



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

INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL: CORN

SEPTEMBER 1990

IWR REPORT 90-R-7

**INTERNATIONAL GRAIN TRANSPORTATION NETWORK MODEL:
CORN**

Prepared By:

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

**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**

	Content	
		page
Introduction.....		i
Programs		
crn1.f.....		1
crn2.f.....		18
crn3.f.....		29
Data files		
f1.crn.....		56
f2.crn.....		63
f3.crn.....		78
f4.crn.....		100
f8.crn.....		105
f9.crn.....		107
Output		
file06_2.crn.....		112
final.out.....		113
Calibration.....		127

INTRODUCTION

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

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

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

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

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

ACKNOWLEDGEMENTS

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

```
C ---- ****
C ---- *          CORN1.F
C ---- *      Documented on      SEPT 1990 *
C ---- *      Run on      .      a 386 machine *
C ---- *      Compiler used      NDP Fortran *
C ---- *      Grain used      CORN (CRN) *
C ---- *      Data files used      F1,F2,F3,F4, and F8. *
C ---- *      Trace file      U6 *
C ---- *      Input file for CORN2      U12 *
C ---- ****
```

```
C ---- Declaration.
C ---- The array size used is 64000.
```

```
COMMON /GO/ K, OPERATION
COMMON /G1/ IARC(64000) /G2/ JARC(64000) /G3/ LOWR(64000)
COMMON /G4/ UPPR(64000) /G5/ KOST(64000)
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.crn.
C ---- F1.crn contains information about the model.
C ---- U2 is the unit number for data file, FILE02.crn.
C ---- F2.crn contains TRUCK mileage information.
C ---- U3 is the unit number for data file, F3.crn.
C ---- F3.crn contains RAIL costs information.
C ---- U4 is the unit number for data file, F4.crn.
C ---- F4.crn contains BARGE costs information.
C ---- U8 is the unit number for data file, F8.crn.
C ---- F8.crn contains information about SUPPLIES and DEMANDS.
C ---- U9 is for data files, FILE09.crn.
C ---- F9.crn 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, CORN2.
```

```
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, CORN1.F:
```

```

C ----
C ----      U1,U2,U3,U4,U8,U9 (input files)
C ----      |
C ----      |
C ----      V
C ----      CORN1.F (program)
C ----      |
C ----      / \ 
C ----      /   \
C ----      V     V
C ----      U6 (trace)    U12 (input to CORN2.F)

```

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

```

OPEN (UNIT = U1, FILE = 'F1.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U2, FILE = 'FILED2.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U3, FILE = 'F3.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U4, FILE = 'F4.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U8, FILE = 'F8.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U6, FILE = 'FILE06_2.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U12, FILE = 'FILE12_2.CRN', 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 11DO K = 1, 64000
LOWR(K)  = 0
UPPR(K)  = INFN
KOST(K)  = 0
11DO CONTINUE
K          = 0
CALL DRIVER
STOP
END

```

C ----- WRITER -----

C ---- Write all nodes, iarc, jarc, 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(= LOWER), L(= UPPER), M(=KOST),
C ---- N(=NODES) to the next program, CORN2.F.

```
      WRITE (U12,510) N, NODE, I, J, K, L, M, N
510 FORMAT ( 20I8 )
      ENDFILE U12
      RETURN
      END
```

C-----

```
C           ***** DRIVER *****
C ---- Reed 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, NOOR, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NOAY(4)
COMMON /A3/ SRNO, REND, ORNO, PEND, FRND, DMND
COMMON /A4/ SINK, SRCE
COMMON /B1/ SRGN(70) /B2/ ORGN(70) /B3/ RIVR(45)
COMMON /B4/ PORT(20) /B5/ FRGN(25)
COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F6/ TR0, RR0, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION
COMMON /G1/ IARC(64000) /G2/ JARC(64000) /G3/ LOWER(64000)
COMMON /G4/ UPPR(64000) /G5/ KOST(64000)
DIMENSION TITL(16)
INTEGER TITL
INTEGER SRND, RENO, DRND, PENO, FRNO, DMND
INTEGER SINK, SRCE
INTEGER SRGN, ORGN, 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 (TITL) of grain.

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

C ---- Read in the number of surplus (NOSR), deficit (NODR), river (MORE),
 C ---- port (NOPE) and foreign regions (NOFR); the time period (NOTP) and
 C ---- the number of days (NDAY) in each time period.
 C ---- Read in the code names of all the surplus regions (SRGN).
 C ---- Check that there is a region before reading the code name.
 C ---- DRGN stores the code numbers for the deficit regions.
 C ---- RIVR stores the code numbers for the river regions.
 C ---- PORT stores the code numbers for the port regions.
 C ---- FRGN stores the code numbers for the foreign regions.

```
READ  (U1,510) NOSR, NODR, MORE, 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 ( MORE .GT. 0 ) READ  (U1,500) ( RIVR(I), I = 1, MORE )
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 * MORE + DRND
PEND     = NOTF * NOPE + REND
FRND     = NOTF * NOFR + PEND
DHND     = NODR + NOFR + FRND
SINK     = DHND + 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) TL0S, RL0S, TL0R, RL0R, BL0R, 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 grain needed.
 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 ( MORE .GT. 0 ) CALL RELVTR
IF ( NOPE .GT. 0 ) CALL PELVTR
CALL DEMAND ( IEXP )

```

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

```

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

```

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

```

SUBROUTINE SURPLS ( IPRD )
COMMON /A1/ NOSR, NODR, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /A3/ SRND, REND, DRND, PEND, FRND, DMND
COMMON /A4/ SINK, SRCE
COMMON /B1/ SRGN(70) /B2/ DRGN(70) /B3/ RIVR(45)
COMMON /B4/ PORT(20)
COMMON /C1/ SDTR(70,70) /C2/ SDRL(70,70)
COMMON /C3/ SRTR(70,45) /C4/ SRRL(70,45)
COMMON /C5/ SPTR(70,20) /C6/ SPRL(70,20)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE
COMMON /E1/ SPLY(70)
COMMON /F1/ STOR(70) /F2/ SCST(4)
COMMON /F3/ TLOS, RLDS, TLOR, RLOR, BLOR, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION
COMMON /G1/ IARC(64000) /G2/ JARC(64000) /G3/ LOWR(64000)
COMMON /G4/ UPPR(64000) /G5/ KOST(64000)
COMMON /H1/ IDEN(70), ALFA(70), BETA(70)
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 (SPLY) and the storage
C ---- capacity (STOR) of each surplus region.

```

READ (U8,520) ( SPLY(I), I = 1, NOSR )
READ (U7,520) ( STOR(I), I = 1, NOSR )

```

C ---- Read in the cost, rail, truck, barge and ship factors, and period
C ---- chosen to be 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 (U7,520) 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, NORR )
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) ( SPRR(I,J), J = 1, NOPE )
1800 CONTINUE
```

C ---- Read in the alphas 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 = NOTP * ( 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 N = 1, NOOR
    NT      = N
    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 + TL0S + 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      = SURL(NF,NT)
IF ( COST .GE. 999. ) GO TO 2400
COST      = ( COST * RAIL + RL0S + 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, MORE, 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 + TL0S + 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 N = 1, MORE
NT      = N
JA      = NOTF * ( NT - 1 ) + DRND
COST    = SRRRL(NF,NT)
IF ( COST .GE. 999. ) GO TO 2700
COST    = ( COST * RAIL + RLDS + 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 N = 1, NOPE
NT      = N
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 + TL0S + 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 + RL0S + 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 (FILE_062.grn)

```

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

```

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

```

SUBROUTINE RELVTR
COMMON /A1/ NOSR, NODR, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /A3/ SRND, REND, DRND, PEND, FRND, DNND
COMMON /B2/ DRGN(70) /B3/ RIVR(45)
COMMON /B4/ PORT(20)
COMMON /C1/ ROTR(70,70) /C2/ RDRL(70,70)
COMMON /C3/ RRBG(70,45) /C5/ RPBG(70,20)
COMMON /D1/ RAIL, TRUCK, BARGE, SNIP, PERIOD
COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION
COMMON /G1/ IARC(64000) /G2/ JARC(64000) /G3/ LOWR(64000)
COMMON /G4/ UPPR(64000) /G5/ KOST(64000)
COMMON /H1/ IDEN(70), ALFA(70), BETA(70)
DIMENSION IDN1(10), IDN2(10), IDN3(25)
INTEGER SRND, REND, DRND, PEND, FRND, DNND
INTEGER DRGN, RIVR, PORT, UPPR
INTEGER U1,U2,U3,U4,U8,U9,U6,U12
LOGICAL ICED
500 FORMAT ( 20A4 )
510 FORMAT ( 20I4 )
520 FORMAT ( 10F8.3 )

```

C ---- Read in the alphas and betas of the deficit regions.
C ---- Reed 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) ( ROTR(I,J), J = 1, NODR )

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

DO 1200 I = 1, MORE
READ  (U3,520) ( RDRL(I,J), J = 1, NOOR )
1200 CONTINUE

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

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

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

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

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

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

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

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

```

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

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

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

```
JA        = NOTF * ( NT - 1 ) + SRND
COST      = ( COST * TRUCK + TLOR + RRID ) * 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 NT = 1, NODR
NT        = M
JA        = NOTF * ( NT - 1 ) + SRND
```

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

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

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

```
DO 2500 NR = 1, NRES
IDNT      = IDN1(M)
```

```

CALL SERIAL ( IDNT, MORE, 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,N) * 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 N = 1, NPES
IDNT      = IDN2(N)
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,N) * 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, MORE, 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(70,45)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD
COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION
COMMON /G1/ IARC(64000) /G2/ JARC(64000) /G3/ LOWR(64000)
COMMON /G4/ UPPR(64000) /G5/ KOST(64000)
DIMENSION IDN3(16)
INTEGER SRND, REND, DRND, PEND, FRND, DMND
INTEGER PORT, FRGN, UPPR
INTEGER U1,U2,U3,U4,U5,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 (skip 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(70) /E3/ FDND(25,4)
COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION
COMMON /G1/ IARC(64000) /G2/ JARC(64000) /G3/ LOWR(64000)
COMMON /G4/ UPPR(64000) /G5/ KOST(64000)
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 (US,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 + NOFR
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.grn)

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(64000) /G2/ JARC(64000) /G5/ KOST(64000)
```

```
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 ---- ****CORN2.F*****
C ---- * DOCUMENTED ON : JUNE 1990 *
C ---- * RUN ON : A 386 machine *
C ---- * COMPILER USED : NODP Fortran *
C ---- * GRAIN USED : CORN *
C ---- * DATA FILES USED : U12 *
C ---- * TRACE FILE : U6 *
C ---- * INPUT FILE FOR CORN3 : UF1 *
C ---- ****
```

C ---- Declaration of all variables used.

```
COMMON /AA/ NR, NN, FSBL, NTIN, TOTL, MAXA
COMMON /B1/ IUV(1500) /B2/ LABL(1500) /B3/ NODE(1500)
COMMON /B4/ MIDL(1500) /B5/ NSAVE(1500)
COMMON /C1/ ILOC(34000) /C2/ ISAVE(34000) /C3/ JSAVE(34000)
COMMON /C4/ JUV(34000)
COMMON /D1/ KOS(68000) /D2/ MIR(68000) /D3/ MA(68000)
COMMON /D4/ MC(68000) /D5/ MF(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, CORN1.
 C ---- This data file contains the relevant information which
 C ---- is necessary for CORN2 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, CORN3.

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

```

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

NTIM      = 0
FSBL      = .TRUE.
MAXA      = 34000

C ---- Read in the data from the output file(generated by) CORN1.F.
C ---- The data read in will be used in this current program.

      READ  (U12,100) NR,NH,( NF(K), K = 1, NR ),( NA(K), K = 1, NR ),
1          ( ILOCK(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 CORN3.F.
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/ IHW(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/ JHW(34000)
COMMON /D1/ KOS(68000) /D2/ MIR(68000) /D3/ MA(68000)
COMMON /D4/ NC(68000) /D5/ NF(68000)
LOGICAL   FSBL
1234 CONTINUE
INFIN=1000000000
IFLOW=0
KLAB=0
KPOT=0
KURK=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 TO 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
M=M+NR
NF(M)=J
NA(M)=I
KOS(M)=KOST
KOS(M)=-KOST
NC(M)=JSIZE(M)-IFLOW
NC(M)=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=N+NR
I = NF(M)
J = NA(M) -
NF(M) = J -
NA(M) = I
IFLOW=NC(M)
KOST=ISAVE(M)+KOS(M)
KOS(M)=KOST
KOS(M)=-KOST
NC(M)=JSAVE(M)-IFLOW
NC(M)=IFLOW-ILO(M)
14 CONTINUE
1401 CONTINUE
C ****
C
C SETUP SECTION
C ****
KL=1
DO 15 K=1,NN1
JK=NODE(K)
NODE(K)=KL
JUV(K)=KL
KL=JK+KL
15 NIDL(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=JUV(I)
MIR(II)=L
JUV(I)=II+1
GO TO 45
44 II=NIDL(I)
MIR(II)=L
NIDL(I)=II-1
45 GO TO(46,47),MIRROR
46 II=JUV(J)
MIR(II)=LL

```

```

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

```

```

IB=NODE(IA)
IE=NIDL(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
INV(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=MIND(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(JHV(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=JHV(I)
155 LABL(J)=0
GO TO 54

C
C      POTENTIAL CHANGE
C
200 KPOT=KPOT+1
201 KSET=NUNS
NEWLAB=0
NUNS=0
INTHRU=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=JHV(K)
KKK=NA(KK)
IF(LABL(KKK)) 310,300,310
300 NONS=NONS-1
JHV(NONS)=KK
310 CONTINUE
C      S-SET RECYCLING FILTER
312 DO 203 K=1,KSET
KK=JHV(K)
KKK=NA(KK)
IF(LABL(KKK)) 203,2021,203
2021 IF(KOS(KK)) 2023,2023,2022
2022 NUNS=NUNS+1
JHV(NUNS)=KK
MIN=MIN0(MIN,KDS(KK))
GO TO 203
2023 NONS=NONS-1
JHV(NONS)=KK
203 CONTINUE

```

```

204 CONTINUE
  IF(IPLL-IPL) 2039,2039,2111
C      FIND MIN(C-BAR) OVER SET S
2039 00 211 LL=IPLL,IPL
  L=IWF(LL)
  JNID=MIDL(L)+1
  JRT=NODE(L+1)-1
  IF(JNID-JRT) 2045,2045,211
2045 DO 210 KK=JNID,JRT
  K=NIR(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  NUNS=NUNS+1
  JWF(NUNS)=K
  MIN=MIN0(MIN,KOS(K))
  GO TO 210
206  NONS=NONS-1
  JWF(NONS)=K
210  CONTINUE
211  CONTINUE
2111 IPLL=IPL+1
  IF(NUNS) 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,IWF(I),I=1,IPL)
  PRINT 2123,(JWF(I),I=NEW,MAXA)
2121 FORMAT(' LABELS, BY NODE'/(5(19,'=',110)) )
2122 FORMAT(' LABELED NODES (IWF)'/
              (10I10))
2123 FORMAT(' THE SET (L,L-), NON-S'/(10I10))
2125 FORMAT('DIS=',IS,' IT=',IS,10X,'INFEASIBLE ARC =',IS)
  RETURN
C      UPDATE RELATIVE COSTS
C
C      UPDATE COST FOR SET S
215  DO 230 I=1,NUNS
  IJ=JWF(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
  IWF(IPL)=NODEB

```

```

IF(NODEB-1S) 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=JI+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(IPL-IPL) 84,200,84
980 DO 981 I=1,IPL
   J=JWV(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, MN, FSBL, NTIM, TOTL, MAXA
COMMON /B1/ IWV(1500) /B2/ LABL(1500) /B3/ NODE(1500)
COMMON /B4/ MIDL(1500) /B5/ NSAVE(1500)
COMMON /C1/ ILO(36000) /C2/ ISAVE(36000) /C3/ JSIZE(36000)
COMMON /C4/ JWV(36000)
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
    IFRM=NODE(I)
    IT0=NODE(I+1)-1
    PRINT 910,IFRM,MIDL(I),IT0,(K,MIR(K),K=IFRM,IT0)
910 FORMAT(3I6/(20I6))
  RETURN
  3  ITEMP=MIR(MID)
    MIR(MID)=INDEX
    MIR(II)=ITEMP
    MIDL(I)=MID-1
    RETURN
    ENTRY LEFT(I,INDEX)
    MID=MIDL(I)+1
    IB=NODE(I+1)-1
    DO 10 II=MID,IB
      IF(MIR(II)-INDEX) 10,12,10
10  CONTINUE
  KWAY=2
  GO TO 2
12  ITEMP=MIR(MID)
    MIR(MID)=INDEX
    MIR(II)=ITEMP
    MIDL(I)=MID
    RETURN
900 FORMAT(5H NODE,I5,5H ARC, I5, 16H LOST ON SHIFT ,I4,4H LOC ,I4
1  )
  ENTRY DUMPO
  NLLINES = 1
  ID = INDEX
  PRINT 1120, ID
  DO 1070 N=1,NLLINES
    N=N+NR
    I=NA(N)
    J=NA(N)
    L=IL0(N)
    K=JSAVE(N)
    KOST=ISAVE(N)
    KBAR=KOS(N)
    IFLOW=K-NC(N)
    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,N,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 ---- *          CORN3.F
C ---- *      Documented on    SEPT 1990
C ---- *      Run on           a 386 machine
C ---- *      Compiler used   NDP Fortran
C ---- *      Grain used      CORN
C ---- *      Data files used F1,F2,F3,F4,F8,F9
C ---- *      Intermediate input file UF1
C ---- *      Final output file FD19
C ---- *
C ---- ****
```

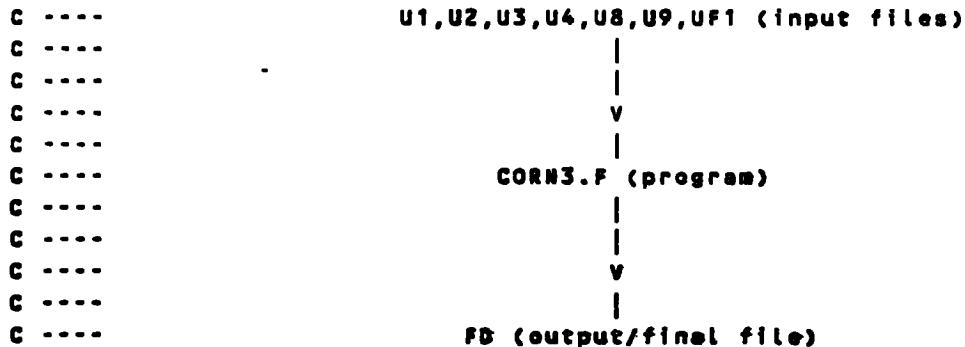
```
C ---- Declaration of 2 common blocks.
C ---- FLOW is the array that stores the output from the previous
C ---- program CORN2.F, it is used as an input for this program.
C ---- K is counter.
C ---- U1,U2,U3,U4,U8,U9,UF1,FD store unit numbers for the files.
```

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

```
C ---- Unit number for each file.
C ---- U1 is the unit number for data file, F1.crn.
C ---- F1.crn contains information about the model.
C ---- U2 is the unit number for data file, FILE02.crn.
C ---- FILE02.crn contains information on TRUCK mileage.
C ---- U3 is the unit number for data file, F3.crn.
C ---- F3.crn contains information on RAIL costs.
C ---- U4 is the unit number for data file, F4.crn.
C ---- F4.crn contains information on BARGE costs..
C ---- U8 is the unit number for data file, F8.crn.
C ---- F8.crn contains information about SUPPLIES and DEMANDS.
C ---- U9 is for the unit number for data files, FILE09.crn.
C ---- FILE09.crn contains the NAMES of all the regions.
C ---- UF1 is the unit number for the input file produced by CORN2.F.
C ---- It is the output file from CORN2.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 ---- To open files for reading and writing

```

OPEN (UNIT = U1, FILE = 'F1.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U2, FILE = 'FILE02.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U3, FILE = 'F3.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U4, FILE = 'F4.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U8, FILE = 'F8.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = U9, FILE = 'FILE09.CRN', STATUS = 'UNKNOWN',
1      ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')
OPEN (UNIT = UF1, FILE = 'FLOW1_2.CRN', 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.CRN with unit number
C ---- UF1. The processed data is the output of CORN2.F.
C ---- N is the total number of data processed in CORN2.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.

```

100  READ  (UF1,100) N, ( FLOW(I), I = 1, N )
100  FORMAT(20I8)
      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, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /B1/ SRGN(70) /B2/ DRGN(70) /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/ SUNG, SUNT, SUNR, SUNB, SUMS, SUNH
COMMON /GB/ TOPR, TODD, TOFD
COMMON /GC/ TGSO(70), TGRD(70), TGRF(70)
COMMON /GD/ TGRR(70,4), TGRP(70,4)
COMMON /R1/ NAM1(70,3), NAM2(70,3), 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(CORN) 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 the title(TITL) of this model.
 C ---- Read in the number of surplus(NOSR), deficit(NODR), river(MORE),
 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, MORE, 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 ( MORE .GT. 0 ) READ  (U1,500) ( RIVR(I), I = 1, MORE )
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 ---- Initialize 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 ---- Initialise the arrays, TGRR and TGRP for each time factor(NOTF).
C ---- TGRR stores Total amount of Grain shipped to a River region.
C ---- TGRP stores Total amount of Grain shipped to a Port region.

