AD/

C^3I ANALYSIS TOOLS FOR DEVELOPMENT PLANNING VOLUME II

APPENDIX F: SOFTWARE LISTINGS

P. A. VAIL
G. H. WEISSMAN
J. G. WOHL

ALPHATECH Inc
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803

27 September 1985

Prepared For

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
DEPUTY FOR DEVELOPMENT PLANS
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731

LEGAL NOTICE

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

"This technical report has been reviewed and is approved for publication."

James D Taylor

JAMES D. TAYLOR, Lt Col,USAF
Project Manager

FOR THE COMMANDER

Charles LaBlonde

CHARLES J. LABLONDE, Colonel, USAF
Assistant Deputy for Development Plans

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) E-10221U	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-86-201 (II)
6a. NAME OF PERFORMING ORGANIZATION ALPHATECH, Inc 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Hq Electronic Systems Division (XR)
6c. ADDRESS (City, State and ZIP Code) 2 Burlington Executive Center 111 Middlesex Turnpike Burlington, MA 01803	7b. ADDRESS (City, State and ZIP Code) Hanscom Air Force Base, MA 01731
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-84-D0016
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) C3I Analysis Tools for Development Planning (Volume II)(Over)	
12. PERSONAL AUTHOR(S) P. A. Vail, G. H. Weissman, J. G. Wohl	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) 1985 September 27 15. PAGE COUNT 113
16. SUPPLEMENTARY NOTATION Refer to ESD-TR-86-201 (I) for Basic Report	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Vanguard NORAD Ballistic Missile Warning Subject Transfer Function (STF) (Over)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The first three programs are the Vanguard Analysts Support Tool (VAST) and two utility programs which support that program. The next two programs are the preprocessor which prepares inputs for the data reduction program (STF), and the data reduction program (STF) itself. The last three programs are utility programs which support the data collection effort.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL James D. Taylor, Lt Col, USAF	22b. TELEPHONE NUMBER (Include Area Code) (617)861-3116 22c. OFFICE SYMBOL ESD/XRX

DD FORM 1473, 83 APR EDITION OF 1 JAN ... OBSOLETE.

Cont of Block 11: Appendix F: Software Listings

Cont of Block 18:

C^3I Command, Control, Communications & Intelligence

CONTENTS

Appendix F

Software Listings

The programs in this appendix fall into three groups.

The first three programs are the Vanguard Analysts Support Tool (VAST) and two utility programs which support that program.

The next two programs are the preprocessor which prepares inputs for the data reduction program (STF), and the data reduction program (STF) itself.

The last three programs are utility programs which support the data collection effort.

Program Name: Vanguard Analyst Support Tool (VAST)

