000

990.0001279.000 833.0001455.0001683.000 852.0001229.0001211.0001271.0001656.000
 766.000 453.0001188.0001097.0001206.0001313.0001419.000 933.000 788.0001101.000
 901.000 980.0001358.000 802.000 522.0001514.0001196.0001830.000 704.000 462.000
 803.0001555.000 565.0001555.000 0.000
 1806.0002619.0002072.0002490.0002780.0001846.0002289.0002075.0001433.0002621.000
 1105.0001996.0002527.0001993.0002245.0002549.0002903.000 913.0001622.0001724.000
 1928.0002523.000 258.000 785.0002104.000 395.0001672.0002786.0001769.0001340.000
 1414.000 335.0001757.0002770.000 0.000
 397.000 974.000 411.000 910.0001177.000 262.000 684.000 621.000 723.0001111.000
 1293.000 445.000 882.000 507.000 661.000 904.0001242.0001428.000 181.000 494.000
 313.000 902.0001445.000 932.000 711.0001459.000 589.0001285.000 111.0001028.000
 396.0001628.000 86.0001135.000 0.000
 598.0001398.000 852.0001269.0001559.000 625.0001068.000 857.000 668.0001400.000
 1224.000 865.0001306.000 772.0001024.0001328.0001683.0001234.000 401.000 605.000
 707.0001351.0001039.000 533.0001048.000 992.000 672.0001565.000 556.0001039.000
 218.0001161.000 555.0001549.000 0.000
 521.0001225.000 671.0001124.0001414.000 454.000 916.000 779.000 768.0001316.000
 1188.000 659.0001133.000 686.000 875.0001155.0001502.0001242.000 319.000 607.000
 536.0001145.0001174.000 662.000 844.0001219.000 702.0001485.000 368.000 968.000
 291.0001371.000 354.0001386.000 0.000
 502.0001360.000 916.0001078.0001377.000 672.000 921.000 664.000 122.0001196.000
 1779.0001066.0001299.000 668.000 877.0001223.0001646.0001752.000 500.000 325.000
 720.0001476.0001210.0001051.0001332.0001048.000 229.0001361.000 639.0001617.000
 414.0001217.000 707.0001378.000 0.000
 836.0001694.0001228.0001412.0001711.0001000.0001255.000 998.000 251.0001530.000
 1614.0001337.0001633.0001002.0001211.0001557.0001980.0001587.000 806.000 659.000
 1054.0001788.000 949.000 875.0001539.000 787.000 563.0001695.000 951.0001523.000
 509.000 956.000 990.0001712.000 0.000

C-----

C Explanation for the Second part

C This links each surplus region (38) to its corresponding river
 C barge-loading locations. Conceptually, a (1 X38) 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*38 -----> corresponding river barge-loading location

C e.g, 1st surplus region, Aberdeen SD (022) to river barge region,
 C Sioux City IA (609) = 288 miles
 C e.g, 3rd surplus region, Altus OK to river barge region,
 C Catoosa OK(630) = 250 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 Abilene were not a surplus region, then 288 would be deleted.

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

288.000	999.000	250.000	361.000	351.000	153.000	436.000	61.000	334.000	836.000
999.000	532.000	97.000	176.000	999.000	658.000	10.000	497.000	200.000	25.000
492.000	326.000	444.000	452.000	278.000	479.000	680.000	381.000	141.000	181.000
425.000	999.000	47.000	251.000	451.000	427.000	301.000	608.000		0.000

C -----

C Explanation for the Third part

C (3) Each surplus region(38) 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 38 -----> port region 1 for export
 C -----> port region 2 for export
 C .
 C .
 C -----> port region 20 for export

C There are 38 surplus regions and 20 port locations thus a (38 by 20)
 C matrix.

C e.g, 1st surplus region, Aberdeen SD(022) to 1st port region,
 C Mobile AL(701) = 1394 miles

C e.g, 1st surplus region, Aberdeen SD(022) to 2nd port region,
 C New Orleans LA(702) = 1388 miles

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

C If a new surplus region were added to the end of the surplus region list,

C then its truck mileages to all the port regions would be added to the list

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

C If a surplus region were deleted, then its truck mileages to all the
C port regions would be removed.
C e.g, if Aberdeen were removed as a surplus region, then the first 2 lines
C (below) would be removed. The file would start with 309.

