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

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

GRAIN SORGHUM

SEPTEMBER 1990

IWR REPORT 90-R-6

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:

GRAIN SORGHUM

Prepared By:

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

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

Commissioned By:

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

September 1990 **IWR Report-90-R-6**

Content

page

Introduction 1

Programs

sorg1.f 1
sorg2.f 18
sorg3.f 29

Data files

f1.sorg 57
f2.sorg 60
f3.sorg 69
f4.sorg 82
f8.sorg 90
f9.sorg 93

Output

file06_2.sorg 98
final.out 99

Calibration 109

INTRODUCTION

This report documents the grain sorghum model. Additional reports document corn, soybeans, hard red winter wheat, soft wheat, hard red spring wheat, and durum wheat 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 ----  *                    SORGHUM1.F                                *
C ----  *        Documented on               Sept  1990               *
C ----  *        Run on                      a 386 machine            *
C ----  *        Compiler used               NDP Fortran              *
C ----  *        Grain used                  Sorghum (SRG).           *
C ----  *        Data files used             F1,F2,F3,F4, and F8.     *
C ----  *        Trace file                  U6                       *
C ----  *        Input file for sorghum2.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.srg.
C ----  F1.srg contains information about the model.
C ----  U2 is the unit number for data file, F2.srg.
C ----  F2.srg contains information on TRUCK mileage.
C ----  U3 is the unit number for data file, F3.srg.
C ----  F3.srg contains information on RAIL costs.
C ----  U4 is the unit number for data file,F4.srg.
C ----  F4.srg contains information on BARGE costs.
C ----  U8 is the unit number for data file, F8.srg.
C ----  F8.srg contains information about SUPPLIES and DEMANDS.
C ----  U9 is for data files, F9.srg.
C ----  F9.srg 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, PROG02.


       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, SORGHUM1.F:
```

```
C ----
C ----                    U1,U2,U3,U4,U8,U9 (input files)
C ----                            |
C ----                            |
C ----                            V
C ----                       SORGHUM1.F  (program)
C ----                            |
C ----                           /   \
C ----                          /     \
C ----                         V       V
C ----                    U6 (trace)   U12 (input to SORGHUM2.F)

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

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


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

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


C-----------------------------------------------------------------
C                 =====    WRITER    ======
C ----  Write all nodes, iarcs, jarcs, lowr, uppr and cost
```

```
C ----  to the data file in the unit number U12.

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

C ----  Write N(= number of arcs), NODE(= number of nodes),
C ----  I(= IARC), J(= JARC), K=(= LOWR), L(= UPPER), M(=KOST),
C ----  N(=NODES) to the next program, SORGHUM2.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', I15, 6X, 'DEMAND', I15, / )
  620 FORMAT ( 6X, '??????????????????????????????????????????', /,
     1         6X, 'INFEASIBLE NETWORK. DEMAND EXCEEDS SUPPLY', /,
```

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

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

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

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

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

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

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

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

       READ  (U1,520) TLDS, RLDS, 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 ( NDRE .GT. 0 ) CALL RELVTR
       IF ( NDPE .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, REND, DRND, PEND, FRND, DMND
      COMMON /A4/ SINK, SRCE
      COMMON /B1/ SRGN(65) /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20)
      COMMON /C1/ SDTR(65,65) /C2/ SDRL(65,65)
      COMMON /C3/ SRTR(65,45) /C4/ SRRL(65,45)
      COMMON /C5/ SPTR(65,20) /C6/ SPRL(65,20)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE
      COMMON /E1/ SPLY(65)
      COMMON /F1/ STOR(65) /F2/ SCST(4)
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     SINK, SRCE
      INTEGER     SRGN, DRGN, RIVR, PORT, FRGN
      INTEGER     UPPR
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12

  500 FORMAT ( 20A4 )
  520 FORMAT ( 10F8.3 )
  610 FORMAT (6X 'SUPPLY', F15.0, )
```

```
  820 FORMAT (6X, 'UNIT = 1 (THOUSAND BUSHEL)', /)
  830 FORMAT (6X, 'UNIT = 2 (THOUSAND SHORT TON )', /)
  840 FORMAT (6X, 'UNIT = 3 (THOUSAND METRIC TON)', /)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

      DO 1600 I = 1, NOSR
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

       CALL GENARC ( IA, JA, COST )
 2300  CONTINUE

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

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

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

       IDNT      = IDEN(I)
       CALL SERIAL ( IDNT, NORE, RIVR, NT )
       IF ( NT .EQ. 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.SRG).

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


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


      SUBROUTINE  RELVTR
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /B2/ DRGN(65) /B3/ RIVR(45)
      COMMON /B4/ PORT(20)
      COMMON /C1/ RDTR(65,65) /C2/ RDRL(65,65)
      COMMON /C3/ RRBG(65,45) /C5/ RPBG(65,20)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      COMMON /H1/ IDEN(65), ALFA(65), BETA(65)
      DIMENSION   IDN1(10), IDN2(10), IDN3(25)
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     DRGN, RIVR, PORT, UPPR
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12
      LOGICAL     ICED
  500 FORMAT ( 20A4 )
  510 FORMAT ( 2014 )
  520 FORMAT ( 10F8.3 )

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

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

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

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

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

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

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

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

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

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

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

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

C ----  Read in the number of river locations above the L&D 26
C ----  and their codes.

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

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

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

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

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

C ----  If the truck cost is greater than 9999 then this cost
C ----  is ignored else the total truck cost is calculated
C ----  by taking into account of the respective alfa and beta,
C ----  and the loading (TLOR) and unloading (TRIO) 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 + TRIO ) * 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      = RORL(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      = ION1(M)
      CALL SERIAL ( IONT, 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.SRG).

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

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

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

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

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

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


      SUBROUTINE  PELVTR
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
      COMMON /A2/ NOTP, NOTF, NDAY(4)
```

```
      COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
      COMMON /B4/ PORT(20) /B5/ FRGN(25)
      COMMON /C3/ PFSP(65,45)
      COMMON /D1/ RAIL, TRUCK, BARGE, SHIP
      COMMON /F3/ TLOS, RLOS, TLOR, RLOR, BLOR, SLOP
      COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
      COMMON /G0/ K, OPERATION
      COMMON /G1/ IARC(27500) /G2/ JARC(27500) /G3/ LOWR(27500)
      COMMON /G4/ UPPR(27500) /G5/ KOST(27500)
      DIMENSION   IDN3(16)
      INTEGER     SRND, REND, DRND, PEND, FRND, DMND
      INTEGER     PORT, FRGN, UPPR
      INTEGER U1,U2,U3,U4,U8,U9,U6,U12
      LOGICAL     ICED
  500 FORMAT ( 20A4 )
  510 FORMAT ( 20I4 )
  520 FORMAT ( 10F8.3 )

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

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

C ----  Read in the number of lakes used for export and their
C ----  code names.

      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.SRG).

       COST       = PFSP(NF,NT) * 100.0
       IF ( COST .GE. 999. ) GO TO 1300
       COST       = ( COST * SHIP + SLOP ) * 1000.0
       CALL GENARC ( IA, JA, COST )
       IF ( ICED ) UPPR(K-NOTP+(PERIOD - 1)) = 0
 1300  CONTINUE
 2000  CONTINUE
       RETURN
       END

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

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

       IF ( NODR .LE. 0 ) GO TO 1400

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

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

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

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

```
      IARC(K)    = IA + N
      JARC(K)    = JA + I
      LOWR(K)    = DDND(I) * 250.0
      IEXP       = IEXP + LOWR(K)
 1200 CONTINUE
      K          = K + 1
      IARC(K)    = JA + I
      JARC(K)    = SINK
 1300 CONTINUE

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

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

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

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

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

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

      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 ----  *                          SORGHUM2.F                                 *
C ----  *          DOCUMENTED ON                  : SEPT 1990                *
C ----  *          RUN ON                         : A 386 machine            *
C ----  *          COMPILER USED                  : NDP Fortran              *
C ----  *          GRAIN USED                     : SORGHUM                  *
C ----  *          DATA FILES USED                : U12                      *
C ----  *          TRACE FILE                     : U6                       *
C ----  *          INPUT FILE FOR SORGHUM3        : 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, SORGHUM1.
C ----  This data file contains the relevant information which
C ----  is necessary for SORGHUM2 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, SORGHUM3.


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


      NTIM       = 0
      FSBL       = .TRUE.
      MAXA       = 34000

C ----  Read in the data from the output file(generated by) SORGHUM1.F.
C ----  The data read in will be used in this current program.

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

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

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

      WRITE (U6,630) CSUM

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

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

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


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

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

```
      JWV(J)=II+1
      GO TO 20
 47   II=MIDL(J)
      MIR(II)=LL
      MIDL(J)=II-1
 20   CONTINUE
C     ******************************************************************
C
C     GO - SUPERKILTER
C
      ND=INFIN
C
C     MAIN LODP (100)
C
      NR2=NR*2
      DO 1000 MAIN=1,NR
      MAINM=MAIN+NR
      DO 1000 MODE=1,2
      GO TO(52,53),MODE
 52   II=MAIN
      JZ=MAINM
      GO TO 54
 53   II=MAINM
      JZ=MAIN
 54   IF(NC(II)) 65,55,56
 55   IF(NC(JZ)) 63,990,990
 56   IF(KOS(II)) 63,55,55
C          IS,IT = START,END NODE NOS, JS,JT = ARC,MIRROR ARC NOS
C               FOR ARC NEEDING FLOW INCREASE
C          WANT TO INCREASE FLOW, START LABLEING AT JJ
 63   IS=NA(JZ)
      JS=II
      IT=NA(II)
      JT=JZ
      GO TO 70
C          WANT TO DECREASE FLOW, START LABELING AT II
 65   IT=NA(JZ)
      IS=NA(II)
      JS=JZ
      JT=II
C
C     LABELING PROCEDURE
C
C     ******************************************************************
 70   IPL=1
      IPLL=1
      IPS=0
      NUMS=0
      LABL(IT)=JS
      IWV(IPL)=IT
 84   KLAB=KLAB+1
      GO TO 86
 85   IF(IPS-IPL)86,200,86
 86   IPS=IPS+1
      IA=IWV(IPS)
```

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

```
      NET=NC(JI)
      NETNU=NET+IALPHA
      NC(JI)=NETNU
      IF(KOS(JI)) 128,1271,128
 1271 IF(NET) 1272,1272,128
 1272 IF(NETNU) 128,128,1273
 1273 CALL LEFT(NOW,JI)
 128  IF(JWV(K)-IALPHA) 129,1281,129
 1281 CALL RIGHT(NEXT,IJ)
 129  IF(NEXT-IS) 130,150,130
 130  NOW=NEXT
      GO TO 125
C
C     ERASE LABELS AND GO FOR O-K CHECK
C
 150  DO 155 I=1,IPL
      J=IWV(I)
 155  LABL(J)=0
      GO TO 54
C
C     POTENTIAL CHANGE
C
200   KPOT=KPOT+1
 201  KSET=NUMS
      NEWLAB=0
      NUMS=0
      IMTHRU=0
      MIN=INFIN
      NEW=NONS
      NONS=MAXA+1
      IF(KSET) 204,204,202
 202  IF(NEW-MAXA) 295,295,312
C       NON-S (L,L-) SET RECYCLING FILTER
 295  MAXNEW=MAXA+NEW
      DO 310 L=NEW,MAXA
      K=MAXNEW-L
      KK=JWV(K)
      KKK=NA(KK)
      IF(LABL(KKK)) 310,300,310
 300  NONS=NONS-1
      JWV(NONS)=KK
 310  CONTINUE
C       S-SET RECYCLING FILTER
 312  DO 203 K=1,KSET
      KK=JWV(K)
      KKK=NA(KK)
      IF(LABL(KKK)) 203,2021,203
 2021 IF(KOS(KK)) 2023,2023,2022
 2022 NUMS=NUMS+1
      JWV(NUMS)=KK
      MIN=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,KOS(K) )
      GO TO 210
 206  NONS=NONS-1
      JWV(NONS)=K
 210  CONTINUE
 211  CONTINUE
 2111 IPLL=IPL+1
      IF(NUMS) 212,212,215
  212 FSBL       = .FALSE.
      CALL DUMPO(NR,II)
      PRINT 2125,IS,IT,II
      IF ( .NOT. FSBL ) RETURN
      PRINT 2121,(I,LABL(I),I=1,NN)
      PRINT 2122,(I,IWV(I),I=1,IPL)
      PRINT 2123,(JWV(I),I=NEW,MAXA)
 2121 FORMAT(' LABELS, BY NODE'/(5(I9,'=',I10)) )
 2122 FORMAT(' LABELED NODES (IWV)'/       (10I10))
 2123 FORMAT(' THE SET (L,L-), NON-S'/(10I10))
 2125 FORMAT('0IS=',I5,'    IT=',I5,10X,'INFEASIBLE ARC =',I5)
      RETURN
C     UPDATE RELATIVE COSTS
C
C       UPDATE COST FOR SET S
 215  DO 230 I=1,NUMS
      IJ=JWV(I)
      JI=IJ-NR
      IF(JI) 216,216,217
 216  JI=IJ+NR
 217  KOST=KOS(IJ)-MIN
      KOS(IJ)=KOST
      KOS(JI)=-KOST
      IF(KOST) 230,218,230
 218  IF(NC(IJ)) 230,230,220
 220  NODEB=NA(IJ)
      CALL LEFT(NA(JI),IJ)
      IF(LABL(NODEB)) 230,223,230
 223  LABL(NODEB) =IJ
      IPL=IPL+1
      IWV(IPL)=NODEB
```