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

C ---- Initialisation of all variables used.
C ---- TOPR Total grain Produced from surplus regions.
C ---- TODD Total grain Demanded by Deficit regions.
C ---- TOFD Total grain Demanded by Foreign regions.
C ---- SUMG SUM of storage cost.
C ---- SUMT SUM of Truck cost.
C ---- SUMR SUM of Rail cost.
C ---- SUMB SUM of Barge cost.
C ---- SUMS SUM of Shipping cost.

```

C ---- SUMH SUM of Handling cost.

```

TOPR      = 0.0
TODD      = 0.0
TOFD      = 0.0
SUNG      = 0.0
SUNT      = 0.0
SUMR      = 0.0
SUMB      = 0.0
SUMS      = 0.0
SUMH      = 0.0

```

C ---- Read in the loading and unloading costs.
C ---- TL0S Truck Loading (country elevator) cost.
C ---- RL0S Railcar Loading (country elevator) cost.
C ---- TL0R Truck LOading (River location) cost.
C ---- RL0R Railcar LOading (River location) cost.
C ---- BL0R 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) TL0S, RL0S, TL0R, RL0R, BL0R, 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, MORE
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
 relating to the surplus regions.
 C ---- Call subroutine RELVTR and PELVTR if there is more than
 one river region and one port region involved.
 C ---- RELVTR and PELVTR deal with the River rEGions and Port rEGions
 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(70), S(70,4)
COMMON /A1/ NOSR, NODR, NORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /B1/ SRGN(70) /B2/ DRGN(70) /B3/ RIVR(45)
COMMON /B4/ PORT(20)
COMMON /C1/ SDTR(70,70) /C2/ SDRL(70,70)
COMMON /C3/ SRTR(70,45) /C4/ SRRL(70,45)
COMMON /C5/ SPTR(70,16) /C6/ SPRL(70,16)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE
COMMON /E1/ SPLYL(70)
COMMON /F1/ STOR(70) /F2/ SCST(4)
COMMON /F3/ TLGS, RLGS, TLGR, RLGR, BLGR, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
COMMON /GA/ SUNG, SUMT, SUMR, SUMB, SUMS, SUMH
COMMON /GB/ TOPR, TODD, TOFD
COMMON /GC/ TGSO(70), TGRD(70), TGRF(70)
COMMON /GD/ TGRR(70,4), TGRP(70,4)
COMMON /R1/ NAM1(70,3), NAM2(70,3), NAM3(45,3)
COMMON /R2/ NAM4(20,3), NAM5(25,3)
COMMON /N1/ IDEN(70), ALFA(70), BETA(70)
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 ( 1HT, //, 9X, 16HORIGIN/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.

```
READ (U8,520) ( SPLY(I), I = 1, NOSR )
```

```

READ  (U1,520) ( STOR(I), I = 1, NOSR )
READ  (U1,520) COST, RAIL, TRUCK, BARGE, SHIP, PERIOD
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, MORE )
1600 CONTINUE

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

DO 1700 I = 1, NOSR

```

```

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

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

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

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

      READ  (U1,520) ( ALFA(I), BETAC(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  NODE 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      = BETAC(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 GENFLQ 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 ---- TL0S 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 N = 1, NODER
NT      = N
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 GENFLQ ( COST, TL0S, 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, NODEF ),
2           TT/OPERATION, COST*OPERATION, TCST, HCST

```

```

C ---- Update the following totals:
C ---- T0DD 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 ---- SUNH 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
SUNH      = SUNH + 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(N)   = TGRD(N) + 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, MORE, 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, TL0S, TRIR, T, TT, TCST, HCST )
SUMH      = SUMH + HCST
SUMT      = SUMT + TCST
  
```

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

```

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

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

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

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

```
DO 2550 N = 1, NOTF
TGRR(NT,N)= TGRR(NT,N) + R(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 GENFL0 to 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 N = 1, MORE
NT      = N
COST    = SRRRL(NF,NT)
IF ( COST .GE. 999. ) GO TO 2700
COST    = COST * RAIL / 100.0
CALL GENFL0 ( COST, RL0S, 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 to 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 N = 1, NOPE
  NT      = N
  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, TLGS, 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(N)) 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(N), ( NAM6(N,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 to 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 M = 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 ABILENE, TX    5702       4424       2212       0        144662

```

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)
  SUNG      = SUNG + 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, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /B2/ DRGN(70) /B3/ RIVR(45)
COMMON /B4/ PORT(20)
COMMON /C1/ RDTR(70,70) /C2/ RDRL(70,70)
COMMON /C3/ RRBG(70,45) /C5/ RPBG(70,16)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE
COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
COMMON /GA/ SUNG, SUNT, SUNR, SUNB, SUNS, SUNH
COMMON /GB/ TOPR, TODD, TOFD
COMMON /GC/ TGSC(70), TGRD(70), TGRF(70)
COMMON /GD/ TGRR(70,4), TGRP(70,4)
COMMON /RT/ NAM1(70,3), NAM2(70,3), NAM3(45,3)
COMMON /R2/ NAM4(20,3), NAM5(25,3)
COMMON /H1/ IDEN(70), ALFA(70), BETA(70)
DIMENSION IDN1(10), IDN2(10), IDN3(25)
INTEGER U1,U2,U3,U4,U8,U9,UF1,FD

```

C ---- The layout of the output.

```

500 FORMAT ( 20A4 )
510 FORMAT ( 20I4 )

```

```

520 FORMAT ( 10F8.3 )
700 FORMAT ( 1H1, //, 9X, 18HORIGIN/DESTN MODE, 6X, 6HSUPPLY, 14X,
    1      16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
    2      7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4I10, 4X,
    3      8HSHIPMENT, 5X, 28HCOST - HAULING COST COSTS, // )
710 FORMAT ( 7, 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 alphas 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(I,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, MORE
READ  (U3,520) ( RBRL(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, MORE
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,N;B,Ch,Port.

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

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

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

C ---- Print out the heading of the output file i.e.,
C ORIGIN/DESTN NODE 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 (NAME3(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, MORE
WRITE (FD,710) RIVR(I), ( NAME3(I,L), L = 1, 3 ),
I           ( 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 ALP.
 C ---- Subroutine GENFLG 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 N = 1, NODR
IDNT      = IDEN(N)
CALL SERIAL ( IDNT, MORE, RIVR, NT )
IF ( NF .NE. NT ) GO TO 2200
NT        = N
COST      = RDTR(1,NT)
IF ( COST .GE. 9999. ) GO TO 2200
ALF       = ALFA(NT)
```

```

BET      = BET(A,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 ---- (SUNT) and the handling cost(SUNH).

IF ( TT .LE. 0.0 ) GO TO 2200
SUNT = SUNT + TCST
SUNH = SUNH + 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 NT = 1, NODR
NT      = N
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(N) is the code number of the deficit region.
 C ---- NAN2(N,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(N), ( NAN2(N,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(N))
 C ---- to this deficit region.

```
TODD      = TODD + RT
TGRD(N)  = TGRD(N) + 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 ATF.
 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 N = 1, NRES
```

```

IDNT      = IDN1(N)
CALL SERIAL ( IDNT, NORE, RIVR, NT )
IF ( NT .EQ. NF ) GO TO 2500
IF ( NT .EQ. 0 ) GO TO 2500
COST      = RRBG(NF,N)
IF ( COST<=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
IONT      = 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, BGOR, 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, NOOR, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(6)
COMMON /B4/ PORT(20) /B5/ FRGN(25)
COMMON /C3/ PFSP(70,45)
COMMON /D1/ RAIL, TRUCK, BARGE, SHIP, PERIOD, CHOICE

```

```

COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F4/ TR0D, RR0D, TR0R, RR0R, BR0R, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
COMMON /GA/ SUNG, SUMT, SUMR, SUMB, SUMS, SUMH
COMMON /GB/ TOPR, TODD, TOFD
COMMON /GE/ TGSO(70), TGRD(70), TGRF(70)
COMMON /GD/ TGRR(70,4), TGRP(70,4)
COMMON /R1/ NAM1(70,3), NAM2(70,3), 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 ( 1N1, //, 9X, 18HORIGIN/DESTN MODE, 6X, 6HSUPPLY, 14X,
    1      16HSHIPMENT BY TIME, 16X, 5HTOTAL, 6X, 4HUNIT,
    2      7X, 5HTOTAL, 5X, 8HHANDLING, /, 39X, 4I10, 4X,
    3      8HSNIPMENT, 5X, 28HCOST HAULING COST COSTS, // )
710 FORMAT ( /, 5X, 1HP, A4, 1X, 3A4, 16X, 6F10.0, / )
720 FORMAT ( 5X, 1HF, A4, 1X, 3A4, 2X, 2HS , 12X, 6F10.0,
    1      F12.0, F10.5, 2F12.0 )

```

C ---- Read in the alphas and bases of the deficit regions.

C ---- Read in the selected (port) barge shipping

C ---- locations(IDEN(I)).

```

READ ..(U1,520) ( ALFA(I), BETAC(I), I = 1, NO0R )
READ (U1,500) ( IDEN(I), I = 1, NO0R )

```

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 GENFL0 is used to calculate the hauling
C ---- skipping 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.CRN).

DO 1300 N = 1, NOFR
NT      = N
COST    = PFSP(NF,NT)
IF ( COST .GE. 999. ) GO TO 1300
COST    = COST * SHIP
CALL GENFL0 ( 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 ---- (SUMS) 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, MORE, NOPE, NOFR
COMMON /A2/ NOTP, NOTF, NDAY(4)
COMMON /B1/ SRGN(70) /B2/ DRGN(70) /B3/ RIVR(45)
COMMON /B4/ PORT(20) /B5/ FRGN(25)
COMMON /E2/ DDND(70) /E3/ FDND(25,4)
COMMON /F3/ TL0S, RL0S, TL0R, RL0R, BL0R, SLOP
COMMON /F4/ TRID, RRID, TRIR, RRIR, BRIR, TRIP, RRIP, BRIP
COMMON /G0/ K, OPERATION /G1/ FLOW(50000)
COMMON /G2/ SUNG, SUMT, SUNR, SUMB, SUMS, SUMH
COMMON /G3/ TOPR, TODD, TOFD
COMMON /GC/ TGSOC(70), TGRD(70), TGRF(70)
COMMON /GD/ TGRR(70,4), TGRP(70,4)
COMMON /R1/ NAM1(70,3), NAM2(70,3), NAM3(45,3)
COMMON /R2/ NAM4(20,3), NAM5(25,3)
INTEGER FLOW
INTEGER UT,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, 8NSHIPMENT, / )
720 FORMAT ( 5X, 1HD, A6, 1X, 3A6, 16X, 6F10.0, F12.0 )
730 FORMAT ( 5X, 1HF, A6, 1X, 3A6, 16X, 6F10.0, 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 (US,520) ( FDND(I,N), N = 1, NOTF )
1500 CONTINUE
```

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

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

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

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

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

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

```
RETURN
END

C-----  
C           ***** GENFLO *****  
C ---- Calculate the amount of grain flow per time period,  
C ---- total hauling cost(TCST) and handling cost (HCST).  
  
SUBROUTINE GENFLO ( COST, COUT, CRIN, A, AT, TCST, HCST )  
DIMENSION A(4)  
COMMON /A2/ NOTP, NOTF, NDAY(4)  
COMMON /G0/ K, OPERATION /G1/ FLOW(50000)  
INTEGER FLOW  
AT      = 0.0  
TCST    = 0.0  
HCST    = 0.0  
DO 1100 N = 1, NOTF  
K      = K + 1  
A(N)   = FLOW(K)  
AT      = AT + A(N)  
1100 CONTINUE  
  
C ---- To find the total hauling cost(TCST) e.g., for Abilene,  
C ---- TCST = 5246*0.20052*1000  
  
TCST      = AT * COST * 1000.0  
  
C ---- COUT and CRIN are the loading and unloading costs.  
C ---- The handling cost, for Abilene,  
C ---- HCST = 5246 * (6.47 + 6.59) * 10  
  