Language: BASIC

Machine: Apple MacIntosh

Purpose: Calculates mission effectiveness measure from capabilities entered into mission hierarchy. Datasets represent sets of capabilities from combinations of programs. User can create, modify or view datasets, perform sensitivity analysis on a dataset or compare two datasets.

```
'**********************************************

GOSUB HouseKeeping      'Put up Menu and First Screen

MainLineLogic:
WHILE 1 = 1
idle:
  menunumber = MENU(0)
  WHILE menunumber = 0
    menunumber = MENU(0)
    menuitem = MENU(1)
  WEND

  IF menunumber > 4 THEN idle   'return to while loop

 FOR m = 1 TO 2 :MENU m,3,0 :NEXT m              'disable menus
 MENU 3,0,1:FOR m =1 TO 3:MENU 3,m,0:NEXT m      'enable print screen

  ON menunumber GOSUB  helpscreens,datamenu,specialmenu,exitmenu

  FOR m = 1 TO 2 :MENU m,0,1:NEXT m          'enable
  FOR m = 1 TO 3 :MENU 3,m,1:NEXT m          'menus

  GOSUB FirstScreen

WEND
'**********************end of Main Line Logic **************************


helpscreens:
  MENU ON      'enables screen copy and exit from other modules
  GOSUB MainHelp
  MENU OFF      'returns menu control to polling routine in Main Line
RETURN

datamenu:
  MENU ON      'enables screen copy and exit from other modules
  ON menuitem GOSUB enternew,editold,showdatesets
  MENU OFF      'returns menu control to polling routine in Main Line
  RETURN

specialmenu:
  MENU ON      'enables screen copy and exit from other modules
  ON menuitem GOSUB sensitivity,compare,setminvalues,screencopy
  MENU OFF
```

```
  RETURN

exitmenu:
  GOTO endit

'eeeeeeThe following modules are the individual menu item seeeeeeeeeeeee

MainHelp:
  WINDOW 1,,(50,50)-(450,325),-2
  PRINT "The Data Function allows you to look at existing Data"
  PRINT "Sets, create new Data Sets or modify existing"
  PRINT "Data Sets."
  PRINT "Each Data Set contains capability ratings, node by node,"
  PRINT "that correspond to a combination of programs."
  PRINT "The Special Functions Menu allows you to do Sensitivity"
  PRINT "Analysis, to Compare Data Sets, and to examine the case"
  PRINT "in which all capabilities are set at a minimum."
  PRINT "You can also print any screen from the Special Functions Menu."
  BUTTON 1,1,"OK",(350,225)-(375,250)
  WHILE DIALOG(0) <>1 : WEND
  WINDOW CLOSE 1
  RETURN

'eeeeeeeeeeeeeeethis routine creates and names a new data record ****

enternew :
  NewNumber=NumberOfDataSets + 1  'save as new record
  NUM$ = STR$(NewNumber)
  NewName$ = "Data Record #" + NUM$ :NewDataSetRequested = 1
  GOSUB NewNameScreen       'allow choice of NewName$
NewDataReentry:
  GOSUB ChooseExistingDataScreen      'allow use of existing data
  ResponseFlag = 0      'Responseflag=1 for OK, 2 for CANCEL
  IF Buttonpushed = 1 THEN GOSUB SelectData :IF ResponseFlag = 2 THEN GOTO NewDataReentry
ELSE GOTO ContinueHere
  'IF ButtonPushed = 2 THEN WINDOW CLOSE 2:FOR i = 1 TO 88 :inval!(i)= 0 :outval!(i) = 0 :NEX
T i
  IF ButtonPushed = 3 THEN WINDOW CLOSE 3:GOSUB setminvalues
ContinueHere:
  NAM$ = NewName$
  WINDOW CLOSE 3
  CHANGEFLAG = 1                  'Changeflag = 1 if change to existing data
  WINDOW CLOSE 1                  'set or new data set requested
  ParentNode=XX : GOSUB FILLTREE
RETURN
```

```
NewNameScreen:
  WINDOW 1,,(30,30)-(290,130),4
  CALL MOVETO(10,25)
  PRINT "The record will be stored"
  PRINT "  with the following name."
  PRINT "  Edit this if you like."
  EDIT FIELD 3,New Name$,(30,80)-(140,95)
  BUTTON 3,1,"NAME IS OK",(150,80)-(260,95),1
  WHILE DIALOG(0)<>1 :WEND
  New Name$ = EDIT$(3) :NAM$ = New Name$
RETURN

ChooseExistingDataScreen:
  WINDOW 3,,(20,150)-(165,250),2
  CALL TEXTSIZE(12)
  PRINT "Do you want to use an"
  PRINT " existing data record"
  PRINT "  to start with?"
  FOR I=1 TO 88:INVAL!(I)=0:OUTVAL!(I)=0:NEXT I
  BUTTON 1,1,"Yes",(10,57)-(50,73),3
  BUTTON 2,1,"No",(80,57)-(120,73),3
  BUTTON 3,1,"Use Minimum Ranges",(0,82)-(145,98),1
  Activity = DIALOG(0)
  WHILE Activity <>1:Activity = DIALOG(0):WEND
  Buttonpushed=DIALOG(1)
RETURN

setminvalues:
  GOSUB CalculatingMessage
  FOR i=1 TO 88
    INVAL!(i)=RV!(i,1)
    NoCalcFlag%(i)=0
  NEXT i
  LL=1:UL=88
  GOSUB INTERP
  GOSUB CALCALL
  WINDOW CLOSE 1
RETURN

CalculatingMessage:
  WINDOW 1,,(95,100)-(410,200),2
  CALL TEXTSIZE(18):CLS:PRINT"VAST CALCULATION IS UNDERWAY"
  PRINT:PRINT"       Please Wait"
RETURN

'**********************end New DataEntry**********************************
editold:
```

```
  GOSUB SelectData
  IF ResponseFlag = 1 THEN GOSUB FillTree
RETURN

show datasets:
  W INDOW 2,"DATASETS",(305,40)-(495,320),4
  CLS
  BUTTON 6,1,"PAGE FORWARD",(10,220)-(190,240),2
  BUTTON 7,1,"PAGE BACK",(10,240)-(190,260),2
  BUTTON 8,1,"OK",(10,260)-(190,280),2
  CALL TEXTFACE(1)
  CALL MOVETO(5,10)
  PRINT "  Existing Data Sets":PRINT
  Page% = 1
Show NextPage:
  CALL MOVETO(1,40)
  First%=10*Page% - 9
  IF Page%=MaxPages% THEN Last%=NumberOfDataSets ELSE Last%=First%+9
  FOR K = First% TO Last%
  GET 2,K        'Get the next Data Set from the Data File
    ONAME$= DA$
    CALL TEXTSIZE(12)
    PRINT K;ONAME$
  NEXT K
  WHILE DIALOG(0)<>1:WEND
  ButtonPushed = DIALOG(1)
  IF ButtonPushed = 8 THEN RETURN
  IF ButtonPushed = 6 THEN IF Page%<MaxPages% THEN Page%=Page%+1 ELSE Page%=1
  IF ButtonPushed = 7 THEN IF Page%>1 THEN Page%=Page%-1 ELSE Page%=MaxPages%
  GOTO Show NextPage
RETURN

'*****************Following Routine Controls Data Set Selection****
SelectData:
  W INDOW 1,,(10,30)-(290,320),2
  CALL TEXTFACE(0):PRINT:PRINT "Enter the ";
  CALL TEXTFACE(1):PRINT "NUMBER ";
  CALL TEXTFACE(0):PRINT "of an"
  CALL TEXTFACE(1):PRINT "EXISTING ";
  CALL TEXTFACE(0):PRINT "Data Set"
  W INDOW 3,,(20,150)-(280,310),2
  PRINT "      DATA SET SELECTION ":Num$ = "2"  'default is BASELINE
  CALL MOVETO(5,33)
  PRINT "Number selected ="
  EDIT FIELD 1,NUM$,(130,20)-(170,35)
  BUTTON 1,0,"OK",(15,60)-(70,76),3
  BUTTON 2,0,"CANCEL",(100,60)-(200,76),3
```

```
  CALL MOVETO(5,120):   PRINT "Click on this window to make selection."
  OldWindow=3:CurrentWindow=2
  GOSUB GETID

LOOP:
  Activity = DIALOG(0)
  WHILE Activity()1 AND Activity()3:Activity=DIALOG(0):WEND
  IF Activity()3 THEN GOTO TestButtons '3means another window selected
  Temp=OldWindow:OldWindow=CurrentWindow:CurrentWindow=Temp
  OldButtons=6-5*(OldWindow-2)
  CurrentButtons=6-5*(CurrentWindow-2)
  FOR i=OldButtons TO OldButtons+1:BUTTON i,0:NEXT i
 WINDOW CurrentWindow
 FOR i=CurrentButtons TO CurrentButtons+1:BUTTON i,1:NEXT i
 GOTO LOOP
TestButtons:
  ButtonPushed = DIALOG(1)
  ON ButtonPushed GOTO OKBUTTON,CANCELBUTTON
  IF ButtonPushed=6 OR ButtonPushed=7 THEN GOSUB GETIDReentry
  GOTO LOOP

OKBUTTON:              'this code reads the proper dataset into the array -
                   'closes the Window and calls the first level of the tree
  NUM$=EDIT$(1)
  NumberEntered=VAL(NUM$)
  IF NumberEntered(1 OR NumberEntered)NumberOfDataSets THEN NumberEntered=NumberOfDataSe
ts:NUM$=STR$(NumberEntered):   EDIT FIELD 1,NUM$,(130,20)-(170,35): GOTO LOOP
  WINDOW CLOSE 3:WINDOW CLOSE 2:WINDOW CLOSE 1
  GOSUB LOADDATA
  ParentNode = XX
  ResponseFlag=1       'set flag for OK response
RETURN

LOADDATA:
  GET 2,VAL(NUM$)
  DATASET$ = DB$
  NAM$ = DA$
  FOR I = 0 TO 87
    NoCalcFlag%(I+1)=0
    INVAL!(I+1) = CVS(MID$(DATASET$,1+(I*8),4))
         'inval is the value of the node input by the user
    IF INVAL!(I+1)=-99 THEN NoCalcFlag%(I+1)=-1:INVAL!(I+1)=0
    IF INVAL!(I+1)<0 THEN NoCalcFlag%(I+1)=-1:INVAL!(I+1)=ABS(INVAL!(I+1))
    OUTVAL!(I+1) = CVS(MID$(DATASET$,5+(I*8),4))
         'outval is the value of the node after the calculation is complete
  NEXT I
RETURN
```

```
CANCELBUTTON:
  W INDOW CLOSE 3:W INDOW CLOSE 2:W INDOW CLOSE 1
  ResponseFlag=2
RETURN
'**********************end SelectData************************

'*****************GET ID lists the existing Data Sets*************
GET ID:
  W INDOW 2,"DATASETS",(305,40)-(495,328),4
  CLS
  BUTTON 6,1,"PAGE FORWARD",(10,240)-(190,260),2
  BUTTON 7,1,"PAGE BACK",(10,260)-(190,280),2
  CALL TEXTFACE(1)
  CALL MOVETO(5,10)
  PRINT "  Existing Data Sets":PRINT
  Page%=1
NextPage:
  CALL MOVETO(1,40)
  First%=10*Page% - 9
  IF Page%=MaxPages% THEN Last%=NumberOfDataSets ELSE Last%=First%+9
  FOR K = First% TO Last%
  GET 2,K        'Get the next Data Set from the Data File
    ONAME$= DA$
    CALL TEXTSIZE(12)
    PRINT K;ONAME$
  NEXT K
RETURN

GETIDReentry:
  IF ButtonPushed=6 THEN IF Page%<MaxPages% THEN Page%=Page%+1 ELSE Page%=1
  IF ButtonPushed=7 THEN IF Page%>1 THEN Page%=Page%-1 ELSE Page%=MaxPages%
  GOTO NextPage
'********************** end of GET ID routine ********************

'**********************end data selection routines***************

'*******************all tree routines*****************************
FILLTREE:

  'THIS MODULE CREATES THE NODAL PICTURE AND THE MOVEMENT AND DATA ENTRY BUTTONS

  W INDOW 1,,(5,5)-(500,350),2
  FOR NC=1 TO 5  :'FIND NUMBER OF CHILDREN FOR THIS NODE