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


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

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

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

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

```
C ----  ************************************************************
C ----  *                      SORGHUM3.F                          *
C ----  *        Documented on              SEPT  1990            *
C ----  *        Run on                     a 386 machine         *
C ----  *        Compiler used              NDP Fortran           *
C ----  *        Grain used                 SORGHUM               *
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 SORGHUM2.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.srg.
C ----  F1.srg contains information about the model.
C ----  U2 is the unit number for data file, F2.srg.
C ----  F2.srg contains information on TRUCK mileage.
C ----  U3 is the unit number for data file, F3.srg.
C ----  F3.srg contains information on RAIL costs.
C ----  U4 is the unit number for data file,F4.srg.
C ----  F4.srg contains information on BARGE costs.
C ----  U8 is the unit number for data file, F8.srg.
C ----  F8.srg contains information about SUPPLIES and DEMAND'S.
C ----  U9 is for the unit number for data files, F9.srg.
C ----  F9.srg contains the NAMES of all the regions.
C ----  UF1 is the unit number for the input file produced by SORGHUM2.F.
C ----  It is the output file from SORGHUM2.F and
C ----  is used as an input file for this program.
C ----  FD is the unit number for the final (output) data file.
C ----  This is the final output file which will show all the solutions
C ----  for this model.


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

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

C ----                    U1,U2,U3,U4,U8,U9,UF1 (input files)
C ----                                  |
C ----                                  |
C ----                                  V
C ----                                  |
C ----                          SORGHUM3.F (program)
C ----                                  |
C ----                                  |
C ----                                  V
C ----                                  |
C ----                        FD (output/final file)


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

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

C ----  The output data (from 2nd program) is required to run this
C ----  3rd and final program.
C ----  Read in the processed data in FLOW1_2.SRG with unit number
C ----  UF1. The processed data is the output of SORGHUM2.F.
C ----  N is the total number of data processed in SORGHUM2.F.
C ----  FLOW is an array which contains relevant data required to get
C ----  the final output of this model.
C ----  Invoke subroutine DRIVER which is responsible for the overall
C ----  supervision of this program.


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

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

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

C ----  Declaration of all variables.

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


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


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


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

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


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


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

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


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


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


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

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


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

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


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


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


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

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

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

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

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

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

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

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

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

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

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

      SUBROUTINE  SURPLS ( IPRD )
      DIMENSION   T(4), R(4), STCT(65), S(65,4)
      COMMON /A1/ NOSR, NODR, NORE, NOPE, NDFR
      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)
      CDMMON /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, NORE )
 1600 CONTINUE

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

      COST       = ( COST * TRUCK ) / 100.0

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

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


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

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


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

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

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

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

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

      IF ( RT .LE. 0.0 ) GO TO 2500

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

      WRITE (FD,730) DRGN(M), ( NAM2(M,L), L = 1, 3 ),
     1             ( R(N)/OPERATION, N = 1, NOTF ),
     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 ----  Updata TGRR, the total amount of grain shipped to a
C ----  river region.

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

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

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

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

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

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

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

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

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

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

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

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

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

```
      ENDIF

      COST       = ( COST * TRUCK) / 100.0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

       WRITE (FD,800)

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

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

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


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

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

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

C ----  The layout of the output.

  500 FORMAT ( 20A4 )
  510 FORMAT ( 20I4 )
  520 FORMAT ( 10F8.3 )
  700 FORMAT ( 1H1, //, 9X, 18HORIGIN/DESTN  MODE, 6X, 6HSUPPLY, 14X,
     1          16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
     2          7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4110, 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 1,e,
C ORIGIN/DESTN MODE SUPPLY SHIPMENT TOTAL   UNIT    TOTAL     HANDLING
C                                 SHIPMENT COST HAULING COST  COSTS

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

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

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

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

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

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

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

      COST      = ( COST * TRUCK ) / 100.0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

      IF ( TT .LE. 0.0 ) GO TO 2500
      SUMB       = SUMB + BCST
      SUMH       = SUMH + HCSB

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

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

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

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

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

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

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

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

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

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

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

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

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

 2800 CONTINUE
 2900 CONTINUE
 3000 CONTINUE
      RETURN
      END

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


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

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

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

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

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

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

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

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

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

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

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

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

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

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

C ----  PORT regions ---> by ShiP ----> FOREIGN regions.
C ----  Read in the ship rate from each of the port region
C ----  to all the foreign regions.
C ----  If the cost(COST) is greater than 999 then it is
C ----  ignored else the cost by truck(COST) is calculated.
C ----  Subroutine GENFLO is used to calculate the hauling
C ----  shipping cost(SCST) and the handling cost (HCSS).
C ----  T is the amount of grain in each shipment by time.
C ----  TT is the amount of grain in total shipment.
C ----  SLOP is the ship loading cost.
C ----  SHIP is the factor for ship rate and can be altered (see F1.SRG).

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

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

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

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

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

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

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

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

  1300 CONTINUE
  2000 CONTINUE
       RETURN
       END

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

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

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

C ---- The layout of the output.

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

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

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

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

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

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

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

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

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

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


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

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

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

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

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

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

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

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

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

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

       TCST       = AT * COST * 1000.0
```

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

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

F1.srg for SORGHUM

NOTE : DO NOT run this file. It is for reference only.

This is the first data file out of 6 files.
Stores all the important information of the model.
Addition of 3 new equations in the programs (first and third)
for calculating truck costs.
So alphas and betas are unused in the model.

This is the title used in file06_2.srg to display the intermediate
results.
SORGHUM SHIPMENT, BAIE COMEAU ADDED, FULLER, GRANT, TEH, FELLIN 0

Number of surplus regions = 31
Number of deficit regions = 35
Number of river regions = 43
Number of port regions = 20
Number of foreign regions = 25
31 35 43 20 25 0

Number of periods per year = 3
Number of days in each period = 121
Number of days in each period = 121
Number of days in each period = 123
3 121 121 123 0

Codes for the surplus regions (31)
064 111 052 043 039 06D 171 094 036 049 036 048 391 055 034 066 047 093 044 062
06F 033 021 042 099 041 411 046 401 062 045 0

Codes for the deficit regions (35)
061 085 038 057 06E 098 063 056 362 051 065 068 092 058 133 095 105 067 068 054
193 053 069 096 091 06A 097 361 032 059 381 211 201 06G 402 0

Codes for the river regions (43)
601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620
621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640
641 642 643 0

Codes for the port regions (20)
701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720

Codes for the foreign regions (25)
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

Loading costs by truck, rail, barge and ship in cents per bushel
Truck Load Country Elevation = 10.422
Rail Load Country Elevation = 10.368
Truck Load River Location = 8.940
Rail Load River Location = 9.330
Barge Load River Location = 7.758

```
Ship Load Port Location       =  5.660

  10.422  10.368   8.940   9.330   7.758   5.660                                  0

Unloading costs by truck, rail and barge.
Truck Unload Deficit   =  8.520
Rail  Unload Deficit   =  6.774
Truck Unload River     =  6.450
Rail  Unload River     =  6.318
River Unload Barge     = 11.076
Port  Unload Truck     =  6.763
Port  Unload Rail      =  4.082
Port  Unload Barge     =  8.097

   8.520   6.774   6.450   6.318  11.076   6.763   4.082   8.097                  0

Quantity of storage available in each surplus region (constraints)
999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9
999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9
999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9999999.9
999999.9                                                                        0

Cost, rail, truck, ship, barge multipliers and period
Storage cost factor (cents/bushel/year) = 11.566
Rail  cost multiplier                   =  1.000
Truck cost multipler                    =  1.000
Ship  cost multipler                    =  1.000
Barge cost multiplier                   =  1.000
Blocked period                          =  1st
  11.566   1.000   1.000   1.000   1.000   1.000                                  0

Conversion factor and choice of measurement
The conversion factor                   =  1.000
Choice of measurement                   =  1.000
Conversion factor for short ton is 2000/56 = 35.714
and for metric ton is 2204/56 = 39.357

Choice of measurement can be one of these:
        1.000 is for bushel
        2.000 is for short ton
        3.000 is for metric ton
   1.000   1.000                                                                0

Alphas and Betas of the surplus regions (not used in new programs)
   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   6.339   0.205   6.508   0.208   7.246   0.233
   6.508   0.208   7.247   0.233   6.339   0.205   5.654   0.191   6.508   0.208
   6.508   0.208                                                                0

Codes for barge shipping locations, receiving from surplus regions
 631 999 630 612 610 999 625 613 610 611 628 613 611 630 611 631 630 612 630 631
 999 611 609 613 621 612 999 613 999 999 613                                    0
```

Alphas and Betas of the deficit regions (not used in new programs)

5.866	0.195	5.654	0.191	6.200	0.202	6.200	0.202	6.508	0.208
7.246	0.233	5.654	0.191	6.200	0.202	6.459	0.207	6.200	0.202
5.654	0.191	6.514	0.201	6.200	0.202	6.200	0.202	6.508	0.208
7.246	0.233	6.350	0.212	5.654	0.191	5.654	0.191	6.200	0.202
6.177	0.202	6.200	0.202	5.654	0.191	7.246	0.233	6.165	0.201
5.654	0.191	7.246	0.233	6.459	0.207	6.508	0.208	6.595	0.210
6.200	0.202	7.246	0.233	6.459	0.207	6.508	0.208	6.595	0.210

Codes for barge shipping locations, sending to deficit regions

```
 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
 999 999 999 999 999 999 999 999 999 999 999 627 628 999 999                    0
```

Number of barge shipping points = 5
Number of port shipping points = 5

```
   5   5                                                                       0
```

Codes of the above points

(625) Nashville, TN
(626) Knoxville, TN
(627) Chatanoo, TN
(628) Guntersv, AL
(629) Florence, AL
(702) New Orle, LA
(701) Mobile, AL
(703) Galvestn, TX
(710) Chicago, IL
(713) Portland, OR

```
 625 626 627 628 629 702 701 703 710 713                                       0
```

F2.SRG

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

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

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

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

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

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

C       First part
C       (1) Each surplus region is linked to each of the deficit regions.
C       In this model there are 31 surplus regions and 35 deficit regions.

C       surplus region 1    ----------> deficit region 1
C                           ----------> deficit region 2
C                                              .
C                                              .
C                           ----------> deficit region 35


C       surplus region 2    ----------> deficit region 1
C                           ----------> deficit region 2
C                                              .
C                                              .
C                           ----------> deficit region 35
```

```
C       .
C       .
C       .

C       surplus region 31  ----------> deficit region 1
C                          ----------> deficit region 2
C                                           .
C                                           .
C                          ----------> deficit region 35

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

C       Second part

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

C       surplus region 1   ----------> corresponding river barge loading region
C       surplus region 2   ----------> corresponding river barge loading region
C                                           .
C                                           .
C       surplus region 31  ----------> corresponding river barge loading region

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

C       Third part

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

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

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

C       .
C       .
C       .

C       surplus region 31  ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                           .
C                                           .
C                          ----------> port region 20 for export

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

```
C       Fourth part

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

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

C       barge-unloading location 2 ----------> corresponding deficit region
C       .
C       .
C       .
C       barge-unloading location 31 ----------> corresponding deficit region

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

C       This includes the truck mileage from each of the 31 surplus regions
C       to each of the 35 deficit regions.

C       A 31 by 35 matrix.
C       The first surplus region is Abilene (064).
C       The distances from Abilene to all deficit regions are given on the first
C       4 lines (see below):
C               Abilene (064) to Amarillo (061) = 267 miles
C               Abilene (064) to Ames (085)     = 862 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 661.