HCST      = AT * ( COUT + CRIN ) * 10.00  
RETURN
END
```

C

F1.CRN

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

C-----

C Explanation for f1.crn.

C This is the first file which gives the overall detail of the model.
 C The explanation for each line of f1.crn is given below.

C-----

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

CORN SHIPMENT PROBLEM, BAIE CONEAU ADDED - FULLER, GRANT, TEH, FELLIN 0

C-----

C This line shows the number of regions involved in this model.
 C If there were any changes in the number of regions, the corresponding
 C number must be changed.
 C e.g. if the number of surplus regions were increased to 59 then change 58
 C to 59.

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

C Maximum number of surplus regions allowed = 70
 C Maximum number of deficit regions allowed = 70
 C Maximum number of river regions allowed = 43
 C Maximum number of port regions allowed = 20
 C Maximum number of foreign regions allowed = 25

C In this model, the

C Number of surplus regions = 58
 C Number of deficit regions = 56
 C Number of river regions = 43
 C Number of port regions = 20

C Number of foreign regions = 25

C The format used is :
C 4 fields i.e. **dd where d is a digit.
C e.g. **58 where * is a space.

58 56 43 20 25

0

C-----

C This line shows the number of periods, days per period in this
C model. There are 3 periods per year.

C The number of days in each period :
C number of days in the 1st period = 121
C number of days in the 2nd period = 121
C number of days in the 3rd period = 123

C If the number of periods were different, alter 3 to the new number and
C add the number of days in the new period.

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

3 121 121 123

0

C-----

C These lines show the codes used for the 58 surplus regions in this
C model.
C For example, 111 is the code number for Alexandria LA, and
C 085 is the code number for Ames, IA
C If there were any changes in the number of surplus regions,
C the code of the new surplus region would be added to the end of the list.
C If there were a deletion, then the code involved would be deleted from
C this list.

111 085 139 068 163 015 024 162 154 351 032 092 153 094 144 083 131 084 130 121
134 164 141 132 066 034 142 113 172 035 138 033 123 073 082 093 021 135 037 088
089 137 099 143 087 391 076 096 074 06A 081 152 122 291 145 151 045 075 0

C-----

C These lines show the codes for the 56 deficit regions in this model.
C For example, 064 is the code number for Abilene TX, and
C 213 is the code number for Albany GA.

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

064 213 061 211 707 301 052 071 351 098 086 261 251 231 391 06H 051 101 201 191

050 192 102 171 342 047 161 181 097 281 140 183 715 282 104 062 054 212 202 182
053 713 242 411 302 042 381 311 221 103 401 066 292 241 222 133 0

C-----

C These lines show the codes for the 63 river barge ports in this
C model.

C For example, 601 is the code number for St Paul MN and
C 602 is the code number for Winona MN.

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

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

C-----

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

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

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

C-----

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

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

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

C-----

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

C Truck Load Country Elevation = 10.422
C Rail Load Country Elevation = 10.368
C -Truck Load River Location = 8.940
C Rail Load River Location . = 9.330
C Barge Load River Location = 7.758
C Ship Load Port Location = 5.660

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

10.422 10.368 8.940 9.330 7.758 5.660

C-----

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

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

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

8.520 **6.774** **6.450** **6.318** **11.076** **6.763** **4.082** **5.097**

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

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

C.....

The grain storage cost, rail, truck, barge, and ship adjustment factors, are as follows:

C	storage cost factor (cents/bushel/year)	=	11.566
C	rail cost multiplier	=	1.000
C	truck mileage multiplier	=	1.000
C	berge cost multiplier	=	1.000
C	ship rate multiplier	=	1.000
C	period blocked	=	1.000

The 1st period is blocked in the regions in the Lakes.

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

C The format used is:

c 8 fields i.e. dddd.ddd where d is a digit.

C If there is no digit in front of the number, then add spaces.
C e.g. 1.000 becomes **1.000 where * is a space.

11.566 **1.000** **1.000** **1.000** **1.000** **1.000**

9

c Conversion factor and choice of measurement

The conversion factor is 1,000.

Choice of measurement

Conversion factor for short ton is 2000/56 = 35.714

and for potato tops 2306/56 = 78.363

C Choices of measurement can be one of these:

1 000 kg per bushel

3,000 is too much too

3,000 is too small ton

1 000 1 000

1

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

5.830	0.195	5.654	0.191	6.486	0.208	6.066	0.200	6.406	0.207
6.440	0.207	6.327	0.205	6.406	0.207	6.597	0.210	6.339	0.205
7.060	0.224	7.246	0.233	6.597	0.210	7.246	0.233	6.434	0.207
6.486	0.208	7.055	0.224	6.486	0.208	7.055	0.224	6.406	0.207
6.434	0.207	7.055	0.223	6.508	0.208	6.339	0.205	6.460	0.207
6.434	0.207	6.353	0.206	6.460	0.207	6.339	0.205	5.654	0.191
6.839	0.215	6.486	0.208	6.339	0.205	5.830	0.195	7.246	0.233
5.165	0.182	7.055	0.224	6.486	0.208	6.486	0.208	7.246	0.233
6.434	0.207	6.486	0.208	6.791	0.214	5.589	0.190	7.246	0.233
6.486	0.208	6.597	0.210	6.792	0.214	6.597	0.210	6.595	0.210
6.791	0.214	6.339	0.205	6.434	0.207	6.434	0.207	7.055	0.224

7.055 0.224 7.055 0.224 6.508 0.208

0

C-----

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

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

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

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

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

C e.g., the 1st entry is the 1st surplus region, Alexandria. It is not linked
C to any barge loading region. So 999 is used to indicate that Alexandria is
C not linked to any barge loading port.

C e.g., the 2nd entry indicates that the 2nd surplus region, Ames IA (085)
C is linked to the barge loading port, Dubuque (604).

999 604 615 631 618 602 609 999 618 610 609 612 618 613 619 603 605 610 615 602
608 999 618 616 630 611 619 999 619 611 608 611 604 602 603 607 609 621 609 611
606 616 621 620 611 999 602 608 601 631 609 618 602 999 620 618 613 609 0

C-----

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

5.654	0.191	5.866	0.195	5.830	0.195	5.654	0.191	6.626	0.211
6.791	0.214	6.188	0.202	7.246	0.233	6.350	0.206	6.514	0.209
6.839	0.215	6.350	0.212	5.654	0.191	5.165	0.182	6.177	0.202
6.177	0.202	6.842	0.215	6.508	0.208	6.406	0.207	7.246	0.233
5.659	0.191	5.866	0.195	6.425	0.207	6.200	0.202	6.327	0.205
5.659	0.191	6.066	0.200	6.200	0.202	6.350	0.212	7.231	0.223
6.248	0.203	6.165	0.201	6.626	0.211	6.508	0.208	6.079	0.200
5.654	0.191	6.840	0.215	6.839	0.215	6.626	0.211	6.350	0.212
5.654	0.191	6.350	0.212	6.508	0.208	5.675	0.192	6.486	0.208
5.659	0.191	6.042	0.199	5.936	0.197	6.353	0.206	6.626	0.211
5.589	0.190	6.275	0.204	5.654	0.191	6.353	0.206	6.626	0.211
5.589	0.190								0

C-----

C These lines show the codes for barge shipping ports with each code
 C number (barge unloading port) associated with a deficit region to which
 C it sends grain.

C For example, the 1st deficit region, Abilene, TX (064), is linked to no
 C other barge region.

C the 19th deficit region, Gadsden AL (201) is linked to
 C Guntersville, AL (628).

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

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

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

999 628 999
 999 999 999 625 999 999 999 999 999 999 999 626 999 999 999 999 999 999 999 999 627 628 625
 999 999 626 999 999 999 999 628 999 999 999 999 626 628 999 0

C-----

C The number of shipping points in this model.

C Number of barge receiving points = 5

C Number of ports which receive from barge = 5

5 5 0

C-----

C The codes for the 5 barge receiving points in this model:

C 625 NASHVILLE, TN

C 626 KNOXVILLE, TN

C 627 CHATANOOGA, TN

C 628 GUNTERSVILLE, AL

C 629 FLORENCE, AL

C and the codes for the 5 ports which receive from barge are:

C 702 NEW ORLEANS, LA

C 701 MOBILE, AL

C 703 GALVESTON, TX

C 710 CHICAGO, IL

C 713 PORTLAND, OR

625 626 627 628 629 702 701 703 710 713

0

C-----

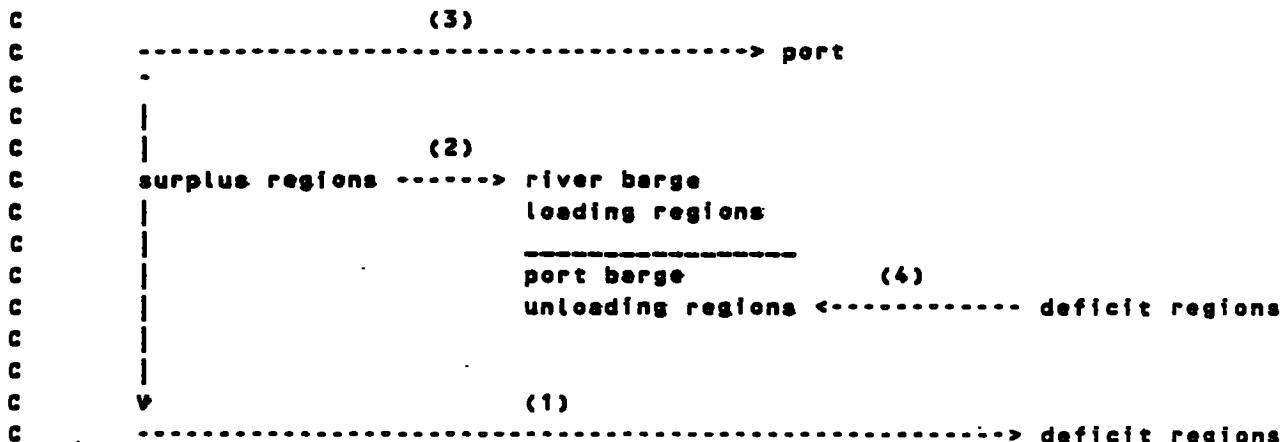
C

F2.CRN

C This file stores all the truck mileages in this model.
 C F2.CRN 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 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 58 surplus regions and 56 deficit regions.

C surplus region 1 -----> deficit region 1
 C -----> deficit region 2
 C .
 C .
 C -----> deficit region 56

C surplus region 2 -----> deficit region 1
 C -----> deficit region 2
 C .
 C .
 C -----> deficit region 56

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 56 -----> corresponding deficit region

C-----

C Explanation for 1st part

C This includes the truck mileage from each of the 58 surplus regions
C to each of the 56 deficit regions. Thus, resulting in a 58 by 56 matrix.

C Diagrammatically,

C surplus region 1 -----> deficit region 1

C -----> deficit region 2

C .

C .

C -----> deficit region 56

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

C -----> deficit region 2

C .

C .

C -----> deficit region 56

C .

C .

C .

C surplus region 58 -----> deficit region 1

C -----> deficit region 2

C .

C .

C -----> deficit region 56

C The first surplus region is Alexandria (111).

C The distances from Alexandria to all deficit regions are given on the first
C 6 lines (see below):

C Alexandria (111) to the 1st deficit region, Abilene (064)

C = 490 miles

C Alexandria (111) to the 2nd deficit region, Albany (213)

C = 549 miles

C The next matrix is for the next surplus region, the 2nd region and its
C mileages to all 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 Alexandria were not a surplus region, then the first 6 lines would
C be removed. The file would start with 862 as the first entry.

490.000	549.000	661.000	562.000	1185.000	1811.000	504.000	1293.000	1618.000	478.000
881.000	922.000	1036.000	777.000	1084.000	927.000	562.000	366.000	471.000	247.000
743.000	212.000	388.000	551.000	1675.000	545.000	1157.000	422.000	507.000	1202.000
802.000	666.000	1694.000	1226.000	263.000	628.000	372.000	643.000	417.000	546.000
479.000	2316.000	989.000	791.000	1567.000	713.000	1549.000	2385.000	557.000	192.000
1223.000	217.000	1435.000	869.000	820.000	733.000				0.000
862.000	1054.000	775.000	900.000	983.000	1479.000	435.000	436.000	910.000	425.000
103.000	771.000	995.000	1066.000	684.000	1159.000	531.000	513.000	819.000	755.000
670.000	916.000	539.000	587.000	1105.000	392.000	495.000	607.000	375.000	822.000
377.000	824.000	1742.000	927.000	594.000	895.000	548.000	1058.000	911.000	658.000
577.000	1788.000	1142.000	990.000	1179.000	397.000	1078.000	1724.000	1115.000	693.000
1417.000	740.000	1034.000	996.000	1357.000	357.000				0.000
1050.000	858.000	1015.000	695.000	701.000	1197.000	648.000	607.000	1196.000	434.000
189.000	487.000	711.000	796.000	962.000	1399.000	771.000	653.000	634.000	648.000
910.000	795.000	494.000	382.000	1393.000	617.000	310.000	465.000	547.000	540.000
104.000	554.000	2020.000	619.000	619.000	1112.000	726.000	853.000	743.000	453.000
765.000	2072.000	858.000	1230.000	897.000	660.000	1356.000	2002.000	945.000	759.000
1657.000	872.000	752.000	712.000	1152.000	124.000				0.000
216.000	909.000	483.000	919.000	1519.000	2145.000	496.000	1336.000	1453.000	685.000
994.000	1198.000	1370.000	1134.000	906.000	584.000	461.000	454.000	828.000	565.000
578.000	576.000	602.000	827.000	1497.000	537.000	1386.000	699.000	562.000	1478.000
1064.000	1000.000	1374.000	1554.000	477.000	381.000	364.000	1000.000	774.000	823.000
389.000	2069.000	1346.000	510.000	1856.000	634.000	1302.000	2138.000	887.000	337.000
903.000	224.000	1711.000	1226.000	1150.000	985.000				0.000
1209.000	855.000	180.000	688.000	552.000	1048.000	807.000	778.000	1373.000	593.000
376.000	393.000	582.000	745.000	1147.000	1568.000	940.000	812.000	647.000	765.000
1079.000	883.000	639.000	452.000	1580.000	776.000	453.000	580.000	706.000	391.000
195.000	503.000	2205.000	496.000	766.000	1271.000	885.000	846.000	764.000	474.000
924.000	2243.000	751.000	1395.000	748.000	811.000	1541.000	2173.000	942.000	904.000
1826.000	1017.000	603.000	618.000	1145.000	266.000				0.000
1138.000	1638.000	948.000	1495.000	1499.000	1995.000	917.000	335.000	418.000	1023.000
692.000	1288.000	1512.000	1600.000	671.000	1304.000	884.000	1068.000	1434.000	1353.000
823.000	1510.000	137.000	1182.000	758.000	881.000	680.000	1223.000	927.000	1338.000
914.000	1358.000	1617.000	1443.000	1166.000	1068.000	1043.000	1653.000	1527.000	1253.000
935.000	1265.000	1659.000	1144.000	1695.000	691.000	916.000	1195.000	1731.000	1214.000
1506.000	1223.000	1550.000	1513.000	1952.000	934.000				0.000
963.000	1355.000	832.000	1207.000	1240.000	1736.000	580.000	307.000	660.000	679.000
390.000	1029.000	1253.000	1341.000	686.000	1196.000	628.000	724.000	1120.000	1009.000
707.000	1166.000	793.000	888.000	994.000	537.000	527.000	908.000	583.000	1079.000
655.000	1099.000	107.000	1184.000	822.000	952.000	715.000	1359.000	1212.000	959.000
678.000	1544.000	1400.000	1034.000	1436.000	435.000	1006.000	1474.000	1416.000	893.000
1454.000	906.000	1291.000	1254.000	1658.000	648.000				0.000

1313.000 988.0001284.000 821.000 675.0001171.000 911.000 699.0001404.000 697.000
 480.000 526.000 705.000 892.0001251.0001672.0001044.000 916.000 780.000 883.000
 1183.0001016.000 752.000 580.0001684.000 880.000 324.000 698.000 810.000 514.000
 299.000 650.0002309.000 593.000 877.0001375.000 989.000 979.000 897.000 607.000
 1028.0002251.000 874.0001499.000 871.000 915.0001645.0002181.0001075.0001017.000
 1930.0001130.000 726.000 751.0001278.000 370.000 0.000
 1269.000 757.0001240.000 594.000 314.000 891.000 867.0001000.0001595.000 653.000
 597.000 131.000 312.000 514.0001307.0001727.0001626.000 872.000 586.000 758.000
 1457.000 858.000 674.000 445.0001801.000 836.000 675.000 573.000 766.000 224.000
 1685.000 407.0002322.0002214.000 786.0001639.000 945.0002424.0002298.0002024.000
 984.000 790.0002430.0001455.000 620.000 910.0001749.0002395.0002502.000 926.000
 1623.0001039.000 482.000 357.000 979.000 368.000 0.000
 957.0001716.000 690.0001574.0001697.0002193.000 872.000 848.000 280.0001028.000
 817.0001485.0001709.0001762.000 275.000 928.000 797.0001030.0001482.0001358.000
 628.0001510.0001142.0001261.000 391.000 849.0001108.0001270.000 919.0001536.000
 1091.0001504.0001104.0001615.0001171.000 810.000 980.0001732.0001564.0001332.000
 861.0002071.0001856.000 792.0001893.000 646.000 403.0001094.0001768.0001149.000
 1059.0001145.0001748.0001710.0002030.0001071.000 0.000
 866.0001570.000 626.0001477.0001513.0002009.000 744.000 710.000 424.000 882.000
 633.0001301.0001525.0001596.000 303.000 914.000 679.000 902.0001335.0001212.000
 529.0001369.000 996.0001104.000 575.000 721.000 924.0001123.000 785.0001352.000
 907.0001347.0001288.0001457.0001025.000 746.000 863.0001575.0001418.0001175.000
 744.0001258.0001672.000 777.0001709.000 518.000 587.0001238.0001622.0001032.000
 1138.0001036.0001564.0001526.0001874.000 887.000 0.000
 724.000 924.000 637.000 781.000 986.0001532.000 297.000 606.0001043.000 263.000
 237.000 727.000 962.000 969.000 637.0001021.000 393.000 374.000 689.000 593.000
 532.000 754.000 377.000 468.0001187.000 254.000 629.000 477.000 237.000 872.000
 375.000 711.0001679.000 963.000 424.000 757.000 410.000 939.000 781.000 539.000
 439.0001828.0001084.000 852.0001232.000 262.0001105.0001857.000 985.000 547.000
 1279.000 602.0001087.000 935.0001238.000 312.000 0.000
 1184.000 662.0001176.000 495.000 413.000 990.000 803.000 956.0001551.000 589.000
 553.000 120.000 355.000 497.0001243.0001595.000 937.000 808.000 487.000 659.000
 1111.000 759.000 575.000 346.0001757.000 772.000 631.000 474.000 702.000 323.000
 282.000 308.0002258.000 390.000 687.0001267.000 881.000 643.000 614.000 350.000
 920.0002408.000 491.0001391.000 714.000 846.0001685.0002351.000 749.000 827.000
 1853.000 940.000 569.000 345.000 939.000 304.000 0.000
 632.000 855.000 587.000 743.0001030.0001590.000 223.000 720.0001104.000 167.000
 346.000 744.000 979.000 950.000 671.000 975.000 343.000 260.000 638.000 497.000
 486.000 654.000 281.000 449.0001246.000 180.000 738.000 424.000 128.000 925.000
 444.000 692.0001633.0001007.000 310.000 694.000 308.000 901.000 703.000 520.000
 2347.0001887.0001083.000 802.0001290.000 244.0001164.0001918.000 907.000 433.000
 1233.000 500.0001145.000 934.0001169.000 381.000 0.000
 1043.000 616.0001014.000 449.000 583.0001153.000 641.000 843.0001432.000 427.000
 426.000 298.000 533.0001255.0001099.0001433.000 775.000 646.000 408.000 544.000
 949.000 644.000 431.000 228.0001627.000 610.000 518.000 359.000 540.000 478.000
 148.000 302.0002096.000 560.000 556.0001105.000 719.000 607.000 525.000 235.000
 758.0002271.000 643.0001229.000 853.000 684.0001548.0002238.000 703.000 696.000
 1691.000 809.000 708.000 494.000 906.000 167.000 0.000
 1016.0001074.000 929.000 911.000 946.0001442.000 589.000 377.000 940.000 522.000
 118.000 735.000 959.0001047.000 856.0001313.000 685.000 667.000 850.000 808.000
 824.000 969.000 622.000 598.0001218.000 546.000 342.000 649.000 529.000 785.000
 359.000 805.0001914.000 890.000 725.0001049.000 702.0001069.000 953.000 669.000