    IF C(ParentNode,NC)<>0 THEN NumberOfChildren=
  NEXT NC
  C1=C(ParentNode,1)
```

```
  C2=C(ParentNode,2)
  C3=C(ParentNode,3)
  C4=C(ParentNode,4)
  C5=C(ParentNode,5)
FILLTREEReEntry:
  CALL TEXTFACE(1)
  CALL TEXTSIZE(9)
  CALL MOVETO(81,28):PRINT NN$(ParentNode)
  CALL MOVETO(75,42):PRINT LEFT$(NA$(ParentNode),14)
  CALL MOVETO(126,28): PRINT "Box";ParentNode;
  CALL MOVETO(100,60) :PRINT USING "###.##";INVAL!(ParentNode)
  CheckBox = ParentNode :GOSUB BoxRealCheck
  IF BoxRealSw = 0 THEN PICTURE(72,30),BOX1$ ELSE PICTURE(72,30),Box2$
  BUTTON 1,1,"UP",(75,65)-(105,80),1
  BUTTON 2,1,"IN",(134,65)-(164,80),1
  BUTTON 20,1,"?",(112,65)-(127,80),1

   BUTTON 13,1,"RECALCULATE",(325,27)-(425,43),1
   BUTTON 14,1,"SAVE",(325,50)-(370,66),1
   BUTTON 15,1,"RETURN",(325,73)-(380,89),1

  IF NumberOfChildren(5 THEN LastNormalChild%=NumberOfChildren ELSE LastNormalChild%=4
  FOR ChildBox = 1 TO LastNormalChild%
    XCorner=10+(ChildBox-1)*Spacing: YCorner = 117
    CALL MOVETO(XCorner+6,YCorner-2)
    PRINT NN$(C(ParentNode,Childbox))
    CALL MOVETO(XCorner+4,YCorner+12)
    PRINT LEFT$(NA$(C(ParentNode,Childbox)),15)
    CALL MOVETO(XCorner+53,YCorner-2)
    PRINT "Box";C(ParentNode,Childbox);
    CALL MOVETO(XCorner+29,YCorner+28)
    PRINT USING "###.##";INVAL!(C(ParentNode,Childbox))
    CheckBox = C(ParentNode,Childbox) :GOSUB BoxRealCheck
    IF BoxRealSw = 0 THEN PICTURE(XCorner,YCorner),BOX1$ ELSE PICTURE(XCorner,YCorner),Bo
x2$
    CALL MOVETO(XCorner+BOXWIDE/2,YCorner-1)
    CALL PENSIZE(2,2)
    CALL LINE(0,-16)
    PENNORMAL
    IF FF$(C(ParentNode,Childbox))(0)="P" AND FF$(C(ParentNode,Childbox))(0)="D" THEN BUTTON (1
+ChildBox)*2-1,1,"DN",(XCorner+3,YCorner+35)-(XCorner+33,YCorner+50)
    BUTTON (1+ChildBox)*2,1,"IN",(XCorner+62,YCorner+35)-(XCorner+92,YCorner+50)
 NEXT ChildBox

 ButtonNum = 22
 FOR ChildBox = 1 TO LastNormalChild%
   XCorner = 10+(ChildBox-1)*Spacing
```

```
    BUTTON ButtonNum ,1,"?",(XCorner+40,YCorner+35)-(XCorner+55,YCorner+50)
    ButtonNum = ButtonNum + 1
NEXT ChildBox

IF ParentNode () 9 GOTO NoScenarioBox
   XCorner=192: YCorner = 44:ChildBox=5
   CALL MOVETO(XCorner+6,YCorner-2)
    PRINT NN$(C(ParentNode,Childbox))
   CALL MOVETO(XCorner+4,YCorner+12)
    PRINT LEFT$(NA$(C(ParentNode,Childbox)),14)
   CALL MOVETO(XCorner+53,YCorner-2)
   PRINT "Box";C(ParentNode,Childbox);
   CALL MOVETO(XCorner+29,YCorner+28)
   PRINT INVAL!(C(ParentNode,Childbox))
   PICTURE(XCorner,YCorner),BOX2$
   CALL MOVETO(119,90)
   CALL PENSIZE(2,2)
   CALL LINE(XCorner-121,0)
   PENNORMAL
   BUTTON 18,1,"LO",(XCorner+3,YCorner+35)-(XCorner+33,YCorner+50)
   BUTTON 19,1,"HI",(XCorner+62,YCorner+35)-(XCorner+92,YCorner+50)
   BUTTON 21,1,"?",(XCorner+40,YCorner+35)-(XCorner+55,YCorner+50)

NoScenarioBox:        'The following draws the lines connecting the boxes
   CALL PENSIZE(2,2)
   CALL MOVETO(10+BOXWIDE/2,100)
   CALL LINE(SPACING*(LastNormalChild%-1),0)
   CALL MOVETO(119,100):
   CALL LINE(0,-17)

   PENNORMAL          'The following sets up current values in the lower left corner.
   CALL MOVETO(5,230):CALL TEXTSIZE(10)
   PRINT "Record Selected is "+NUM$
   PRINT " "+NAM$
   CALL MOVETO(5,270)
   PRINT "SELECTED NODE is "+NN$(SelectedNode)
   PRINT " Value = ";:PRINT USING "###.##";INVAL!(SelectedNode)
   BUTTON 16,1,"Select node",(7,300)-(130,312),1

        'The following waits for a button and then branches accordingly
READBUTTONS:
   WHILE DIALOG(0) () 1:WEND      'loop until a button is pushed
    BUTTONPUSHED = DIALOG(1)
    IF BUTTONPUSHED = 15 THEN W INDOW CLOSE 3:GOSUB Checkit:W INDOW CLOSE 1:RETURN
    IF BUTTONPUSHED = 14 THEN  W INDOW CLOSE 3: GOSUB savit:GOTO FILLTREE
    IF BUTTONPUSHED = 13 THEN GOSUB CalculatingMessage:GOSUB CALCALL:GOTO FILLTREE
    IF BUTTONPUSHED = 16 THEN SelectedNode=ParentNode:CALL MOVETO(5,270):CALL TEXTSIZE
```

```
(10):PRINT "SELECTED NODE is "+NN$(SelectedNode):PRINT" Value = ";:PRINT USING "###.##";IN
VAL!(SelectedNode):GOTO READBUTTONS
   IF BUTTONPUSHED = 18 THEN INVAL!(7) = 1:CALL MOVETO(XCorner+29,YCorner+28):PRINT 1
   IF BUTTONPUSHED = 19 THEN INVAL!(7) = 2:CALL MOVETO(XCorner+29,YCorner+28):PRINT 2
   IF BUTTONPUSHED = 18 OR  BUTTONPUSHED = 19 THEN GOTO READBUTTONS
   IF ButtonPushed ) 19 THEN GOSUB Boxscreen :GOTO ReadButtons
   IF BUTTONPUSHED MOD 2 = 1 THEN GOTO UPDOWN      'odd button pushed
        'even button pushed m eans enter data value
ENTERNODE:
  ChildSelected% = INT((BUTTONPUSHED-1)/2)   'find w hich box w as picked
  IF ChildSelected% = 0 THEN EE = ParentNode  ELSE  EE = C(ParentNode,ChildSelected%)
  IF FF$(C(ParentNode,ChildSelected%))="0" THEN NE = S(C(ParentNode,ChildSelected%),1):GOTO NO
ENTRY
  GOTO MYPOINTER

UPDOWN:
  IF BUTTONPUSHED = 1 AND ParentNode ( BB THEN ParentNode = P(ParentNode,1):GOTO FILLTREE
  IF BUTTONPUSHED = 1 THEN MOUSE OFF :GOSUB Checkit :W INDOW  CLOSE 1:RETURN
  IF BUTTONPUSHED = 3 THEN ParentNode = C1:GOTO FILLTREE
  IF BUTTONPUSHED = 5 THEN ParentNode = C2:GOTO FILLTREE
  IF BUTTONPUSHED = 7 THEN ParentNode = C3:GOTO FILLTREE
  IF BUTTONPUSHED = 9 THEN ParentNode = C4:GOTO FILLTREE

MYPOINTER:     'open the data entry w indow , Initialize w indow  values and text
  W INDOW  3,,(155,200)-(465,320),2
   VLOW != RV!(EE,1)
   FOR TEMP = 6 TO 2 STEP -1   'FIND HIGHEST RANGE VALUE
     IF RV!(EE,TEMP) ) 0 THEN GOTO HIGHEST
   NEXT TEMP
HIGHEST:
   VHIGH!= RV!(EE,TEMP)
   VCURRENT!= INVAL!(EE)
   IF VHIGH!) VLOW !THEN VMAX!=VHIGH!:VMIN!=VLOW !
   IF VHIGH!( VLOW !THEN VMAX!=VLOW !:VMIN!=VHIGH!
   IF VCURRENT! ) VMAX!THEN VCURRENT! = VMAX!
   IF VCURRENT! ( VMIN!THEN VCURRENT! = VMIN!
  BUTTON 15,1,"OK",(10,103)-(50,119),3
  BUTTON 16,1,"CANCEL",(70,103)-(150,119),3
  IF NoCalcFlag%(EE) = -1 THEN BUTTON 17,1,"Release Value",(180,100)-(300,116),1
  CALL MOVETO(3,18)
  CALL TEXTSIZE(18)
  PRINT "NODE NUMBER =";NN$(EE)
  CALL MOVETO(3,40)
  PRINT "VALUE IS";:PRINT USING "###.##"; INVAL!(EE)
  IF NoCalcFlag%(EE) = -1 THEN CALL MOVETO (170,40):CALL TEXTSIZE(12):PRINT "Value is held
 constant";
```

```
 CALL MOVETO(40,75)      'DRAW RANGE LINE
 CALL PENSIZE(1,3)
 CALL LINE(0,5)
 CALL LINE(102,0)
 CALL LINE(0,-5)

 CALL TEXTSIZE(10)      'WRITE END VALUES OF RANGE
 CALL MOVETO(30,74)
  PRINT VLOW!;
  CALL MOVETO(132,74)
  PRINT VHIGH!;

 IF VHIGH!-VLOW! = 0 THEN XPOSITION =0 ELSE XPOSITION=100*(VCURRENT!-VLOW!)/(VHIGH!-VLOW
!)
 CALL PENSIZE(3,3)                          'draw pen
 CALL MOVETO(40+XPOSITION,84)
 CALL LINE(0,15)
 CALL MOVETO(40+XPOSITION,84)
 CALL LINE(5,7)
 CALL MOVETO(40+XPOSITION,84)
 CALL LINE(-5,7)
 GET(35+XPOSITION,84)-(47+XPOSITION,101),PP



IDLE.POINTER:      'RESPOND TO MOUSE ROUTINE
   WHILE MOUSE(0)=)0
      0=DIALOG(0): ButtonPushed = DIALOG(1)
      IF ButtonPushed = 15 THEN GOSUB ChangeValue:WINDOW CLOSE 3:GOTO FILLTREEReEntry
      IF ButtonPushed = 16 THEN WINDOW CLOSE 3:GOTO FILLTREEReEntry
      IF ButtonPushed = 17 THEN NoCalcFlag%(EE)=0:CHANGEFLAG=1:WINDOW CLOSE 3:GOTO F
ILLTREEReEntry
   WEND
    IF (MOUSE(3)(XPOSITION+35) OR (MOUSE(3))XPOSITION+47) THEN GOTO IDLE.POINTER
    IF (MOUSE(4)(83) OR (MOUSE(4))102) THEN GOTO IDLE.POINTER
    IF MOUSE(5))140 OR MOUSE(5)(40 GOTO IDLE.POINTER
    OLDMOUSE=MOUSE(3)
    IF ABS(MOUSE(5)-OLDMOUSE)(1 THEN GOTO IDLE.POINTER

Move.Mouse:
     IF MOUSE(0)=)0 GOTO IDLE.POINTER      'is button still down?
     IF ABS(MOUSE(5)-OLDMOUSE)(1  GOTO Move.Mouse   'has it moved again?
     PUT(35+XPOSITION,84)-(47+XPOSITION,101),PP     'erase old pointer
     XPOSITION=XPOSITION+MOUSE(5)-OLDMOUSE          'get new position
     IF XPOSITION(0 THEN XPOSITION=0
     IF XPOSITION)100 THEN XPOSITION=100
     PUT(35+XPOSITION,84)-(47+XPOSITION,101),PP       'draw new pointer
```

```
    VCURRENT!=VLOW!+XPOSITION*(VHIGH!-VLOW!)/100        'recalculate value
    CALL MOVETO(3,40)
    CALL TEXTSIZE(18)
    PRINT "VALUE IS";:PRINT USING "###.##";VCURRENT!
    OLDMOUSE=MOUSE(5)
    GOTO Move.Mouse

ChangeValue:
  INVAL!(EE)=VCURRENT!:CHANGEFLAG=1
  IF FF$(EE)()"P" THEN NoCalcFlag%(EE)=-1      'changing value of a non-primitive sets the No Calc
ulation Flag
  RETURN


NOENTRY: 'THIS MODULE CHECKS FOR THE NODE SELECTED FOR INPUT BEING
         'A DUPLICATE,AND ASKS IF THE USER WANTS TO CANCEL ENTRY
         'OR JUMP TO THE ORIGINAL NODE FOR ENTRY

  WINDOW 3,,(155,200)-(465,320),2
  PRINT:PRINT "THIS NODE IS A DUPLICATE OF NODE ";NN$(NE);" !!":PRINT "SELECT THE BUTTON OF
YOUR CHOICE"
  BUTTON 13,1,"JUMP TO ORIGINAL",(10,50)-(200,66),3
  BUTTON 14,1,"CANCEL",(10,80)-(75,96),3

  WHILE DIALOG(0)()1:WEND
  THISBUTTON=DIALOG(1)
  IF THISBUTTON=14 THEN WINDOW CLOSE 3:GOTO READBUTTONS
  IF BUTTONPUSHED=2 THEN WINDOW CLOSE 3:GOTO FILLTREE
  ParentNode=NE:WINDOW CLOSE 3:GOTO FILLTREE

'**********************end all tree routines*****************************

BoxRealCheck:
  BoxRealSw=0
  FOR i=1 TO 37
    IF CheckBox=RealBox(i) THEN BoxRealSw=1:RETURN
  NEXT i
RETURN

CheckIt:
  IF CHANGEFLAG=0 THEN RETURN