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
309.000 189.000 229.000 450.000 585.000 839.0001185.000 993.0001092.000 869.000
1226.0002385.0002316.0002060.0001694.0001638.000 993.0001092.000 869.0001226.000
807.000 713.000 508.000 551.000 680.0001261.0001443.0001099.0001186.000 932.000
1079.0001881.0001812.0001556.0001248.0001217.0001099.0001186.000 932.0001079.000
944.000 850.000 645.000 662.000 791.0001386.0001554.0001210.0001297.0001043.000
1128.0001724.0001655.0001399.0001091.0001067.0001210.0001297.0001043.0001128.000
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
498.000 578.000 829.0001036.0001171.000 575.000 693.000 411.000 526.000 393.000
851.0002348.0002336.0002365.0002073.0002044.000 411.000 526.000 393.000 851.000
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
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
646.000 707.000 893.0001076.0001211.000 777.000 701.000 357.000 452.000 205.000
632.0002128.0002134.0002163.0001935.0001911.000 357.000 452.000 205.000 632.000
831.000 751.000 581.000 657.000 792.0001186.0001315.000 971.0001057.000 802.000
900.0001857.0001788.0001661.0001383.0001329.000 971.0001057.000 802.000 900.000
596.000 555.000 505.000 462.000 566.0001488.0001772.000 842.000 929.0001309.000
1468.0001853.0001784.0001408.0001034.000 978.000 842.000 929.0001309.0001468.000
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
2458.0002428.0002236.0002228.0002346.0002697.0002488.0002052.0002056.0001820.000
1442.000 326.000 325.000 865.0001102.0001194.0002052.0002056.0001820.0001442.000
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
1105.0001069.000 910.000 997.0001132.0001365.0001271.000 839.000 885.000 617.000
625.0001575.0001527.0001556.0001447.0001495.000 839.000 885.000 617.000 625.000
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

```

885.000 750.000 820.000 1208.000 1284.000 1333.000 1504.000 1499.000 1272.000 885.000
1000.000 920.000 732.000 750.000 879.000 1382.000 1450.000 1090.000 1157.000 902.000
964.000 1664.000 1595.000 1494.000 1186.000 1162.000 1090.000 1157.000 902.000 964.000
1952.000 1922.000 1730.000 1747.000 1876.000 2199.000 1996.000 1560.000 1587.000 1328.000
973.000 702.000 772.000 1071.000 1147.000 1196.000 1560.000 1587.000 1328.000 973.000
911.000 817.000 566.000 544.000 673.000 1393.000 1645.000 1301.000 1388.000 1134.000
1248.000 1757.000 1688.000 1432.000 1098.000 1042.000 1301.000 1388.000 1134.000 1248.000
1197.000 1129.000 941.000 982.000 1110.000 1474.000 1401.000 971.000 1017.000 745.000
746.000 1527.000 1462.000 1479.000 1405.000 1381.000 971.000 1017.000 745.000 746.000
1665.000 1659.000 1536.000 1623.000 1758.000 1778.000 1569.000 1133.000 1093.000 901.000
479.000 1194.000 1264.000 1631.000 1699.000 1748.000 1133.000 1093.000 901.000 479.000
1748.000 1735.000 1568.000 1589.000 1718.000 1952.000 1743.000 1307.000 1327.000 1075.000
713.000 938.000 1008.000 1331.000 1407.000 1456.000 1307.000 1327.000 1075.000 713.000
1487.000 1481.000 1331.000 1418.000 1553.000 1662.000 1453.000 1017.000 1062.000 785.000
475.000 1228.000 1298.000 1603.000 1552.000 1601.000 1017.000 1062.000 785.000 475.000
475.000 668.000 498.000 585.000 720.000 1128.000 1298.000 954.000 1041.000 787.000
946.000 1910.000 1841.000 1657.000 1349.000 1325.000 954.000 1041.000 787.000 946.000
1316.000 1312.000 1224.000 1339.000 1474.000 1453.000 1244.000 808.000 853.000 576.000
268.000 1437.000 1507.000 1805.000 1754.000 1803.000 808.000 853.000 576.000 268.000
1520.000 1507.000 1337.000 1364.000 1493.000 1767.000 1564.000 1128.000 1173.000 896.000
677.000 1134.000 1204.000 1414.000 1363.000 1412.000 1128.000 1173.000 896.000 677.000
1074.000 980.000 714.000 668.000 781.000 1556.000 1769.000 1425.000 730.000 1258.000
1330.000 2562.000 1569.000 2443.000 1234.000 1283.000 1425.000 730.000 1258.000 1330.000
2605.000 2523.000 2272.000 2250.000 2368.000 2899.000 2785.000 2353.000 2366.000 2128.000
1752.000 219.000 47.000 590.000 913.000 1038.000 2353.000 2366.000 2128.000 1752.000
977.000 902.000 743.000 830.000 965.000 1254.000 1224.000 861.000 928.000 673.000
723.000 1725.000 1656.000 1663.000 1428.000 1404.000 861.000 928.000 673.000 723.000
1418.000 1351.000 1176.000 1194.000 1323.000 1678.000 1564.000 1132.000 1178.000 910.000
811.000 1270.000 1250.000 1285.000 1234.000 1283.000 1132.000 1178.000 910.000 811.000
1225.000 1145.000 970.000 990.000 1119.000 1505.000 1438.000 1054.000 1100.000 832.000
836.000 1454.000 1385.000 1414.000 1242.000 1291.000 1054.000 1100.000 832.000 836.000
1465.000 1476.000 1364.000 1451.000 1586.000 1573.000 1364.000 928.000 923.000 696.000
309.000 1326.000 1396.000 1735.000 1752.000 1801.000 928.000 923.000 696.000 309.000
1794.000 1788.000 1647.000 1685.000 1814.000 1907.000 1698.000 1262.000 1222.000 1030.000
608.000 1065.000 1135.000 1512.000 1587.000 1636.000 1262.000 1222.000 1030.000 608.000

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 1st deficit region Ames IA (085) = 176 miles
 C e.g, barge-unloading river location, Chatanoo TN(627) to
 C 2nd 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 Amarillo were not a deficit region, then 392 would be deleted.

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

176.000	114.000	193.000	183.000	443.000	74.000	67.000	58.000	404.000	387.000
999.000	316.000	34.000	49.000	111.000	2.000	999.000	999.000	54.000	82.000
66.000	2.000	65.000	999.000	999.000	137.000	73.000	1.000	65.000	1069.000
237.000	132.000	197.000	999.000						0.000

C
C
C
C
C
C
C

surplus region 38 -----> deficit region 1
-----> deficit region 2
.
.
-----> deficit region 34

C
C
C
C
C

For example,
the rail cost from the 1st surplus region, Aberdeen S0 (022) to
1st deficit region, Ames IA (085) = 41.400 cents/bushel
the rail cost from Aberdeen S0 (022) to 3rd deficit region,
Cabool MO(098) = 55.800 cents/bushel

41.400	999.999	55.800	78.000	999.999	49.800	72.000	48.000	31.800	999.999
999.999	999.999	999.999	48.600	66.000	999.999	999.999	999.999	34.200	31.200
52.800	999.999	999.999	999.999	999.999	999.999	999.999	30.000	105.600	999.999
41.400	999.999	999.999	999.999						0.000
999.999	46.200	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	39.600	999.999	999.999	51.600	55.200	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						0.000
999.999	80.400	45.600	999.999	999.999	52.800	999.999	999.999	999.999	999.999
87.600	23.400	73.800	999.999	999.999	86.400	97.800	99.600	999.999	999.999
59.400	60.600	999.999	87.000	43.800	999.999	999.999	999.999	999.999	68.400
999.999	999.999	31.200	999.999						0.000
999.999	87.000	56.400	999.999	999.999	48.600	999.999	999.999	999.999	999.999
82.800	30.000	78.600	999.999	999.999	87.600	97.200	83.400	999.999	999.999
66.000	67.200	999.999	73.200	53.400	999.999	999.999	999.999	55.200	61.800
999.999	999.999	33.000	999.999						0.000
56.400	999.999	999.999	999.999	999.999	65.400	999.999	55.800	33.600	999.999
999.999	999.999	999.999	57.600	999.999	999.999	999.999	999.999	999.999	44.400
67.800	999.999	999.999	999.999	999.999	999.999	31.800	999.999	999.999	999.999
57.000	999.999	999.999	999.999						0.000
46.200	999.999	999.999	999.999	999.999	999.999	999.999	49.200	26.400	999.999
999.999	999.999	999.999	49.800	999.999	999.999	999.999	999.999	999.999	32.400
999.999	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						0.000
56.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	26.400	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	39.600
999.999	999.999	999.999	78.000	999.999	999.999	30.000	999.999	999.999	999.999
999.999	999.999	999.999	999.999						0.000
999.999	27.000	999.999	44.400	57.600	999.999	31.800	49.800	999.999	53.400
999.999	999.999	27.600	46.200	25.200	25.800	55.800	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	40.800						0.000