 731.0001824.0001106.0001144.0001142.000 554.0001230.0001754.0001157.000 847.000
 1571.000 894.000 997.000 960.0001368.000 353.000 0.000
 1058.000 881.000 996.000 718.000 725.0001221.000 656.000 586.0001166.000 442.000

170.000	514.000	738.000	825.000	943.000	1380.000	752.000	661.000	657.000	666.000
891.000	815.000	502.000	405.000	1374.000	613.000	304.000	485.000	555.000	564.000
133.000	583.000	2001.000	669.000	627.000	1116.000	734.000	876.000	766.000	476.000
773.000	2050.000	885.000	1211.000	921.000	621.000	1337.000	1980.000	968.000	767.000
1638.000	880.000	776.000	739.000	1175.000	144.000				0.000
811.000	1140.000	710.000	986.000	1075.000	1571.000	414.000	445.000	818.000	481.000
195.000	863.000	1087.000	1158.000	592.000	1074.000	478.000	526.000	905.000	811.000
585.000	968.000	595.000	673.000	1013.000	371.000	587.000	693.000	385.000	914.000
469.000	916.000	1650.000	1019.000	624.000	830.000	549.000	1144.000	997.000	744.000
526.000	1696.000	1234.000	912.000	1271.000	305.000	986.000	1632.000	1201.000	706.000
1332.000	738.000	1126.000	1088.000	1443.000	449.000				0.000
1038.000	885.000	953.000	722.000	761.000	1257.000	613.000	574.000	1132.000	422.000
127.000	569.000	773.000	856.000	900.000	1337.000	709.000	661.000	661.000	654.000
848.000	808.000	483.000	409.000	1331.000	570.000	316.000	478.000	535.000	600.000
162.000	612.000	1958.000	705.000	608.000	1073.000	714.000	880.000	770.000	480.000
753.000	2014.000	920.000	1168.000	957.000	578.000	1294.000	1946.000	972.000	1929.000
1595.000	861.000	812.000	774.000	1179.000	156.000				0.000
1143.000	1145.000	1056.000	981.000	975.000	1471.000	716.000	311.000	913.000	648.000
249.000	764.000	988.000	1076.000	930.000	1428.000	812.000	794.000	921.000	910.000
939.000	1068.000	726.000	669.000	1253.000	673.000	241.000	738.000	656.000	814.000
390.000	834.000	1978.000	919.000	851.000	1176.000	829.000	1139.000	1030.000	740.000
858.000	1776.000	1135.000	1266.000	1171.000	659.000	1277.000	1706.000	1232.000	974.000
1686.000	1021.000	1026.000	989.000	1438.000	410.000				0.000
882.000	671.000	853.000	508.000	701.000	1261.000	480.000	780.000	1314.000	266.000
308.000	432.000	667.000	669.000	945.000	1272.000	614.000	485.000	447.000	452.000
788.000	599.000	310.000	195.000	1493.000	449.000	493.000	269.000	379.000	596.000
155.000	421.000	1935.000	678.000	435.000	944.000	558.000	666.000	556.000	266.000
597.000	2134.000	774.000	1068.000	961.000	523.000	1411.000	2128.000	758.000	575.000
1530.000	688.000	816.000	625.000	965.000	73.000				0.000
1306.000	910.000	1277.000	743.000	551.000	1047.000	904.000	801.000	1470.000	690.000
473.000	402.000	581.000	781.000	1244.000	1665.000	1037.000	909.000	733.000	859.000
1176.000	973.000	733.000	544.000	1677.000	873.000	426.000	674.000	803.000	390.000
292.000	570.000	2302.000	495.000	858.000	1368.000	982.000	901.000	854.000	564.000
1021.000	2340.000	750.000	1492.000	747.000	908.000	1638.000	2270.000	997.000	998.000
1923.000	1111.000	602.000	627.000	1200.000	363.000				0.000
1138.000	760.000	1109.000	593.000	515.000	1026.000	736.000	774.000	1369.000	522.000
371.000	301.000	525.000	648.000	1135.000	1517.000	870.000	741.000	557.000	679.000
1028.000	793.000	553.000	364.000	1575.000	705.000	449.000	494.000	635.000	367.000
133.000	406.000	2175.000	472.000	678.000	1200.000	814.000	751.000	674.000	384.000
853.000	2239.000	672.000	1324.000	726.000	760.000	1529.000	2169.000	847.000	818.000
1775.000	931.000	581.000	526.000	1050.000	205.000				0.000
949.000	857.000	882.000	694.000	807.000	1323.000	542.000	596.000	1130.000	339.000
124.000	560.000	795.000	848.000	857.000	1266.000	638.000	552.000	633.000	595.000
777.000	756.000	411.000	381.000	1325.000	499.000	411.000	432.000	446.000	664.000
165.000	606.000	1915.000	768.000	536.000	1002.000	625.000	852.000	736.000	452.000
664.000	1975.000	931.000	1097.000	1023.000	507.000	1252.000	1944.000	940.000	676.000
1524.000	789.000	878.000	785.000	1151.000	139.000				0.000
421.000	1224.000	231.000	1112.000	1425.000	1985.000	350.000	890.000	842.000	536.000
685.000	1143.000	1378.000	1326.000	303.000	595.000	253.000	490.000	1007.000	823.000
106.000	955.000	650.000	1086.000	894.000	328.000	1070.000	793.000	398.000	1320.000
830.000	1080.000	1252.000	1402.000	644.000	351.000	425.000	1270.000	1072.000	908.000
306.000	1535.000	1471.000	433.000	1685.000	201.000	794.000	1604.000	1276.000	594.000
853.000	598.000	1540.000	1322.000	1538.000	767.000				0.000
650.000	1227.000	514.000	1085.000	1271.000	1771.000	394.000	615.000	761.000	539.000
409.000	1023.000	1258.000	1273.000	389.000	878.000	346.000	552.000	993.000	869.000

266.000	867.000	1833.000	952.000	683.000	984.000	637.000	1118.000	979.000	718.000
666.000	1757.000	1169.000	1079.000	1204.000	488.000	1144.000	1687.000	1183.000	782.000
1506.000	829.000	1059.000	1022.000	1417.000	405.000				0.000
760.000	840.000	686.000	697.000	927.000	1473.000	346.000	652.000	1125.000	198.000
222.000	665.000	900.000	885.000	719.000	1070.000	442.000	377.000	605.000	528.000
581.000	689.000	312.000	384.000	1269.000	303.000	608.000	393.000	256.000	813.000
117.000	627.000	1728.000	904.000	413.000	806.000	436.000	855.000	697.000	455.000
675.000	1910.000	1007.000	901.000	1173.000	313.000	1187.000	1939.000	901.000	544.000
1328.000	628.000	1028.000	858.000	1154.000	253.000				0.000
1036.000	1594.000	846.000	1431.000	1453.000	1949.000	815.000	401.000	435.000	921.000
618.000	1242.000	1466.000	1554.000	606.000	1210.000	782.000	966.000	1360.000	1251.000
721.000	1408.000	1035.000	1118.000	775.000	779.000	690.000	1148.000	825.000	1292.000
737.000	1312.000	1552.000	1397.000	1066.000	966.000	941.000	1589.000	1452.000	1189.000
833.000	1298.000	1613.000	1048.000	1649.000	589.000	851.000	1228.000	1656.000	1112.000
1441.000	1121.000	1504.000	1467.000	1888.000	876.000				0.000
865.000	624.000	836.000	461.000	767.000	1327.000	463.000	831.000	1343.000	249.000
359.000	435.000	670.000	649.000	937.000	1255.000	597.000	468.000	400.000	386.000
778.000	535.000	244.000	148.000	1499.000	432.000	559.000	205.000	362.000	662.000
244.000	391.000	1925.000	744.000	369.000	927.000	541.000	619.000	509.000	422.000
580.000	2398.000	552.000	1051.000	1027.000	513.000	1417.000	2157.000	832.000	509.000
1880.000	622.000	882.000	628.000	918.000	139.000				0.000
830.000	1246.000	639.000	1126.000	1189.000	1711.000	453.000	422.000	713.000	604.000
335.000	1063.000	1227.000	1298.000	552.000	1003.000	436.000	649.000	1045.000	934.000
514.000	1091.000	718.000	780.000	920.000	462.000	634.000	833.000	508.000	1054.000
397.000	1056.000	610.000	1133.000	701.000	759.000	640.000	1251.000	1094.000	851.000
545.000	1582.000	1348.000	900.000	1411.000	301.000	932.000	1513.000	1298.000	818.000
1261.000	829.000	1266.000	1228.000	1583.000	563.000				0.000
784.000	1006.000	697.000	854.000	1003.000	1512.000	357.000	515.000	957.000	370.000
165.000	754.000	989.000	1087.000	661.000	1081.000	453.000	435.000	771.000	700.000
592.000	861.000	484.000	541.000	1136.000	314.000	557.000	559.000	297.000	855.000
162.000	784.000	1719.000	960.000	518.000	817.000	470.000	1012.000	863.000	612.000
499.000	1779.000	1125.000	912.000	1212.000	322.000	1056.000	1771.000	1067.000	615.000
1339.000	662.000	1067.000	979.000	1311.000	333.000				0.000
850.000	939.000	763.000	785.000	930.000	1439.000	423.000	534.000	1019.000	319.000
101.000	681.000	916.000	964.000	734.000	1147.000	519.000	490.000	704.000	647.000
658.000	808.000	433.000	472.000	1207.000	380.000	493.000	492.000	363.000	782.000
89.000	715.000	1792.000	887.000	534.000	883.000	536.000	943.000	796.000	543.000
565.000	1852.000	1052.000	978.000	1139.000	388.000	1129.000	1833.000	1000.000	657.000
1405.000	728.000	997.000	906.000	1242.000	260.000				0.000
943.000	804.000	893.000	641.000	767.000	1285.000	541.000	653.000	1187.000	327.000
181.000	508.000	743.000	796.000	900.000	1277.000	649.000	546.000	580.000	557.000
788.000	718.000	388.000	328.000	1378.000	508.000	413.000	388.000	440.000	625.000
107.000	554.000	1935.000	729.000	513.000	1005.000	619.000	799.000	689.000	399.000
658.000	2019.000	879.000	1108.000	985.000	520.000	1296.000	2001.000	891.000	653.000
1535.000	766.000	840.000	733.000	1098.000	87.000				0.000
695.000	597.000	734.000	467.000	908.000	1486.000	361.000	912.000	1377.000	116.000
431.000	541.000	776.000	681.000	951.000	1132.000	495.000	319.000	362.000	270.000
685.000	431.000	73.000	185.000	1521.000	333.000	722.000	141.000	263.000	821.000
308.000	428.000	1825.000	897.000	198.000	813.000	412.000	625.000	454.000	256.000
478.000	2162.000	819.000	949.000	1186.000	527.000	1439.000	2191.000	658.000	338.000
1411.000	451.000	1041.000	771.000	911.000	311.000				0.000
920.000	591.000	915.000	428.000	701.000	1266.000	542.000	877.000	1412.000	328.000
406.000	356.000	591.000	593.000	1016.000	1334.000	676.000	545.000	367.000	424.000
857.000	563.000	298.000	115.000	1578.000	511.000	556.000	239.000	441.000	598.000
309.000	337.000	2007.000	678.000	423.000	1006.000	620.000	586.000	476.000	186.000

659.0002219.000	698.0001130.000	966.000	592.0001496.0002226.000	678.000	563.000
1592.000	676.000	821.000	549.000	885.000	157.000
749.0001088.000	662.000	936.0001086.0001595.000	352.000	528.000	876.000
235.000	837.0001072.0001120.000	580.0001028.000	418.000	454.000	853.000
539.000	896.000	523.000	623.0001055.000	309.000	627.000
444.000	866.0001638.0001043.000	552.000	782.000	483.0001094.000	945.000
464.0001696.0001208.000	866.0001295.000	258.000	975.0001690.0001149.000	634.000	
1286.000	675.0001150.0001062.0001393.000	416.000			0.000
1512.0002472.0001314.0002374.0002567.0003067.0001612.0001706.000	957.0001798.000				
1704.0002327.0002562.0002588.0001011.0001133.0001515.0001752.0002269.0002057.000					
1320.0002161.0001912.0002088.000	619.0001590.0002034.0002055.0001660.0002410.000				
1725.0002331.000	469.0002515.0001906.0001347.0001687.0002532.0002318.0002159.000				
1568.000	538.0002698.0001202.0002767.0001439.000	523.000	710.0002504.0001803.000		
848.0001765.0002622.0002552.0002767.0001906.000					0.000
1048.0001144.000	961.000	981.0001005.0001501.000	621.000	307.000	895.000
179.000	794.0001018.0001106.000	845.0001362.000	717.000	699.000	920.000
853.0001026.000	671.000	668.0001181.000	578.000	336.000	710.000
298.000	864.0001894.000	949.000	759.0001081.000	734.0001139.0001014.000	739.000
763.0001761.0001165.0001176.0001201.000	573.0001193.0001691.0001218.000	897.000			
1600.000	926.0001056.0001019.0001438.000	423.000			0.000
682.000	743.000	653.000	613.000	902.0001462.000	280.000
339.000	613.000	848.000	801.000	809.0001072.000	414.000
599.000	556.000	179.000	300.0001375.000	249.000	670.000
216.000	543.0001744.000	879.000	280.000	744.000	358.000
397.0002016.000	934.000	868.0001162.000	385.0001292.0002045.000	804.000	420.000
1330.000	533.0001017.000	785.0001057.000	270.000		0.000
1108.0001298.000	994.0001135.0001136.0001632.000	718.000	153.000	773.000	697.000
332.000	925.0001149.0001237.000	856.0001358.000	773.000	796.0001074.0001018.000	
869.0001179.000	811.000	822.000	811.000	675.000	352.000
451.000	995.0001877.0001080.000	877.0001114.000	831.0001293.0001167.000	893.000	
823.0001625.0001296.0001196.0001332.000	589.0001176.0001555.0001371.000	976.000			
1616.0001023.0001187.0001150.0001592.000	571.000				0.000
250.000	942.000	516.000	983.0001596.0002222.000	573.0001413.0001486.000	762.000
1071.0001275.0001447.0001198.000	939.000	573.000	522.000	531.000	892.000
612.000	609.000	679.000	904.0001530.000	614.0001463.000	776.000
989.0001077.0001363.0001631.000	554.000	398.000	441.0001037.000	838.000	900.000
466.0002086.0001410.000	522.0001933.000	711.0001319.0002155.000	920.000	414.000	
892.000	301.0001788.0001290.0001183.0001062.000				0.000
906.0001203.000	805.0001049.0001113.0001609.000	509.000	350.000	771.000	571.000
252.000	902.0001126.0001214.000	671.0001169.000	573.000	616.000	968.000
680.0001058.000	685.000	736.0001027.000	466.000	514.000	756.000
353.000	972.0001729.0001057.000	714.000	925.000	644.0001207.0001060.000	807.000
621.0001655.0001273.0001007.0001309.000	400.0001039.0001585.0001264.000	796.000			
1427.000	833.0001164.0001127.0001506.000	508.000			0.000
1148.000	679.0001119.000	512.000	435.0001003.000	746.000	896.0001491.000
493.000	188.000	415.000	564.0001196.0001538.000	880.000	751.000
1054.000	750.000	563.000	337.0001697.000	715.000	571.000
428.000	325.0002201.000	412.000	678.0001210.000	824.000	670.000
863.0002351.000	559.0001334.000	703.000	789.0001628.0002291.000	766.000	818.000
1796.000	931.000	558.000	413.000	969.000	247.000
1190.0001076.0001103.000	909.000	903.0001399.000	763.000	412.0001020.000	630.000
269.000	692.000	916.0001004.0001016.0001487.000	859.000	841.000	857.000
998.0001004.000	691.000	605.0001352.000	720.000	166.000	674.000
382.000	762.0002065.000	847.000	816.0001223.000	876.0001067.000	966.000
905.0001883.0001063.0001318.0001099.000	728.0001364.0001813.0001163.000	956.000			

1745.0001068.000	956.000	917.0001366.000	346.000	0.000
1664.0001060.0001615.000	933.000	318.000	501.0001242.0001452.0001878.0001028.000	
881.000	550.000	439.000	797.0001651.0002034.0001376.0001247.000	982.000 215.000
1550.0001252.0001059.000	839.0002085.0001211.000	958.000	967.0001141.000	210.000
660.000	779.000	893.000	184.0001180.0001706.0001320.000	984.0001099.000 843.000
1359.0002478.000	575.0001830.000	201.0001285.0002045.0002678.0001137.0001320.000		
2292.0001433.000	71.000	625.0001259.000	728.000	0.000
968.000	672.000	919.000	509.000	635.0001195.000 566.000 797.0001368.000 332.000
342.000	376.000	611.000	635.0001002.0001338.000	680.000 551.000 468.000 498.000
854.000	643.000	372.000	196.0001539.000	515.000 472.000 313.000 445.000 530.000
256.000	393.0002001.000	612.000	497.0001010.000	624.000 667.000 557.000 267.000
663.0002180.000	740.0001136.000	895.000	589.0001457.0002162.000	759.000 637.000
1596.000	750.000	750.000	591.000	966.000 85.000 0.000
1226.000	778.0001197.000	611.000	412.000	917.000 824.000 884.0001479.000 610.000
681.000	252.000	439.000	631.0001245.0001616.000	958.000 829.000 601.000 749.000
1132.000	849.000	641.000	436.0001685.000	793.000 559.000 564.000 723.000 258.000
475.000	424.0002279.000	363.000	766.0001288.000	902.000 769.000 728.000 440.000
941.0002369.000	610.0001412.000	617.000	867.0001639.0002279.000	865.000 906.000
1874.0001019.000	472.000	477.0001068.000	310.000	0.000
466.0001015.000	350.000	903.0001264.0001804.000	161.000	788.0001008.000 327.000
509.000	956.0001191.0001147.000	509.000	742.000	109.000 299.000 798.000 641.000
253.000	797.000	441.000	628.0001098.000	119.000 901.000 584.000 189.0001139.000
644.000	871.0001600.0001617.000	453.000	470.000	267.0001061.000 863.000 699.000
159.0001739.0001262.000	565.0001504.000	86.0001003.0001808.0001067.000	438.000	
1000.000	467.0001359.0001113.0001329.000	589.000		0.000
915.0001254.000	801.0001100.0001159.0001655.000	529.000	319.000	724.000 607.000
298.000	968.0001172.0001260.000	667.0001163.000	580.000	652.0001019.000 937.000
676.0001094.000	721.000	787.0001003.000	486.000	497.000 807.000 511.000 998.000
404.0001018.0001716.0001103.000	750.000	921.000	664.0001258.0001111.000	858.000
630.0001608.0001319.0001003.0001355.000	396.0001015.0001538.0001315.000			832.000
1623.000	853.0001210.0001173.0001557.000	556.000		0.000

C-----

C Explanation for the Second part

C This links each surplus region (58) to its corresponding river barge-loading locations. Conceptually, a (1 x 58) 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 58 -----> corresponding river barge-loading location

C e.g., the 2nd surplus region, Ames (085) to its corresponding river barge region, Dubuque (064) = 173 miles

C e.g., the 3rd surplus region, Aurora (139) to its corresponding river barge region, Ottawa (615) = 53 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 Ames were not a surplus region, then 173 would be deleted.

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

999.999	173.000	53.000	427.000	250.000	436.000	139.000	999.999	183.000	616.000
375.000	74.000	92.000	79.000	67.000	65.000	84.000	71.000	64.000	66.000
97.000	999.999	153.000	49.000	378.000	150.000	111.000	999.999	72.000	50.000
53.000	274.000	97.000	133.000	153.000	66.000	381.000	97.000	76.000	136.000
75.000	10.000	77.000	31.000	53.000	999.999	49.000	104.000	73.000	504.000
95.000	72.000	132.000	999.999	112.000	171.000	197.000	91.000		0.000

C-----

C Explanation for the Third part

C (3) Each surplus region(58) 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 -----> port region 20 for export

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

C .
C -----> port region 20 for export

C surplus region 58 -----> port region 1 for export
C -----> port region 2 for export

C .
C -----> port region 20 for export

C There are 58 surplus regions and 20 port locations thus a (58 by 20)
C matrix.

C e.g., the 1st surplus region, Alexandria (111) to the 1st port region,
C Mobile = 309 miles

C e.g., the 1st surplus region, Alexandria (111) to the 2nd port region,
C New Orleans = 189 miles

C The next matrix is for the next surplus region, the 2nd region and its
C mileages to all 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 Alexandria were removed as a surplus region, then the first 2 lines
(below) would be removed. The file would start with 984, the truck mileage
C from Ames to Mobile.

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

309.000	189.000	229.000	450.000	585.000	839.000	1185.000	993.000	1092.000	869.000
1226.000	2385.000	2316.000	2060.000	1715.000	1638.000	993.000	1092.000	869.000	1226.000
984.000	1011.000	987.000	1106.000	1241.000	1174.000	986.000	550.000	596.000	324.000
373.000	1745.000	1808.000	1854.000	1819.000	1793.000	550.000	596.000	324.000	373.000
842.000	903.000	1089.000	1260.000	1395.000	904.000	701.000	265.000	310.000	40.000
456.000	2002.000	2072.000	2108.000	2045.000	2033.000	265.000	310.000	40.000	456.000
639.000	517.000	207.000	197.000	332.000	1196.000	1519.000	1256.000	1355.000	1110.000
1279.000	2138.000	2069.000	1748.000	1395.000	1318.000	1256.000	1355.000	1110.000	1279.000