  WINDOW 3,,(175,200)-(450,330),2
   CALL MOVETO(8,10):CALL TEXTSIZE(10)
   CALL TEXTFACE(1):PRINT "This dataset has been changed,"
   PRINT "Or a New Data Set was Requested.":CALL TEXTFACE(0)
   BUTTON 15,1,"Save Changes",(10,35)-(250,50),2
   BUTTON 16,1,"Abandon Changes",(10,60)-(250,75),2
```

```
    Activity=DIALOG(0)
    WHILE Activity <>1:Activity=DIALOG(0):WEND
    Buttonpushed = DIALOG(1):WINDOW CLOSE 3
    IF Buttonpushed = 15 THEN GOSUB SAVIT
    RETURN

'**************save data set modules*****************************

SAVIT:              'MAKE DATAFILE
  IF NewDataSetRequested = 1 THEN NUM$ = STR$(NewNumber):GOSUB NewNameScreen :GOTO sa
vit2
  WINDOW 3,,(175,200)-(450,330),2
   CALL MOVETO(8,10):CALL TEXTSIZE(12)
   CALL TEXTFACE(1):PRINT"Please choose a save option"
   CALL TEXTFACE(0)
   BUTTON 15,1,"Save with current record number and name",(10,35)-(250,50),2
    BUTTON 17,1,"Change the record name",(10,85)-(250,100),2
   BUTTON 16,1,"Create a new data record",(10,60)-(250,75),2
    Activity=DIALOG(0)
    WHILE Activity <>1:Activity=DIALOG(0):WEND
    Buttonpushed = DIALOG(1)
    IF Buttonpushed = 15 THEN GOTO Savit2
    IF ButtonPushed = 16 THEN NewNumber = NumberOfDataSets + 1 :  Num$ = STR$(NewNumber) :
GOSUB NewNameScreen :GOTO savit2
    IF ButtonPushed = 17 THEN NewNames = "Data Record #" + Num$
    WINDOW CLOSE 3
    GOSUB NewNameScreen :Name$ = NewName$
Savit2:
    WINDOW 3,,(175,200)-(450,330),2
   CALL MOVETO(8,10):CALL TEXTSIZE(12)
   CALL TEXTFACE(1)
    PRINT"SAVING DATA RECORD":PRINT"  Please Wait"
    GOSUB WriteFile :WINDOW CLOSE 3
RETURN

WriteFile:
  D$ = ""
  ID$ = NAM$
  FOR I = 1 TO 88
    IF NoCalcFlag%(I)=-1 THEN INVAL!(I)= -INVAL!(I)
    IF INVAL!(I)=0 THEN IF NoCalcFlag%(I)=-1 THEN INVAL!(I)= -99
    A$ = MKS$(INVAL!(I)):B$ = MKS$(OUTVAL!(I))
    C$ = A$ + B$
    D$ = D$ + C$
  NEXT I
  LSET DA$ = ID$:LSET DES = D$
  PUT 2,VAL(NUM$)
```

```
  NumberOfDataSets = LOF(2)/730
 CHANGEFLAG = 0 :NewDataSetRequested = 0
RETURN

'****************end save data modules*****************************
BoeScreen:
   IF ButtonPushed = 20 THEN ClickBoe = ParentNode
   IF ButtonPushed = 21 THEN ClickBox = 7
   IF ButtonPushed >21 THEN ClickBoe = C(ParentNode, ButtonPushed - 21)
   GET #4,Clickboe
  TextString$ = TotA$
  Screen2$ = MID$(TextString$,451,450)
  linecount = 0
  WINDOW 4,,(155,200)-(465,320),2
  FOR k = 1 TO 2
     IF k = 2 AND Screen2$ = BlankString$ THEN GOTO NeeK
     CALL MOVETO (10,10)
     FOR i = 1 TO 6
        test$ = MID$(TextString$,linecount*75 + 1,75)
        PRINT test$
        linecount = linecount + 1
     NEXT i
     BUTTON 1,1,"OK",(10,103)-(50,119)
     WHILE DIALOG(0) <> 1 :WEND
  NeeK: NEXT k
  WINDOW CLOSE 4
RETURN

**************special functions*************************************

sensitivity:
   IF sensitivityfirsttime <> 0 THEN GOTO beginSensitivity
   sensitivityfirsttime = 1
   WINDOW 1,,(25,25)-(500,325),-2
   CALL MOVETO (50,50)
   PRINT "Sensitivity Analysis is concerned with the effect that improvements"
   PRINT "in primitive nodes at the bottom of the tree have on higher level nodes,"
   PRINT "specifically, on the following:"
   PRINT "   Node 1111 :MWC Data (% Complete)"
   PRINT "   Node 1121 :CINC Assessment (Completeness of Information to support)"
   PRINT "   Node 2321 :Attack Characterization (% Complete)"
   PRINT "   Node 1 :Force Warning "
   PRINT "   Node 2 :EAM"
   PRINT "   Node 0 :SAC Mission Value"
   BUTTON 1,1,"ok",(275,150)-(300,175)
   WHILE DIALOG (0) <> 1  : WEND
   CALL MOVETO (50,50) :CLS
```

```
  PRINT "For the data set you will select, this function prints the current value"
  PRINT "of each of these nodes.
  PRINT
  PRINT "Then, for each primitive node that is involved in the calculation, it"
  PRINT "calculates the improvement in each of the above nodes that the best"
  PRINT "possible value of the primitive node would produce, with all other nodes"
  PRINT "unchanged.  It prints the non-zero improvements."
  PRINT
  PRINT "This information will not be displayed again, but is printed."
  BUTTON 1,1,"ok",(275,150)-(300,175)
  WHILE DIALOG(0)<>1 : WEND
  WINDOW CLOSE 1

beginSensitivity:
  WINDOW 1,,(100,180)-(400,300),-2
  CALL TEXTSIZE(14)
  CALL MOVETO(1,20)
  PRINT "You must pick a  Data Set"
  PRINT " for sensitivity analysis."
  CALL TEXTFACE(1)
  BUTTON 1,1,"OK",(170,110)-(190,125)
  CALL TEXTFACE(0)
  BUTTON 2,1,"Cancel",(120,150)-(220,170)
  Activity=0
  WHILE Activity<>1:Activity=DIALOG(0):WEND