42.000	97.200	999.999	81.000	109.200	39.600	75.000	52.800	40.800	97.200
999.999	999.999	999.999	55.800	62.400	999.999	999.999	999.999	999.999	17.400
999.999	999.999	999.999	999.999	999.999	999.999	25.800	98.000	999.999	999.999
36.600	999.999	999.999	999.999						0.000
65.400	999.999	82.200	999.999	999.999	78.000	999.999	96.600	73.800	999.999
115.200	999.999	999.999	98.400	999.999	999.999	999.999	102.000	57.000	999.999
999.999	999.999	99.000	29.400	999.999	999.999	999.999	999.999	61.800	78.600

50.400	73.800	61.200	999.999							0.000
54.600	999.999	68.400	999.999	999.999	56.400	999.999	72.600	96.600	999.999	
82.000	999.999	999.999	72.000	999.999	999.999	999.999	95.400	37.200	999.999	
59.400	999.999	72.600	44.400	999.999	97.800	999.999	999.999	42.600	73.200	
41.400	106.800	43.800	999.999							0.000
62.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	62.400	31.800	999.999
999.999	999.999	999.999	63.000	999.999	999.999	999.999	999.999	118.800	999.999	45.000
999.999	999.999	88.200	72.000	999.999	96.000	36.000	999.999	999.999	999.999	
63.000	102.000	999.999	999.999							0.000
999.999	57.600	999.999	40.200	63.600	999.999	30.000	32.400	999.999	999.999	56.400
999.999	999.999	51.600	21.600	17.400	46.200	86.400	999.999	999.999	999.999	
999.999	55.800	999.999	999.999	999.999	999.999	999.999	999.999	60.600	999.999	999.999
999.999	999.999	999.999	61.800							0.000
999.999	83.400	33.000	90.600	91.800	38.400	84.600	67.200	999.999	999.999	106.200
94.200	30.600	77.400	58.200	72.000	76.200	97.800	100.200	999.999	999.999	
39.600	63.600	999.999	87.600	52.800	999.999	999.999	125.400	39.000	999.999	
999.999	999.999	15.000	100.200							0.000
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
69.600	40.800	999.999	999.999	999.999	999.999	999.999	999.999	81.000	999.999	999.999
999.999	999.999	999.999	105.600	42.000	999.999	999.999	999.999	999.999	999.999	49.200
999.999	999.999	999.999	999.999							0.000
71.400	999.999	108.600	142.800	999.999	97.200	100.800	67.800	30.600	999.999	
999.999	999.999	999.999	66.000	90.000	999.999	999.999	999.999	82.800	57.000	
90.600	999.999	59.400	61.800	999.999	40.200	46.800	999.999	88.800	999.999	
74.400	45.600	111.000	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
112.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	105.600	999.999	999.999
999.999	999.999	35.400	58.200	999.999	15.000	999.999	999.999	999.999	999.999	999.999
999.999	25.200	999.999	999.999							0.000
97.800	999.999	999.999	114.000	999.999	123.600	104.400	142.200	43.200	999.999	
126.600	999.999	999.999	139.800	155.400	999.999	999.999	115.800	108.000	61.200	
125.400	999.999	57.600	60.000	999.999	36.000	68.400	999.999	113.400	999.999	
106.200	40.800	117.000	999.999							0.000
32.400	85.800	40.800	85.200	107.400	33.600	79.200	56.400	999.999	100.200	
114.600	999.999	79.800	46.800	66.600	84.600	999.999	99.000	15.000	999.999	
36.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999	19.800	999.999	
27.600	999.999	27.600	100.200							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	94.800	999.999	999.999
999.999	999.999	28.200	35.400	999.999	19.800	999.999	999.999	999.999	999.999	999.999
999.999	30.600	999.999	999.999							0.000
81.000	999.999	106.200	131.400	999.999	91.800	113.400	94.800	58.200	999.999	
122.400	103.200	999.999	93.600	111.600	999.999	999.999	105.000	69.600	80.400	
97.200	999.999	999.999	57.600	999.999	38.400	75.600	999.999	79.800	999.999	
55.200	37.800	96.600	999.999							0.000
55.800	81.600	59.400	88.200	110.400	50.400	82.200	69.600	999.999	103.800	
81.000	51.000	75.000	60.000	69.600	84.600	103.800	92.400	999.999	999.999	
49.200	81.600	999.999	84.600	71.400	999.999	999.999	126.600	27.600	61.800	
999.999	999.999	25.800	99.600							0.000
82.200	999.999	92.400	126.000	999.999	88.200	120.600	87.600	49.200	999.999	
94.200	999.999	999.999	95.400	108.000	999.999	999.999	86.400	65.400	72.600	
99.000	999.999	62.400	47.400	999.999	36.600	63.000	999.999	86.400	114.600	
45.600	49.800	97.200	999.999							0.000
999.999	85.800	52.800	85.200	111.600	62.400	101.400	72.000	999.999	117.600	
81.600	28.800	79.800	70.200	82.800	95.400	107.400	82.800	999.999	999.999	

64.800	66.000	999.999	80.400	42.000	999.999	999.999	138.000	54.600	66.000
999.999	999.999	37.800	114.000						0.000
39.600	93.600	48.000	92.400	114.600	41.400	86.400	64.200	999.999	108.000
122.400	999.999	87.000	54.000	73.800	91.800	999.999	106.800	22.200	999.999
43.800	91.800	999.999	59.400	80.400	999.999	999.999	130.800	31.800	103.200
35.400	999.999	35.400	107.400						0.000
58.800	999.999	999.999	999.999	999.999	67.200	999.999	58.800	15.600	999.999
999.999	999.999	999.999	59.400	999.999	999.999	999.999	999.999	999.999	41.400
65.400	999.999	999.999	999.999	999.999	999.999	999.999	32.400	999.999	999.999
58.800	999.999	999.999	999.999						0.000
92.400	999.999	118.800	999.999	999.999	106.800	88.200	58.800	63.000	999.999
119.400	999.999	999.999	76.200	82.800	999.999	999.999	108.600	88.800	55.800
109.200	999.999	82.800	61.200	999.999	56.400	58.200	999.999	97.200	999.999
91.200	47.400	102.000	999.999						0.000
47.400	999.999	61.800	999.999	999.999	999.999	77.400	54.000	37.800	999.999
999.999	999.999	999.999	54.600	432.000	999.999	999.999	999.999	35.400	36.600
58.200	999.999	999.999	999.999	999.999	999.999	999.999	36.000	111.000	999.999
47.400	999.999	999.999	999.999						0.000
999.999	69.600	32.400	77.400	84.000	44.400	66.000	50.400	107.400	93.000
96.000	21.600	58.800	49.200	61.800	71.400	84.600	106.800	999.999	999.999
46.800	58.800	999.999	999.999	43.800	999.999	999.999	117.000	36.600	80.400
999.999	999.999	999.999	90.000						0.000
45.600	999.999	999.999	999.999	999.999	999.999	999.999	44.400	27.000	999.999
999.999	999.999	999.999	46.800	999.999	999.999	999.999	999.999	999.999	24.600
999.999	999.999	999.999	999.999	999.999	999.999	999.999	28.200	999.999	999.999
35.400	999.999	999.999	999.999						0.000
58.800	999.999	145.200	108.000	999.999	67.800	97.200	79.200	57.600	999.999
999.999	999.999	999.999	79.800	96.600	999.999	999.999	999.999	47.400	61.200
70.200	999.999	999.999	999.999	999.999	999.999	999.999	60.600	135.600	999.999
22.800	999.999	999.999	999.999						0.000
999.999	86.400	71.400	999.999	999.999	70.200	999.999	999.999	999.999	999.999
93.000	29.400	75.600	999.999	999.999	90.000	115.800	85.800	999.999	999.999
72.600	81.600	999.999	81.600	63.000	999.999	999.999	999.999	58.200	55.800
999.999	999.999	45.600	999.999						0.000
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
89.400	999.999	999.999	999.999	999.999	999.999	999.999	81.600	999.999	999.999
999.999	999.999	33.000	999.999	999.999	39.000	999.999	999.999	999.999	999.999
999.999	36.000	999.999	999.999						0.000
39.000	73.200	51.000	62.400	97.200	28.800	56.400	59.400	999.999	70.800
999.999	999.999	73.800	41.400	56.400	71.400	82.200	124.200	999.999	999.999
31.200	79.800	999.999	76.200	999.999	999.999	999.999	88.800	15.000	95.400
999.999	999.999	16.200	999.999						0.000
49.800	103.800	63.000	102.600	125.400	51.600	96.600	73.800	999.999	118.800
109.800	999.999	102.000	64.200	84.000	102.000	143.400	93.600	32.400	999.999
54.000	120.000	999.999	55.200	999.999	999.999	999.999	141.000	36.600	96.600
25.200	999.999	45.600	117.600						0.000
51.600	97.800	58.200	99.000	121.200	46.800	91.800	66.000	999.999	107.400
139.200	999.999	91.800	65.400	79.200	98.400	120.600	127.200	30.000	999.999
49.200	96.600	999.999	75.600	999.999	999.999	999.999	137.400	37.200	114.000
43.200	999.999	45.600	112.200						0.000
48.600	999.999	999.999	999.999	999.999	57.000	999.999	64.200	25.800	999.999
999.999	999.999	999.999	49.200	999.999	999.999	999.999	999.999	999.999	31.200
59.400	999.999	999.999	999.999	999.999	999.999	22.200	999.999	999.999	999.999
48.600	999.999	999.999	999.999						0.000
66.000	999.999	80.400	95.400	130.200	74.400	89.400	66.000	999.999	999.999