 267.000 862.000 679.000 231.000 378.000 647.000 155.000 251.0001049.000 339.000
 152.000 216.000 724.000 307.000 905.000 624.000 497.000 307.000 347.000 339.000
 680.000 285.000 240.000 682.000 653.000  90.000 574.0001543.000 791.000 390.000
1165.0001012.000 828.000 442.000 770.000                                       0
 661.000 857.000 963.000 400.000 544.000 478.000 547.000 561.0001527.000 562.000
 138.000 368.000 687.000 303.000 720.000 530.000 263.000 183.000 157.000 372.000
 264.000 479.000 730.000 543.000 704.000 539.000 482.0002006.0001108.000 548.000
1595.000 596.000 412.000 927.0001248.000                                       0
 157.000 710.000 511.000 155.000 550.000 507.000  67.000  73.0001106.000 184.000
 197.000 360.000 572.000 261.000 765.000 484.000 466.000 343.000 338.000 233.000
 714.000 145.000 394.000 542.000 501.000 268.000 434.0001502.000 641.000 250.000
1091.0001046.000 830.000 555.000 841.000                                       0
 544.000 234.000 321.000 433.000 857.000 298.000 539.000 406.0001451.000 300.000
 546.000 727.000  96.000 432.000 364.000 236.000 441.000 567.000 641.000 368.000
```

731.000 346.0008565.000 260.000 22.000 721.000 222.0001733.000 451.000 277.000
1036.000 925.000 749.000 928.0001211.000 0
545.000 257.000 193.000 471.000 993.000 429.000 555.000 427.0001432.000 324.000
574.000 763.000 202.000 501.000 470.000 367.000 572.000 636.000 710.000 437.000
862.000 374.000 864.000 390.000 131.000 759.000 353.0001602.000 336.000 346.000
905.0001055.000 880.000 909.0001192.000 0
629.000 928.000 971.000 382.000 393.000 611.000 515.000 549.0001450.000 560.000
298.000 245.000 790.000 313.000 868.000 648.000 396.000 192.000 104.000 382.000
407.000 477.000 637.000 676.000 752.000 448.000 600.0001946.0001116.000 468.000
1563.000 739.000 555.000 827.0001171.000 0
982.000 624.000 896.000 720.0001106.000 364.000 910.000 808.0001933.000 743.000
744.000 876.000 516.000 621.000 283.000 420.000 399.000 628.000 702.000 635.000
384.000 716.0001136.000 363.000 580.000 966.000 440.0002304.0001009.000 622.000
1615.000 381.000 257.0001333.0001654.000 0
587.000 284.000 449.000 421.000 900.000 167.000 559.000 274.0001498.000 343.000
501.000 670.000 114.000 375.000 339.000 105.000 310.000 510.000 584.000 308.000
600.000 347.000 857.000 159.000 131.000 722.000 91.0001839.000 577.000 243.000
1166.000 823.000 618.000 975.0001258.000 0
593.000 224.000 115.000 537.0001059.000 530.000 603.000 485.0001461.000 390.000
640.000 829.000 302.000 605.000 531.000 468.000 673.000 740.000 812.000 548.000
963.000 440.000 912.000 490.000 235.000 815.000 454.0001556.000 226.000 461.000
859.0001151.000 981.000 957.0001240.000 0
456.000 346.000 203.000 392.000 914.000 441.000 466.000 343.0001343.000 245.000
495.000 684.000 245.000 460.000 513.000 379.000 584.000 959.000 667.000 403.000
874.000 295.000 775.000 411.000 171.000 670.000 365.0001590.000 348.000 318.000
898.0001079.000 892.000 820.0001103.000 0
1311.0001066.0001349.0001044.0001313.000 790.0001218.0001137.0002230.0001111.000
1041.0001134.000 926.000 945.000 732.000 858.000 723.000 886.000 925.000 959.000
513.0001053.0001433.000 801.0001033.0001263.000 866.0002656.0001458.000 986.000
2068.000 186.000 365.0001630.0001951.000 0
437.000 338.000 355.000 336.000 853.000 309.000 432.000 299.0001344.000 193.000
434.000 623.000 200.000 339.000 460.000 247.000 436.000 474.000 548.000 275.000
742.000 239.000 749.000 302.000 135.000 614.000 233.0001685.000 499.000 184.000
1048.000 965.000 760.000 821.0001104.000 0
462.000 655.000 326.000 743.0001084.000 851.000 576.000 585.0001075.000 595.000
796.000 945.000 634.000 822.000 902.000 789.000 986.000 942.000 987.000 767.000
1284.000 648.000 729.000 820.000 560.000 763.000 775.0001181.000 365.000 716.000
484.0001485.0001302.000 705.000 936.000 0
125.000 670.000 405.000 361.000 725.000 569.000 209.000 203.0001012.000 213.000
429.000 578.000 532.000 440.000 792.000 541.000 604.000 575.000 620.000 385.000
910.000 266.000 444.000 599.000 461.000 404.000 496.0001401.000 596.000 338.000
872.0001185.000 980.000 489.000 772.000 0
514.000 310.000 89.000 509.0001019.000 539.000 525.000 407.0001380.000 346.000
612.000 801.000 312.000 577.000 579.000 477.000 682.000 712.000 784.000 520.000
972.000 412.000 833.000 500.000 241.000 750.000 463.0001503.000 234.000 435.000
806.0001165.000 990.000 878.0001161.000 0
390.000 777.000 746.000 161.000 363.000 552.000 276.000 310.0001230.000 328.000
 56.000 133.000 639.000 188.000 810.000 529.000 366.000 150.000 167.000 231.000
523.000 256.000 422.000 587.000 601.000 229.000 479.0001724.0001329.000 317.000
1324.000 855.000 671.000 627.000 951.000 0
425.000 692.000 454.000 271.000 767.000 217.000 404.000 274.0001369.000 169.000
366.000 534.000 254.000 242.000 472.000 191.000 337.000 377.000 451.000 178.000
650.000 192.000 702.000 249.000 213.000 567.000 141.0001750.000 596.000 87.000
1136.000 873.000 668.000 844.0001127.000 0
707.000 169.000 434.000 592.0001072.000 256.000 702.000 569.0001614.000 463.000

673.000 842.000 78.000 547.000 221.000 209.000 471.000 678.000 756.000 480.000
689.000 509.0001019.000 191.000 144.000 884.000 252.0001854.000 547.000 408.000
703.000 820.000 655.0001091.0001374.000   0
164.000 631.000 336.000 349.000 732.000 538.000 201.000 191.0001051.000 201.000
421.000 570.000 493.000 428.000 753.000 502.000 592.000 563.000 612.000 373.000
898.000 254.000 483.000 560.000 422.000 429.000 465.0001440.000 437.000 326.000
1979.0001173.000 968.000 528.000 811.000   0
120.000 895.000 650.000 296.000 504.000 709.000 147.000 277.000 931.000 382.000
290.000 381.000 757.000 405.000 867.000 686.000 615.000 445.000 488.000 411.000
818.000 347.000 201.000 744.000 686.000 183.000 636.0001378.000 721.000 452.000
1000.0001150.000 954.000 345.000 652.000   0
756.0001212.0001156.000 596.000 142.000 985.000 645.000 718.0001370.000 762.000
491.000 302.0001074.000 623.0001242.000 964.000 770.000 549.000 463.000 666.000
835.000 691.000 562.0001022.0001036.000 457.000 914.0001866.0001281.000 752.000
1604.0001167.000 983.000 747.0001091.000   0
401.000 430.000 142.000 547.000 969.000 629.000 438.000 389.0001267.000 388.000
632.000 802.000 412.000 626.000 680.000 567.000 772.000 761.000 823.000 571.000
1062.000 452.000 720.000 598.000 338.000 666.000 553.0001401.000 200.000 505.000
704.0001263.0001080.000 764.0001048.000   0
737.000 467.000 261.000 834.0001305.000 838.000 774.000 710.0001537.000 686.000
937.0001126.000 610.000 902.000 808.000 776.000 981.0001037.0001109.000 845.000
1271.000 737.0001056.000 798.000 543.0001002.000 762.0001522.000 136.000 760.000
825.0001441.0001278.0001101.0001383.000   0
406.000 397.000 251.000 342.000 864.000 411.000 416.000 293.0001293.000 195.000
445.000 634.000 262.000 410.000 524.000 349.000 538.000 545.000 617.000 353.000
844.000 245.000 725.000 385.000 188.000 620.000 335.0001599.000 396.000 291.000
935.0001054.000 862.000 770.0001053.000   0
781.000 494.000 744.000 565.000 951.000 163.000 737.000 607.0001737.000 542.000
589.000 721.000 364.000 466.000 240.000 228.000 244.000 473.000 551.000 459.000
369.000 525.000 981.000 182.000 428.000 811.000 239.0002121.000 872.000 421.000
1463.000 531.000 344.0001178.0001483.000   0
352.000 521.000 233.000 574.000 964.000 671.000 441.000 416.0001198.000 419.000
659.000 810.000 454.000 653.000 722.000 609.000 806.000 788.000 850.000 598.000
1104.000 479.000 668.000 640.000 380.000 643.000 595.0001354.000 291.000 536.000
664.0001305.0001122.000 710.000 993.000   0
215.000 990.000 732.000 472.000 581.000 833.000 299.000 389.000 766.000 477.000
453.000 510.000 852.000 581.0001091.000 810.000 790.000 608.000 651.000 587.000
981.000 473.000 661.000 868.000 789.000 305.000 760.0001233.000 803.000 719.000
881.0001313.0001129.000 204.000 487.000   0
256.000 587.000 301.000 483.000 868.000 607.000 346.000 325.0001106.000 335.000
566.000 715.000 500.000 562.000 762.000 554.000 726.000 697.000 757.000 507.000
1032.000 388.000 572.000 612.000 426.000 547.000 521.0001374.000 372.000 460.000
739.0001263.0001058.000 614.000 897.000   0
677.0001417.0001152.000 923.000 924.0001295.000 761.000 851.000 304.000 939.000
885.000 903.000 338.0001032.000 516.0001272.0001230.0001040.0001083.0001038.000
1413.000 935.000 527.0001330.0001208.000 726.0001222.000 834.0001223.0001038.000
843.0001745.0001561.000 319.000 65.000   0
605.0001013.000 982.000 397.000 187.000 763.000 493.000 546.0001312.000 564.000
292.000 122.000 875.000 418.0001020.000 765.000 548.000 329.000 241.000 467.000
613.000 492.000 499.000 823.000 837.000 327.000 715.0001808.0001120.000 553.000
1503.000 945.000 761.000 689.0001033.000   0
350.000 425.000 336.0002556.000 778.000 327.000 345.000 212.0001265.000 109.000
359.000 548.000 287.000 324.000 547.000 288.000 453.000 459.000 531.000 267.000
760.000 159.000 669.000 346.000 216.000 534.000 251.0001631.000 481.000 182.000
1018.000 983.000 778.000 742.0001025.000   0

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

C       Explanation for the Second part

C       This links each surplus region (31) to its corresponding river
C       barge-loading locations. Conceptually, a (1 X31) 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 31  --------> corresponding river barge-loading location

C       e.g, surplus region, Abilene (064) to river barge region, Muskogee (631)
C           = 402 miles
C       e.g, surplus region, Altus (052) to river barge region, Catoosa (630)
C           = 264 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 402 would be deleted.

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

 402.000 999.000 264.000  22.000 100.000 999.000  61.000  79.000  85.000 159.000
 319.000 110.000 506.000 352.000 200.000 294.000 103.000 144.000 326.000 474.000
 999.000 274.000 299.000 251.000  30.000 380.000 999.000 414.000 999.000 999.000
 336.000                                                                       0

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

C       Explanation for the Third part

C        (3) Each surplus region(31) 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 31  ----------> port region 1 for export
C                          ----------> port region 2 for export
C                                      .
C                                      .
C                          ----------> port region 20 for export

C       There are 31 surplus regions and 20 port locations thus a (31 by 20)
C       matrix.

C       e.g, surplus region, Abilene(064) to port region, Mobile = 773 miles
C       e.g, surplus region, Abilene(064) to port region, New Orleans = 675 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 Abilene 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.

 773.000 675.000 401.000 396.000 525.0001255.0001539.0001239.0001326.0001072.000
1231.0001922.0001853.0001590.0001216.0001160.0009999.9999999.9999999.9999999.999
 309.000 189.000 229.000 450.000 585.000 839.0001185.000 993.0001092.000 869.000
1226.0002385.0002316.0002060.0001694.0001638.0009999.9999999.9999999.9999999.999
 807.000 713.000 508.000 551.000 680.0001261.0001443.0001099.0001182.000 932.000
1079.0001881.0001812.0001556.0001248.0001217.0009999.9999999.9999999.9999999.999
 864.000 858.000 804.000 923.0001058.0001125.0001080.000 701.000 768.000 508.000
 608.0001792.0001757.0001786.0001586.0001562.0009999.9999999.9999999.9999999.999
 995.0001067.0001791.0001462.0001318.0001255.0001186.000 779.000 825.000 557.000
 583.0001678.0001626.0001655.0001532.0001543.0009999.9999999.9999999.9999999.999
 394.000 272.000  78.000 299.000 434.000 970.0001328.0001144.0001243.0001020.000
1297.0002345.0002276.0001991.0001617.0001571.0009999.9999999.9999999.9999999.999
 498.000 578.000 829.0001036.0001171.000 575.000 693.000 411.000 526.000 393.000
 851.0002348.0002336.0002365.0002073.0002044.0009999.9999999.9999999.9999999.999
 733.000 727.000 747.000 866.0001001.0001032.0001030.000 676.000 743.000 488.000
 653.0001918.0001887.0001903.0001633.0001609.0009999.9999999.9999999.9999999.999
```

```
1096.0001090.000 938.0001025.0001160.0001351.0001190.000 758.000 804.000 536.000
 536.0001567.0001580.0001609.0001521.0001570.0009999.9999999.9999999.9999999.999
1007.000 952.000 793.000 880.0001015.0001279.0001229.000 850.000 914.000 646.000
 672.0001688.0001619.0001648.0001478.0001454.0009999.9999999.9999999.9999999.999
 533.000 677.000 998.0001219.0001354.0009999.9999999.999 676.000 815.000 769.000
1234.0002782.0002789.0002710.0002397.0002341.0009999.9999999.9999999.9999999.999
 875.000 831.000 711.000 819.000 954.0001174.0001157.000 797.000 864.000 609.000
 707.0001838.0001769.0001749.0001479.0001455.0009999.9999999.9999999.9999999.999
1394.0001312.0001107.0001124.0001253.0001685.0001618.0001189.0001235.000 967.000
 965.0001274.0001205.0001234.0001113.0001162.0009999.9999999.9999999.9999999.999
1012.000 932.000 740.000 758.000 887.0001394.0001489.0001129.0001196.000 941.000
1003.0001662.0001593.0001455.0001147.0001123.0009999.9999999.9999999.9999999.999
1105.0001069.000 910.000 997.0001132.0001365.0001271.000 839.000 885.000 617.000
 625.0001575.0001527.0001556.0001447.0001495.0009999.9999999.9999999.9999999.999
 616.000 496.000 260.000 329.000 646.0001098.0001408.0001144.0001231.000 977.000
1146.0002103.0002034.0001771.0001397.0001341.0009999.9999999.9999999.9999999.999
 778.000 734.000 614.000 733.000 868.0001082.0001150.000 806.000 893.000 639.000
 761.0001926.0001857.0001812.0001504.0001480.0009999.9999999.9999999.9999999.999
 822.000 849.000 919.0001038.0001173.0001020.000 937.000 554.000 607.000 349.000
 525.0001886.0001878.0001907.0001749.0001725.0009999.9999999.9999999.9999999.999
1000.000 920.000 732.000 750.000 879.0001382.0001450.0001090.0001157.000 902.000
 964.0001664.0001595.0001494.0001186.0001162.0009999.9999999.9999999.9999999.999
 911.000 817.000 566.000 544.000 673.0001393.0001645.0001301.0001388.0001134.000
1248.0001757.0001688.0001432.0001098.0001042.0009999.9999999.9999999.9999999.999
 822.000 700.000 379.000 152.0009999.9991398.0001756.0001549.0001648.0001410.000
1581.0002361.0002292.0001911.0001537.0001481.0009999.9999999.9999999.9999999.999
1195.0001118.000 943.000 987.0001116.0001463.0001395.000 963.0001009.000 741.000
 745.0001494.0001425.0001454.0001319.0001368.0009999.9999999.9999999.9999999.999
1404.0001394.0001235.0001322.0001452.0001640.0001431.000 995.0001040.000 763.000
 505.0001283.0001353.0001575.0001524.0001573.0009999.9999999.9999999.9999999.999
 977.000 902.000 743.000 830.000 965.0001254.0001224.000 861.000 928.000 673.000
 723.0001725.0001656.0001663.0001428.0001404.0009999.9999999.9999999.9999999.999
 502.000 532.000 710.000 885.0001020.000 731.000 861.000 535.000 634.000 398.000
 800.0002212.0002184.0002180.0001872.0001848.0009999.9999999.9999999.9999999.999
1225.0001145.000 970.000 990.0001119.0001505.0001438.0001054.0001100.000 832.000
 836.0001454.0001385.0001414.0001242.0001291.0009999.9999999.9999999.9999999.999
1074.000 980.000 714.000 668.000 781.0001556.0001769.0001425.0001512.0001258.000
1330.0001638.0001569.0001287.000 987.000 877.0009999.9999999.9999999.9999999.999
1134.0001054.000 877.000 895.0001024.0001472.0001462.0001099.0001166.000 898.000
1226.0001529.0001460.0001438.0001252.0001221.0009999.9999999.9999999.9999999.999
1506.0001412.000 110.0001020.0001124.0001988.0002231.0001876.0001943.0001688.000
1750.0001553.0001382.000 879.0001414.000 415.0009999.9999999.9999999.9999999.999
 600.000 478.000 159.000  93.000 228.0001176.0001534.0001327.0001426.0001188.000
1382.0002260.0002191.0001853.0001479.0001423.0009999.9999999.9999999.9999999.999
 893.000 816.000 657.000 744.000 879.0001192.0001244.000 884.000 951.000 696.000
 794.0001808.0001739.0001695.0001400.0001376.0009999.9999999.9999999.9999999.999

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 35 deficit regions, thus a 1 by 35 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 31     ----------> corresponding deficit region



C       e.g, barge-unloading river location, Muskogee(631) to
C            deficit region Amarillo (061) = 392 miles
C       e.g, barge-unloading river location, Omaha (610) to
C            deficit region Ames (085)      = 163 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.

 392.000 163.000 210.000 178.000 999.000 193.000 317.000 187.000 999.000 125.000
 999.000 999.000  74.000 132.000 120.000  90.000  43.000 161.000 202.000  65.000
 135.000 105.000 999.000 104.000  20.000 999.000 185.000 999.000 237.000  10.000
 999.000 120.000  95.000 999.000 999.000                                       0

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

```
C                                F3.SRG

C       F3.SRG is the third file in this model and includes linking rail costs.
C       Four sets of rail costs are estimated. These include:
C       (1) costs from each surplus region to each deficit region
C       (2) costs from each surplus region to all barge-loading river locations
C       (3) costs from each surplus region to all ports and
C       (4) costs from each river barge-unloading location to all deficit regions.