911.000	991.000	1207.000	1405.000	1540.000	853.000	551.000	107.000	119.000	159.000
628.000	2189.000	2252.000	2322.000	2254.000	2209.000	107.000	119.000	159.000	628.000
1589.000	1583.000	1433.000	1520.000	1655.000	1708.000	1499.000	1063.000	1058.000	831.000
444.000	1195.000	1265.000	1604.000	1632.000	1666.000	1063.000	1058.000	831.000	444.000
1245.000	1239.000	1151.000	1263.000	1398.000	1469.000	1249.000	813.000	859.000	583.000
334.000	1494.000	1557.000	1778.000	1744.000	1768.000	813.000	859.000	583.000	334.000
1044.000	1124.000	1325.000	1518.000	1653.000	1000.000	672.000	228.000	104.000	266.000
547.000	2198.000	2261.000	2429.000	2361.000	2316.000	228.000	104.000	266.000	547.000
878.000	966.000	1216.000	1423.000	1558.000	618.000	312.000	182.000	322.000	383.000
852.000	2413.000	2476.000	2532.000	2344.000	2299.000	182.000	322.000	383.000	852.000
1594.000	1527.000	1335.000	1352.000	1481.000	1854.000	1697.000	1261.000	1306.000	1036.000
912.000	1094.000	1074.000	1153.000	1125.000	1153.000	1261.000	1306.000	1036.000	912.000
1448.000	1410.000	1235.000	1262.000	1391.000	1697.000	1515.000	1079.000	1125.000	853.000
760.000	1258.000	1281.000	1352.000	1318.000	1342.000	1079.000	1125.000	853.000	760.000
829.000	841.000	849.000	968.000	1103.000	1061.000	989.000	614.000	682.000	422.000
536.000	1883.000	1864.000	1881.000	1699.000	1654.000	614.000	682.000	422.000	536.000
779.000	867.000	1117.000	1324.000	1459.000	605.000	412.000	178.000	317.000	337.000
806.000	2367.000	2430.000	2467.000	2275.000	2230.000	178.000	317.000	337.000	806.000
733.000	727.000	747.000	866.000	753.000	1032.000	1027.000	680.000	748.000	488.000
650.000	1943.000	1919.000	1936.000	1654.000	1609.000	680.000	748.000	488.000	650.000
672.000	752.000	997.000	1197.000	1332.000	652.000	583.000	258.000	357.000	231.000
700.000	2261.000	2315.000	2332.000	2118.000	2073.000	258.000	357.000	231.000	700.000
1026.000	1071.000	1141.000	1260.000	1395.000	1155.000	950.000	514.000	560.000	284.000
282.000	1768.000	1831.000	2014.000	1980.000	1951.000	514.000	560.000	284.000	282.000
862.000	923.000	1100.000	1268.000	1403.000	933.000	726.000	290.000	336.000	100.000

438.0001999.0002062.0002117.0002060.0002015.000 290.000 336.000 100.000 438.000
 1047.0001041.000 985.0001104.0001239.0001266.0001078.000 642.000 688.000 416.000
 640.0001653.0001716.0001762.0001727.0001704.000 642.000 688.000 416.000 440.000
 855.000 916.0001081.0001249.0001384.000 962.000 761.000 325.000 370.000 100.000
 422.0001946.0002014.0002046.0001983.0001971.000 325.000 370.000 100.000 422.000
 1115.0001173.0001268.0001387.0001522.0001184.000 975.000 539.000 584.000 307.000
 158.0001706.0001776.0002029.0001999.0002027.000 539.000 584.000 307.000 158.000
 646.000 707.000 893.0001076.0001211.000 777.000 701.000 357.000 454.000 206.000
 637.0002146.0002174.0002191.0001955.0001910.000 357.000 454.000 206.000 637.000
 1001.0001081.0001301.0001499.0001634.000 888.000 551.000 107.000 33.000 257.000
 648.0002287.0002350.0002420.0002352.0002307.000 107.000 33.000 257.000 648.000
 821.000 901.0001121.0001319.0001454.000 756.000 518.000 104.000 202.000 155.000
 624.0002185.0002248.0002310.0002201.0002156.000 104.000 202.000 155.000 624.000
 809.000 858.0001009.0001177.0001312.000 956.000 811.000 395.000 442.000 179.000
 480.0001968.0002014.0002031.0001943.0001898.000 395.000 442.000 179.000 480.000
 1052.000 972.000 797.000 817.000 946.0001401.0001423.0001071.0001139.000 878.000
 901.0001649.0001577.0001525.0001271.0001230.0001071.0001139.000 878.000 901.000
 1105.0001069.000 910.000 997.0001132.0001365.0001279.000 848.000 894.000 622.000
 626.0001598.0001562.0001579.0001518.0001494.000 848.000 894.000 622.000 626.000
 716.000 796.0001003.0001201.0001336.000 696.000 564.000 220.000 318.000 187.000
 656.0002217.0002271.0002288.0002092.0002047.000 220.000 318.000 187.000 656.000
 768.000 835.0001024.0001222.0001357.000 759.000 617.000 214.000 302.000 118.000
 583.0002131.0002170.0002199.0002083.0002038.000 214.000 302.000 118.000 583.000
 639.000 727.000 977.0001184.0001319.000 525.000 550.000 277.000 414.000 358.000
 827.0002388.0002434.0002451.0002205.0002160.000 277.000 414.000 358.000 827.000
 1022.0001016.000 913.0001000.0001135.0001281.0001175.000 746.000 792.000 520.000
 543.0001664.0001646.0001663.0001618.0001587.000 746.000 792.000 520.000 543.000
 682.000 723.000 875.0001043.0001178.000 871.000 848.000 415.000 486.000 219.000
 610.0002131.0002110.0002139.0001920.0001871.000 415.000 486.000 219.000 610.000
 1195.0001118.000 943.000 987.0001116.0001463.0001401.000 971.0001017.000 745.000
 746.0001527.0001462.0001479.0001405.0001381.000 971.0001017.000 745.000 746.000
 945.0001006.0001183.0001351.0001486.0001017.000 808.000 372.000 418.000 142.000
 334.0001895.0001958.0002126.0002088.0002043.000 372.000 418.000 142.000 334.000
 1146.0001184.0001158.0001277.0001412.0001298.0001089.000 653.000 698.000 421.000
 230.0001607.0001677.0001864.0001838.0001862.000 653.000 698.000 421.000 230.000
 1052.0001092.0001076.0001195.0001330.0001217.0001007.000 571.000 617.000 341.000
 289.0001707.0001770.0001929.0001895.0001887.000 571.000 617.000 341.000 289.000
 764.000 791.000 875.000 994.0001129.000 977.000 934.000 559.000 627.000 367.000
 583.0001967.0001948.0001965.0001768.0001703.000 559.000 627.000 367.000 583.000
 1487.0001481.0001331.0001418.0001553.0001662.0001453.0001017.0001062.000 785.000
 475.0001228.0001298.0001603.0001571.0001601.0001017.0001062.000 785.000 475.000
 582.000 643.000 827.0001010.0001145.000 741.000 768.000 424.000 521.000 273.000
 697.0002182.0002175.0002192.0001946.0001901.000 424.000 521.000 273.000 697.000
 1170.0001164.0001043.0001130.0001265.0001406.0001214.000 778.000 824.000 548.000
 444.0001513.0001576.0001700.0001666.0001680.000 778.000 824.000 548.000 444.000
 936.000 935.000 909.0001028.0001163.0001134.0001001.000 587.000 633.000 361.000
 447.0001793.0001820.0001837.0001764.0001719.000 587.000 633.000 361.000 447.000
 869.000 910.000 975.0001094.0001229.0001065.000 927.000 513.000 559.000 287.000
 467.0001853.0001894.0001911.0001825.0001780.000 513.000 559.000 287.000 467.000
 765.000 820.000 986.0001154.0001289.000 904.000 767.000 360.000 413.000 158.000
 520.0002001.0001808.0002048.0001960.0001911.000 360.000 413.000 158.000 520.000
 506.000 533.000 671.000 839.000 974.000 756.000 918.000 581.000 679.000 446.000
 819.0002219.0002197.0002139.0001845.0001800.000 581.000 679.000 446.000 819.000
 610.000 671.000 866.0001064.0001199.000 701.000 701.000 362.000 460.000 264.000
 726.0002244.0002254.0002271.0002025.0001980.000 362.000 460.000 264.000 726.000

975.000 969.000 922.0001041.0001176.0001216.0001083.000 669.000 715.000 443.000
 510.0001711.0001738.0001755.0001709.0001664.000 669.000 715.000 443.000 510.000
 1710.0002164.0001913.0001834.0001938.0002663.0002567.0002135.0002181.0001913.000
 1815.000 710.000 538.000 229.000 490.000 561.0002135.0002181.0001913.0001815.000
 1087.0001128.0001173.0001292.0001427.0001214.0001015.000 579.000 625.000 349.000
 229.0001712.0001775.0001988.0001954.0001960.000 579.000 625.000 349.000 229.000
 631.000 658.000 753.000 916.0001051.000 893.000 904.000 560.000 651.000 396.000
 721.0002073.0002051.0002060.0001766.0001721.000 560.000 651.000 396.000 721.000
 1240.0001281.0001270.0001389.0001524.0001345.0001132.000 696.000 738.000 466.000
 139.0001575.0001638.0001941.0001907.0001931.000 696.000 738.000 466.000 139.000
 879.000 550.000 236.000 146.000 275.0001248.0001596.0001333.0001432.0001187.000
 1356.0002155.0002086.0001737.0001384.0001307.0001333.0001432.0001187.0001356.000
 1133.0001131.0001080.0001199.0001334.0001332.0001113.000 677.000 723.000 447.000
 345.0001606.0001669.0001823.0001789.0001789.000 677.000 723.000 447.000 345.000
 778.000 858.0001108.0001315.0001450.000 672.000 436.000 128.000 265.000 277.000
 746.0002307.0002370.0002410.0002220.0002175.000 128.000 265.000 277.000 746.000
 1051.0001112.0001289.0001434.0001569.0001112.000 903.000 467.000 512.000 235.000
 259.0001813.0001883.0002116.0002086.0002114.000 467.000 512.000 235.000 259.000
 974.0001340.0001610.0001817.0001952.000 850.000 318.000 441.000 582.000 665.000
 1130.0002678.0002748.0002797.0002718.0002673.000 441.000 582.000 665.0001130.000
 690.000 751.000 940.0001138.0001273.000 743.000 635.000 291.000 386.000 179.000
 644.0002162.0002180.0002209.0002026.0001977.000 291.000 386.000 179.000 644.000
 877.000 957.0001207.0001407.0001542.000 738.000 415.000 48.000 188.000 265.000
 734.0002295.0002358.0002420.0002298.0002253.000 48.000 188.000 265.000 734.000
 893.000 816.000 657.000 764.000 879.0001192.0001244.000 884.000 951.000 696.000
 794.0001808.0001739.0001695.0001425.0001376.000 884.000 951.000 696.000 794.000
 1173.0001167.0001100.0001215.0001350.0001368.0001158.000 722.000 768.000 492.000
 327.0001596.0001619.0001813.0001779.0001785.000 722.000 768.000 492.000 327.000

C-----

C Explanation for the Fourth part

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

C There are 56 deficit regions, thus a 1 by 56 matrix.

C Diagrammatically,

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

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

C .

C .

C .

C deficit region 56 -----> corresponding barge-unloading location

C e.g., the 19th deficit region Gadsen, AL (201) to the 28th

C barge-unloading river location, Gunsterville, AL (628)

C = 34 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 If a barge-unloading river location were deleted, make sure that there
C is no deficit region linked to it.

404.000	999.999	371.000	114.000	497.000	999.999	60.000	999.999	999.999	194.000
70.000	208.000	643.000	999.999	999.999	811.000	176.000	123.000	34.000	2.000
386.000	999.999	91.000	71.000	499.000	90.000	999.999	85.000	114.000	395.000
119.000	2.000	999.999	474.000	43.000	474.000	65.000	272.000	161.000	2.000
141.000	0.000	391.000	999.999	999.999	176.000	999.999	172.000	356.000	156.000
1069.000	197.000	628.000	242.000	587.000	90.000				0.000

F3.crd

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

C In this model, 4 sets of rail cost must be found.

C This includes

(1) rail coast from each surplus region to each deficit region

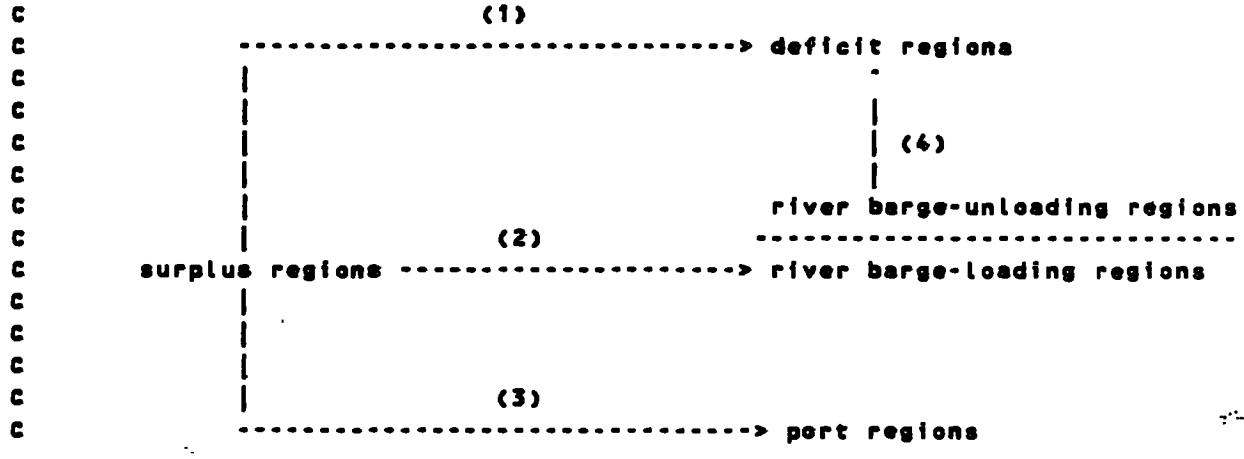
(2) rail cost from each surplus region to each river barge-loading region

(3) rail cost from each surplus region to each port region

(4) sail coast from each barge-unloading river location to

C each deficit region

C Diagrammatically,



C First part

(1) Each surplus region is linked to each of the deficit regions.

In this model there are 58 surplus regions and 56 deficit regions.

c Each surplus region is linked to all deficit regions(56). Thus

c resulting in a 58 by 56 matrix.

C surplus region 1 -----> deficit region 1

-----> deficit region 2

C

6

6

-----> deficit

-----> deficit region 56

surplus region 2 -----> deficit region 1

-----> deficit region 2

C

2

C

-----> deficit

C surplus region 58 -----> deficit region 1
 C -----> deficit region 2
 C
 C
 C -----> deficit region 56

C For example,
 C the rail cost from Alexandria (111) to second deficit region,
 C Albany (213) = 46.480 cents/bushel
 C the rail cost from Alexandria (111) to fourth deficit region,
 C Atlanta (211) = 48.720 cents/bushel

999.999	46.480	999.999	48.720	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	34.720	41.440	30.240	
999.999	28.000	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	999.999	25.200	999.999	999.999	999.999	50.960	39.200	55.440		
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	49.840	999.999		
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0.000	
64.960	78.960	54.880	64.960	73.360	118.720	38.080	40.320	72.800	35.840			
19.040	999.999	999.999	999.999	39.200	85.120	45.920	49.840	999.999	59.920			
54.320	67.760	54.320	49.280	85.120	36.400	47.040	49.840	36.960	999.999			
999.999	64.400	99.120	999.999	55.440	61.600	45.360	999.999	69.440	52.640			
43.680	123.200	999.999	65.520	999.999	36.960	80.080	133.840	999.999	54.880			
103.600	67.760	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0.000	
76.720	76.160	67.200	59.920	59.920	999.999	47.040	999.999	999.999	999.999	32.480		
26.880	43.120	64.400	75.600	999.999	95.200	54.320	56.560	58.240	60.480			
67.760	60.480	43.680	64.400	101.360	44.800	999.999	42.560	41.440	999.999			
25.200	56.000	999.999	60.480	52.640	73.976	55.440	71.680	61.040	45.360			
54.880	999.999	84.000	67.200	999.999	47.600	95.760	999.999	75.600	60.480			
999.999	48.160	999.999	75.040	89.040	24.640	999.999	999.999	999.999	999.999	999.999	0	
36.400	999.999	43.120	999.999	999.999	999.999	47.600	999.999	999.999	999.999	999.999		
999.999	999.999	999.999	999.999	59.360	52.080	43.680	53.200	999.999	999.999	999.999		
54.880	999.999	50.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999		
999.999	999.999	124.880	999.999	39.200	36.400	44.240	999.999	999.999	999.999	999.999		
34.160	999.999	999.999	49.840	999.999	999.999	102.480	999.999	999.999	999.999	30.800		
69.440	22.96	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	0	
999.999	76.720	999.999	59.360	48.160	98.560	999.999	999.999	999.999	999.999	999.999		
999.999	42.560	61.600	66.640	999.999	999.999	999.999	999.999	999.999	57.120	61.600		
999.999	81.200	999.999	42.560	999.999	999.999	45.920	51.520	999.999	38.640			
20.720	49.280	999.999	49.840	999.999	999.999	999.999	999.999	71.120	59.920	43.680		
999.999	999.999	80.640	999.999	88.320	999.999	999.999	999.999	999.999	74.480	999.999		
999.999	999.999	72.080	67.760	88.480	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	34.720	34.160	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	999.999	999.999	999.999	63.280	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	87.920	999.999	999.999	999.999	999.999	999.999	999.999	72.240	91.840	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	
88.480	999.999	58.240	999.999	999.999	999.999	66.080	42.000	59.360	999.999	999.999		
999.999	999.999	999.999	999.999	58.800	110.320	70.000	66.080	999.999	999.999	999.999		
78.960	999.999	72.240	999.999	82.320	58.240	999.999	999.999	999.999	999.999	999.999		
999.999	999.999	100.800	999.999	86.800	86.800	73.920	110.320	98.000	999.999	999.999		
66.080	115.920	999.999	96.880	999.999	64.960	82.320	113.120	999.999	86.800			

118.160	90.720	999.999	999.999	999.999	999.999	999.999		0
999.999	88.480	999.999	64.400	56.560	106.400	999.999	999.999	999.999
999.999	50.400	69.440	73.360	999.999	999.999	999.999	999.999	67.200
999.999	91.840	999.999	54.320	999.999	999.999	43.120	63.280	999.999
31.920	61.040	999.999	57.120	999.999	999.999	999.999	82.880	70.560
999.999	999.999	80.640	999.999	75.600	999.999	999.999	999.999	86.240
999.999	999.999	59.920	71.120	100.240	999.999			0
999.999	63.280	999.999	58.240	30.240	92.400	999.999	999.999	999.999
999.999	34.160	53.200	54.880	999.999	999.999	999.999	999.999	53.200
999.999	76.720	999.999	39.760	999.999	999.999	999.999	48.160	999.999
37.520	39.200	999.999	31.920	999.999	999.999	999.999	63.840	56.000
999.999	999.999	63.840	999.999	63.840	999.999	999.999	999.999	70.560
999.999	999.999	49.840	49.840	84.000	999.999			0
999.999	999.999	57.680	999.999	999.999	999.999	999.999	999.999	27.440
999.999	999.999	999.999	999.999	27.440	72.800	999.999	999.999	999.999
44.240	999.999	999.999	999.999	59.360	999.999	999.999	999.999	999.999
999.999	999.999	96.880	999.999	999.999	64.960	999.999	999.999	999.999
999.999	999.999	999.999	69.440	999.999	999.999	50.960	999.999	999.999
94.640	999.999	999.999	999.999	999.999	999.999			0
109.200	999.999	61.600	999.999	999.999	999.999	63.840	66.640	36.960
47.040	999.999	999.999	999.999	31.360	72.800	59.920	72.240	999.999
50.960	101.920	71.680	999.999	61.040	67.200	85.120	82.880	61.600
999.999	99.680	100.240	999.999	84.000	67.760	76.160	999.999	999.999
66.080	100.240	999.999	72.800	999.999	54.320	67.760	90.160	999.999
94.640	86.800	999.999	999.999	999.999	999.999			0
71.680	71.120	62.160	66.640	999.999	999.999	42.560	999.999	999.999
29.680	77.280	94.640	84.000	54.320	87.360	42.560	39.760	61.040
52.640	67.760	37.520	50.960	999.999	34.160	999.999	34.160	30.800
999.999	67.760	140.560	999.999	45.920	68.320	44.240	79.520	61.600
42.560	999.999	999.999	73.920	999.999	31.920	91.280	999.999	999.999
64.400	58.240	999.999	86.800	999.999	999.999			0
999.999	54.880	999.999	37.520	36.400	75.600	999.999	999.999	999.999
999.999	22.960	40.880	68.320	999.999	999.999	999.999	999.999	30.800
999.999	999.999	999.999	38.640	999.999	999.999	999.999	35.280	999.999
49.280	26.880	999.999	56.560	999.999	999.999	999.999	50.960	38.080
999.999	999.999	54.320	999.999	72.240	999.999	999.999	999.999	62.720
999.999	999.999	73.920	50.960	81.760	999.999			0
57.120	999.999	43.120	999.999	999.999	999.999	24.640	999.999	83.440
999.999	999.999	999.999	999.999	57.680	82.320	36.960	29.120	999.999
51.520	66.640	35.840	47.600	999.999	24.080	999.999	49.840	24.640
999.999	64.400	138.320	999.999	38.080	49.840	30.240	74.480	62.160
36.960	999.999	999.999	54.880	999.999	34.720	94.080	999.999	999.999
106.400	53.760	999.999	999.999	999.999	999.999			0
999.999	52.640	999.999	40.880	48.160	100.800	999.999	999.999	999.999
999.999	29.120	48.160	48.720	999.999	999.999	999.999	999.999	39.200
999.999	62.720	999.999	25.200	999.999	999.999	999.999	33.040	999.999
18.480	31.360	999.999	52.080	48.720	999.999	999.999	52.640	33.600
999.999	999.999	62.720	999.999	70.000	999.999	999.999	999.999	56.000
999.999	999.999	56.560	54.320	70.000	26.860			0
78.960	97.440	81.200	75.600	95.200	999.999	50.960	33.040	78.960
22.960	71.680	90.720	91.280	64.400	104.160	62.160	62.720	74.480
69.440	73.360	64.400	56.000	95.760	49.840	50.400	63.280	49.840
999.999	71.680	138.320	76.720	71.120	86.240	63.840	91.840	80.640
60.480	125.440	999.999	87.360	999.999	50.400	90.720	143.920	999.999
122.080	89.040	999.999	87.360	999.999	999.999			0

81.200	77.280	67.760	61.040	61.600	999.999	52.640	999.999	999.999	48.160
41.440	44.800	63.840	65.520	999.999	96.880	59.920	54.320	59.920	53.760