```

999.999 999.999 999.999 66.600 84.000 999.999 999.999 999.999 54.000 48.600
72.000 999.999 76.800 74.400 999.999 58.800 39.600 999.999 91.800 999.999
66.000 69.600 89.400 999.999
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 38 -----> river barge-loading region 1
C -----> river barge-loading region 2
C .
C .
C -----> river barge-loading region 43

C A 38 by 43 matrix containing the rail costs from each surplus region
C to 43 river ports.

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

```

22.800 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
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
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 26.400

```



```

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
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
44.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999

```

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 38 -----> 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 38 by 20 matrix.

C For example,
C from the 1st surplus region, Aberdeen SD (022) to
C 1st port region, Mobile, AL (701) = 97.200 cents/bushel

```

97.200 999.999 64.200 108.000 999.999 999.999 999.999 999.999 999.999 999.999 49.800
34.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 49.800 34.200
31.800 19.891 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 48.000 62.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999

```

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	55.800	65.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	102.000	84.600	91.200	999.999	999.999	999.999	999.999	999.999
120.000	999.999	75.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
31.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	31.800
117.000	999.999	106.800	999.999	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	999.999	19.200
999.999	999.999	103.200	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
42.000	96.000	106.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	42.000
38.400	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
109.200	999.999	59.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	44.400
30.720	121.200	132.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999	30.720
999.999	999.999	96.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	89.400	105.000	98.400	82.000	89.400	999.999	999.999	999.999	999.999	999.999
999.999	999.999	57.000	94.200	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	105.000	93.600	93.000	96.600	102.600	999.999	999.999	999.999	999.999	999.999
999.999	999.999	81.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
47.900	89.400	100.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	48.000
49.800	60.000	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	43.800	70.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	117.600	100.200	106.200	999.999	999.999	999.999	999.999	999.999
999.999	999.999	54.000	51.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	104.400	77.400	75.600	999.999	999.999	999.999	999.999	999.999
999.999	999.999	132.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	61.300	58.900	114.600	119.400	125.400	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	34.800	37.800	999.999	999.999	999.999	999.999	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	999.999
999.999	55.800	53.400	95.400	108.600	120.600	999.999	999.999	999.999	999.999	999.999
999.999	999.999	75.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	118.200	123.000	115.800	121.800	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	34.200	30.000	999.999	999.999	999.999	999.999	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	999.999
999.999	57.100	56.900	106.200	110.400	118.200	999.999	999.999	999.999	999.999	999.999
999.999	999.999	49.800	78.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	121.200	94.200	91.800	999.999	999.999	999.999	999.999	999.999
999.999	999.999	127.200	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	60.000	59.000	103.800	96.000	102.000	999.999	999.999	999.999	999.999	999.999
999.999	999.999	40.800	61.200	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	113.400	84.000	92.400	999.999	999.999	999.999	999.999	999.999
999.999	999.999	81.600	96.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	108.000	121.800	114.600	120.600	999.999	999.999	999.999	999.999	999.999
999.999	999.999	111.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
38.400	999.999	98.400	124.200	127.800	136.200	999.999	999.999	999.999	999.999	38.400
999.999	999.999	126.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	87.800	87.700	100.200	109.200	121.200	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	999.999	999.999	999.999
999.999	102.000	113.400	124.200	127.800	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	39.600	55.200	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	132.000	105.000	102.600	999.999	999.999	999.999	999.999	999.999
96.600	92.400	92.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	49.200
28.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	49.200	28.800

999.999	999.999	110.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	92.400	121.200	113.400	120.000	999.999	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	999.999
999.999	999.999	999.999	112.800	117.000	123.000	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	12.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	57.000	75.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	138.000	126.000	123.600	126.600	134.400	999.999	999.999	999.999	999.999	999.999
999.999	999.999	86.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	109.200	97.800	112.200	95.400	101.400	999.999	999.999	999.999	999.999	999.999
999.999	999.999	67.200	104.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	121.200	109.800	123.600	115.800	123.600	999.999	999.999	999.999	999.999	999.999
999.999	999.999	100.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
	27.600	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	27.600
999.999	999.999	112.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	90.600	116.400	120.600	999.999	999.999	999.999	999.999	999.999	999.999

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

F4.hrw

C This is the 4th data file data dealing with the barge costs and
C shiprates for Hard Red Winter Wheat.

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) = 0.289 \$/bushel
- C St. Paul, MN (601) to Knoxville (701) = 0.263 \$/bushel
- C St. Paul, MN (601) to Chatanoo (703) = 0.231 \$/bushel
- C St. Paul, MN (601) to Guntersv (710) = 0.213 \$/bushel
- C St. Paul, MN (601) to Florence (713) = 0.195 \$/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
999.000	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
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 11th river region (St. Paul)

C St. Paul (601) to New Orle, LA (702) = 0.185 \$/bushel

C St. Paul (601) to Mobile, AL (701) = 0.397 \$/bushel

C St. Paul (601) to Galvestn, TX (703) = 0.491 \$/bushel

C St. Paul (601) to Chicago, IL (710) = 999 \$/bushel

C St. Paul (601) to Portland, OR (713) = 999 \$/bushel

0.283	0.345	0.398	0.234	999.000	0
999.999	999.999	999.999	999.999	999.999	0
0.229	0.291	0.344	0.180	999.000	0
0.219	0.281	0.334	0.170	999.000	0
0.210	0.272	0.325	0.161	999.000	0
0.182	0.244	0.297	0.133	999.000	0
0.164	0.226	0.280	0.116	999.000	0
0.125	0.187	0.240	0.109	999.000	0
0.334	0.333	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

0.932	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 Number of ports at Great Lakes for export

8

0

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

F8.hrw

C
C

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

C-----

C
C
C

These lines show the amount of surplus produced by each surplus region.
For example, the 1st surplus region,
Aberdeen SD (022) produces 8.731 million/bushel.