C       Diagrammatically,
C
C                                    (1)
C                                  -------> deficit regions
C                                 |
C                                 |  (2)
C       Surplus regions ---------> -------> river barge-loading regions
C                                 |
C                                 |  (3)
C                                  -------> port regions

C                                          (4)
C       River barge loading regions -------------------> deficit regions
C

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

C       The format used is :
C       8 fields i.e. dddd.ddd where d is a digit.
C       If there is no digit in front of the number, then add spaces.
C       e.g. 30.443 becomes **30.443 where * is a space.

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

C       Explanation for the first part

C       This part shows the rail costs from each surplus region to
C       each of the 35 deficit regions. There are 31 surplus regions
C       35 deficit regions, thus a 31 by 35 matrix.

C       See below,
C       e.g.  from Abilene (064) to Amarillo (061) = 38.640 cents/bushel
C       e.g.  from Abilene (064) to Ames (085)     = 71.120 cents/bushel

C       If a new surplus region were added, then rail costs to all
C       deficit regions must be added at the end of this data section.

C       If a new surplus region were added at position 10 of the
C       matrix, then all associated rail costs would be added in the
C       10th matrix.

C       If a surplus region were deleted, then all its rail costs would
C       be removed.
```

```
C       If Abilene were no longer a surplus region, then the first 4 lines
C       would be removed. Then the first entry would be 59.920.

C       If a new deficit region were added then the rail cost to this region
C       would depend on the position of the deficit region in the list of
C       codes of the deficit regions in the first file, F1.SRG.
C       If the new deficit region were in position 9, then a new value would
C       be added to position 9.

C       If the new region were added to the end of the list, then the value
C       would be added to the end.
C       Similarly for deletion, if a deficit region were removed from the
C       list then find out the position of this region and delete the value
C       from that position.
```

```
  38.640   71.120   72.800   32.480   49.840   57.680   30.800   31.360   87.920   39.760
  22.400   31.920   65.520   35.280   78.400   62.720   61.040   30.800   39.760   37.520
  54.320   36.960   26.880   66.640   61.600   21.840   59.360  107.520   92.400   43.120
 103.040   78.960   62.160   39.200   65.520                                        0.000
  59.920   80.080   92.960   44.240   56.000   51.520   49.280   52.640  113.120   52.080
  33.040   47.040   79.520   42.000   59.360   54.880   29.680   24.080   29.680   41.440
  34.160   48.160   56.000   58.800   69.440   51.520   53.200  137.200   98.000   51.520
 121.520   57.120   40.320   67.760   99.120                                        0.000
  30.800   61.040   48.160   30.800   56.560   43.680   20.720   16.800   98.000   26.320
  28.560   44.240   51.520   36.960   62.720   42.000   41.440   37.520   42.560   31.920
  62.720   21.280   36.400   25.200   44.240   21.280   54.880  113.120   63.840   28.000
  95.760   85.120   68.320   48.720   76.160                                        0.000
  47.600   30.800   32.480   37.520   71.680   30.240   55.440   39.200  112.000   34.160
  43.680   58.240   18.480   48.160   42.000   29.120   58.240   63.280   68.320   43.120
  53.760   32.480   72.240   33.040   14.000   61.040   26.320  137.760   45.920   31.360
  99.680   73.360   60.480   73.920   95.200                                        0.000
  62.160   32.480   28.560   47.040   81.760   39.760   73.360   48.720  138.880   39.200
  53.760   68.320   24.640   57.120   47.600   38.640   58.240   66.080   71.120   48.160
  70.560   41.440   89.040   42.000   20.720   82.880   35.280  148.960   36.960   47.600
  88.480   80.640   64.960  101.360  122.640                                        0.000
  50.960   92.400   87.920   40.320   41.440   62.160   50.960   53.760  110.320   50.960
  34.720   30.240   78.960   50.400   74.480   61.040   39.760   24.080   19.600   56.560
  43.120   45.920   55.440   64.960   76.720   28.560   57.680  140.560   99.680   57.120
 124.320   68.320   51.520   64.400   91.840                                        0.000
  76.160   62.160   76.720   60.480   87.920   42.000   75.600   66.080  145.040   62.160
  62.160   74.480   59.920   54.880   33.040   48.160   36.400   56.560   58.240   63.280
  36.400   56.560   89.600   52.080   56.560   77.280   44.800  158.480   81.200   55.440
 130.480   36.400   24.640  101.920  123.200                                        0.000
  50.400   41.440   46.480   42.560   73.920   31.920   56.560   43.120  123.200   37.520
  45.920   60.480   34.720   40.880   37.520   30.800   38.640   50.400   59.360   39.760
  58.240   37.520   73.360   34.720   30.800   66.080   27.440  132.720   57.680   30.800
 105.280   71.680   52.080   85.680  115.920                                        0.000
  66.640   34.160   24.080   52.640   87.360   44.240   68.880   57.120  144.000   43.680
  59.920   73.920   29.680   63.280   53.200   42.560   63.840   72.240   76.720   54.320
  78.400   47.040   94.080   46.480   25.760   88.480   39.200  148.960   30.800   51.520
  81.760   84.560   70.000  107.520  129.360                                        0.000
  43.680   38.640   28.000   37.520   71.680   44.800   44.800   39.200  123.760   29.680
  43.680   58.240   31.920   48.160   45.920   43.120   56.560   56.560   65.520   43.120
  74.480   32.480   70.000   47.040   22.960   58.800   40.320  127.680   45.920   37.520
  90.160   89.040   67.200   82.320  103.040                                        0.000
```

101.360	93.000	104.160	89.600	94.080	67.200	87.920	86.800	150.640	88.480
78.400	77.280	81.760	83.440	68.880	74.480	61.040	68.880	74.480	87.920
42.560	81.200	99.120	78.400	86.240	96.880	71.680	183.120	116.480	84.000
156.240	24.080	34.720	112.560	137.200					0.000
42.000	36.400	45.920	31.920	65.520	35.840	48.720	33.600	106.400	28.000
37.520	52.080	28.560	42.560	46.480	34.720	49.280	50.400	59.360	37.520
67.760	26.320	66.080	38.640	25.200	55.440	31.360	120.960	67.200	35.840
90.160	91.280	76.000	68.320	89.600					0.000
44.800	64.400	51.520	66.640	90.720	81.760	51.520	61.040	99.120	56.000
64.400	78.960	57.680	76.160	64.400	80.080	84.560	76.160	85.680	67.760
99.680	64.400	77.840	84.000	53.760	64.960	77.280	114.240	51.520	67.200
62.720	126.000	107.520	65.520	90.160					0.000
24.080	57.680	56.000	49.280	90.160	56.560	30.240	38.080	82.880	39.760
48.160	76.720	52.080	55.440	75.600	54.880	63.840	59.920	64.960	50.400
71.120	43.680	50.400	57.680	46.480	39.760	52.080	104.160	69.440	46.480
94.080	95.760	78.960	45.360	70.560					0.000
58.800	35.280	20.160	47.040	80.640	44.800	64.400	44.240	126.000	39.200
53.760	67.200	29.680	58.800	50.400	43.680	63.840	65.520	64.960	48.160
80.080	41.440	81.200	47.600	26.320	74.480	40.320	140.560	35.840	43.120
77.840	86.240	70.560	88.480	110.320					0.000
39.200	84.560	64.960	26.880	38.080	58.800	32.480	41.440	102.480	36.960
16.240	26.880	63.280	28.000	74.000	57.120	40.880	28.560	31.360	36.960
50.960	32.480	52.640	61.040	62.160	36.400	54.320	133.280	77.840	43.120
108.080	75.600	58.240	64.400	86.240					0.000
48.720	42.560	52.080	34.720	67.760	31.360	45.360	35.280	104.160	29.680
42.560	54.320	35.840	35.840	42.560	29.680	48.160	51.520	57.120	29.120
58.240	29.680	75.040	31.920	27.440	64.960	26.880	127.120	78.400	18.480
95.760	71.120	56.560	65.520	90.720					0.000
73.360	27.440	44.240	63.840	94.080	47.040	79.520	63.840	132.720	58.800
70.560	79.520	23.520	68.880	35.840	45.360	49.280	70.000	75.600	55.440
77.280	53.760	91.840	49.280	27.440	84.560	42.560	170.800	56.560	47.040
103.040	86.800	69.440	98.560	120.400					0.000
26.320	57.120	60.480	43.120	92.960	62.720	35.840	40.320	93.520	33.040
49.840	65.520	49.280	53.200	67.200	61.600	60.480	61.600	75.040	48.160
86.240	37.520	55.440	65.520	45.360	44.800	58.240	122.080	77.280	43.680
91.840	98.000	82.880	47.600	76.720					0.000
19.040	73.360	71.680	44.800	52.080	61.040	21.280	35.840	85.120	41.440
35.280	38.640	62.160	50.960	80.080	59.360	57.120	44.240	53.200	44.800
57.680	38.640	35.280	66.640	58.240	24.640	56.560	120.400	85.680	49.280
89.040	82.880	65.520	47.600	72.800					0.000
70.560	106.400	107.520	58.240	31.920	86.800	65.520	89.040	106.400	71.680
47.600	38.640	99.680	57.680	106.400	85.680	65.520	52.080	42.560	63.840
68.320	90.160	68.880	89.600	90.160	58.240	82.320	146.720	120.960	70.000
140.000	96.880	80.640	68.880	90.720					0.000
46.480	38.080	22.400	48.160	86.800	43.120	52.640	42.560	119.840	36.960
54.880	67.200	34.720	62.160	50.400	38.640	52.080	58.240	64.400	45.920
48.160	42.560	72.240	42.560	31.360	57.120	35.280	109.760	26.320	43.680
61.600	60.480	64.960	66.080	91.840					0.000
76.160	42.000	39.760	71.680	94.640	60.480	73.920	68.320	125.440	59.360
77.840	85.120	39.760	76.720	61.600	58.800	71.680	83.440	90.160	70.560
81.200	66.080	95.760	62.720	36.400	82.880	56.000	118.160	32.480	63.280
77.280	96.320	87.360	100.240	108.080					0.000
42.560	42.560	31.920	38.080	68.880	43.680	39.760	32.480	99.120	26.880
44.240	56.000	43.120	39.200	46.480	42.560	53.760	45.920	52.640	36.400
68.880	32.480	64.400	46.480	24.080	50.400	39.200	122.080	39.200	30.240

```
  75.600   76.160   65.520   62.160   83.440                                    0.000
  60.480   52.080   57.680   55.440   63.840   26.320   56.560   54.880  123.200  50.960
  43.120   68.320   33.040   46.480   27.440   22.960   21.840   34.160   43.680  33.600
  36.960   35.280   66.080   20.160   36.960   59.360   25.200  150.640   64.960  36.960
 101.360   44.240   27.440   78.400  101.920                                    0.000
  46.480   45.920   29.680   55.440   85.120   52.080   52.640   48.720  104.160  44.800
  61.600   73.920   37.520   54.880   57.680   50.960   72.800   66.640   72.800  54.880
  80.640   49.840   73.360   54.880   36.400   69.440   47.600  115.360   35.840  54.320
  69.440   92.960   78.960   72.800   94.080                                    0.000
  20.720   70.560   68.880   48.720   59.360   64.400   30.800   33.600   77.840  37.520
  42.560   49.840   60.480   53.760   73.360   63.280   63.280   51.520   56.560  56.560
  87.360   42.560   18.480   67.200   59.920   31.920   59.920   93.520   75.600  49.280
  78.960  108.640   95.200   25.200   50.960                                    0.000
  49.840   58.240   61.040   50.400   84.000   63.840   68.320   52.080  115.360  47.040
  56.000   70.560   54.880   61.600   58.240   62.720   53.760   68.880   78.400  56.560
  81.760   45.360   85.120   66.640   46.480   73.920   91.840  106.960   85.120  52.080
  76.160   97.440   81.760   70.000   99.680                                    0.000
  56.560  102.480  106.400   79.520   78.400   98.000   66.640   65.520   33.040  73.360
  70.000   73.920   92.960   81.200  105.840   96.880   92.400   78.400   84.560  84.560
 105.840   72.240   47.040  100.800   92.400   69.440   93.520   71.680  113.120  82.880
  94.640  127.680  114.240   30.800   16.800                                    0.000
  54.880   96.320   80.640   42.000   29.680   67.200   46.480   60.480   94.640  52.640
  31.920   22.960   86.800   42.000   79.520   66.080   52.640   36.400   26.880  51.520
  51.520   47.600   62.160   70.000   85.680   42.560   62.720   90.720   95.200  47.600
 120.960   76.160   59.920   56.560   78.400                                    0.000
  43.680   42.000   46.480   28.560   61.600   41.440   45.360   30.240  109.760  24.640
  34.160   48.720   34.160   38.640   52.080   40.320   48.160   47.040   56.000  34.160
  62.160   22.960   62.720   44.240   30.240   48.720   37.520  124.880   67.760  29.680
  94.080   75.600   51.520   63.840   92.960                                    0.000

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

C       Explanation for the second part

C       This links  each surplus region (31) to each river barge-unloading(43)
C       locations, thus a (31 by 43) matrix.