  ButtonPushed = DIALOG(1)
  IF ButtonPushed = 1 THEN GOSUB SelectData ELSE RETURN
  IF ResponseFlag = 2 THEN RETURN
  WINDOW CLOSE 1:WINDOW CLOSE 2:WINDOW CLOSE 3

  WINDOW 1,,(50,100)-(450,300),2:CALL TEXTFACE(9)
  CALL TEXTSIZE(14):CLS:PRINT:PRINT "  SENSITIVITY ANALYSIS CALCULATION"
  PRINT                              "          IS UNDERWAY"
  PRINT:PRINT "         Please Wait":CALL TEXTSIZE(12):CALL TEXTFACE(0)

  B9!=INVAL!(9):B23!=INVAL!(23):B32!=INVAL!(32)
  B59!=INVAL!(59):B86!=INVAL!(86):B8B!=INVAL!(8B)  'hold "baseline" values

  LPRINT CHR$(12):LPRINT:LPRINT "SENSITIVITY ANALYSIS":LPRINT
  LPRINT "Sensitivity Analysis is concerned with the effect that improvements"
  LPRINT "in primitive nodes at the bottom of the tree have on higher level nodes,"
  LPRINT "specifically, on the following:"
  LPRINT "   Node 1111 :MWC Data (% Complete)"
  LPRINT "   Node 1121 :CINCNORAD Assessment (Completeness of Information to support)"
  LPRINT "   Node 2321 :NORAD Attack Characterization (% Complete)"
  LPRINT "   Node 1 :SAC Force Warning"
  LPRINT "   Node 2 :SAC EAM"
```

```
  LPRINT "   Node 0 :SAC Mission Value"
  LPRINT
  LPRINT "For the data set you will select, this function prints the current value"
  LPRINT "of each of these nodes."
  LPRINT
  LPRINT "Then, for each primitive node that is involved in the calculation, it"
  LPRINT "calculates the improvement in each of the above nodes that the best"
  LPRINT "possible value of the primitive node would produce, with all other nodes"
  LPRINT "unchanged.  It prints the non-zero improvements."
  LPRINT
  FOR i = 26 TO 1 STEP -1
    IF MID$(nam$,i,1) <> " " THEN GOTO shiftname1
  NEXT i
shiftname1:nam x$ = LEFT$(nam$,i)
  LPRINT:LPRINT "The Baseline Values for Data Set ";NAMX$," are:"
  LPRINT "   MWC (node 1111) =";INVAL!(9)
  LPRINT "   CINC Assess.(node 1121) =";INVAL!(23)
  LPRINT "   Attack Characterization (node 2321) =";INVAL!(32)
  LPRINT "   Force Warning % (node 1) =";INVAL!(59)
  LPRINT "   EAM (node 2) =";INVAL!(86)
  LPRINT "   Mission Value (node 0) =";INVAL!(88)
   LPRINT
   LPRINT "The following are the improvements that can be expected in the above"
   LPRINT "if the indicated primitive node is improved to its best possible value."
   FOR sidx = 1 TO LOF(3)/6       'read to end of indexfile
       GET 3,sidx
       Index=ABS(VAL(IX$))
       IF FF$(Index)<>"P" THEN NextIndex
       Temp!= INVAL!(Index)     'hold "baseline" value
       j=6
       WHILE RV!(Index,j) = 0 :j=j-1 :WEND      'find "best" range value
       INVAL!(Index) = RV!(Index,j):GOSUB CALCALL   'set value and calculate
       FOR i = 38 TO 1 STEP -1
         IF MID$(na$(index),i,1) <> " " THEN GOTO shiftname2
       NEXT i
shiftname2:nam x$ = LEFT$(na$(index),i)

       LPRINT:LPRINT "Improving the value of node ";NN$(Index);": "nam x$;" should produce the fol
low ing:"
       IF B9!<>Inval!(9) THEN LPRINT "   MWC (Node 1111) increased by: "; inval!(9) - B9!
       IF B23!<>Inval!(23) THEN LPRINT "   CINC Assess.(node 1121) increased by: "; inval!(23) -
 B23!
       IF B32!<>Inval!(32) THEN LPRINT "   Attack Characterization (node 2321) increased by: ";
inval!(32) - B32!
       IF B59!<>Inval!(59) THEN LPRINT "   Force Warning (node 1) increased by: "; inval!(59) - B
59!
       IF B86!<>Inval!(86) THEN LPRINT "   EAM (node 2) increased by: "; inval!(86) - B86!
```

```
      IF 888!() inval!(88) THEN LPRINT "   Mission Value (node 0) increased by: "; inval!(88) - 8
88!
      IF 89!= INVAL!(9) AND 823!= INVAL!(23) AND 832!= INVAL!(32) AND 859!= INVAL!(59) AND
886!= inval!(86) AND 888!= INVAL!(88) THEN LPRINT STRING$(25," "),"NO DIFFERENCE"
   LPRINT
      INVAL!(index)=Temp!      'restore "baseline" value
   NextIndex:NEXT sidx
   LPRINT CHR$(12):WINDOW CLOSE 1
   RETURN

compare:
   WINDOW CLOSE 1:WINDOW CLOSE 2:WINDOW CLOSE 3
   FOR icomp=1 TO 2
      WINDOW 1,,(100,100)-(400,300),-2
      CALL TEXTSIZE(14)
      CALL MOVETO(1,20)
      PRINT "You must pick two Data Sets"
      PRINT " for node by node comparison."
      CALL MOVETO(5,100)
      PRINT "Are you ready to pick Data Set #"+STR$(icomp)+" ?"
      CALL TEXTFACE(1)
      BUTTON 1,1,"OK",(170,110)-(190,125)
      CALL TEXTFACE(0)
      BUTTON 2,1,"Cancel",(120,150)-(220,170)
      Activity=0
      WHILE Activity()1:Activity=DIALOG(0):WEND
      ButtonPushed = DIALOG(1)
      IF ButtonPushed = 1 THEN GOSUB SelectData ELSE RETURN
      IF ResponseFlag = 2 THEN RETURN