8.731	0.674	32.913	31.220	0.100	1.320	0.607	0.341	22.852	20.722
77.314	1.101	0.368	35.113	0.230	8.363	12.766	26.641	25.930	15.155
2.672	68.866	17.454	17.929	23.156	0.146	7.203	5.915	13.858	0.708
23.286	1.154	1.232	120.125	27.958	92.084	0.060	0.300		0.000

C-----

C
C
C

These lines show the amount of corn demanded by the 34 deficit regions.
For example, the 1st deficit region,
Ames IA (085) requires 4.320 million/bushel.

4.320	1.329	0.102	0.433	1.921	22.194	2.004	3.424	0.106	1.683
1.960	7.156	3.803	5.425	1.128	11.902	8.859	33.375	2.937	3.491
2.883	4.261	1.972	2.571	5.993	3.100	1.950	19.316	8.182	5.645
1.254	2.956	0.000	3.330						0.000

C-----

C
C
C
C
C
C

The remaining lines show the amount demanded by 25 foreign regions
(demand per quarter).
For example, the 1st foreign region, Scandinavia,
in the 1st quarter requires 0 million bushel
in the 2nd quarter requires 0 million bushel
in the 3rd quarter requires 0 million bushel
in the 4th quarter requires 0.070 million bushel

0.000	0.000	0.000	0.070						0.000
0.000	0.000	0.000	0.000						0.000
0.000	0.000	0.000	0.000						0.000
0.000	0.000	0.000	0.000						0.000
1.166	5.524	0.000	0.230						0.000
0.000	26.129	109.277	26.948						0.000
8.338	10.438	4.947	12.298						0.000
6.561	5.877	6.736	6.673						0.000
18.539	16.321	14.219	15.254						0.000
0.689	2.908	7.490	1.983						0.000
0.055	0.443	3.256	3.223						0.000
1.794	2.673	1.543	1.482						0.000
16.675	33.190	21.898	12.102						0.000
2.420	2.420	2.420	2.420						0.000
0.257	0.257	0.257	0.257						0.000
2.561	2.879	3.012	3.030						0.000
4.509	5.738	6.568	5.395						0.000

10.590	8.810	9.112	9.454	0.000
0.000	2.639	6.542	10.035	0.000
0.000	0.000	0.000	0.000	0.000
0.391	2.966	0.291	0.363	0.000
0.389	2.969	0.291	0.361	0.000
2.673	5.445	13.328	11.437	0.000
2.077	1.645	3.083	2.167	0.000
1.548	1.101	1.497	1.002	0.000

C

F9.hrw

C This is the last and 6th data file with the location names and code
C numbers of all involved regions. The grain involved is Hard Red Winter
C Wheat.

C-----

C 38 Surplus regions

Code number	Location name	State	
022	Aberdeen	SD	0
111	Alexandria	LA	0
052	Altus	OK	0
061	Amarillo	TX	0
017	Ashley	ND	0
071	Bemidgi	MN	0
015	Bismarck	ND	0
171	Bowling G	KY	0
024	Brookings	SD	0
351	Casper	WY	0
391	Denver	CO	0
014	Dickinson	ND	0
134	Effingham	IL	0
051	Enid	OK	0
06H	Ft Stock	TX	0
323	Glasgow	MT	0
312	Moses Lake	WA	0
322	Havre	MT	0
034	Hastings	NE	0
638	Lewiston	ID	0
321	Lewistown	MT	0
044	Liberal	KS	0
324	Livingston	MT	0
062	Lubbock	TX	0
033	McCook	NE	0
012	Minot	ND	0
325	Miles City	MT	0
021	Mobridge	SD	0
053	Ok City	OK	0
072	Ortonville	MN	0
026	Rapid City	SD	0
411	Roswell	NM	0
713	Salem	OR	0
042	Salina	KS	0
031	Scottsbluf	NE	0
041	StFrancis	KS	0
016	Valley Cty	ND	0
011	Williston	ND	0

C-----

C 34 Deficit regions

Code number	Location name	State
-------------	---------------	-------

085	Ames	IA	0
211	Atlanta	GA	0
098	Cabool	MO	0
154	Cambrid	OH	0
251	Charlty	VA	0
092	Chillico	MO	0
144	Columbus	IN	0
131	De Kalb	IL	0
013	Devils	ND	0
281	Du Bois	PA	0
362	El Centro	CA	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
702	New Orlea	LA	0
331	Pendelton	OR	0
381	Salt Lake	UT	0
06A	San Anton	TX	0
311	Spokane	WA	0
074	St Cloud	MN	0
291	Buffalo	NY	0
043	Topeka	KS	0
401	Tucson	AZ	0
032	Valentine	NE	0
313	Wenatch	WA	0
045	Blank	KS	0
241	Winston	NC	0

C 43 River regions

C	Code number	Location name	State	
	601	ST. PAUL,	MN	0
	602	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	CHATTANO,	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 2D 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	TOLEDD,	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 8831818752.

HARD RED WINTER WHEAT, FULLER, GRANT, TEH, FELLIN

UNIT = 1 (THOUSAND BUSHEL)

SUPPLY 746567.

DEMAND 744474.

700 701 701 14996

1 Network Generator For Grain Shipment Problem

HARD RED WINTER WHEAT, FULLER, GRANT, TEH, FELLIN

1

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME				TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
			1	2	3	4				
S 022 Aberdeen	SD	8731.								
P 703 GALVESTN, TX	R	4067.	3175.	1489.	00.	8731.	0.64200	5605302.	1261630.	
S 111 Alexandria	LA	674.								
D 702 New Orlea	LA T	674.	00.	00.	00.	674.	0.19891	134065.	127669.	
S 052 Altus	OK	32913.								
D 06A San Anton	TX T	1498.	1498.	1498.	1268.	5762.	0.38741	2232236.	1091438.	
P 703 GALVESTN, TX	R	00.	00.	17042.	10109.	27151.	0.48000	13032480.	3923320.	
S 061 Amarillo	TX	31220.								
D 065 Ft Worth	TX R	1789.	1789.	1789.	1789.	7156.	0.30000	2146800.	1226682.	
D 715 Los Ang	CA R	1341.	00.	00.	00.	1341.	0.83400	1118394.	229874.	
D 702 New Orlea	LA R	391.	1065.	1065.	1065.	3586.	0.67200	2409792.	614712.	
P 703 GALVESTN, TX	R	00.	4149.	11920.	3068.	19137.	0.55800	10678445.	2765297.	
S 017 Ashley	ND	100.								
P 720 DULUTH,	BC R	00.	100.	00.	00.	100.	0.31800	31800.	14450.	
S 071 Bemidji	MN	1320.								
R 644 DULUTH,	RV T	00.	1320.	00.	00.	1320.	0.16115	212716.	222710.	
S 015 Bismarck	ND	607.								
D 074 St Cloud	MN R	200.	407.	00.	00.	607.	0.30000	182100.	104052.	
S 171 Bowling G	KY	341.								
D 241 Winston	NC R	341.	00.	00.	00.	341.	0.40800	139128.	58454.	
S 024 Brookings	SD	22852.								
D 073 Mankato	MN R	872.	872.	872.	872.	3488.	0.17400	606912.	597913.	
P 703 GALVESTN, TX	R	1929.	2908.	7490.	7037.	19364.	0.59400	11502217.	2798098.	
S 351 Casper	WY	20722.								
D 715 Los Ang	CA R	614.	00.	00.	3459.	4073.	1.02000	4154460.	698194.	