C       e.g. from Abilene (064) to Muskogee (631) = 41.440 cents/bushel

C       If a new surplus region were added, then its rail costs to all
C       river regions would be added to the end of this data section.

C       If a new surplus region were added to the position 10 of the
C       matrix, then all its rail costs would be in the 10th matrix.

C       If a surplus region were deleted, then all its rail costs must
C       be removed.

C       If Abilene were removed as a surplus region, then the first 4 lines
C       would be removed. Then the first entry would be 999.999.

C       If a new river barge-unloading location were added then the rail cost
C       to this region would depend on the position of the
C       river barge-unloading location in the list of codes representing
C       the barge-unloading locations. See the first file, F1.SRG.
```

```
C       If the new surplus region were in position 9, then a new value would
C       be added to position 9.
C       If the new region were added to the end of the list, then the value
C       would be added to the end.

C       Similarly for deletion, if a surplus region were removed from the
C       list then identify the position of this region. Then delete the values
C       from that position.

 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
  41.440 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999  13.440 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999  16.240
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999  13.44  999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999  23.520 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 055.000 999.999 999.999  17.920
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999                                                       0
```

999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 24.640 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 025.257
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 39.760 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 19.600 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 47.040 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 48.160
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 21.840 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 31.920 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 18.480
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 26.320 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 42.560
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 42.560 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999   0

```
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 29.680 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999  40.320 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 023.520 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
021.280 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 036.400 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
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                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999  38.080 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999                                                       0
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999  25.200 999.999 999.999 999.999 999.999 999.999 999.999 999.999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999 999
999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999 999
999.999 999.999 999.999                                                       0
```

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

C       Explanation for the third part

C       This part shows the rail costs from each surplus region(31) to
C       all 20 port regions, thus a 31 by 20 matrix.

C       e.g. from Abilene (064) to Mobile (701) = 61.600 cents/bushel

C       If a new surplus region were added, then its rail costs to all
C       port regions would be added to the end of this data section.

C       If a new surplus region were added to the position 10 of the
C       matrix, then all its rail costs would be in the 10th matrix.

C       If a surplus region were deleted, then all its rail costs must
C       be removed.

C       If Abilene were removed as a surplus region, then the first 2 lines
C       (below) would be removed. Then the first entry would be 44.240

C       If a new port region were added then the rail cost to this region
C       would depend on the position of the port region in the list of
C       codes representing the port shown in the first file, F1.SRG.
C       If the new port region were in position 9, then a new value would
C       be added to position 9. If the new region were added to the end of
C       the list, then the value would be added to the end.

C       Similarly for deletion, if a port region were removed from the
C       list then determine the position of this region and delete the value
C       from that position.

  61.600  52.640  42.000  53.200  61.040  99.120 123.760 100.800 110.320  85.120
 104.160 163.520 165.760 112.000  90.160  88.480 999.999 999.999 999.999 999.999
  44.240  24.080  29.120  40.320  48.160  69.440 108.640  85.120  95.200  70.000
  98.000 173.600 175.840 141.680 125.440 117.600 999.999 999.999 999.999 999.999
  72.800  56.560  44.800  59.920  67.760  94.640 106.400  83.440  92.960  71.120
  80.640 136.080 138.880 111.440  95.760  87.920 999.999 999.999 999.999 999.999
  48.200  51.500  47.600  62.160  68.320  84.560  85.680  62.160  72.240  46.480
  49.280 109.200 111.440 117.040 110.880 116.480 999.999 999.999 999.999 999.999
  63.280  57.680  60.000  66.640  73.360  88.480  91.280  62.160  64.420  41.440
  48.160 105.280 108.080 127.120 119.840 125.440 999.999 999.999 999.999 999.999
  38.080  29.120  21.280  32.480  40.320  74.480  94.640  87.360  99.680  78.400
 112.560 175.280 177.520 133.280 112.560 110.880 999.999 999.999 999.999 999.999
  39.760  47.040  76.160  87.360  95.200  48.720  60.480  36.960  47.040  38.640
  71.680 162.400 164.640 168.000 161.280 166.880 999.999 999.999 999.999 999.999
  62.720  61.040  64.400  75.600  83.440  77.280  81.200  52.080  67.760  42.560
  65.520 142.240 144.480 140.000 132.720 138.880 999.999 999.999 999.999 999.999
  57.680  64.400  63.840  73.360  79.520  99.680  99.120  62.720  53.200  42.560
  44.240 104.720 106.960 115.360 108.080 113.680 999.999 999.999 999.999 999.999
  63.280  58.240  56.560  65.520  71.680  88.480  92.400  62.720  70.000  46.480
  48.160 108.640 100.240 117.040 109.760 115.360 999.999 999.999 999.999 999.999
  47.600  54.880  84.000  95.200 103.040 999.999  62.720  62.720  71.120  64.960
 105.280 192.640 195.440 201.040 194.320 199.920 999.999 999.999 999.999 999.999
```

```
 59.920  57.680  44.800  53.760  59.920  86.240  93.520  58.800  62.720  39.200
 57.680 120.400 110.320 120.960 104.720 110.320 999.999 999.999 999.999 999.999
 89.040  92.400  71.120  83.440  90.720 117.040 120.960  87.360  91.280  67.760
 69.440  88.480  78.400  94.640  87.360  92.960 999.999 999.999 999.999 999.999
 87.920  96.880  62.160  73.360  81.200 108.640 113.120  89.600  99.680  67.760
 86.240 135.520 138.320 114.800  89.600  87.360 999.999 999.999 999.999 999.999
 57.680  62.720  60.480  69.440  76.160  93.520  97.440  63.280  67.200  48.720
 45.360  99.120 101.360 111.440 104.720 110.320 999.999 999.999 999.999 999.999
 55.440  46.480  29.120  40.320  48.160  91.840 117.040  87.920  97.440  78.400
 97.440 156.800 159.600 128.240 103.040 101.360 999.999 999.999 999.999 999.999
 53.760  42.560  42.000  50.960  57.120  87.920  91.840  58.240  65.520  42.000
 65.520 134.960 125.440 130.480 114.240 119.840 999.999 999.999 999.999 999.999
 61.040  72.240  77.280  88.480  96.320  71.120  79.520  46.480  59.920  36.960
 59.360 128.800 131.600 145.040 138.320 143.920 999.999 999.999 999.999 999.999
 75.040  70.560  49.280  59.920  66.640 101.360 105.280  76.160  91.840  59.920
 78.400 123.760 126.000 113.120  87.920  85.680 999.999 999.999 999.999 999.999
 89.040  66.080  46.480  61.040  68.880 118.160 122.640  93.520 109.200  87.360
 97.440 143.360 145.600  99.680  83.440  89.040 999.999 999.999 999.999 999.999
 66.080  56.560  34.160  22.400 999.999 101.920 119.840 113.120 128.800 104.160
134.400 195.440 184.800 142.800 117.600 115.920 999.999 999.999 999.999 999.999
 74.480  80.640  78.400  89.040  96.020 102.480 106.400  71.120  75.040  51.520
 53.760  93.000  91.000 113.680 106.960 112.560 999.999 999.999 999.999 999.999
 87.360 103.600 101.360 112.000 119.280 113.680 104.720  70.000  82.320  50.400
 44.240  82.000  80.000 126.560 119.280 124.880 999.999 999.999 999.999 999.999
 60.480  59.360  57.120  66.080  72.240  89.040  92.960  63.840  79.520  47.600
 53.200 115.920 105.840 123.200 116.480 122.080 999.999 999.999 999.999 999.999
 35.280  50.400  55.440  66.640  74.480  59.360  76.160  47.040  62.720  39.200
 68.320 161.840 151.200 160.720 153.440 159.600 999.999 999.999 999.999 999.999
 74.480  75.600  73.360  82.320  88.480 112.000 115.920  77.840  91.280  58.800
 59.920 113.120 115.360 121.520 114.240 119.840 999.999 999.999 999.999 999.999
 86.240  77.840  58.240  72.800  80.640 123.200 145.600 101.360 105.280  82.880
104.720 146.720 148.960 100.240  84.000  89.600 999.999 999.999 999.999 999.999
 80.080  72.240  70.000  78.960  85.120 104.720 108.640  79.520  81.760  63.280
 71.680 106.400  96.320 113.120 105.840 111.440 999.999 999.999 999.999 999.999
107.520  98.560  81.200  78.400  86.800 143.920 164.080 130.480 146.160 121.520
132.160 123.200 109.200  68.320  43.120  41.440 999.999 999.999 999.999 999.999
 50.400  40.880  20.720  18.480  26.320  86.240 109.200 100.240 115.920  90.720
114.800 183.680 169.120 128.800 103.040 101.360 999.999 999.999 999.999 999.999
 57.120  54.880  49.840  58.800  64.960  91.280  95.200  64.400  68.320  44.800
 71.120 122.640 112.560 129.920 122.640 128.800 999.999 999.999 999.999 999.999

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

C       Explenation for the fourth part

C       This shows rail costs linking each river barge-unloading locations(43)
C       to each of the 35 deficit regions, thus a 43 by 35 matrix.

C       e.g.  from St. Paul (601) to Amarillo (061) = 999.999 cents/bushel
C       e.g.  from St. Louis (608) to Decatur (133) = 19.040 cents/bushel

C       If a new river barge-loading region were added, then its rail costs to all
C       deficit regions must added at the end of this data section.

C       If a new river barge-loading region were added to the position 10 of the
```

```
C        matrix, then all its rail costs would be in the 10th matrix.

C        If a river barge-loading region were deleted, then all associated rail costs m
C        be removed.

C        If St. Paul were no longer a river barge-unloading location,
C        then the first 4 lines (below) would be removed.

C        If a new deficit region were added then the rail cost to this region
C        would depend on the position of the deficit region in the list of
C        codes representing the deficit regions in file, F1.SRG.

C        If the new deficit region were in position 9, then a new value would
C        be added to position 9. If the new region were added to the end of
C        the list, then the value would be added to the end.

C        Similarly for deletion, if a deficit region were removed from the
C        list then find out the position of this region and delete the value
C        from that position.

 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999  19.040 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999  19.600 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                       0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999  36.960 999.999
```

999.999	999.999	999.999	999.999	999.999						0
999.999	22.400	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						0
999.999	999.999	25.760	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	17.920	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	40.320	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	33.600	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999						0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	