65.520	53.200	43.680	43.120	98.560	45.360	999.999	43.680	48.160	999.999
20.160	57.680	999.999	999.999	49.840	74.480	61.040	73.360	59.360	46.480
55.440	999.999	85.680	78.400	999.999	53.200	100.800	999.999	76.720	58.240
999.999	70.560	999.999	76.720	90.720	23.520				0
61.040	90.160	47.040	83.440	103.040	147.280	42.560	45.360	66.080	42.560
19.040	74.480	93.520	98.560	47.600	85.680	45.920	61.040	87.920	75.600
57.680	68.320	49.280	63.840	78.960	39.200	50.400	65.520	36.960	77.280
47.040	75.040	96.320	87.920	64.960	53.200	50.400	95.760	84.560	63.840
48.720	117.040	115.360	70.560	112.560	38.640	70.560	127.680	99.120	58.800
106.400	70.000	90.720	98.560	112.560	42.000				0
77.280	71.120	66.080	56.000	71.120	999.999	49.840	999.999	84.000	45.920
15.120	47.040	66.080	58.800	65.520	98.000	57.680	51.520	53.760	52.080
66.080	50.960	40.880	36.400	999.999	48.160	40.320	39.200	47.600	58.800
22.400	52.080	999.999	64.960	47.600	72.800	61.600	67.760	56.560	39.760
55.440	999.999	80.080	76.720	999.999	48.720	91.280	999.999	71.120	55.440
115.360	67.760	999.999	70.560	85.120	16.240				0
96.880	999.999	80.640	999.999	999.999	999.999	62.720	35.840	64.400	59.920
33.040	999.999	999.999	999.999	71.120	103.040	69.440	61.600	999.999	999.999
73.920	999.999	72.800	999.999	97.440	53.200	26.320	57.680	56.000	999.999
999.999	30.240	125.440	999.999	63.840	86.800	68.880	999.999	999.999	86.800
64.400	999.999	999.999	87.360	999.999	54.880	92.400	999.999	999.999	64.400
115.920	68.320	999.999	999.999	999.999	42.560				0
82.880	71.120	73.920	48.160	60.480	104.720	54.880	72.800	104.160	41.440
39.760	47.040	45.360	58.240	68.320	99.680	54.320	47.040	42.000	49.280
68.880	67.760	34.720	25.200	999.999	45.920	39.200	24.080	42.000	64.400
25.200	45.360	999.999	58.800	43.120	77.840	55.440	79.520	45.920	29.120
53.760	999.999	73.360	85.120	88.480	63.840	109.200	999.999	80.080	51.520
117.600	63.280	68.880	66.080	98.000	15.680				0
999.999	87.360	999.999	68.320	53.200	106.400	999.999	999.999	999.999	999.999
999.999	43.120	62.720	39.760	999.999	999.999	999.999	999.999	67.760	70.560
999.999	90.160	999.999	52.640	999.999	999.999	40.880	61.600	999.999	45.920
29.680	59.920	999.999	57.120	999.999	999.999	999.999	81.200	69.440	53.760
999.999	155.120	80.640	999.999	75.600	999.999	999.999	999.999	84.560	999.999
999.999	999.999	48.720	66.640	98.560	999.999				0
999.999	67.200	999.999	49.280	38.080	96.320	999.999	999.999	999.999	999.999
999.999	29.680	48.720	49.280	999.999	999.999	999.999	999.999	47.600	51.520
999.999	71.120	999.999	33.600	999.999	999.999	999.999	41.440	999.999	35.840
15.680	34.720	999.999	47.040	57.120	999.999	999.999	61.040	49.840	33.600
999.999	999.999	57.120	999.999	64.960	999.999	999.999	999.999	64.400	999.999
999.999	999.999	49.840	49.840	78.400	23.520				0
63.280	74.480	54.320	58.240	70.560	999.999	34.160	43.680	999.999	38.080
20.160	48.720	67.760	72.800	61.600	83.440	50.960	43.680	56.560	50.400
52.080	58.240	38.080	39.760	92.400	31.920	37.520	39.760	31.360	62.720
999.999	58.800	999.999	68.880	44.800	61.040	47.040	67.760	59.360	43.680
42.000	999.999	82.320	64.960	999.999	39.760	87.360	999.999	999.999	52.640
101.920	64.960	999.999	73.360	999.999	21.840				0
55.440	999.999	32.480	999.999	999.999	999.999	28.560	999.999	67.760	999.999
999.999	999.999	999.999	999.999	27.440	62.160	24.080	999.999	999.999	999.999
23.520	999.999	999.999	999.999	117.600	30.800	999.999	999.999	999.999	999.999
999.999	999.999	109.760	999.999	999.999	39.760	43.680	999.999	999.999	999.999
30.800	108.640	999.999	44.800	999.999	25.760	72.800	84.560	999.999	54.880
84.560	59.920	999.999	999.999	999.999	999.999				0
62.160	999.999	34.160	999.999	999.999	999.999	35.280	999.999	64.400	999.999

999.999	999.999	999.999	999.999	999.999	30.800	71.680	33.600	40.320	999.999	999.999
36.960	999.999	57.120	999.999	72.240	25.200	999.999	999.999	25.760	999.999	
999.999	999.999	95.760	999.999	50.400	39.760	34.720	999.999	999.999	999.999	999.999
36.400	90.160	999.999	58.240	999.999	17.920	57.120	100.240	999.999	999.999	63.280
89.600	67.200	999.999	999.999	999.999	999.999					0
999.999	61.040	999.999	43.120	46.480	98.560	999.999	999.999	999.999	999.999	999.999
999.999	29.680	48.720	48.720	999.999	999.999	999.999	999.999	41.440	45.360	
999.999	64.960	999.999	26.320	999.999	999.999	999.999	35.280	999.999	38.640	
15.680	33.600	999.999	49.840	50.960	999.999	999.999	54.880	40.320	27.440	
999.999	999.999	61.040	999.999	67.760	999.999	999.999	999.999	58.240	999.999	
999.999	999.999	53.760	51.520	72.240	20.160					0
999.999	45.920	999.999	46.480	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	66.080	999.999	999.999	999.999	999.999	37.520	31.920	
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	31.360	999.999	999.999	48.720	33.600	999.999	
999.999	999.999	75.600	999.999	999.999	999.999	999.999	999.999	43.680	26.880	
999.999	31.360	999.999	67.760	57.680	999.999					0
999.999	44.240	999.999	27.440	59.920	95.760	999.999	999.999	999.999	48.160	
999.999	22.960	42.000	33.040	999.999	999.999	999.999	999.999	28.560	41.440	
999.999	44.240	42.000	15.680	999.999	999.999	69.280	35.840	51.520	46.480	
27.440	17.360	999.999	53.200	999.999	999.999	999.999	39.760	36.400	20.720	
999.999	999.999	45.360	999.999	78.960	999.999	999.999	999.999	48.160	999.999	
999.999	999.999	64.480	33.040	61.600	999.999					0
58.240	999.999	37.520	999.999	999.999	999.999	35.280	999.999	70.000	33.600	
30.240	999.999	999.999	999.999	36.960	72.240	41.440	39.200	999.999	999.999	
33.040	999.999	40.320	999.999	78.960	28.560	999.999	999.999	28.560	999.999	
999.999	999.999	95.200	999.999	52.640	43.120	39.760	999.999	999.999	999.999	999.999
43.120	108.080	999.999	62.160	999.999	19.600	64.960	118.720	999.999	54.880	
93.520	68.880	999.999	999.999	999.999	999.999					0
75.600	59.920	67.200	46.800	67.200	999.999	47.600	999.999	999.999	33.600	
26.320	61.600	88.640	59.360	999.999	89.600	46.480	39.200	43.120	36.400	
64.960	44.240	29.120	25.200	106.400	45.920	999.999	26.320	39.200	71.120	
25.200	48.880	999.999	77.280	35.280	77.840	47.600	56.560	45.360	29.120	
46.480	999.999	68.880	81.760	999.999	45.920	100.240	999.999	64.480	43.680	
999.999	50.960	999.999	59.920	73.920	19.600					0
80.080	999.999	39.200	999.999	999.999	999.999	48.720	999.999	59.360	999.999	
999.999	999.999	999.999	999.999	21.280	66.080	36.960	49.280	999.999	999.999	
38.080	999.999	44.800	999.999	65.520	43.120	999.999	999.999	42.000	999.999	
999.999	999.999	82.320	999.999	52.080	43.680	57.120	999.999	999.999	999.999	999.999
42.560	100.800	999.999	60.480	999.999	28.560	60.480	111.440	999.999	70.000	
70.000	63.840	999.999	999.999	999.999	999.999					0
999.999	82.880	999.999	64.960	76.720	999.999	999.999	42.560	81.760	62.160	
25.760	52.080	71.680	82.320	999.999	999.999	999.999	999.999	63.280	61.600	
999.999	59.920	999.999	45.920	104.160	999.999	28.560	47.040	62.720	64.400	
999.999	57.120	999.999	71.680	999.999	999.999	999.999	79.520	66.080	44.800	
999.999	999.999	91.840	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	82.880	96.880	24.640					0
89.600	999.999	71.120	999.999	999.999	999.999	58.240	29.680	64.400	999.999	
21.840	999.999	999.999	999.999	52.640	87.360	62.160	58.240	999.999	999.999	
63.840	999.999	999.999	999.999	87.920	45.920	44.800	999.999	999.999	999.999	
999.999	999.999	999.999	999.999	999.999	77.840	68.320	999.999	999.999	999.999	999.999
57.680	103.040	999.999	82.880	999.999	46.480	78.400	103.600	999.999	999.999	
82.880	73.360	999.999	999.999	999.999	999.999					0
70.000	999.999	59.360	999.999	999.999	999.999	43.120	28.560	74.480	40.880	
17.920	999.999	999.999	999.999	40.880	90.160	50.960	54.880	999.999	64.960	

58.800	72.800	53.760	53.760	86.800	40.880	999.999	54.320	41.440	999.999
999.999	69.440	117.040	999.999	60.480	66.640	50.400	999.999	73.920	57.120
48.720	125.440	999.999	70.560	999.999	42.000	81.760	135.520	999.999	59.360
108.640	68.320	999.999	999.999	999.999	999.999				0
74.480	61.600	53.760	58.240	999.999	999.999	30.800	999.999	999.999	38.640
999.999	999.999	999.999	999.999	55.440	80.080	43.120	40.880	999.999	45.920
50.960	70.560	44.800	35.840	999.999	27.440	999.999	38.640	29.680	999.999
999.999	44.800	999.999	999.999	54.880	60.480	43.680	67.200	58.800	36.400
43.120	999.999	87.920	65.520	999.999	30.800	90.720	999.999	72.240	51.520
98.000	55.440	999.999	999.999	86.240	999.999				0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	44.240	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	62.720	999.999	999.999	999.999	999.999	999.999
999.999	999.999	120.960	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	94.080	999.999	999.999	999.999	999.999	75.600	98.000	999.999	999.999
149.520	999.999	999.999	999.999	999.999	999.999				0
76.160	51.520	67.760	37.520	60.480	999.999	48.720	999.999	999.999	34.720
33.600	58.240	77.280	51.520	999.999	90.720	47.040	39.760	33.040	34.160
64.960	42.560	29.680	18.480	106.960	46.480	999.999	24.080	35.280	57.680
26.880	30.800	999.999	71.680	36.400	78.960	48.160	48.720	35.280	19.040
47.040	999.999	61.040	82.320	999.999	47.040	105.840	999.999	52.080	44.240
999.999	51.520	999.999	51.520	66.080	20.160				0
73.920	999.999	35.840	999.999	999.999	999.999	43.120	999.999	71.120	45.920
28.560	999.999	999.999	999.999	38.080	99.120	36.400	47.600	999.999	999.999
47.600	999.999	63.280	999.999	82.880	35.840	999.999	999.999	38.640	999.999
999.999	999.999	97.440	999.999	56.560	43.120	44.800	999.999	999.999	999.999
49.840	110.320	999.999	59.920	999.999	22.400	73.920	124.880	999.999	71.120
64.960	90.160	999.999	999.999	999.999	999.999				0
71.680	999.999	41.440	999.999	999.999	999.999	39.760	53.760	70.000	39.200
23.520	999.999	999.999	999.999	50.960	86.240	48.720	45.920	999.999	44.240
57.120	59.920	57.120	34.160	82.880	37.520	999.999	40.320	33.040	999.999
35.840	48.720	999.999	999.999	54.320	46.480	45.920	999.999	999.999	37.520
47.040	999.999	999.999	67.200	999.999	38.640	73.920	999.999	999.999	50.960
103.040	66.640	999.999	999.999	999.999	999.999				0
68.880	999.999	55.440	999.999	999.999	999.999	36.960	53.200	999.999	39.760
19.040	999.999	999.999	999.999	55.440	83.440	46.480	45.920	57.680	51.520
54.880	65.520	40.880	40.320	999.999	33.040	999.999	41.440	33.600	999.999
999.999	55.440	999.999	999.999	47.600	62.160	43.680	999.999	60.480	44.240
44.240	999.999	999.999	999.999	999.999	34.160	999.999	999.999	999.999	50.960
101.360	62.160	999.999	999.999	999.999	999.999				0
73.360	62.160	71.680	53.200	54.880	999.999	46.480	999.999	999.999	38.640
25.760	46.480	64.400	50.960	999.999	98.560	50.960	49.840	45.920	47.040
68.320	49.280	34.160	37.720	108.080	47.600	999.999	31.360	39.200	999.999
14.560	52.080	999.999	999.999	40.880	78.400	53.760	56.560	54.320	37.520
49.280	999.999	69.440	81.760	999.999	47.040	108.640	999.999	69.440	49.280
999.999	52.640	999.999	67.200	81.200	11.760				0
50.400	52.080	63.280	53.760	999.999	999.999	34.720	999.999	999.999	20.720
999.999	55.440	74.480	71.120	67.760	75.600	48.720	27.440	41.440	28.560
64.400	44.240	12.320	36.960	999.999	33.600	999.999	27.440	29.680	81.760
999.999	55.440	999.999	999.999	19.040	62.160	33.040	55.440	33.600	42.000
39.760	999.999	61.040	75.040	999.999	47.600	102.480	999.999	62.720	26.880
96.880	30.800	999.999	87.360	78.400	999.999				0
72.240	74.480	62.720	58.240	63.840	108.640	42.560	999.999	999.999	41.440
22.960	47.600	66.640	73.360	999.999	90.720	59.360	52.080	56.560	50.960
63.280	58.800	42.000	40.320	96.320	40.320	999.999	40.880	37.520	53.760

24.080	54.320	999.999	63.840	48.720	69.640	50.960	70.560	59.360	43.680
50.400	999.999	82.880	62.720	84.000	43.120	91.280	999.999	73.920	56.560
999.999	68.880	999.999	73.360	87.360	24.640				0
67.200	999.999	52.080	999.999	999.999	999.999	35.280	57.120	65.520	35.280
26.320	999.999	999.999	999.999	46.480	81.760	44.800	45.920	999.999	999.999
53.200	62.160	52.640	38.080	77.840	33.600	999.999	44.800	29.120	999.999
999.999	53.200	109.200	999.999	54.880	58.800	42.000	999.999	999.999	42.000
42.560	116.480	999.999	63.280	999.999	34.160	73.360	127.120	999.999	50.960
99.680	62.160	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	53.760	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	63.840	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	63.840	999.999	999.999	999.999	999.999	45.360	75.600	999.999	999.999
65.520	999.999	999.999	999.999	999.999	999.999				0
94.640	999.999	75.600	999.999	999.999	999.999	63.280	34.720	68.880	57.680
24.640	999.999	999.999	999.999	57.680	92.400	67.200	62.720	999.999	77.280
68.320	86.240	63.840	82.320	92.960	50.400	45.360	70.000	50.400	999.999
999.999	70.560	999.999	999.999	76.720	82.320	73.360	999.999	999.999	83.440
62.160	999.999	999.999	87.920	999.999	51.520	83.440	999.999	999.999	83.440
87.920	78.400	999.999	999.999	999.999	999.999				0
54.880	59.920	53.200	70.560	999.999	999.999	34.160	999.999	999.999	19.600
999.999	999.999	999.999	999.999	61.040	85.120	34.160	24.640	41.440	39.200
56.560	56.000	25.200	36.960	999.999	25.760	999.999	36.400	24.080	999.999
999.999	50.960	999.999	999.999	36.400	52.080	32.480	84.560	52.080	39.760
33.600	999.999	72.800	68.320	999.999	40.880	98.000	999.999	75.600	44.800
103.040	48.720	999.999	62.720	95.200	999.999				0
90.720	999.999	82.320	999.999	999.999	999.999	61.600	18.480	52.640	999.999
999.999	999.999	999.999	999.999	54.880	96.880	60.480	59.360	999.999	87.360
67.760	89.040	68.320	67.760	80.080	51.520	39.760	72.800	47.040	999.999
999.999	78.400	105.840	999.999	73.360	78.400	70.000	999.999	999.999	70.560
59.920	108.080	999.999	72.800	999.999	52.640	91.840	94.640	999.999	74.480
114.800	78.400	999.999	999.999	999.999	999.999				0
38.080	999.999	46.480	999.999	999.999	999.999	48.160	999.999	999.999	999.999
999.999	999.999	999.999	999.999	68.320	43.680	45.920	45.360	999.999	999.999
54.320	999.999	54.320	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	69.440	999.999	47.600	40.880	33.600	999.999	999.999	999.999
40.880	999.999	999.999	62.720	999.999	999.999	117.600	999.999	999.999	39.200
61.040	35.840	999.999	999.999	999.999	999.999				0
87.920	999.999	50.960	999.999	999.999	999.999	34.320	38.640	72.800	53.200
24.640	999.999	999.999	999.999	42.560	72.240	37.680	59.360	999.999	999.999
49.280	999.999	59.920	999.999	90.160	52.080	999.999	999.999	48.160	999.999
999.999	999.999	99.680	999.999	71.120	46.480	65.520	999.999	999.999	999.999
62.720	128.240	999.999	80.640	999.999	53.200	85.680	142.800	999.999	71.120
87.360	82.320	999.999	999.999	999.999	999.999				0
999.999	64.960	999.999	50.400	38.080	91.840	999.999	999.999	999.999	999.999
999.999	27.440	46.480	47.040	999.999	999.999	999.999	999.999	49.280	53.760
999.999	68.880	999.999	31.360	999.999	999.999	999.999	39.760	999.999	31.920
24.640	31.360	999.999	43.120	999.999	999.999	999.999	56.000	47.600	31.360
999.999	999.999	61.600	999.999	61.040	999.999	999.999	999.999	62.720	999.999
999.999	999.999	45.360	47.600	76.160	999.999				0
999.999	999.999	999.999	999.999	999.999	999.999	999.999	36.960	72.800	999.999
44.240	59.920	78.960	81.200	999.999	999.999	999.999	999.999	74.480	76.720
999.999	86.240	999.999	56.560	105.840	999.999	16.800	61.600	64.960	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	75.040	58.800

999.999	999.999	999.999	999.999	999.999	999.999	999.999	101.360	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	106.400	999.999	999.999	47.600	54.320	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	65.520	84.560	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	999.999	999.999	999.999	35.840
999.999	999.999	999.999	28.000	999.999	999.999	999.999	999.999	101.360	999.999	999.999	999.999
999.999	999.999	87.360	999.999	29.120	999.999	999.999	999.999	999.999	108.640	999.999	
999.999	999.999	16.800	80.080	119.840	999.999						0
999.999	52.640	999.999	35.280	55.440	103.040	999.999	999.999	999.999	999.999	999.999	999.999
999.999	34.160	53.200	53.760	999.999	999.999	999.999	999.999	33.600	39.760		
999.999	59.360	999.999	19.600	999.999	999.999	999.999	999.999	29.680	999.999	43.120	
17.360	35.840	146.720	54.320	45.920	999.999	999.999	49.840	35.280	20.160		
999.999	999.999	65.520	999.999	71.120	999.999	999.999	999.999	53.200	999.999		
999.999	999.999	58.240	56.000	67.200	15.680						0
999.999	76.720	999.999	58.240	34.160	91.280	999.999	999.999	999.999	999.999	999.999	
999.999	31.920	50.960	56.000	999.999	999.999	999.999	999.999	53.200	57.120		
999.999	76.720	999.999	39.760	999.999	999.999	999.999	47.600	999.999	30.800		
30.800	39.200	999.999	39.760	999.999	999.999	999.999	63.840	56.000	39.760		
999.999	999.999	70.000	999.999	60.480	999.999	999.999	999.999	70.560	999.999		
999.999	999.999	46.240	56.000	84.000	999.999						0
44.240	999.999	28.560	999.999	999.999	999.999	28.560	999.999	88.480	999.999		
999.999	999.999	999.999	999.999	46.480	54.880	13.440	27.440	999.999	999.999		
21.840	999.999	29.120	999.999	88.480	17.360	999.999	999.999	17.920	999.999		
999.999	999.999	103.040	999.999	36.960	35.280	27.440	999.999	999.999	999.999		
18.480	127.120	999.999	40.320	999.999	15.120	84.000	137.200	999.999	44.800		
72.800	52.080	999.999	999.999	999.999	999.999						0
84.560	999.999	66.080	999.999	999.999	999.999	53.200	35.280	69.440	999.999		
24.080	999.999	999.999	999.999	48.160	82.320	57.120	999.999	999.999	999.999		
58.800	999.999	999.999	999.999	82.880	40.880	999.999	999.999	45.360	999.999		
999.999	999.999	117.600	999.999	999.999	72.800	63.280	999.999	999.999	999.999		
52.640	97.440	999.999	77.840	999.999	41.440	73.360	98.560	999.999	999.999		
77.840	68.320	999.999	999.999	999.999	999.999						0

C-----

C Second part

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

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

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

C .

6

C Third part

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

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

ISSN 1062-1024 • 1999 • Volume 22 Number 22 • 10

• per le regole di gioco

• port regular to fit export

C surplus region 2 \rightarrow part E

-----> port region 2 for export