D 381 Salt Lake UT R	642.	642.	642.	642.	2568.	0.29400	754992.	440207.
D 401 Tucson AZ R	257.	1411.	1411.	1411.	4490.	0.78600	3529140.	769676.
P 715 LONG BEA, CA R	00.	00.	137.	9454.	9591.	0.82000	7864620.	1385900.
S 391 Denver CO	77314.							
D 362 El Centro CA R	490.	490.	490.	490.	1960.	0.82000	1607200.	335983.
P 703 GALVESTN, TX R	26877.	26758.	15705.	6014.	75354.	0.57000	42951780.	10888653.
S 014 Dickinson ND	1101.							
D 074 St Cloud MN R	287.	00.	346.	00.	633.	0.36000	227880.	108509.
P 720 DULUTH, BC R	00.	00.	00.	468.	468.	0.48000	224640.	67626.
S 134 Effingham IL	368.							
D 142 Indy IN T	282.	86.	00.	00.	368.	0.14541	53513.	69707.
S 051 Enid OK	35113.							
P 703 GALVESTN, TX R	00.	10959.	24054.	100.	35113.	0.43800	15379494.	5073829.
S 06H Ft Stock TX	230.							
D 06A San Anton TX T	00.	00.	00.	230.	230.	0.31598	72676.	43567.
S 323 Glasgow MT	8363.							
P 713 PORTLAND, OR R	00.	00.	5277.	3086.	8363.	0.58900	4925807.	1208454.
S 312 Moses Lake WA	12766.							
R 639 PASCO, WA T	4211.	4598.	3957.	00.	12766.	0.01115	142378.	2153880.
S 322 Havre MT	26641.							
P 713 PORTLAND, OR R	1242.	13774.	3886.	7739.	26641.	0.53400	14226295.	3849625.
S 034 Hastings NE	25930.							
D 092 Chillicothe MO T	3503.	3503.	4305.	3503.	14814.	0.30001	4444331.	2806068.
D 035 Lincoln NE T	734.	734.	734.	734.	2936.	0.10975	322230.	556137.
D 043 Topeka KS R	2045.	2045.	2045.	2045.	8180.	0.19800	1619640.	1402216.
S 638 Lewiston ID	15155.							
D 331 Pendelton OR T	493.	493.	493.	493.	1972.	0.14961	295032.	373536.
P 713 PORTLAND, OR R	11232.	1951.	00.	00.	13183.	0.30000	3954900.	1904944.
S 321 Lewistown MT	2672.							
P 712 SEATTLE, WA R	00.	00.	739.	1933.	2672.	0.57100	1525712.	386104.
S 044 Liberal KS	68866.							

P 703 GALVESTN, TX R	658.	00.	60649.	7559.	68866.	0.49800	34295268.	9951137.
S 326 LivingstonMT	17454.							
P 713 PORTLAND, OR R	00.	00.	11495.	5959.	17454.	0.59000	10297860.	2522103.
S 062 Lubbock TX	17929.							
P 703 GALVESTN, TX R	999.	16010.	920.	00.	17929.	0.40800	7315032.	2590741.
S 033 McCook NE	23156.							
D 085 Ames IA R	1080.	1080.	1080.	1080.	4320.	0.39600	1710720.	740534.
D 098 Cabool MO R	25.	25.	25.	25.	100.	0.48000	48000.	17142.
D 092 ChillicotheMO T	2045.	2045.	1243.	2045.	7378.	0.39398	2906820.	1397541.
D 201 Gadsen AL R	950.	950.	950.	950.	3800.	0.87000	3306000.	651396.
D 132 Galesburg IL R	00.	1024.	566.	00.	1590.	0.54000	858600.	272558.
D 142 Indy IN R	00.	00.	00.	101.	101.	0.73800	74538.	17313.
D 096 Moberley MO R	720.	720.	720.	720.	2880.	0.43800	1261440.	493690.
D 241 Winston NC R	491.	832.	832.	832.	2987.	1.07400	3208038.	512032.
S 012 Minot ND	146.							
P 720 DULUTH, BC R	00.	42.	00.	104.	146.	0.38400	56064.	21097.
S 325 Miles CityMT	7203.							
D 131 Dekalb IL R	854.	332.	157.	00.	1343.	0.58800	789684.	230217.
D 311 Spokane WA R	775.	680.	674.	775.	2904.	0.56400	1637856.	497804.
D 313 Wenatch WA R	739.	739.	739.	739.	2956.	0.47400	1401144.	506718.
S 021 Mobridge SD	5915.							
D 131 Dekalb IL R	2.	524.	699.	856.	2081.	0.54000	1123740.	356725.
D 132 Galesburg IL R	1356.	332.	790.	1356.	3834.	0.54600	2093364.	657224.
S 053 Ok City OK	13858.							
P 703 GALVESTN, TX R	3134.	1717.	3057.	5950.	13858.	0.39600	5487768.	2002481.
S 072 OrtonvilleMN	708.							
D 074 St Cloud MN T	00.	80.	141.	487.	708.	0.13597	96270.	134109.
S 023 Rapid CitySD	23286.							
D 032 Valentine NE R	313.	313.	313.	313.	1252.	0.22800	285456.	214618.
R 609 SIOUX CT, IA R	00.	5713.	7836.	6392.	19941.	0.34200	6819822.	3327355.
S 411 Roswell NM	1154.							
D 401 Tucson AZ T	1154.	00.	00.	00.	1154.	0.44097	508883.	218591.
S 713 Salem OR	1232.							

P 713	PORTLAND, OR	T	1232.	00.	00.	00.	1232.	0.04996	61554.	211719.
S 042	Salina	KS	120125.							
D 211	Atlanta	GA R	332.	332.	332.	332.	1328.	0.73200	972096.	227646.
D 154	Cambrid	OH R	108.	108.	108.	108.	432.	0.62400	269568.	74053.
D 251	Charlty	VA R	480.	480.	480.	480.	1920.	0.97200	1866240.	329126.
D 144	Columbus	IN R	501.	501.	501.	501.	2004.	0.56400	1130256.	343526.
D 281	Dubois	PA R	420.	420.	420.	420.	1680.	0.70800	1189440.	287986.
D 142	Indy	IN R	00.	196.	282.	181.	659.	0.56400	371676.	112966.
D 183	Knoxville	TN R	2975.	2975.	2975.	2975.	11900.	0.71400	8496600.	2039898.
D 222	Lake City	FL R	2214.	2214.	2214.	2214.	8856.	0.82200	7279632.	1518096.
D 291	Buffalo	NY R	4829.	3509.	4829.	4829.	17996.	0.88800	15980448.	3084874.
P 703	GALVESTN,	TX R	6229.	17027.	15485.	34609.	73350.	0.57000	41809500.	10599075.
S 031	Scottsbluff	NE	27958.							
D 715	Los Ang	CA R	6388.	8343.	8343.	4884.	27958.	0.93600	26168688.	4792561.
S 041	StFrancis	KS	92084.							
P 703	GALVESTN,	TX R	19422.	31490.	24629.	16543.	92084.	0.67200	61880448.	13306138.
S 016	Valley Cty	ND	60.							
P 720	DULLUTH,	BC R	00.	00.	00.	60.	60.	0.27600	16560.	8670.
S 011	Williston	ND	300.							
D 013	Devils	ND T	26.	26.	26.	26.	104.	0.24268	25239.	19700.
D 311	Spokane	WA R	00.	95.	101.	00.	196.	0.58800	115248.	33598.