999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	21.840	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	14.560	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	32.380	26.320	999.999	20.720
999.999	999.999	999.999	28.560	999.999	999.999	999.999	999.999	999.999	999.999
999.999	20.720	999.999	999.999	999.999	999.999	22.960	999.999	999.999	12.880
999.999	999.999	999.999	999.999	999.999					0
42.560	999.999	999.999	28.000	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	16.240
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	15.680	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
21.280	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	22.400	28.560	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999					0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999

```
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999 999.999
 999.999 999.999 999.999 999.999 999.999                                        0
C-------------------------------------------------------------------------------
```

```
C                                F4.srg

C       F4.SRG is the 4th file used in this model.
C       This data file includes barge and ship costs.
C       Included are 5 data sets. These include:
C       (1) barge costs linking each river barge-loading location with five
C           barge-unloading locations
C       (2) barge costs linking each river barge-loading location with five
C           port locations
C       (3) barge-loading locations above L&D 26
C       (4) ship rates from 20 ports to 25 world demand locations
C       (5) ports located in Great Lakes.
C
C       Diagrammatically,
C                                   (1)
C       43 river barge loading ----------------------> 5 barge shipping
C          regions                                       regions

C                                   (2)
C       43 river barge loading ----------------------> 5 port shipping
C                regions                                regions

C       (3) The codes for 16 river shipping point above L&D 26

C                                   (4)
C       20 ports regions ----------------------> 25 foreign regions

C       (5) The number and codes of ports at Great Lakes for export.


C       A zero '0' must be inserted on the 80th column on each line.
C       The letter 'C' at the start of a statement is an explanation or
C       a comment. It is not in the actual file used in the programs.
C       The format used is:
C       8 fields dddd.ddd where d stands for a digit
C       e.g. ***0.234 where * is a space.

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

C       Explanation for first part

C       This includes barge costs linking the 43 river barge-loading locations
C       to 5 barge-unloading locations. Thus a 43 by 5 matrix.


C       The 5 barge-unloading 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 costs
C                from St. Paul, MN (601) to Nashville (702) =   0.282  $/bushel
```

```
C                from St. Paul, MN (601) to Knoxville (701) =   0.305  $/bushel
C                from St. Paul, MN (601) to Chatanoo  (703) =   0.279  $/bushel
C                from St. Paul, MN (601) to Guntersv  (710) =   0.265  $/bushel
C                from St. Paul, MN (601) to Florence  (713) =   0.251  $/bushel

C        If a new barge-unloading location were added, then there would be 6
C        entries on each line.

C        If a river barge-unloading location were deleted, then there would be 4
C        entries on each line. Deletion of a region will depend on its position of
C        in the list which represents unloading locations. This can be found in
C        the first file, F1.SRG.

C        If a river barge-loading location were added, then there would be one
C        more region, 44 river regions and thus a 44 by 5 matrix.
C        If a river region were deleted then there would be one less region,
C        42 regions and thus a 42 by 5 matrix.

C        Deletions and additions depend on the position of the region in
C        the code list on F1.SRG.

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. 0.443 becomes ***0.443 where * is a space.

   0.282   0.305   0.279   0.265   0.251                                        0
   0.248   0.271   0.246   0.231   0.218                                        0
   0.232   0.255   0.230   0.215   0.201                                        0
   0.222   0.246   0.220   0.206   0.192                                        0
   0.214   0.237   0.212   0.197   0.184                                        0
   0.188   0.211   0.186   0.171   0.157                                        0
   0.171   0.195   0.169   0.155   0.141                                        0
   0.135   0.158   0.125   0.118   0.104                                        0
   0.329   0.353   0.327   0.312   0.299                                        0
   0.302   0.325   0.300   0.285   0.271                                        0
   0.396   0.427   0.393   0.373   0.314                                        0
   0.356   0.388   0.353   0.333   0.314                                        0
   0.332   0.363   0.329   0.309   0.290                                        0
   0.287   0.319   0.284   0.264   0.245                                        0
   0.271   0.302   0.268   0.248   0.229                                        0
   0.256   0.287   0.253   0.233   0.214                                        0
   0.239   0.270   0.236   0.216   0.197                                        0
   0.217   0.206   0.174   0.155   0.137                                        0
   0.198   0.191   0.158   0.139   0.122                                        0
   0.151   0.153   0.120   0.102   0.084                                        0
   0.133   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
```

```
   0.273    0.247    0.215    0.196    0.173                                  0
   0.264    0.241    0.208    0.190    0.172                                  0
   0.213    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
C-------------------------------------------------------------------------------

C       Explanation for the second part
C
C       Included are barge costs from 43 barge-loading locations to
C       5 port locations and thus a 43 by 5 matrix.

C       The 5 ports 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,
C               St. Paul, MN (601) to New Orle, LA (702) =   0.264 $/bushel
C               St. Paul, MN (601) to Mobile,   AL (701) =   0.322 $/bushel
C               St. Paul, MN (601) to Galvestn, TX (703) =   0.371 $/bushel
C               St. Paul, MN (601) to Chicago,  IL (710) =   0.218 $/bushel
C               St. Paul, MN (601) to Portland, OR (713) = 999.000 $/bushel

C       If a new port location were added, then there would be 6 entries on
C       on each line and thus a 43 by 6 matrix

C       If a port location were deleted, then there would be 4 entries on
C       each line. Deletion of a port will depend on its position.
C       This can be found in the first file, F1.SRG.

C       If a river barge-loading location were added, then there would be one
C       more region, 44 river regions and thus a 44 by 5 matrix. If a river
C       region were deleted then there would be one less region, 42 regions.

C       Deletions and additions depend on the position of the barge-loading
C       location in the code list in F1.SRG.

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. 0.443 becomes ***0.443 where * is a space.

   0.264    0.322    0.371    0.218 999.000                                  0
```

```
   0.230    0.288    0.337    0.184 999.000                                  0
   0.214    0.272    0.321    0.168 999.000                                  0
   0.204    0.262    0.312    0.159 999.000                                  0
   0.196    0.254    0.304    0.150 999.000                                  0
   0.170    0.227    0.277    0.124 999.000                                  0
   0.153    0.211    0.261    0.108 999.000                                  0
   0.117    0.175    0.224    0.102 999.000                                  0
   0.311    0.255    0.419    0.280 999.000                                  0
   0.284    0.341    0.391    0.252 999.000                                  0
   0.272    0.401    0.518    0.328 999.000                                  0
   0.243    0.371    0.478    0.288 999.000                                  0
   0.225    0.353    0.454    0.264 999.000                                  0
   0.193    0.321    0.409    0.220 999.000                                  0
   0.246    0.309    0.393    0.056 999.000                                  0
   0.231    0.297    0.377    0.073 999.000                                  0
   0.251    0.285    0.361    0.093 999.000                                  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
   0.150    0.281    0.393  999.000 999.000                                  0
   0.123    0.216    0.327  999.000 999.000                                  0
   0.104    0.170    0.273  999.000 999.000                                  0
   0.094    0.144    0.242  999.000 999.000                                  0
   0.083    0.118    0.213  999.000 999.000                                  0
   0.098    0.229    0.287  999.000 999.000                                  0
   0.093    0.217    0.273  999.000 999.000                                  0
   0.063    0.144    0.188  999.000 999.000                                  0
 999.000  999.000  999.000  999.000 999.000                                  0
   0.042    0.122    0.176  999.000 999.000                                  0
   0.034    0.102    0.153  999.000 999.000                                  0
 999.000  999.000  999.000  999.000 999.000                                  0
 999.000  999.000  999.000  999.000   0.170                                  0
 999.000  999.000  999.000  999.000   0.145                                  0
 999.000  999.000  999.000  999.000   0.113                                  0
 999.000  999.000  999.000  999.000   0.063                                  0
 999.000  999.000  999.000  999.000   0.052                                  0
   0.000  999.000  999.000  999.000 999.000                                  0
 999.000  999.000  999.000  999.000   0.000                                  0

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

C       Explanation for the third part.

C       The number of river barge-loading locations above L&D 26 is 16
C       The codes for the barge-loading points are given.

C       Deletion and addition of a region will depend on the position
C       on this list and the code list in F1.SRG.

C       The format used is :
C       4 fields i.e. *ddd where d is a digit.
```

```
C       e.g. *601 where * is a space.

  16                                                                    0
 601 602 603 604 605 606 607 609 610 611 612 613 614 615 616 617        0

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

C       Explanation for the fourth part.
C       This shows the ship costs from 20 ports to 25 world demand locations

C       For example, the ship cost
C               from Mobile, AL (701) to Scandinavia (801) = 0.443 $/bushel
C               from Mobile, AL (701) to N. C. Europe(802) = 0.349 $/bushel

C       If a new foreign region were added, then there would be 26 entries in
C       each block. Thus a 20 by 26 matrix.
C       Remember that the maximum number of foreign demand regions allowed is 25.

C       If a foreign region were deleted, then there would be 24 entries
C       in each block. Deletion of this region will depend on its position.
C       This can be found in the first file, F1.SRG.
C       A 43 by 24 matrix would result.

C       If a port were added, there would be 21 ports. A 21 by 25 matrix would
C       result.
C       If a port were deleted then there would be 19 regions. A 19 by 25 matrix
C       would result.

C       Deletions and additions depend on the position of the region in
C       the code list on F1.SRG.

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. 0.443 becomes ***0.443 where * is a space.

   0.443   0.349   0.299   0.321   0.423   0.502   0.448   0.462   0.445   0.685
   0.956   0.390   0.477   0.484   0.773   0.430   0.590   0.405   0.660 999.000
   0.266   0.077   0.286   0.150   0.201                                       0
   0.443   0.349   0.299   0.321   0.423   0.502   0.448   0.462   0.445   0.685
   0.956   0.390   0.477   0.484   0.773   0.430   0.590   0.405   0.660 999.000
   0.266   0.077   0.286   0.150   0.201                                       0
   0.457   0.361   0.310   0.331   0.434   0.516   0.466   0.474   0.461   0.701
   0.975   0.406   0.485   0.490   0.784   0.430   0.580   0.389   0.650 999.000
   0.259   0.055   0.254   0.165   0.214                                       0
   0.457   0.361   0.310   0.331   0.434   0.516   0.466   0.474   0.461   0.700
   0.975   0.406   0.485   0.490   0.784   0.430   0.580   0.389   0.650 999.000
   0.259   0.055   0.254   0.165   0.214                                       0
   0.466   0.368   0.318   0.339   0.441   0.525   0.473   0.480   0.470   0.710
   0.986   0.415   0.490   0.490   0.799   0.334   0.798   0.389   0.903 999.000
   0.292   0.045   0.255   0.158   0.222                                       0
   0.405   0.349   0.310   0.325   0.471   0.431   0.499   0.515   0.372   0.602
   0.905   0.329   0.786   0.746   1.076   0.935   0.869   0.824   0.891 999.000
   0.316   0.155   0.281   0.138   0.138                                       0
   0.350   0.304   0.268   0.287   0.419   0.408   0.450   0.457   0.343   0.565
```

```
 0.908   0.308   0.539   0.511   0.842   0.884   0.828   0.782   0.917 999.000
 0.324   0.189   0.291   0.160   0.158                                       0
 0.691   0.691   0.582   0.594   0.872   0.783   0.932   0.909   0.664   1.080
 1.442   0.658   1.425   1.607   2.087   2.038   1.954   1.802   1.983 999.999
 0.870   0.616   0.814   0.595   0.505                                       0
 0.741   0.741   0.632   0.644   0.922   0.833   0.982   0.959   0.714   1.111
 1.492   0.708   1.475   1.657   2.137   2.088   2.004   1.852   2.033 999.999
 0.920   0.666   0.864   0.645   0.555                                       0
 0.796   0.711   0.687   0.699   0.977   0.888   1.037   1.018   0.769   1.167
 1.548   0.764   1.531   1.712   2.193   2.143   1.961   1.907   2.089 999.999
 0.929   0.723   0.913   0.700   0.610                                       0
 0.809   0.668   0.644   0.712   0.990   0.901   1.050   1.031   0.782   1.180
 1.560   0.776   1.544   1.725   2.205   2.156   2.070   1.920   2.046 999.999
 0.988   0.735   0.871   0.657   0.513                                       0
 0.777   0.703   0.723   0.709   0.822   0.841   0.643   0.609   0.779   1.130
 1.384   0.696   0.426   0.500   0.350   0.272   0.402   0.213   0.466 999.000
 0.296   0.532   0.240   0.417   0.588                                       0
 0.765   0.718   0.710   0.697   0.810   0.829   0.744   0.704   0.766   1.117
 1.319   0.679   0.494   0.579   0.355   0.317   0.401   0.217   0.459 999.000
 0.291   0.517   0.270   0.405   0.571                                       0
 0.713   0.668   0.660   0.646   0.761   0.776   0.823   0.819   0.710   1.062
 1.251   0.722   0.692   0.651   0.604   0.479   0.430   0.376   0.517 999.000
 0.230   0.459   0.357   0.351   0.512                                       0
 0.686   0.641   0.633   0.619   0.733   0.748   0.572   0.677   0.637   1.033
 1.416   0.767   0.456   0.538   0.388   0.270   0.399   0.210   0.459 999.000
 0.198   0.427   0.187   0.321   0.463                                       0
 0.741   0.696   0.687   0.672   0.788   0.796   0.572   0.853   0.688   1.173
 1.384   0.751   0.979   0.917   0.667   0.480   0.427   0.369   0.453 999.000
 0.202   0.421   0.337   0.312   0.455                                       0
 0.761   0.620   0.660   0.666   0.783   0.815   0.718   0.713   0.762   0.984
 1.369   0.763   0.789   1.094   1.071   1.056   1.402   0.984   1.504 999.999
 0.877   0.757   0.843   0.720   0.747                                       0
 0.801   0.666   0.700   0.706   0.823   0.855   0.758   0.753   0.802   1.024
 1.409   0.803   0.829   1.134   1.111   1.096   1.442   1.024   1.544 999.999
 0.917   0.797   0.883   0.760   0.787                                       0
 0.899   0.757   0.797   0.793   0.921   0.953   0.857   0.851   0.900   1.122
 1.507   0.901   0.927   1.232   1.209   1.994   1.540   1.122   1.642 999.999
 1.015   0.895   0.981   0.858   0.885                                       0
 0.814   0.685   0.713   0.605   0.836   0.868   0.772   0.766   0.815   1.037
 1.422   0.816   0.842   1.147   1.134   1.109   1.455   1.037   1.556 999.999
 0.930   0.810   0.806   0.773   0.800                                       0
```

C----------

C Explanation for the fifth part.

C Number of ports at Great Lakes for export are 4.

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, 8C
C                (720) DULUTH,  8C

C        If a new port were added, then change 8 to 9 and
C        add the code for that region. (see below)
C        If a port were deleted, then change 8 to 7 and
C        delete the code from the list.

   8
 708 709 710 711 717 718 719 720                                                0
                                                                                0

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

```
C                                F8.SRG

C       F8.SRG is the 5th file in this model.
C       This file contains the grain supplies and demand of the regions
C       included in this model.

C       This file deals with
C               (1) the amount of sorghum produced by each surplus region
C               (2) the amount of sorghum demanded by each deficit region
C               (3) the amount of sorghum required by each foreign region
C                   (per quarter)

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. 0.443 becomes ***0.443 where * is a space.
C       ***0.400 will be the first entry,
C       **11.456 will be the fourth entry.

C       The letter 'C' at the start of a statement is used as a comment.
C       Refer to F1.srg for the position of the regions.

C       A zero '0'  must be included on the 80th column on each line.

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

C       Explanation for first part

C       This shows the quantity of sorghum produced by each surplus region.
C       e.g. Abilene produces 0.400 million bushels.
C       Surpluses are in 1000's of bushels.

C       If a surplus region were added then the value would be added to the position
C       as shown in the code list in F1.SRG. If it were added to the end of the
C       list then add the surplus value to the end of this list.

C       If a surplus region were removed then the value associated with this
C       region must be deleted. See F1.SRG for the actual position of the
C       deleted region.

   0.400   9.365   0.468  11.456  50.648   9.159   2.256   5.288  22.118  19.873
   1.130  11.708   4.831  10.263  13.940  20.651  16.270   4.500  33.092   1.935
  17.139   4.007  14.570  20.707  20.207  11.376   1.741   9.131   1.014  20.692
  21.590                                                                      0

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

C       Explanation for second part

C       This shows the value of sorghum demanded by each deficit region.
C       e.g. Amarillo requires 52.771 million bushels.

C       If a deficit region were added then the value would be added to the
C       position as shown in the code list in F1.SRG. If it were added to the
```

```
C       end of the list then add the surplus value to the end of this list.

C       If a deficit region were removed then the value associated with this
C       region must be deleted. See F1.SRG for the actual position of the
C       deleted region.

  52.771   3.819   1.558   1.564  23.000   3.014   1.897   0.320  10.079   0.080
   6.249   2.809   0.048   0.500   4.310   2.929  23.556   8.824   2.809   1.228
   2.882   2.308   2.955   2.776   0.516   1.646   0.231   0.871   3.629   0.113
   7.801   4.606  22.174   8.172   1.769                                       0

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

C       Explanation for third part

C       This shows the amount of sorghum demanded by each foreign region
C       per quarter.

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. 0.443 becomes ***0.443 where * is a space.
C            0     becomes ***0.000

C       e.g. N.C. Europe requires 0.348 million bushels in the 1st quarter
C       and 0.000 on the 2nd, 3rd and 4th quarters, respectively.
C       e.g. Japan requires 26.933 million bushels in the 1st quarter
C                           19.857 million bushels in the 2nd quarter
C                           25.850 million bushels in the 3rd quarter
C                           17.184 million bushels in the 4th quarter.

C       Addition and deletion of a region will depend on its position in
C       the code list in F1.SRG.

C       If a foreign region were added then the value were added to the
C       position as shown in the code list in F1.SRG. If it were added to
C       the end of the list then add the value to the end of this list.

C       If a foreign region were removed then the value associated with this
C       region must be deleted. See F1.SRG for the actual position of the
C       deleted region.

   0.000   0.000   0.000   0.000                                               0
   0.348   0.000   0.000   0.000                                               0
   0.000   0.000   0.000   0.000                                               0
   0.000   0.000   0.000   0.000                                               0
   0.000   0.000   0.000   0.000                                               0
   0.000   0.000   0.000   0.000                                               0
   1.456   1.571   1.175   1.597                                               0
   2.750   3.603   2.577   2.491                                               0
   0.000   0.000   0.000   0.000                                               0
   0.000   0.000   0.000   0.000                                               0
   0.303   0.000   0.123   0.123                                               0
   0.068   0.029   0.358   0.322                                               0
   0.000   0.000   0.000   0.000                                               0
```

0.000	0.000	0.000	0.000	0
0.000	0.000	0.000	0.580	0
2.866	3.002	2.605	6.146	0
0.000	0.000	0.273	0.000	0
26.933	19.857	25.850	17.184	0
0.000	0.000	0.000	0.000	0
0.000	0.000	0.000	0.000	0
2.832	0.532	4.290	0.583	0
2.821	0.532	4.290	0.593	0
0.000	0.000	0.742	0.774	0
0.000	14.086	9.851	11.276	0
0.000	0.000	0.000	0.000	0

C----------

C F9.SRG

C This data file includes the location names.
C The code numbers are used for explanation and are not included in
C the actual file.