c **f** **g** **h** **i** **j** **k** **l** **m** **n** **o** **p** **q** **r** **s** **t** **u** **v** **w** **x** **y** **z**

-----> port region 2d for export

•

•
•
•
•
•

C surplus region 58 -----> port region 1 for export
C . -----> port region 2 for export
C .
C .
C .-----> port region 20 for export

The rail costs from each surplus region to the 20 ports. Thus, a 58 by 20 matrix.

C For example,
C from the 1st surplus region, Alexandria LA(111) to Mobile, AL (701)
C = 24.080 cents/bushel

24.080	17.360	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	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
69.460	57.680	64.400	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	25.760
35.860	132.720	122.080	121.520	124.320	133.280	999.999	999.999	999.999	999.999	999.999	999.999	25.760	35.840
62.720	60.480	83.460	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	8.960	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	8.960	999.999
999.999	999.999	999.999	19.040	21.000	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
64.480	71.120	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	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
36.400	82.320	85.120	111.440	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	92.400	82.880	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
36.960	83.770	84.720	114.240	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	114.240
76.160	84.560	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
64.960	73.360	999.999	999.999	999.999	999.999	999.999	62.160	29.680	20.720	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	20.720	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	76.720	84.560	105.280	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	96.320	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	93.520	96.320	98.000	91.280	98.560	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
63.280	62.720	58.800	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
57.120	61.040	999.999	999.999	999.999	73.360	23.520	14.560	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	14.560	999.999	999.999	999.999	999.999	999.999	999.999
59.920	63.280	52.080	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
50.960	57.680	999.999	999.999	999.999	53.200	47.040	34.160	999.999	999.999	999.999	999.999	999.999	22.960
999.999	999.999	999.999	999.999	999.999	999.999	999.999	34.160	999.999	999.999	999.999	999.999	999.999	999.999
76.160	71.680	80.080	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	23.520
37.520	143.360	133.280	139.440	136.960	141.120	999.999	999.999	999.999	999.999	999.999	999.999	23.520	37.520
68.320	64.960	81.760	999.999	999.999	999.999	999.999	999.999	56.560	30.240	999.999	999.999	10.640	
999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	10.640	999.999
75.040	66.640	74.480	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	31.360	
37.520	105.280	106.960	122.640	119.280	128.240	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
62.720	57.680	80.080	999.999	999.999	999.999	999.999	999.999	58.800	33.040	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	83.460	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	30.240	
17.360	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	30.240	17.360
48.160	46.480	70.560	999.999	999.999	999.999	999.999	999.999	70.560	35.840	31.920	19.600		

84.000	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	106.960	109.760	125.440	129.360	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	17.360	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	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
78.960	65.520	72.240	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	36.400
39.760	116.480	106.960	134.960	129.920	139.440	999.999	999.999	999.999	999.999	36.400	39.760			
57.680	66.960	999.999	999.999	999.999	54.320	38.080	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	15.680	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	88.080	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	17.024
20.160	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	17.024	20.160			
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	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
43.120	49.840	82.320	999.999	999.999	999.999	999.999	53.760	35.280	999.999	20.720	999.999	20.720		
999.999	999.999	999.999	999.999	999.999	999.999	999.999	35.280	999.999	20.720	999.999	999.999	999.999	999.999	999.999
65.520	73.360	999.999	999.999	999.999	62.160	34.960	10.520	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	10.520	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	50.960	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	999.999	999.999	999.999	122.640	111.440	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999
999.999	999.999	88.080	999.999	999.999	999.999	999.999	999.999	999.999	999.999	999.999	43.680			
30.240	85.000	95.000	120.400	113.680	131.040	999.999	999.999	999.999	43.680	30.240				

C-----

C Fourth part

C (4) Each barge-unloading river location is linked to 56
C deficit regions.

C barge-unloading river location 1 -----> deficit region 1
C -----> deficit region 2
C
C
C
C -----> deficit region 56

C barge-unloading river location 2 -----> deficit region 1
C -----> deficit region 2
C
C
C
C -----> deficit region 56

C
C
C
C
C barge-unloading river location 43 -----> deficit region 1
C -----> deficit region 2
C
C
C
C
C -----> deficit region 56

C The rail costs from each river port to the 56 deficit regions.
C There are 43 river ports. Thus a 43 by 56 matrix.

C For example,

C from the 1st barge-unloading river location,
C St. Paul, MN(601) to the 1st deficit region, Abilene (064)
C - 999.999 cents/bushel

C For example,

C from the 4th barge-unloading river location,
C Dubuque, IA(604) to the 11th deficit region, Cedar Rapids (086)
C = 14.000 cents/bushel

F4.CRN

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

C The barge costs from 43 river regions to 5 river shipping locations.
C The 5 river shipping locations are :

C Nashville, TN (625)
C Knoxville, TN (626)
C Chatanooga, TN (627)
C Guntersv, AL (628)
C Florence, AL (629)

C For example,

C the barge cost from the 1st river barge region,

C St. Paul, MN (601) to Nashville TN (625) = 0.282 \$/bushel
C St. Paul, MN (601) to Knoxville TN (626) = 0.305 \$/bushel
C St. Paul, MN (601) to Chatanooga TN (627) = 0.279 \$/bushel
C St. Paul, MN (601) to Guntersv AL (628) = 0.265 \$/bushel
C St. Paul, MN (601) to Florence AL (629) = 0.251 \$/bushel

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	0.178	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	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0
999.000	999.000	999.000	999.000	999.000	999.000	0

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

C For example, the 1st river region (St. Paul)
C St. Paul MN (601) to New Orle, LA (702) = 0.164 \$/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

0.164	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	0.369	0.419	0.280	999.000	0
0.284	0.341	0.391	0.252	999.000	0
0.371	0.401	0.518	0.328	999.000	0
0.332	0.371	0.478	0.288	999.000	0
0.307	0.353	0.454	0.264	999.000	0
0.263	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
0.086	0.147	0.244	999.000	999.000	0
0.077	0.125	0.211	999.000	999.000	0
0.102	0.101	0.225	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 The number of river shipping ports above L&D 26
13

The codes for the river shipping points.
For example, 601 is the code for St Paul MN.

c Shipments from 20 ports to 25 world demand locations.

For example, the shiprate from the 1st port
Mobile, AL (701) to 1st world demand location, Scandinavia (801)
= 0.443 \$/bushel
Mobile, AL (701) to 2nd world demand location, N.C. Europe(802)
= 0.349 \$/bushel

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	0.586						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	0.407	0.317	0.401	0.249	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 Number of ports at Great Lakes for export

8		0
---	--	---

C-----

C Codes for the above ports

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

C (719) Chicago, SC
C (720) Duluth, SC

708 709 710 711-717 718 719 720

0

C-----

F8.CRN

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

C-----

C These lines show the amount of surplus produced by each surplus region.

C For example, the 1st surplus region,

C Alexandria (111) produces 0.314 million/bushel.

0.314	161.917	23.468	4.035	47.719	42.029	135.166	10.053	14.224	1.116
3.342	79.220	22.535	22.107	23.062	94.602	0.000	142.148	148.080	13.917
160.172	24.962	69.460	53.652	47.342	150.510	105.739	9.646	51.702	181.239
138.346	70.598	77.423	111.318	148.192	24.793	110.007	90.644	72.595	24.005
26.115	144.679	31.974	42.936	69.730	0.008	60.568	14.919	89.884	5.203
115.326	56.230	17.974	28.141	64.257	83.648	5.034	75.112		0

C-----

C These lines show the amount of corn demanded by the 56 deficit regions.

C For example, the 1st deficit region,

C Abilene (064) requires 18.512 million/bushel.

18.512	67.704	60.958	50.414	72.211	16.424	10.413	5.166	4.431	12.700
154.968	13.486	88.287	18.610	1.445	41.339	6.422	115.060	104.124	31.330
2.214	34.242	37.571	8.190	7.362	10.884	2.972	23.433	9.148	1.977
5.747	94.124	166.786	77.612	22.854	17.203	8.701	74.967	51.683	21.584
30.637	10.130	64.614	2.804	42.449	21.856	2.269	33.494	33.859	23.574
21.161	3.849	3.812	81.095	31.579	45.430				0

C-----

C The remaining lines show the amount demanded by 25 foreign regions

C (demand per quarter).

C For example, the 1st foreign region, Scandinavia (801)

C in the 1st quarter, requires 4.017 million bushel

C in the 2nd quarter, requires 1.731 million bushel

C in the 3rd quarter, requires 3.823 million bushel

C in the 4th quarter, requires 4.170 million bushel.

4.017	1.731	3.823	4.170		0
26.864	4.449	3.793	2.702		0
9.616	13.325	19.245	15.573		0
3.728	0.000	0.308	0.000		0
1.083	1.086	1.669	1.115		0
12.440	140.733	0.000	53.372		0
0.902	6.308	8.567	15.173		0
24.693	15.072	33.549	7.173		0
9.714	14.876	10.377	5.999		0
0.232	0.000	0.117	0.018		0
0.000	0.000	0.216	0.308		0
0.055	0.405	0.134	0.419		0
4.789	13.160	9.135	10.797		0
0.210	0.210	0.210	0.210		0

0.000	0.044	0.814	14.250	0
28.189	28.734	26.100	34.350	0
40.238	31.499	56.198	46.950	0
111.428	119.557	134.040	145.373	0
17.237	22.121	3.731	8.472	0
0.000	0.000	0.000	0.000	0
8.044	11.168	11.324	8.512	0
17.916	24.874	25.222	18.960	0
4.548	7.179	5.018	9.479	0
15.155	9.725	4.056	2.637	0
5.209	6.407	7.912	6.510	0

C-----

C

File09.crn

C This is the last and 6th data file with the location names and code
 C number of all involved regions. The grain involved is corn.

C-----

C 58 Surplus regions:

C	Code number	Location name	State
	111	ALEXANDR,	LA
	085	ANES,	IA
	139	AURORA,	IL
	068	AUSTIN,	TX
	163	BATTLE C,	MI
	015	BISMARCK,	ND
	024	BROOKING,	SD
	162	CADILLAC,	MI
	154	CAMBRIDGE,	OH
	351	CASPER,	WY
	032	CHADRON,	NE
	092	CHILlico,	MO
	153	CHILlico,	OH
	094	CLINTON,	MO
	144	COLUMBUS,	IN
	083	DECORAH,	IA
	131	DE KALB,	IL
	084	DENISON,	IA
	130	DIXON,	IL
	121	EAUCLAIR,	WS
	134	EFFINGHA,	IL
	164	FLINT,	MI
	141	FORT WAY,	IN
	132	GALESBUR,	IL
	046	GARDEN C,	KS
	034	HASTINGS,	NE
	142	INDIANAP,	IN
	113	LAFAYETT,	LA
	172	LEXINGTO,	KY
	035	LINCOLN,	NE
	138	LITCHFLO,	IL
	033	MCCOOK,	NE
	123	MADISON,	WI
	073	MANKATO,	MN
	082	MASON CI,	IA
	093	MOBERLY,	MO
	021	MOBRIDGE,	SD
	135	MOUNT VE,	IL
	037	NORFOLK,	NE
	088	OSCEOLA,	IA
	089	OTTUMWA,	IA
	137	PEKIN,	IL
	099	POPLAR B ,	MO
	143	PRINCETO,	IN

087	RED OAK,	IA	0
391	RENO,	NE	0
076	ROCHESTE,	MN	0
096	ROLLA,	MO	0
074	ST.CLOUD,	MN	0
06A	SANANTON,	TX	0
081	SPENCER,	IA	0
152	SPRINGFI,	OH	0
122	STEVENS P,	WS	0
291	SYRACUSE,	NY	0
145	TER HAUTE,	IN	0
131	TIFFIN,	OH	0
045	WICHITA,	KS	0
075	WORTHING,	MN	0

C-----
 C 56 Deficit regions

C	Code number	Location name	State
064	ABILENE	TX	0
213	ALBANY	GA	0
061	AMARILLO	TX	0
211	ATLANTA	GA	0
707	BALTIMORE	MD	0
301	BANGOR	ME	0
052	BARTESVILL	OK	0
071	BEMIDJI	MN	0
351	BILLINGS	MT	0
098	CASOOL	MO	0
086	CEDAR RAPI	IA	0
261	CHARLEST	WV	0
251	CHARLOTTE	VA	0
231	COLUMBIA	SC	0
391	DENVER	CO	0
06H	EL PASO	TX	0
051	ENID	OK	0
101	FT SMTH	AR	0
201	GADSDEN	AL	0
191	GREENWOOD	MS	0
050	GUYMON	OK	0
192	HATTIESBUR	MS	0
102	HOXIE	AR	0
171	HOPKINS	KY	0
342	IDANO FA	ID	0
047	INDEPEND	KS	0
161	IRON MT	WI	0
181	JACKSON	TN	0
097	JOPLIN	MO	0
281	KANE	PA	0
140	KENTLAND	IN	0
183	KNOXVILL	TN	0
715	L.A.	CA	0
282	LEWISBURG	PA	0
104	LITTLE ROC	AR	0

062	LUBBOCK	TX	0
054	MCALISTER	OK	0
212	MCRAE	GA	0
202	MONTGO	AL	0
182	NASHVILLE	TN	0
053	OK CITY	OK	0
713	PORTLAND	OR	0
242	ROCKY MT.	NC	0
411	ROSWELL	NM	0
302	RUTLAND	VT	0
042	SALINA	KS	0
381	SALT LAK	UT	0
311	SEATTLE	WA	0
221	TALLAHASSE	FL	0
103	TEXARKAN	AR	0
401	TUCSON	AZ	0
066	TYLER	TX	0
292	WATERTUN	NY	0
241	WINSTON	NC	0
222	WINT HAV	FL	0
133	CHAMPAIGNE	IL	0

C-----
 C 63 River regions

C Code number Location name State

601	ST. PAUL,	MN	0
602	WINOMA,	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	OMANA,	NE	0
611	NE CITY,	NE	0
612	ST JOSEP.	MO	0
613	KANSAS C.	MO	0
614	GLASCO,	MO	0
615	OTTAWA,	IL	0
616	PEORIA,	IL	0
617	BEARDSTO,	IL	0
618	CINCINNA,	OH	0
619	LOUISVIL,	KY	0
620	EVANSVIL,	IN	0
621	CAIRO,	IL	0
622	HICKMAN,	KY	0
623	OSCEOLA,	TN	0
624	MEMPHIS,	TN	0
625	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
637	LEWISTON,	ID	0
638	CENTRL F,	WA	0
639	PASCO,	WA	0
640	ROOSEVELT,	WA	0
641	THE DALL,	OR	0
642	NEW ORLE,	LA	0
643	PORTLAND,	OR	0

C 20 Port regions

C	Code number	Location name	State
701	MOBILE,	AL	0
702	NEW ORLE,	LA	0
703	GALVESTN,	TX	0
704	CORPUS C,	TX	0
705	BROWNSVI,	TX	0
706	CHARLEST,	SC	0
707	BALTINOR,	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 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 322688163%

CORN SHIPMENT PROBLEM, BAIE COMEAU ADDED - FULLER, GRANT, TEN,
UNIT = 1 (THOUSAND-BUSHEL.)

SUPPLY 3625116.

Demand 3622857.

890 891 891 30238

1 Network Generator For Grain (CORN) Shipment Problem

CORN SHIPMENT PROBLEM, BAIE COMENU ADDED - FULLER, GRANT, TEH.

P 707 BALTIMOR, MD R	526.	13700.	00.	00.	14226.	0.29680	4221686.	2055368.
S 351 CASPER, WY	1116.							
D 381 SALT LAK UT T	00.	00.	549.	567.	1116.	0.38553	430248.	211393.
S 092 CHICAGO, IL	3342.							
D 381 SALT LAK UT T	567.	567.	18.	00.	1152.	0.55864	643326.	218212.
D 311 SEATTLE WA R	00.	722.	1468.	00.	2190.	0.90160	1974506.	373410.
S 092 CHILLICO, MO	79220.							
D 213 ALBANY GA R	00.	10348.	4608.	9237.	26193.	0.71120	17206062.	4147164.
D 052 BROOKVILLE T	2603.	2603.	2603.	2605.	10412.	0.26591	2976920.	1972261.
D 197 GREENWOOD MS R	00.	00.	00.	7323.	7323.	0.50400	3690792.	1255309.
D 102 NODIE AR R	00.	00.	00.	3375.	3375.	0.37520	1266300.	578543.
D 047 INDEPEND KS T	00.	00.	592.	00.	592.	0.24580	1453388.	112137.
D 181 JACKSON TN R	5858.	5858.	5858.	5858.	23632.	0.36160	8004371.	4016714.
D 097 JOPLIN MO T	2287.	2287.	2287.	2287.	9148.	0.23670	2165332.	1732814.
D 106 LITTLE ROCKAR T	00.	00.	00.	745.	745.	0.40526	301920.	1411118.
S 153 CHILLICO, OH	2235.							
P 707 BALTIMOR, MD R	2102.	538.	117.	18.	2235.	0.23520	5300232.	3256308.
S 096 CLINTON, MO	22107.							
D 106 LITTLE ROCKAR T	5713.	5713.	5713.	4968.	22107.	0.29813	6590744.	4187508.
S 144 COLUMBUS, OH	2302.							
D 202 MONTGOM AL R	10650.	12178.	00.	234.	2302.	0.33600	774352.	3953288.
S 085 DODGE, IA	94602.							
R 603 MCGREGOR, IA T	00.	48119.	34617.	11866.	94602.	0.06886	6512737.	15961249.
S 131 DE KALB IL	00.							
S 086 DODGE, IA	142148.							
R 610 OMAHA, NE T	00.	22056.	22490.	97822.	142148.	0.07514	10680585.	25983210.
S 130 DODGE, IL	148020.							
D 192 HATTIESBURG R	8560.	8560.	8560.	8560.	34260.	0.50960	17448704.	5869421.
D 221 TALLAWASSEFL R	864.	864.	864.	864.	33856.	0.71120	24072988.	5803596.
D 222 WINT HAW FL R	7894.	7894.	7894.	7894.	31576.	0.85120	26877492.	5412758.
D 133 CHAMPAIGNEIL R	11357.	11357.	11357.	11357.	45428.	0.16240	7377507.	7787268.
P 710 CHICAGO, IL T	00.	00.	770.	2210.	2980.	0.10556	314356.	512113.
S 121 ENCLAIR, MS	13917.							

D 185 KNOXVILLE TN R	00.	4889.	9028.	00.	13917.	0.30240	4208801.	2385652.
S 134 EFFINGHAM IL	140172.							
D 231 CHARLOTTESVILLE VA R	22071.	22071.	22071.	22071.	88284.	0.45360	40045624.	15133643.
D 201 GOSDEN AL R	4129.	20211.	13231.	14317.	51888.	0.42000	21792960.	8894641.
S 164 FLINT MI	24952.							
D 231 COLUMBIA SC R	482.	482.	482.	482.	18808.	0.39760	7398541.	3189784.
D 292 WATERTOWN NY R	953.	953.	953.	953.	3912.	0.48720	1857207.	653453.
P 718 SEDGWICK BC T	00.	2542.	00.	00.	2542.	0.05528	88677.	436843.
S 141 FORT WAYNE IN	69460.							
D 707 BALTIMORE MD R	15304.	18052.	18052.	18052.	69460.	0.38080	26450368.	11906533.
S 132 GALESBURG IL	53652.							
R 616 PEORIA IL T	45257.	00.	00.	895.	53652.	0.05206	2798167.	9052165.
S 016 GARDEN CITY KS	47342.							
D 056 ABILENE TX T	4628.	4628.	4628.	4628.	18512.	0.40244	7450019.	3504543.
D 061 AMARILLO TX T	00.	00.	10718.	6676.	17394.	0.23670	4117160.	3294772.
D 051 ENID OK R	1605.	1605.	1605.	1605.	6420.	0.24080	1545536.	1100516.
D 050 GLASSON OK T	553.	553.	553.	553.	2212.	0.11185	247411.	418997.
D 411 ROSWELL NM T	701.	701.	701.	701.	2804.	0.41372	1160070.	531134.
S 034 HASTINGS NE	150510.							
P 713 PORTLAND OR R	57473.	22121.	45613.	25301.	150510.	0.61710	92879720.	21748694.
S 142 INDIANAPOLIS IN	103739.							
D 185 KNOXVILLE TN R	10175.	6853.	14503.	10746.	42277.	0.33600	14205072.	7247126.
D 202 MONTGOMERY AL R	2270.	00.	7978.	4525.	14773.	0.40520	5956474.	2532388.
D 302 RUTLAND VT R	00.	00.	00.	966.	966.	0.67760	654562.	165592.
P 707 BALTIMORE MD R	00.	23489.	00.	24234.	47723.	0.43680	20845406.	6895974.
S 113 LAFAYETTE LA	9846.							
P 702 NEW ORLEANS LA R	9846.	00.	00.	00.	9846.	0.20000	1929200.	1395847.
S 172 LEXINGTON KY	51702.							
D 185 KNOXVILLE TN R	13356.	00.	00.	00.	13356.	0.17360	2518802.	2289486.
D 261 WINSTON NC R	3371.	3371.	11331.	20273.	33346.	0.33040	12669519.	6573272.
S 035 LINCOLN NE	181239.							
D 101 FT SMITH AR R	28765.	28765.	28765.	28765.	115060.	0.39200	45103524.	19723586.
D 047 INDEPENDENCE KS R	2721.	2721.	2129.	2721.	10292.	0.28560	2393355.	1764255.