1

	SURPLUS REGION	SUPPLY	STORAGE	STORAGE COST
S 022	Aberdeen SD	8731.	4664.	1489. 00. 235919.
S 111	Alexandria LA	674.	00.	00. 00. 00.
S 052	Altus OK	32913.	31415.	29917. 11377. 2795025.
S 061	Amarillo TX	31220.	27699.	20696. 5922. 2086380.
S 017	Ashley ND	100.	100.	00. 00. 3834.
S 071	Bemidji MN	1320.	1320.	00. 00. 50612.
S 015	Bismarck ND	607.	407.	00. 00. 15605.
S 171	Bowling G KY	341.	00.	00. 00. 00.
S 024	Brookings SD	22852.	20051.	16271. 7909. 1700921.
S 351	Casper WY	20722.	19209.	17156. 14966. 1977622.
S 391	Denver CO	77314.	49947.	22699. 6504. 3038898.
S 014	Dickinson ND	1101.	814.	814. 468. 80662.
S 134	Effingham IL	368.	86.	00. 00. 3297.
S 051	Enid OK	35113.	35113.	24154. 100. 2276318.
S 064	Ft Stock TX	230.	230.	230. 230. 26602.
S 323	Glasgow MT	8363.	8363.	8363. 3086. 761589.
S 312	Moses Lake WA	12766.	8555.	3957. 00. 479736.
S 322	Havre MT	26641.	25399.	11625. 7739. 1721211.
S 034	Hastings NE	25930.	19648.	13366. 6282. 1510672.
S 638	Lewiston ID	15155.	3430.	986. 493. 188534.

S 321 Lewistown MT	2672.	2672.	2672.	1933.	280240.
S 044 Liberal KS	68866.	68208.	68208.	7559.	5525092.
S 324 Livingston MT	17454.	17454.	17454.	5959.	1570702.
S 062 Lubbock TX	17929.	16930.	920.	00.	684406.
S 033 McCook NE	23156.	17845.	11169.	5753.	1336685.
S 012 Minot ND	146.	146.	104.	104.	13639.
S 325 Miles City MT	7203.	4835.	3084.	1514.	362640.
S 021 Mobridge SD	5915.	4557.	3701.	2212.	402843.
S 053 Ok City OK	13858.	10724.	9007.	5950.	988434.
S 072 Ortonville MN	708.	708.	628.	487.	70206.
S 023 Rapid City SD	23286.	20880.	14854.	6705.	1631449.
S 411 Roswell NM	1154.	00.	00.	00.	00.
S 713 Salem OR	1232.	00.	00.	00.	00.
S 042 Salina KS	120125.	102037.	74275.	46649.	8578353.
S 031 Scottsbluff NE	27958.	21570.	13227.	4884.	1524548.
S 041 St Francis KS	92084.	72662.	41172.	16543.	5009410.
S 016 Valley City ND	60.	60.	60.	60.	6940.
S 011 Williston ND	300.	274.	153.	26.	17385.

1

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME				TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
			1	2	3	4				
R 601 ST. PAUL, MN			00.	00.	00.	00.				
R 644 DULUTH, RV			00.	1320.	00.	00.				
R 645 BUFFALO, NY B			00.	1320.	00.	00.	1320.	0.27000	356400.	248609.
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.	5713.	7836.	6392.				
P 701 MOBILE, AL B			00.	5713.	7836.	6392.	19941.	0.33300	6640353.	3161646.
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.	1320.	00.	00.				
D 291 Buffalo NY T	00.	1320.	00.	00.	1320.	0.00171	2261.	230472.
R 626 KNOXVILL, TN	00.	00.	00.	00.				
R 627 CHATANO, TN	00.	00.	00.	00.				
R 628 GUNTERS, AL	00.	00.	00.	00.				

R 629 FLORENCE, AL	00.	00.	00.	00.				
R 630 CATOOSA, OK	00.	00.	00.	00.				
R 631 MUSKOGEE, OK	00.	00.	00.	00.				
R 632 PINEBLUFF, AR	00.	00.	00.	00.				
R 633 DES ARC, AR	00.	00.	00.	00.				
R 634 GREENWOOD, MS	00.	00.	00.	00.				
R 635 VICKSBURG, 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	4211.	4598.	3957.	00.				
P 713 PORTLAND, OR B	4211.	4598.	3957.	00.	12766.	0.11300	1442558.	2024049.
R 640 ROOSEVELT, 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 SHIPMENT	UNIT COST	TOTAL HALLING COST	HANDLING COSTS
			1	2	3	4				
P 701 MOBILE,	AL	00.	5713.	7836.	6392.					

F 807 E BLOCK EURO	S	00.	5270.	4580.	3169.	13019.	0.48000	6249120.	736875.
F 811 E AFRICA	S	00.	443.	3256.	3223.	6922.	1.02400	7088129.	391785.
P 702 NEW ORLE, LA		00.	00.	00.	00.				
P 703 GALVESTN, TX		63315.	114193.	182440.	90989.				
F 801 SCANDINAVIA	S	00.	00.	00.	70.	70.	0.49000	34300.	3962.
F 805 ADRIATIC	S	1166.	5524.	00.	230.	6920.	0.46500	3217800.	391672.
F 806 USSR	S	00.	26129.	109277.	26948.	162354.	0.55300	89781760.	9189236.
F 807 E BLOCK EURO	S	8338.	5168.	367.	9129.	23002.	0.49900	11477998.	1301913.
F 808 E MEDITERRAN	S	6561.	5735.	6736.	6041.	25073.	0.50800	12737084.	1419132.
F 809 N AFRICA	S	18539.	16320.	14219.	15254.	64332.	0.49400	31780008.	3641191.
F 810 RED SEA	S	689.	2908.	7490.	1983.	13070.	0.75100	9815570.	739762.
F 811 E AFRICA	S	55.	00.	00.	00.	55.	1.04500	57475.	3113.
F 812 W AFRICA	S	1794.	2673.	1543.	1482.	7492.	0.43500	3259020.	424047.
F 813 PERSIAN GULF	S	16675.	33190.	21898.	12102.	83865.	0.52000	43609800.	4746759.
F 814 W ASIA	S	2420.	2420.	2420.	2420.	9680.	0.52500	5082000.	547888.
F 821 W MEXICO	S	391.	2966.	291.	363.	4011.	0.27700	11111047.	227023.
F 822 E MEXICO	S	389.	2969.	291.	361.	4010.	0.05900	236590.	226966.
F 823 MS AMERICA	S	2673.	5445.	13328.	11437.	32883.	0.27200	8944176.	1861178.
F 824 CENT AMERICA	S	2077.	1645.	3083.	2167.	8972.	0.17700	1588044.	507815.
F 825 CARIBBEAN	S	1548.	1101.	1497.	1002.	5148.	0.22900	1178892.	291377.
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.	00.	00.	00.				
P 712 SEATTLE, WA		00.	00.	739.	1933.				
F 815 SE ASIA	S	00.	00.	257.	00.	257.	0.37500	96375.	14546.
F 816 TAIWAN	S	00.	00.	482.	1933.	2415.	0.29100	702765.	136689.