C The number of regions are:
C (1) 31 surplus regions
C (2) 35 deficit regions
C (3) 43 river regions
C (4) 20 port regions
C (5) 25 foreign regions

C If there are any changes in the number of regions, then either add a
C new name or remove the name of that region.
C If the surplus region, Abilene were deleted then remove Abilene, TX
C from the list of surplus regions. Then the list will start with
C Alexandria, LA.
C If a new region were added, the position of this region must be the
C the same as the list of regions in F1.SRG.
C Refer to F1.SRG for positions of the regions.

C This file is used for explanation and reference only.
C Do not include the code numbers.
C The format or layout of this file is incorrect and must not be used
C to run the programs.

C Inclusion of new names (for all regions) must follow the format :
C 12 characters including the state code(if applicable),
C e.g. 012345678901
C Abilene TX
C Upper and/or lower cases can be used for the names.

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

C---

C (1) 31 Surplus regions

C Code number	Location name	State	
064	Abilene	TX	0
111	Alexandria	LA	0
052	Altus	OK	0
043	Atchison	KS	0
039	Beatrice	NE	0
06D	Beaumont	TX	0
171	Bowling Gr	KY	0
094	Clinton	MO	0
036	Columbus	NE	0
049	Concordia	KS	0
036	Columbia	SC	0

048	Emporia	KS	0
391	Greeley	CO	0
055	Guymon	OK	0
034	Hastings	NE	0
066	Hillsboro	TX	0
047	Independen	KS	0
093	Kirksville	MO	0
044	Liberal	KS	0
062	Lubbock	TX	0
06F	McAllen	TX	0
033	McCook	NE	0
021	Pierre	SD	0
042	Salina	KS	0
099	Sikeston	MO	0
041	St. Franci	KS	0
411	Roswell	NM	0
046	Tribune	KS	0
401	Tuscon	AZ	0
062	Victoria	TX	0
045	Wichita	KS	0

C----------

C 35 Deficit regions

C Code number	Location name	State	
061	Amarillo	TX	0
085	Ames	IA	0
038	Ansley	NE	0
057	Ardmore	OK	0
06E	Laredo	TX	0
098	Cabool	MO	0
063	Childress	TX	0
056	Clinton	OK	0
362	El Centro	CA	0
051	Enid	OK	0
065	Ft. Worth	TX	0
068	Austin	TX	0
092	Chillicoth	MO	0
058	Hugo	OK	0
133	Decatur	IL	0
095	Lebanon	MO	0
105	Little Roc	AR	0
067	Longview	TX	0
068	Lufkin	TX	0
054	McAlester	OK	0
193	Meridian	MS	0
053	OklahomaC	OK	0
069	Pecos	TX	0
096	Rolla	MO	0
091	St. Joseph	MO	0
06A	San Angelo	TX	0
097	Springfiel	MO	0
361	Stockton	CA	0

032	Valentine	NE	0
059	Tulsa	OK	0
381	Logan	UT	0
211	Macon	GA	0
201	Birmingham	AL	0
06G	El Paso	TX	0
402	Nogales	AZ	0

C----------

C 43 River Port regions

C Code number	Location name	State	
601	ST. PAUL,	MN	0
602	WINONA,	MN	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,	AR	0
624	MEMPHIS,	TN	0
625	NASHVILL,	TN	0
626	KNOXVILL,	TN	0
627	CHATANOO,	TN	0
628	GUNTERSV,	AL	0
629	FLORENCE,	AL	0
630	CATOOSA,	OK	0
631	MUSKOGEE,	OK	0
632	PINEBLUF,	AR	0
633	DES ARC,	AR	0
634	GREENWOO,	MS	0
635	VICKSBUR,	MS	0
636	MONROE,	LA	0
638	LEWISTON,	ID	0
639	CENTRL F,	WA	0
640	PASCO,	WA	0
641	ROOSEVEL,	WA	0
643	THE DALL,	OR	0

702	NEW ORLE,	LA	0
713	PORTLAND,	OR	0

C----------

20 Port Regions

701	MOBILE,	AL	0
702	NEW ORLE,	LA	0
703	GALVESTN,	TX	0
704	CORPUS C,	TX	0
705	BROWNSVI,	TX	0
706	CHARLEST,	SC	0
707	BALTIMOR,	MD	0
708	TOLEDO,	OH	0
709	SAGINAW,	MI	0
710	CHICAGO,	IL	0
711	DULUTH,	MN	0
712	SEATTLE,	WA	0
713	PORTLAND,	OR	0
714	SAN FRAN,	CA	0
715	LONG BEA,	CA	0
716	SAN DIEG,	CA	0
717	TOLEDO,	BC	0
718	SAGINAW,	BC	0
719	CHICAGO,	BC	0
720	DULUTH,	BC	0

C----------

C 25 Foreign regions

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

822	E MEXICO	0
823	WS AMERICA	0
824	CENT AMERICA	0
825	CARIBBEAN	0

C----------

******* OPTIMAL SOLUTION *******
TOTAL COST 3039468032.
SORGHUM SHIPMENT, BAIE COMEAU ADDED, FULLER, GRANT, TEH, FELLI
UNIT = 1 (THOUSAND BUSHEL)

SUPPLY 391525.

DEMAND 391164.

677 678 678 16773

1 Network Generator For Grain Shipment Problem

SORGHUM SHIPMENT, FULLER, GRANT, TEH, FELLIN

1

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME 1	2	3	4	TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
S 064 Abilene TX		400.								
D 06A San AngeloTX	T		312.	88.	00.	00.	400.	0.09507	38027.	75768.
S 111 AlexandriaLA		9365.								
D 193 Meridian MS	T		00.	00.	00.	91.	91.	0.25490	23196.	17237.
P 702 NEW ORLE, LA	T		9274.	00.	00.	00.	9274.	0.19891	1844688.	1593737.
S 052 Altus OK		468.								
D 06A San AngeloTX	R		00.	00.	345.	123.	468.	0.21280	99590.	80225.
S 043 Atchison KS		11456.								
R 612 ST JOSEP, MO	T		00.	3451.	1584.	6421.	11456.	0.02374	271965.	1932856.
S 039 Beatrice NE		50648.								
D 085 Ames IA	T		954.	954.	954.	954.	3816.	0.24832	947597.	722827.
D 098 Cabool MO	R		753.	753.	753.	753.	3012.	0.39760	1197571.	516317.
D 092 ChillicothMO	T		12.	12.	12.	12.	48.	0.21255	10202.	9092.
D 096 Rolla MO	T		694.	502.	694.	694.	2584.	0.37331	964633.	489461.
D 091 St. JosephMO	T		129.	129.	129.	129.	516.	0.13807	71245.	97741.
R 610 OMAHA, NE	T		00.	17971.	10367.	12334.	40672.	0.10556	4293163.	6862180.
S 060 Beaumont TX		9159.								
P 703 GALVESTN, TX	T		4153.	5006.	00.	00.	9159.	0.08248	755430.	1573974.
S 171 Bowling GrKY		2256.								
D 211 Macon GA	R		21.	1151.	1084.	00.	2256.	0.36400	821184.	386724.
S 094 Clinton MO		5288.								
D 095 Lebanon MO	T		732.	732.	732.	732.	2928.	0.11080	324423.	554622.
D 105 Little RocAR	T		00.	00.	2074.	58.	2132.	0.29813	635612.	403843.
D 097 SpringfielMO	T		57.	57.	57.	57.	228.	0.09612	21914.	43188.
S 036 Columbus NE		22118.								
R 610 OMAHA, NE	T		00.	4754.	7400.	9964.	22118.	0.08982	1986682.	3731749.

S 049	Concordia	KS		19873.								
P 703	GALVESTN,	TX	R		00.	3990.	14929.	954.	19873.	0.56560	11240170.	2871649.
S 036	Columbia	SC		1130.								
D 211	Macon	GA	T		1130.	00.	00.	00.	1130.	0.19576	221212.	214045.
S 048	Emporia	KS		11708.								
P 703	GALVESTN,	TX	R		11708.	00.	00.	00.	11708.	0.44800	5245184.	1691806.
S 391	Greeley	CO		4831.								
D 381	Logan	UT	T		1950.	1950.	931.	00.	4831.	0.46165	2230219.	915088.
S 055	Guymon	OK		10263.								
D 061	Amarillo	TX	T		10263.	00.	00.	00.	10263.	0.13178	1352445.	1944018.
S 034	Hastings	NE		13940.								
D 038	Ansley	NE	T		389.	389.	389.	389.	1556.	0.09402	146291.	294738.
P 701	MOBILE,	AL	R		1349.	00.	9760.	1275.	12384.	0.57680	7143091.	1789488.
S 066	Hillsboro	TX		20651.								
D 06E	Laredo	TX	T		00.	00.	111.	5750.	5861.	0.34794	2039256.	1110191.
D 065	Ft. Worth	TX	T		1562.	1562.	1562.	1562.	6248.	0.05940	371152.	1183496.
D 068	Austin	TX	T		702.	702.	702.	702.	2808.	0.14017	393598.	531891.
D 067	Longview	TX	T		2176.	750.	00.	00.	2926.	0.15800	462313.	554243.
D 068	Lufkin	TX	T		702.	702.	702.	702.	2808.	0.17583	493740.	531891.
S 047	Independen	KS		16270.								
D 105	Little Roc	AR	T		5889.	5788.	3815.	778.	16270.	0.32350	5263391.	3081864.
S 093	Kirksville	MO		4500.								
D 133	Decatur	IL	T		1077.	1077.	1077.	1077.	4308.	0.23248	1001503.	816021.
D 096	Rolla	MO	T		00.	192.	00.	00.	192.	0.20101	38593.	36369.
S 044	Liberal	KS		33092.								
D 061	Amarillo	TX	T		00.	10627.	10675.	3665.	24967.	0.17269	4311467.	4729249.
D 063	Childress	TX	T		474.	474.	474.	474.	1896.	0.21150	400998.	359140.
D 06G	El Paso	TX	R		100.	2043.	2043.	2043.	6229.	0.47600	2965004.	1067775.
S 062	Lubbock	TX		1935.								
D 069	Pecos	TX	T		635.	446.	441.	413.	1935.	0.21150	409246.	366528.
S 06F	McAllen	TX		17139.								

D 06E	Laredo	TX	T		5750.	5750.	5639.	00.	17139.	0.14961	2564172.	3246470.
S 033	McCook	NE		4007.								
D 193	Meridian	MS	R		720.	720.	720.	629.	2789.	0.48160	1343182.	478090.
D 211	Macon	GA	R		00.	00.	67.	1151.	1218.	0.60480	736646.	208790.
S 021	Pierre	SD		14570.								
D 361	Stockton	CA	R		217.	217.	217.	217.	868.	1.18160	1025629.	148793.
D 032	Valentine	NE	T		907.	907.	907.	907.	3628.	0.14332	519954.	687216.
P 713	PORTLAND,	OR	R		00.	00.	273.	9440.	9713.	0.80000	7770400.	1403529.
S 042	Salina	KS		20707.								
D 057	Ardmore	OK	T		391.	391.	391.	391.	1564.	0.32820	513307.	296253.
D 056	Clinton	OK	T		80.	80.	80.	80.	320.	0.28215	90289.	60614.
D 362	El Centro	CA	R		2519.	2519.	2519.	2519.	10076.	0.99120	9987332.	1727228.
D 058	Hugo	OK	R		00.	00.	00.	100.	100.	0.39200	39200.	17142.
D 067	Longview	TX	R		30.	1456.	2206.	2206.	5898.	0.45920	2708362.	1011035.
D 06A	San Angelo	TX	R		99.	323.	66.	288.	776.	0.50400	391104.	133022.
D 06G	El Paso	TX	R		1219.	00.	00.	00.	1219.	0.62160	757730.	208961.
D 402	Nogales	AZ	R		00.	00.	312.	442.	754.	0.83440	629138.	129251.
S 099	Sikeston	MO		20207.								
D 201	Birmingham	AL	R		5543.	5543.	5543.	3578.	20207.	0.27440	5544801.	3463884.
S 041	St. Franci	KS		11376.								
D 061	Amarillo	TX	T		808.	2565.	2517.	2517.	8407.	0.33760	2838197.	1592454.
D 381	Logan	UT	T		00.	00.	1019.	1950.	2969.	0.63080	1872858.	562388.
S 411	Roswell	NM		1741.								
D 069	Pecos	TX	R		103.	292.	297.	325.	1017.	0.18480	187942.	174334.
D 06G	El Paso	TX	T		724.	00.	00.	00.	724.	0.21464	155402.	137140.
S 046	Tribune	KS		9131.								
D 061	Amarillo	TX	T		2121.	00.	00.	7010.	9131.	0.24738	2258847.	1729594.
S 401	Tuscon	AZ		1014.								
D 402	Nogales	AZ	T		442.	442.	130.	00.	1014.	0.06884	69807.	192072.
S 062	Victoria	TX		20692.								
P 704	CORPUS C,	TX	T		8546.	6012.	4953.	1181.	20692.	0.09821	2032230.	3555920.
S 045	Wichita	KS		21590.								
D 051	Enid	OK	T		20.	20.	20.	20.	80.	0.11500	9200.	15154.
D 058	Hugo	OK	T		125.	125.	125.	25.	400.	0.31129	124514.	75768.
D 105	Little Roc	AR	T		00.	101.	00.	5053.	5154.	0.43251	2229182.	976271.