D 054 McALESTER OK R	2175.	2175.	2175.	2175.	8700.	0.39760	3459120.	1491354.
D 053 OK CITY OK T	3529.	4840.	7659.	2625.	18653.	0.39680	7401587.	3533251.

D 042 SALINA KS R		5464.	5464.	5464.	5464.	21856.	0.19600	4265776.	3746556.
P 703 GALVESTN, TX R		00.	2490.	4188.	00.	6678.	0.56100	3746358.	964971.
S 138 LITCHFLD, IL T	138346.								
R 603 ST LOUIS, MO T	138346.	00.	00.	00.	138346.	0.05626	7782862.	23341736.	
S 033 MCCOOK, NE	70598.								
D 391 DENVER CO R		361.	361.	361.	361.	1444.	0.21200	307283.	267530.
D 715 LA. CA R		40366.	27466.	722.	00.	69154.	0.56320	56927572.	11654379.
S 123 NEWTON, WI	77621.								
D 171 HOPKINS KY T		2047.	2047.	2047.	2047.	8185.	0.21989	1800443.	1550971.
R 606 DUBLIN, IA T		00.	11210.	23675.	34350.	69235.	0.10361	7090286.	11681329.
S 073 MINNEAPOLIS, MN	111318.								
R 602 WIXOMA, MS T		00.	00.	82173.	29145.	111318.	0.14017	15603455.	15781572.
S 082 MEXICO CI, IA	148192.								
R 603 MCEWEN, IA R		00.	138604.	9588.	00.	148192.	0.14560	21576756.	24727316.
S 053 MCHENRY, MD	26795.								
R 607 HARRISON, MD T		00.	10348.	00.	14445.	26795.	0.06989	1732844.	4183075.
S 021 MORRIDGE, SD	110007.								
D 342 IDAHO FA ID R		1840.	1840.	1840.	1840.	7360.	0.52720	4616192.	1251651.
D 713 PORTLAND OR R		00.	00.	145.	2532.	2677.	0.94080	2518522.	458891.
D 311 SEATTLE WA R		00.	233.	919.	00.	1152.	0.98000	1128960.	197476.
P 712 SEATTLE, WA R		29601.	4560.	15290.	45357.	98818.	0.80880	79924000.	14279201.
S 135 MOUNT VIE, IL	90644.								
D 201 GROSVENOR AL R		14543.	3090.	6706.	1683.	26032.	0.33040	8600973.	4462406.
D 242 ROCKY MT. NC T		16153.	16153.	16153.	16153.	64612.	0.52555	33956916.	12259905.
S 057 NORFOLK, NE	72935.								
D 061 AMARILLO TX R		15239.	15239.	4521.	8563.	43562.	0.35840	15612620.	7467398.
D 401 TUCSON AZ R		5290.	5290.	5290.	5290.	21160.	0.64980	13745535.	3527247.
P 713 PORTLAND, OR R		00.	00.	7873.	00.	7873.	0.70220	5528421.	1137849.
S 089 OCEOLA, IA	24005.								
D 191 GREENWOOD MS R	.	7832.	7832.	7832.	509.	24005.	0.44240	10619812.	4114937.
S 089 OTTUMWA, IA	26115.								
D 140 KENTLAND IN T		1436.	1436.	1436.	1436.	5744.	0.09402	540037.	1088029.
R 605 BURLINGT, IA T		00.	8582.	11789.	00.	20371.	0.07933	1616087.	3436995.

S 122 STEVENS P, MS	179.6.	
D 261 CHAMBERS W T	16902. 357. 00.	427.6. 0.3962. 16881446. 256139.
D 261 CHAMBERS W T	16902. 357. 00.	427.6. 0.3962. 16881446. 256139.
S 125 SPRINGER, OH		56250.
R 609 SIOUX CT, IA T	00. 26750. 17470. 4300. 4300. 4300. 17200. 0.46680 7994560. 291826.	0 052 LUMBER TX R
D 059 EL PROD TX R	10354. 10354. 10354. 10354. 10354. 10354. 0.2220 29061126. 705887.	R 609 SIOUX CT, IA T
S 061 SPRINGER, IA		115350.
P 706 COOPER C, TX R	00. 269. 269. 269. 269. 269. 0.15680 815890. 751824.	S 060 SUMMERS, TX
R 601 ST. PAUL, MN T	00. 26550. 16902. 16902. 16902. 16902. 16902. 0.07725 691212. 15165226.	S 061 ST. CLAUD, MN
D 102 HODGE MN T	375. 375. 375. 375. 375. 375. 0.06880 824313. 2610526.	D 058 CHERRY ND T
S 060 HODGE, ND		16919.
R 602 MUNSON, MN T	00. 2720. 2720. 2720. 2720. 2720. 0.05205 3153220. 10219035.	P 716 SWEENEY, CO R
S 065 NOONAN, MN		69200.
P 716 SWEENEY, CO R	00. 26. 26. 26. 26. 26. 0.21860 1727. 1156.	S 341 NEMO, MN
S 067 NEMO, IA		35269.
D 102 MUNSON MN R	00. 11720. 11720. 11720. 11720. 11720. 0.05080 1201221. 401072.	D 102 HODGE MN R
D 102 HODGE MN R	00. 11720. 11720. 11720. 11720. 11720. 0.05080 1201221. 401072.	D 102 HODGE MN R
R 620 EMMERT, IA T	35253. 69. 210. 350. 429. 210. 0.05218 1626162.	R 620 EMMERT, IA T
S 163 PUNTERO, IN		42624.
D 102 HODGE MN T	35252. 35252. 35252. 35252. 35252. 35252. 0.07725 266909. 6916515.	D 213 ALBANY GA R
S 069 POWELL S, GA	35252.	
D 281 GROSSEIN AL R	18741. 18741. 18741. 18741. 18741. 18741. 0.45920 1205257. 1191890.	D 212 HOUSE GA R
D 212 HOUSE GA R	18741. 18741. 18741. 18741. 18741. 18741. 0.45920 1205257. 1191890.	D 213 ALBANY GA R
S 177 REEDIN IL	166279.	

D 161 100A RT	ME	R	753.	753.	753.	753.	2572.	0.16200	499226.	502460.	
P 720 000H, SC	R		00.	4559.	1043.	00.	1502.	0.20160	322445.	216768.	
S 291 STOOLIE W	-										
D 312 HULLAND VT	T										
S 145 TEE HULTE, IN			6257.								
D 211 ATLANTA GA	R			1652.	1652.	1652.	2012.	0.35200	1776532.	866163.	
D 232 KENTON AL	R			00.	752.	4932.	6161.	1386.	0.35200	488516.	257510.
S 151 TIPPIK, OR			8542.								
D 707 BARTHOLOME W	R			2752.	00.	00.	2752.	0.34160	98577.	471082.	
D 222 WILMING PA	T			1942.	1942.	1942.	17612.	0.34778	2700466.	1470726.	
D 312 RUMBLE VT	T			00.	00.	00.	3252.	0.35666	192653.	62613.	
S 005 WINTON, KS			3052.								
D 080 0C CITY OR	R			00.	00.	00.	5052.	0.18680	98025.	82828.	
S 005 WINTON, KS			7512.								
R 609 000 CT, IA	T			00.	4257.	2257.	952.	7512.	0.0612	721925.	1257266.
<hr/>											
SUPPLY NUMBER	SUPPLY	STONE	STONE	STONE	STONE	STONE	STONE	STONE	STONE	STONE	
S 111 ALEXANDER, LA	374.	00.	00.	00.	00.	00.	00.	00.	00.	00.	
S 005 ALEX, LA	16577.	11935.	7752.	3572.	905533.						
S 109 ALBION, IL	2542.	2262.	1151.	5132.	151615.						
S 009 ALBION, TX	4035.	4035.	00.	00.	15672.						
S 163 BATTLE C, MI	4779.	4319.	3133.	1460.	327526.						
S 005 BIRMING, MD	42029.	3007.	1680.	9420.	26737.						
S 024 BOSTON, SD	13516.	13266.	10572.	52134.	1110175.						
S 162 BRUNAC, HI	10023.	10023.	10023.	6352.	101874.						
S 154 CINCINNATI, OH	14226.	13702.	00.	00.	52237.						
S 351 COEUR, ID	1116.	1116.	1116.	567.	10767.						
S 002 CRANBROOK, BC	3542.	2752.	1425.	00.	16356.						
S 002 CRYSTAL, ND	79220.	6852.	4756.	31428.	566576.						
S 155 CRYSTAL, ND	2253.	672.	15.	18.	3142.						
S 004 CRYSTAL, ND	22107.	1652.	1059.	4928.	123176.						
S 164 CRYSTAL, ND	2302.	1262.	29.	29.	42352.						
S 005 CRYSTAL, ND	94402.	9442.	6632.	1182.	527190.						
S 151 DE HAB, IL	00.	00.	00.	00.	00.						
S 005 DE HAB, IL	14218.	14218.	1212.	97522.	138026.						
S 165 DIXON, IL	2462.	11165.	75321.	3945.	882799.						
S 121 EAUCLAI, WI	13917.	13917.	9028.	00.	897579.						
S 154 EFRYTHE, IL	14072.	11372.	7160.	3638.	826621.						
S 161 FLINT, MI	2462.	19557.	11210.	3625.	135042.						
S 141 FORT WAY, IN	6940.	54156.	3610.	1802.	416234.						
S 152 GALESBUR, IL	5362.	5555.	5555.	5555.	97066.						
S 005 GREEN C, KS	4732.	3985.	3268.	14163.	321193.						
S 005 HARTMAN, KS	15050.	9305.	7074.	2501.	70722.						

	QUANTITY	LINE	SUPPLY	1	2	3	4	TOTAL SUPPLY	UNIT COST	TOTAL HANDLING COSTS
S 142 INDIANAP, IN	10579.	9528.	6252.	40671.	756518.					
S 113 LAFAYETTE, LA	966.	0.	0.	0.	0.	0.	0.	0.	0.	
S 172LEXINGTON, KY	51702.	34075.	31606.	20273.	334235.					
S 05 LINCOLN, NE	181259.	138885.	92130.	61750.	1067335.					
S 139 LITCHFIELD, IL	13826.	0.	0.	0.	0.	0.	0.	0.	0.	
S 033 MEXICO, MO	70593.	2877.	1444.	351.	1191767.					
S 123 MOISIE, MI	77623.	7536.	6219.	36577.	660048.					
S 073 MARSHALL, MI	111318.	111318.	2945.	962279.						
S 022 NEWTON CT, IA	145182.	145182.	9583.	0.	608616.					
S 055 NORMAN, OK	26755.	26755.	14445.	14445.	265745.					
S 021 OCEANSIDE, CA	110007.	78566.	7853.	55739.	78468.					
S 135 OELMIER, IL	90264.	59763.	45705.	17866.	452450.					
S 057 OMEROK, NE	72555.	52066.	31537.	13853.	376546.					
S 038 OSEOLA, IA	2605.	16173.	8561.	509.	99957.					
S 029 OTTAWA, IA	2615.	26579.	14661.	1636.	196637.					
S 137 PEKIN, IL	14459.	101653.	7386.	34451.	514027.					
S 029 PELLETIER B, MD	31195.	2582.	15192.	3793.	151963.					
S 143 PONCA, IA	4235.	4235.	3579.	3369.	42368.					
S 027 RED COK, IA	69731.	58641.	35353.	2607.	453645.					
S 391 REED, NE	8.	8.	8.	8.	925.					
S 076 ROBBERE, MN	60568.	60568.	35358.	44791.	6113827.					
S 026 ROLLA, MO	14975.	1774.	8569.	5794.	98878.					
S 074 ST. CLAIR, MI	8655.	8655.	4062.	2165.	56400.					
S 014 SUMMIT, TX	5205.	65.	0.	0.	2114.					
S 021 SPACER, IA	15556.	10082.	59860.	27156.	719721.					
S 132 SPRINGER, GA	56230.	35537.	15656.	3371.	211411.					
S 122 STEVENS P, MS	17978.	17251.	11929.	763.	114704.					
S 201 STRASBURG, NY	26141.	17539.	6917.	0.	95311.					
S 145 TEE HAUKE, WI	64257.	51654.	38209.	20766.	423663.					
S 151 TIPPIN, GA	85648.	61497.	42054.	22891.	4856256.					
S 065 WICHITA, KS	5054.	5054.	5054.	59222.						
S 075 WILMINGTON, NC	75112.	3545.	9873.	450863.						

1 QUANTITY LINE SUPPLY 1 2 3 4 TOTAL SUPPLY UNIT COST TOTAL HANDLING COSTS

	R 601 ST. PAUL, MN	R 602 NEW ORLEANS, LA	R 602 NEW YORK, NY							
P 702 NEW ORLEANS, LA	0.	0.	45932.	18830.	2165.					
P 702 NEW YORK, NY	0.	0.	49582.	18603.	2165.	8866.	0.16400	14740576.	1425108.	
P 702 NEW YORK, NY	0.	0.	7210.	90740.	7386.					
P 702 NEW YORK, NY	0.	0.	7210.	90740.	7386.	17186.	0.25000	3535760.	272236.	
P 702 NEW YORK, NY	0.	0.	185725.	44205.	11866.	24275.	0.21400	5195776.	384498.	
P 702 NEW YORK, NY	0.	0.	11210.	25675.	3430.					
P 702 NEW YORK, NY	0.	0.	11210.	25675.	3430.	6925.	0.20400	1412560.	1057209.	

R 605 CLINTON, IA	00.	00.	00.	00.				
R 606 BURLINGT, IA	00.	5532.	11789.	00.				
P 702 NEW ORLE, LA S	00.	5532.	11789.	00.	20371.	0.17000	3463070.	3229822.
R 607 MUNICIPAL, MD	00.	10343.	00.	14445.				
P 702 NEW ORLE, LA S	00.	10343.	00.	14445.	20793.	0.15300	3793329.	3550930.
R 608 ST LOUIS, MO	132346.	00.	00.	00.				
P 702 NEW ORLE, LA S	132346.	00.	00.	00.	132346.	0.11700	16186482.	21954758.
R 609 SIOUX CT, IA	00.	51031.	51367.	31347.				
P 702 NEW ORLE, LA S	00.	51031.	51367.	31347.	163993.	0.31100	51002444.	26001408.
R 610 OMAHA, NE	00.	22036.	22490.	97622.				
P 702 NEW ORLE, LA S	00.	22036.	22490.	97622.	142148.	0.28400	40570052.	22537566.
R 611 NE CITY, NE	00.	00.	00.	00.				
R 612 ST JOSEP, MO	00.	00.	00.	00.				
R 613 KIMBER 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	45257.	00.	00.	5535.				
P 702 NEW ORLE, LA S	45257.	00.	00.	5535.	53652.	0.23100	12955612.	8506525.
R 617 BENEDICT, IL	00.	00.	00.	00.				
R 618 CINCINNATI, OH	00.	00.	00.	00.				
R 619 LOUISVIL, KY	00.	00.	00.	00.				
R 620 EVANSVIL, IN	38733.	624.	210.	3369.				
P 701 MOBILE, AL S	38733.	624.	210.	3369.	42356.	0.10100	4336536.	6807503.

R 621 CAIRO, IL	00.	00.	00.	00.
R 622 HICKMAN, KY	00.	00.	00.	00.
R 623 OSCEOLA, TN	00.	00.	00.	00.
R 624 MEMPHIS, TN	00.	00.	00.	00.
R 625 NASHVILLE, TN	00.	00.	00.	00.
R 626 KNOXVILLE, TN	00.	00.	00.	00.
R 627 CHATANOOGA, TN	00.	00.	00.	00.
R 628 GLENCOE, AL	00.	00.	00.	00.
R 629 FLORENCE, AL	00.	00.	00.	00.
R 630 CATOOSA, OK	00.	00.	00.	00.
R 631 MUSKOGEE, OK	00.	00.	00.	00.
R 632 PINEBLUFF, AR	00.	00.	00.	00.
R 633 DES ARK, AR	00.	00.	00.	00.
R 634 GREENWOOD, MS	00.	00.	00.	00.
R 635 VICKSBURG, MS	00.	00.	00.	00.
R 636 MONROE, LA	00.	00.	00.	00.
R 637 LEONSTON, ID	00.	00.	00.	00.
R 638 CENTRAL F, WA	00.	00.	00.	00.
R 639 PASCO, WA	00.	00.	00.	00.

R 640 ROSENTHAL, LA 00. 00. 00.

R 661 THE DALL, OR 00. 00. 00.

R 662 NEW ORLE, LA 00. 00. 00.

R 663 PORTLAND, OR 00. 00. 00.

1

CONTRACTORS NAME	SUPPLY	1	2	3	4	TOTAL	UNIT	COAT	HULLING COAT	HANDLING COSTS
------------------	--------	---	---	---	---	-------	------	------	--------------	----------------

P 707 NEW ORL, LA	3973.	654.	210.	339.						
F 813 PERUVIAN GULF S	4789.	414.	00.	5159.	882.	0.47700	393857.	47289.		
F 814 U ARA	210.	210.		210.	80.	0.48000	40650.	47244.		
F 815 TAHITI	1239.	00.	00.	1230.	0.43000	522450.	69760.			
F 822 E HOND	9257.	00.	00.	9257.	0.07000	71269.	522914.			
F 826 CENT AMERICA S	1297.	00.	00.	1297.	0.15000	182020.	698940.			
P 702 NEW ORL, LA	19363.	37882.	26976.	28846.						
F 801 SCHILLWATA S	4017.	00.	00.	4017.	0.44300	777531.	22722.			
F 802 N.C. BACRE S	2664.	449.	376.	272.	3762.	0.39000	1514992.	219935.		
F 803 S.A. BACRE S	9616.	1325.	1285.	13573.	5779.	0.29000	1726940.	3269199.		
F 804 ISLA DE S	5720.	00.	00.	5720.	0.32100	1193638.	211005.			
F 805 AFRICAT S	1023.	1026.	1169.	1115.	493.	0.42000	2025119.	20340.		
F 806 USAR S	00.	103006.	00.	29138.	13244.	0.50000	6633236.	747950.		
F 807 E BLOCK BLDG S	902.	603.	8567.	15173.	30250.	0.44800	1383900.	1751770.		
F 808 E MEDITERRAN S	2463.	7609.	18864.	982.	52328.	0.44500	2617236.	2361725.		
F 811 E AFRICA S	00.	216.	308.	524.	0.95600	50394.				
F 812 U AFRICA S	55.	405.	156.	419.	1013.	0.35000	35370.	57336.		
F 813 PERUVIAN GULF S	00.	1276.	9135.	7638.	2819.	0.47700	1402053.	1670775.		
F 814 TAHITI S	2873.	2610.	33380.	10523.	0.43000	4526382.	5955622.			
F 817 HOND S	00.	31493.	6443.	30113.	69875.	0.59000	4019246.	3851975.		
F 818 JAPAN S	81827.	115016.	119366.	110256.	423653.	0.40500	17208560.	2619340.		
F 821 U HOND S	5053.	11168.	11324.	8512.	35047.	0.26000	1028592.	221060.		
F 822 E HOND S	8679.	26378.	23222.	18860.	7775.	0.07000	598895.	4399801.		
F 823 US AMERICA - S	00.	00.	800.	9479.	1059.	0.26000	284857.	582469.		
F 824 CENT AMERICA S	2013.	9725.	4026.	2857.	19226.	0.15000	283900.	1081192.		
F 825 CHINAN S	5209.	6407.	7912.	6710.	2658.	0.20100	523638.	1473531.		
P 705 CHILOE, TX	00.	6526.	4188.	00.						
F 826 US AMERICA S	00.	6526.	4188.	00.						
P 706 CHILOE C, TX	4933.	655.	00.							

F 823 W AMERICA	S	4548.	655.	00.	00.	5203.	0.25400	1321562.	294490.
P 705 BROWNSVI, TX	-	00.	00.	00.	00.				
P 706 CHARLEST, SC		00.	00.	00.	00.				
P 707 BALTIMORE, MD		22986.	37727.	117.	24252.				
F 806 USSR	S	12440.	37727.	00.	24254.	74401.	0.40800	30555608.	4211097.
F 809 N AFRICA	S	9714.	00.	00.	00.	9714.	0.34300	3331902.	549812.
F 810 RED SEA	S	252.	00.	117.	18.	367.	0.56500	207355.	20772.
P 708 TOLEDO, OH		00.	6926.	7655.	3789.				
F 809 N AFRICA	S	00.	6926.	7655.	3789.	16596.	0.66400	12214944.	1041214.
P 709 SINGINAW, ME		00.	00.	00.	00.				
P 710 CHICAGO, IL		00.	9655.	6517.	6380.				
F 801 SCANDINAVIA	S	00.	1731.	3823.	4170.	9726.	0.79600	7740304.	550378.
F 809 N AFRICA	S	00.	7952.	2694.	2210.	12856.	0.76900	9885264.	727650.
P 711 DULUTH, MN		00.	00.	00.	00.				
P 712 SEATTLE, WA		29601.	4560.	15290.	48367.				
F 813 SE ASIA	S	00.	44.	814.	14250.	15108.	0.35000	5267800.	855113.
F 818 JAPAN	S	29601.	4516.	14476.	35117.	85710.	0.21300	17830250.	4737986.
P 713 PORTLAND, OR		57475.	22121.	53486.	25301.				
F 817 KOREA	S	40258.	00.	49755.	16829.	106822.	0.40100	42885620.	6046125.
F 819 CHINA	S	17257.	22121.	3731.	8472.	51561.	0.45900	23666498.	2918353.
P 714 SAN FRANC, CA		00.	00.	00.	8.				
F 817 KOREA	S	00.	00.	00.	8.	8.	0.43000	3440.	453.
P 715 LONG BEA, CA		00.	00.	00.	00.				
P 716 SAN DIEG, CA		00.	00.	00.	00.				
P 717 TOLEDO, BC		00.	162.	4550.	6211.				
F 808 E MEDITERRAN	S	00.	162.	4550.	6211.	10923.	0.71300	7788099.	618242.
P 718 SINGINAW, BC		00.	2542.	00.	00.				

F 300 E MEDITERRANEI S.	00.	252.	00.	252.	0.7530	191426.	143877.
P 719 CHICAGO, IL -	00.	00.	00.	00.			
P 720 CHICAGO, IL	00.	459.	1043.	00.			
P 804 15 NOV 85	S	00.	302.	00.	0.4650	185340.	17433.
P 805 15 NOV 85	S	00.	455.	1012.	0.7650	112550.	851680.

	TOTAL	4. SUPPORT	3.	2.	1.	DENOM IN TIME
D 036 ALBEE	15					
D 213 ALBY	69					
D 061 ANDOLI	74					
D 211 ANTHONY	65					
D 707 BALTIC	10					
D 311 BATES	16					
D 038 BEAUMONT	16					
D 032 BENNETT	10					
D 077 BERNARD	10					
D 311 BURKE	16					
D 036 CANTRELL	10					
D 036 CAROL	10					
D 036 CECIL	10					
D 036 CECILIA	10					
D 211 CHALMERS	14					
D 211 CHARLES	10					
D 211 COLEMAN	35					
D 036 CORNELL	10					
D 211 CRISTY	14					
D 036 E. PEG	14					
D 031 ERICKSON	16					
D 031 ERIC	10					
D 101 FETTER	10					
D 201 FRANCIS	10					
D 036 GEORGE	10					
D 191 GIBBARD	10					
D 030 GILMAN	10					
D 192 HANFORD	10					
D 102 HEDDE	10					
D 171 HERCLES	17					
D 362 HOMER	10					
D 037 HOBBS	10					
D 101 HOWE	10					
D 101 JACKSON	10					
D 037 JAMES	10					
D 201 KANE	10					
D 149 KENNEDY	10					
D 101 KIRKLAND	10					
D 037 LARSEN	10					
D 232 LEHESKE	10					
D 104 LITTLE ROCK	10					
D 032 WEDDIX	15					
D 054 HOLISTER	10					
D 212 HOWE	10					
D 224 KIRK	10					
D 101 LITTLE ROCK	10					

D 085 CK CITY OR		3086.
D 713 PORTLAND OR		2332.
D 262 ROCKY MT. NC	16133.	16133.
D 411 ROSEBELL MN -	701.	701.
D 322 RUTLAND VT	10612.	10612.
D 042 SALINA KS	5466.	5466.
D 351 SALT LAK UT	567.	567.
D 311 SEATTLE WA	5573.	5573.
D 221 TALLAWASSEE FL	8644.	8644.
D 103 TEGALOGONIA AL	5953.	5953.
D 407 TELTON IA	5280.	5280.
D 066 TIBER ID	982.	982.
D 232 WATERTOWN NY	955.	955.
D 261 WILMINGTON NC	20273.	20273.
D 222 WINT HAM FL	7834.	7834.
D 135 CHIAPAS MEX	11357.	11357.
F 801 SOROCABA BRA	4017.	15731.
F 812 N.C. BAHREIN	26864.	4449.
F 813 S.L. BAHREIN	9016.	13325.
F 804 ISLAMABAD PAK	3728.	01.
F 815 AMAZONIC	1033.	1033.
F 825 LEBANON	12440.	140733.
F 817 E BLOCK BLDG	902.	683.
F 818 E NEW YORK	2685.	15072.
F 819 N AFRICA	974.	14573.
F 810 RED SEA	222.	00.
F 811 E AFRICA	01.	00.
F 812 W AFRICA	55.	405.
F 813 PERSIAN GLP	4789.	13160.
F 814 W AFRICA	210.	210.
F 815 SE ASIA	01.	44.
F 816 TAIWAN	26189.	28734.
F 817 KOREA	40238.	31499.
F 818 JAPAN	111428.	119357.
F 819 CHINA	17257.	22121.
F 820 CHINA	01.	00.
F 821 W MEXICO	803.	1168.
F 822 E MEXICO	17916.	24870.
F 823 W AMERICA	4548.	7179.
F 824 CENT AMERICA	15135.	9725.
F 825 CARIBBEAN	509.	6407.
		7889.
		2332.
		16133.
		701.
		10612.
		5466.
		567.
		5573.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		4449.
		13325.
		01.
		1033.
		140733.
		683.
		15072.
		14573.
		222.
		00.
		405.
		13160.
		210.
		44.
		28734.
		31499.
		119357.
		17257.
		22121.
		1168.
		10612.
		5466.
		567.
		8644.
		5953.
		20273.
		11357.
		44

GRAIN SHIPPED TO DEFICIT REGIONS 2025785.

GRAIN SHIPPED TO FOREIGN REGIONS 1597049.

Validation of Grain Corn Model

Port	FGIS Recorded Exports	Model Solution
Million Bu.		
Mobile	42.20	42.90
New Orleans	1124.70	1127.07
Galveston	10.40	10.70
Corpus Christi	5.76	5.20
Brownsville	0	0
Charleston	0	0
Baltimore	84.00	84.40
Toledo	21.93	18.40
Saginaw	0	0
Chicago	22.38	0
Duluth	1.30	0
Seattle	99.25	98.80
Portland	158.10	158.30
California	0	0
San Francisco	-	-
Long Beach	-	-
San Diego	-	-
Baie Comeau	27.42	28.30
Toledo	-	10.90
Saginaw	-	2.50
Chicago	-	-
Duluth	-	14.90
Total	1597.440	1597.049

Special adjustments made in the calibration process:

- 1) 95% railroad cost from Indianapolis to Baltimore
- 2) 80% railroad cost from Stevens Point to Chicago

— A Kodak Company —

MD2-D

Formatted for IBM PC's, XT's and
other 360 KB systems

disk no:

DataLife®

IWR - Report
90-B-7

Verdication

A Kodak Company

DataLife