P 713 PORTLAND, OR		17917.	20323.	24615.	16784.				
F 815 SE ASIA	S	257.	257.	00.	257.	771.	0.38000	292980.	43639.
F 816 TAIWAN	S	2561.	2879.	2530.	1097.	9067.	0.29600	2683832.	513192.
F 817 KOREA	S	4509.	5738.	6568.	5395.	22210.	0.43000	9550300.	1257086.
F 818 JAPAN	S	10590.	8810.	8975.	00.	28375.	0.23300	6611375.	1606025.
F 819 CHINA	S	00.	2639.	6542.	10035.	19216.	0.25200	4842432.	1087626.
P 714 SAN FRAN, CA		00.	00.	00.	00.				
P 715 LONG BEA, CA		00.	00.	137.	9454.				
F 818 JAPAN	S	00.	00.	137.	9454.	9591.	0.22500	2157975.	542851.
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.	142.	00.	632.				
F 808 E MEDITERRAN	S	00.	142.	00.	632.	774.	0.82000	634680.	43808.

1

	DEMAND BY TIME			TOTAL	
	1	2	3	4	SHIPMENT
D 085 Ames IA	1080.	1080.	1080.	1080.	4320.
D 211 Atlanta GA	332.	332.	332.	332.	1328.
D 098 Cabool MO	25.	25.	25.	25.	100.
D 154 Cambrid OH	108.	108.	108.	108.	432.
D 251 Charltv VA	480.	480.	480.	480.	1920.
D 092 ChillothMO	5548.	5548.	5548.	5548.	22192.
D 144 Columbus IN	501.	501.	501.	501.	2004.
D 131 Dekalb IL	856.	856.	856.	856.	3424.
D 013 Devils ND	26.	26.	26.	26.	104.
D 281 Dubois PA	420.	420.	420.	420.	1680.
D 362 El Centro CA	490.	490.	490.	490.	1960.
D 065 Ft Worth TX	1789.	1789.	1789.	1789.	7156.
D 201 Gadsen AL	950.	950.	950.	950.	3800.
D 132 Galesburg IL	1356.	1356.	1356.	1356.	5424.
D 142 Indy IN	282.	282.	282.	282.	1128.
D 183 Knoxville TN	2975.	2975.	2975.	2975.	11900.
D 222 Lake City FL	2214.	2214.	2214.	2214.	8856.
D 715 Los Ang CA	8343.	8343.	8343.	8343.	33372.
D 035 Lincoln NE	734.	734.	734.	734.	2936.
D 073 Markato MN	872.	872.	872.	872.	3488.

D 096 Moberley MO	720.	720.	720.	720.	2880.
D 702 New Orleans LA	1065.	1065.	1065.	1065.	4260.
D 331 Pendleton OR	493.	493.	493.	493.	1972.
D 381 Salt Lake UT	642.	642.	642.	642.	2568.
D 06A San Anton TX	1498.	1498.	1498.	1498.	5992.
D 311 Spokane WA	775.	775.	775.	775.	3100.
D 074 St Cloud MN	487.	487.	487.	487.	1948.
D 291 Buffalo NY	4829.	4829.	4829.	4829.	19316.
D 043 Topeka KS	2045.	2045.	2045.	2045.	8180.
D 401 Tucson AZ	1411.	1411.	1411.	1411.	5644.
D 032 Valentine NE	313.	313.	313.	313.	1252.
D 313 Wenatch WA	739.	739.	739.	739.	2956.
D 045 Blank	00.	00.	00.	00.	00.
D 241 Winston NC	832.	832.	832.	832.	3328.
F 801 SCANDINAVIA	00.	00.	00.	70.	70.
F 802 N.C. EUROPE	00.	00.	00.	00.	00.
F 803 S.W. EUROPE	00.	00.	00.	00.	00.
F 804 ISLANDS	00.	00.	00.	00.	00.
F 805 ADRIATIC	1166.	5524.	00.	230.	6920.
F 806 USSR	00.	26129.	109277.	26948.	162354.
F 807 E BLOCK EURO	8338.	10438.	4947.	12298.	36021.
F 808 E MEDITERRAN	6561.	5877.	6736.	6673.	25847.
F 809 N AFRICA	18539.	16320.	14219.	15254.	64332.
F 810 RED SEA	689.	2908.	7490.	1983.	13070.
F 811 E AFRICA	55.	443.	3256.	3223.	6977.
F 812 W AFRICA	1794.	2673.	1543.	1482.	7492.
F 813 PERSIAN GULF	16675.	33190.	21898.	12102.	83865.
F 814 W ASIA	2420.	2420.	2420.	2420.	9680.
F 815 SE ASIA	257.	257.	257.	257.	1028.
F 816 TAIWAN	2561.	2879.	3012.	3030.	11482.
F 817 KOREA	4509.	5738.	6568.	5395.	22210.
F 818 JAPAN	10590.	8810.	9112.	9454.	37966.
F 819 CHINA	00.	2639.	6542.	10035.	19216.
F 820 CANADA	00.	00.	00.	00.	00.
F 821 W MEXICO	391.	2966.	291.	363.	4011.
F 822 E MEXICO	389.	2969.	291.	361.	4010.
F 823 US AMERICA	2673.	5445.	13328.	11437.	32883.
F 824 CENT AMERICA	2077.	1645.	3083.	2167.	8972.
F 825 CARIBBEAN	1548.	1101.	1497.	1002.	5148.

1

STORAGE COST 46956408.
 TRUCK COST 11510203.
 RAIL COST 400016768.
 BARGE COST 8439311.
 SHIP COST 264821536.
 HANDLING CST 151443776.

GRAIN SHIPPED FROM SURPLUS REGIONS 744474.

GRAIN SHIPPED TO DEFICIT REGIONS 180920.

GRAIN SHIPPED TO FOREIGN REGIONS

563554.

Validation of Red Winter Wheat Model

<u>Port</u>	<u>FGIS Recorded Exports</u>	<u>Model Solution</u>
	Million Bu.	
Mobile	19.4	19.9
New Orleans	0.5	0
Galveston	431.0	450.9
Corpus Christi	16.4	0
Brownsville	0	0
Charleston	0	0
Baltimore	0	0
Toledo	0	0
Saginaw	0	0
Chicago	1.4	0
Duluth	0.3	0
Seattle	2.6	2.7
Portland	79.9	79.6
California	12.7	-
San Francisco	-	-
Long Beach	-	9.6
San Diego	-	-
Baie Comeau	0	-
Toledo	-	0
Saginaw	-	0
Chicago	-	0
Duluth	-	0.8
Total	564.2	563.6

Special adjustments made in the calibration process:

- 1) Exports through Corpus Christi, usually supplied from the the midwest by rail via Houston, at a special rail rate, were considered in the model as part of the exports from Galveston. Rail rates could be adjusted to force 16.4 million bushels to be exported at Corpus Christi.