D 054 McAlester OK	T		307.	307.	307.	307.	1228.	0.25772	316480.	232608.
D 053 Oklahoma COK	T		577.	577.	577.	577.	2308.	0.16744	386456.	437181.
D 059 Tulsa OK	T		28.	28.	28.	28.	112.	0.19157	21456.	21215.
D 201 BirminghamAL	R		00.	00.	00.	1965.	1965.	0.51520	1012368.	336840.
P 703 GALVESTN, TX	R		5347.	2028.	2868.	100.	10343.	0.49840	5154951.	1494564.

1

SURPLUS REGION	SUPPLY	STORAGE			STORAGE COST
S 064 Abilene TX	400.	88.	00.	00.	3374.
S 111 AlexandriaLA	9365.	91.	91.	91.	10525.
S 052 Altus OK	468.	468.	468.	123.	40682.
S 043 Atchison KS	11456.	11456.	8005.	6421.	996439.
S 039 Beatrice NE	50648.	48106.	27785.	14876.	3489624.
S 060 Beaumont TX	9159.	5006.	00.	00.	191940.
S 171 Bowling GrKY	2256.	2235.	1084.	00.	127257.
S 094 Clinton MO	5288.	4499.	3710.	847.	347763.
S 036 Columbus NE	22118.	22118.	17364.	9964.	1902177.
S 049 Concordia KS	19873.	19873.	15883.	954.	1408143.
S 056 Columbia SC	1130.	00.	00.	00.	00.
S 048 Emporia KS	11708.	00.	00.	00.	00.
S 391 Greeley CO	4831.	2881.	931.	00.	146160.
S 055 Guymon OK	10263.	00.	00.	00.	00.
S 034 Hastings NE	13940.	12202.	11813.	1664.	985641.
S 066 Hillsboro TX	20651.	15509.	11793.	8716.	1386529.
S 047 IndependenKS	16270.	10381.	4593.	778.	604458.
S 093 KirksvilleMO	4500.	3423.	2154.	1077.	255811.
S 044 Liberal KS	33092.	32518.	19374.	6182.	2230596.
S 062 Lubbock TX	1935.	1300.	854.	413.	98686.
S 06F McAllen TX	17139.	11389.	5639.	00.	652889.
S 033 McCook NE	4007.	3287.	2567.	1780.	293852.
S 021 Pierre SD	14570.	13085.	11961.	10564.	1372057.
S 042 Salina KS	20707.	16369.	11600.	6026.	1307258.
S 099 Sikeston MO	20207.	14664.	9121.	3578.	1051422.
S 041 St. FranciKS	11376.	10568.	8003.	4467.	886156.
S 411 Roswell NM	1741.	914.	622.	325.	71561.
S 046 Tribune KS	9131.	7010.	7010.	7010.	810777.
S 401 Tucson AZ	1014.	572.	130.	00.	26916.
S 062 Victoria TX	20692.	12146.	6134.	1181.	746924.
S 045 Wichita KS	21590.	15186.	12000.	8075.	1357098.

1

ORIGIN/DESTN MODE	SUPPLY	SHIPMENT BY TIME				TOTAL	UNIT	TOTAL	HANDLING
		1	2	3	4	SHIPMENT	COST	HAULING COST	COSTS
R 601 ST. PAUL, MN		00.	00.	00.	00.				
R 602 WINONA, MN		00.	00.	00.	00.				
R 603 MCGREGOR, IA		00.	00.	00.	00.				

R 604 DUBUQUE, IA	00.	00.	00.	00.				
R 605 CLINTON, IA	00.	00.	00.	00.				
R 606 BURLINGT, IA	00.	00.	00.	00.				
R 607 HANNIBAL, MO	00.	00.	00.	00.				
R 608 ST LOUIS, MO	00.	00.	00.	00.				
R 609 SIOUX CT, IA	00.	00.	00.	00.				
R 610 OMAHA, NE	00.	22725.	17767.	22298.				
P 702 NEW ORLE, LA B	00.	22725.	17767.	22298.	62790.	0.28400	17832360.	9955355.
R 611 NE CITY, NE	00.	00.	00.	00.				
R 612 ST JOSEP, MO	00.	3451.	1584.	6421.				
P 702 NEW ORLE, LA B	00.	3451.	1584.	6421.	11456.	0.24300	2783808.	1816349.
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 625	NASHVILL, TN	00.	00.	00.	00.
R 626	KNOXVILL, TN	00.	00.	00.	00.
R 627	CHATANOO, TN	00.	00.	00.	00.
R 628	GUNTERSV, AL	00.	00.	00.	00.
R 629	FLORENCE, AL	00.	00.	00.	00.
R 630	CATOOSA, OK	00.	00.	00.	00.
R 631	MUSKOGEE, OK	00.	00.	00.	00.
R 632	PINEBLUF, AR	00.	00.	00.	00.
R 633	DES ARC, AR	00.	00.	00.	00.
R 634	GREENWOO, MS	00.	00.	00.	00.
R 635	VICKSBUR, MS	00.	00.	00.	00.
R 636	MONROE, LA	00.	00.	00.	00.
R 637	LEWISTON, ID	00.	00.	00.	00.
R 638	CENTRL F, WA	00.	00.	00.	00.
R 639	PASCO, WA	00.	00.	00.	00.

R 640 ROOSEVEL, WA			00.	00.	00.	00.				
R 641 THE DALL, OR			00.	00.	00.	00.				
R 642 NEW ORLE, LA			00.	00.	00.	00.				
R 643 PORTLAND, OR			00.	00.	00.	00.				

1

ORIGIN/DESTN	MODE	SUPPLY	SHIPMENT BY TIME 1	2	3	4	TOTAL SHIPMENT	UNIT COST	TOTAL HAULING COST	HANDLING COSTS
P 701 MOBILE, AL			1349.	00.	9760.	1275.				
F 821 W MEXICO	S		1349.	00.	00.	00.	1349.	0.26600	358834.	76353.
F 824 CENT AMERICA	S		00.	00.	9760.	1275.	11035.	0.15000	1655250.	624581.
P 702 NEW ORLE, LA			9274.	26176.	19351.	28719.				
F 802 N.C. EUROPE	S		348.	00.	00.	00.	348.	0.34900	121452.	19697.
F 807 E BLOCK EURO	S		1456.	1571.	1175.	1597.	5799.	0.44800	2597952.	328223.
F 808 E MEDITERRAN	S		2750.	3603.	2577.	2491.	11421.	0.46200	5276502.	646429.
F 811 E AFRICA	S		303.	00.	123.	123.	549.	0.95600	524844.	31073.
F 812 W AFRICA	S		68.	29.	358.	322.	777.	0.39000	303030.	43978.
F 816 TAIWAN	S		2866.	3002.	2605.	6146.	14619.	0.43000	6286170.	827435.
F 818 JAPAN	S		00.	3353.	8132.	7456.	18941.	0.40500	7671105.	1072061.
F 821 W MEXICO	S		1483.	532.	4290.	583.	6888.	0.26600	1832208.	389861.
F 824 CENT AMERICA	S		00.	14086.	91.	10001.	24178.	0.15000	3626700.	1368475.
P 703 GALVESTN, TX			21208.	11024.	17797.	1054.				
F 818 JAPAN	S		18387.	10492.	12765.	100.	41744.	0.38900	16238416.	2362710.
F 822 E MEXICO	S		2821.	532.	4290.	180.	7823.	0.05500	430265.	442782.
F 823 WS AMERICA	S		00.	00.	742.	774.	1516.	0.25400	385064.	85806.
P 704 CORPUS C, TX			8546.	6012.	4953.	1181.				
F 818 JAPAN	S		8546.	6012.	4953.	768.	20279.	0.38900	7888531.	1147791.
F 822 E MEXICO	S		00.	00.	00.	413.	413.	0.05500	22715.	23376.
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.	00.	00.			
P 713 PORTLAND,	OR		00.	00.	273.	9440.			
F 815 SE ASIA		S	00.	00.	00.	580.	580.	0.35500	205900.	32828.
F 817 KOREA		S	00.	00.	273.	00.	273.	0.40100	109473.	15452.
F 818 JAPAN		S	00.	00.	00.	8860.	8860.	0.21700	1922620.	501476.
P 714 SAN FRAN,	CA		00.	00.	00.	00.			
P 715 LONG BEA,	CA		00.	00.	00.	00.			
P 716 SAN DIEG,	CA		00.	00.	00.	00.			
P 717 TOLEDO,	BC		00.	00.	00.	00.			
P 718 SAGINAW,	BC		00.	00.	00.	00.			
P 719 CHICAGO,	BC		00.	00.	00.	00.			
P 720 DULUTH,	BC		00.	00.	00.	00.			

1

		DEMAND BY TIME				TOTAL
		1	2	3	4	SHIPMENT
D 061 Amarillo	TX	13192.	13192.	13192.	13192.	52768.
D 085 Ames	IA	954.	954.	954.	954.	3816.
D 038 Ansley	NE	389.	389.	389.	389.	1556.
D 057 Ardmore	OK	391.	391.	391.	391.	1564.
D 06E Laredo	TX	5750.	5750.	5750.	5750.	23000.
D 098 Cabool	MO	753.	753.	753.	753.	3012.
D 063 Childress	TX	474.	474.	474.	474.	1896.
D 056 Clinton	OK	80.	80.	80.	80.	320.

D 362	El Centro CA	2519.	2519.	2519.	2519.	10076.
D 051	Enid OK	20.	20.	20.	20.	80.
D 065	Ft. Worth TX	1562.	1562.	1562.	1562.	6248.
D 068	Austin TX	702.	702.	702.	702.	2808.
D 092	ChillicothMO	12.	12.	12.	12.	48.
D 058	Hugo OK	125.	125.	125.	125.	500.
D 133	Decatur IL	1077.	1077.	1077.	1077.	4308.
D 095	Lebanon MO	732.	732.	732.	732.	2928.
D 105	Little RocAR	5889.	5889.	5889.	5889.	23556.
D 067	Longview TX	2206.	2206.	2206.	2206.	8824.
D 068	Lufkin TX	702.	702.	702.	702.	2808.
D 054	McAlester OK	307.	307.	307.	307.	1228.
D 193	Meridian MS	720.	720.	720.	720.	2880.
D 053	Oklahoma COK	577.	577.	577.	577.	2308.
D 069	Pecos TX	738.	738.	738.	738.	2952.
D 096	Rolla MO	694.	694.	694.	694.	2776.
D 091	St. JosephMO	129.	129.	129.	129.	516.
D 06A	San AngeloTX	411.	411.	411.	411.	1644.
D 097	SpringfielMO	57.	57.	57.	57.	228.
D 361	Stockton CA	217.	217.	217.	217.	868.
D 032	Valentine NE	907.	907.	907.	907.	3628.
D 059	Tulsa OK	28.	28.	28.	28.	112.
D 381	Logan UT	1950.	1950.	1950.	1950.	7800.
D 211	Macon GA	1151.	1151.	1151.	1151.	4604.
D 201	BirminghamAL	5543.	5543.	5543.	5543.	22172.
D 06G	El Paso TX	2043.	2043.	2043.	2043.	8172.
D 402	Nogales AZ	442.	442.	442.	442.	1768.
F 801	SCANDINAVIA	00.	00.	00.	00.	00.
F 802	N.C. EUROPE	348.	00.	00.	00.	348.
F 803	S.W. EUROPE	00.	00.	00.	00.	00.
F 804	ISLANDS	00.	00.	00.	00.	00.
F 805	ADRIATIC	00.	00.	00.	00.	00.
F 806	USSR	00.	00.	00.	00.	00.
F 807	E BLOCK EURO	1456.	1571.	1175.	1597.	5799.
F 808	E MEDITERRAN	2750.	3603.	2577.	2491.	11421.
F 809	N AFRICA	00.	00.	00.	00.	00.
F 810	RED SEA	00.	00.	00.	00.	00.
F 811	E AFRICA	303.	00.	123.	123.	549.
F 812	W AFRICA	68.	29.	358.	322.	777.
F 813	PERSIAN GULF	00.	00.	00.	00.	00.
F 814	W ASIA	00.	00.	00.	00.	00.
F 815	SE ASIA	00.	00.	00.	580.	580.
F 816	TAIWAN	2866.	3002.	2605.	6146.	14619.
F 817	KOREA	00.	00.	273.	00.	273.
F 818	JAPAN	26933.	19857.	25850.	17184.	89824.
F 819	CHINA	00.	00.	00.	00.	00.
F 820	CANADA	00.	00.	00.	00.	00.
F 821	W MEXICO	2832.	532.	4290.	583.	8237.
F 822	E MEXICO	2821.	532.	4290.	593.	8236.
F 823	WS AMERICA	00.	00.	742.	774.	1516.
F 824	CENT AMERICA	00.	14086.	9851.	11276.	35213.
F 825	CARIBBEAN	00.	00.	00.	00.	00.

1

STORAGE COST 22802692.

TRUCK COST	47326552.
RAIL COST	66000576.
BARGE COST	20616168.
SHIP COST	57457032.
HANDLING CST	89746896.

GRAIN SHIPPED FROM SURPLUS REGIONS	391164.
GRAIN SHIPPED TO DEFICIT REGIONS	213772.
GRAIN SHIPPED TO FOREIGN REGIONS	177392.

Validation of Grain Sorghum Model

Port	FGIS Recorded Exports	Model Solution
	Million Bu	
Mobile	12 2	12 4
New Orleans	83 5	83.5
Galveston	51 3	51.1
Corpus Christi	20 2	20 7
Brownsville	0	0
Charleston	0	0
Baltimore	0	0
Toledo	0	0
Saginaw	0	0
Chicago	0	0
Duluth	0	0
Seattle	0 3	0
Portland	9 9	9 7
California	0	0
San Francisco	-	-
Long Beach	-	-
San Diego	-	-
Baie Comeau	0	0
Toledo	-	-
Saginaw	-	-
Chicago	-	-
Duluth	-	-
Total	177 4	177 4

Special adjustments made in the calibration process:

1) Rail rate from Pierre, South Dakota to Portland set to 85 percent of Reebie estimated total cost to force flows to Portland.