      IF icomp = 1 THEN nam1$= nam$ ELSE nam2$ = nam$
      IF icomp = 1 THEN FOR J=1 TO 88:INVAL1!(J)=INVAL!(J):NEXT J
   NEXT icomp
      WINDOW CLOSE 1:WINDOW CLOSE 2:WINDOW CLOSE 3
      WINDOW 1,,(95,100)-(410,200),2
      CALL TEXTSIZE(14):CLS:PRINT "DATA SET COMPARISON IS UNDERWAY"
      PRINT:PRINT "       Please Wait":LPRINT CHR$(12)
      LPRINT "DATA SET COMPARISON":LPRINT
      pline$= STRING$(79," ")
      MID$(pline$,1) = "NODE":MID$(pline$,9) = "Node Name"
      FOR i= 26 TO 1 STEP -1
         IF MID$(nam1$,i,1) () " " THEN GOTO shiftname
      NEXT i
shiftname:namx$ = LEFT$(nam1$,i)
      MID$(pline$,45 - i) = namx$
      MID$(pline$,54) = nam2$
      LPRINT pline$
      FOR I = 1 TO LOF(3)/6      'read to end of indexfile
```

```
        GET 3,I
        LPRINT
        plines = STRING$(38,"")
        Index=ABS(VAL(IX$))
        IF INVALI!(Index) > INVAL!(Index) THEN IN$=GT$ ELSE IF INVALI!(Index) = INVAL!(Index) THE
N IN$="  =  " ELSE IN$=LT$
        IF rv!(index,1) > rv!(index,2) AND in$ = GT$ THEN in$ =LT$ ELSE IF rv!(index,1) > rv!(index,2)
 AND in$ = LT$ THEN in$ =GT$
        MID$(plines,1) = NN$(index) :MID$(plines,9) = na$(index) :LPRINT plines;
        LPRINT USING "###.##";INVALI!(Index);
        LPRINT IN$;
        LPRINT USING "###.##";INVAL!(Index);
        IF rv!(index,1) > rv!(index,2) THEN LPRINT " (Less is better)" ELSE LPRINT ""
      NEXT I
    LPRINT CHR$(12) :WINDOW CLOSE 1
RETURN

screencopy:
  LCOPY
RETURN

endit:
  MENU RESET
  CLOSE
  SYSTEM
  END

'***************END MENU STUFF*******************


'***************node calculation routines**********

CALCALL:

'******set all scenario/time nodes********
IF inval!(7) = 0 THEN inval!(7) = 1
INVAL!(18) = INVAL!(7) :LL=18 :UL=18 :GOSUB INTERP
INVAL!(29) = INVAL!(7) :LL=29 :UL=29 :GOSUB INTERP
'*********************************

'In order to calculate box 9, need to load appropriate scale values for boxes
'4,5,6 and 8.

  IF inval!(7) = 1 THEN i = 1 ELSE i = 2
  FOR j = 1 TO 4
    sc!(4,j) = S4!(i,j):sc!(5,j) = S5!(i,j):sc!(6,j) = S6!(i,j):sc!(8,j) = S8!(i,j)
  NEXT j
```

```
CALCN1:
  LET LL = 1:LET UL = 6 :GOSUB INTERP

  'aaaaa CALCULATE Box 8 RAW SCORE aaaaa
  IF NoCalcFlag%(8) () -1 THEN INVAL!(8) = .811*OUTVAL!(1)*OUTVAL!(3)
  LET LL = 8 :LET UL = 8 :GOSUB INTERP

  'aaaaa CALCULATE BOX 9  (mode 1111, % data) aaaaa
  ' Recall that box 9 has two different models, depending on whether box
  ' 7 (time) is 1 (10 mins) or 2 (20 mins). Note that in the latter case
  ' the model includes a range term.

  IF NoCalcFlag%(9) = -1 THEN GOTO Next9

  IF ineal!(7) = 1 THEN ineal!(9) = .0036*outval!(8)*outeal!(4)+.0068*outeal!(4)*outeal!(5)+.0018*o
utval!(4)*outval!(6) :GOTO next9

  smax!= .00001 :smin!= 100000!
  FOR i = 4 TO 6
    IF outval!(i)) smax!THEN smax!= outeal!(i)
    IF outeal!(i)( smin!THEN smin!= outeal!(i)
  NEXT i
  IF outeal!(8) ) smax!THEN smax!= outeal!(8)
  IF outval!(8) ( smin!THEN smin!= outval!(8)
  ineal!(9) = .0098*Outval!(8)*outeal!(4) + .86*outval!(5) + .21*oeteal!(6) - .38*(smax!- smin!)

  Next9:
  IF Ineal!(9) ) 95!THEN ineal!(9) = 95!
  LL = 9 :UL= 9: GOSUB INTERP
  INVAL!(19) = INVAL!(9) :LL=19 :UL=19 :GOSUB INTERP

  'aaaaaaaEND BOX 9 CALCULATIONS aaaaaa
CALCN2:
  LET LL = 10 :LET UL = 14 :GOSUB INTERP
  IF NoCalcFlag%(20) () -1 THEN INVAL!(20) = 1.83 * OUTVAL!(10) * OUTVAL!(11) + .98 *ABS (OUTV
AL!(10) - OUTVAL!(11))
  IF ineal!(20) ) 90!THEN inval!(20) = 90!
  IF NoCalcFlag%(21) () -1 THEN INVAL!(21) = .28 * OUTVAL!(12) + .134 * OUTVAL!(13) * OUTVAL!(1
4)
  IF ineal!(21) ( 1!THEN inval!(21) = 1!
  IF INVAL!(13) = 0!THEN IF NoCalcFlag%(21) () -1 THEN INVAL!(21) = 1
  IF inval!(13) () 0!THEN ineal!(21) = (inval!(21) + 3)/3    'convert 0-9 scale to 1-4 scale
  IF NoCalcFlag%(22)= -1 THEN GOTO KEEPON
                        'Assume MWC to NCP Voice is there
   IF INVAL!(16) ( 1.5THEN IF INVAL!(19) ( 20 THEN INVAL!(22) = 5 + (1.5*INVAL!(19)) ELSE INVAL
!(22) = 35
```

```
   IF INVAL!(16) =) 1.5 THEN INVAL!(22) = 20 + (.5 * INVAL!(19))
   IF INVAL!(16) =) 2.5 THEN INVAL!(22) = 22 + (.6 * INVA_ .9))

KEEPON:
  LET LL = 17 :LET UL = 17 :GOSUB INTERP
  LET LL = 20 :LET UL = 22 :GOSUB INTERP
  IF NoCalcFlag%(23) () -1 THEN INVAL!(23) = .4*OUTVAL!(20) + .0032*OUTVAL!(21) + .57*OUTVAL!
(17) + .21*OUTVAL!(22) + .21*outval!(18) - 42!

CALCN3:
  INVAL!(26) = INVAL!(20) :INVAL!(27) = INVAL!(21) :INVAL!(28) = INVAL!(17) :INVAL!(31) = INVAL!(2
3)
  LL = 26 :UL = 31 :GOSUB INTERP
   'aaaaa CALCULATE Box 32 RAW SCORE aaaaa
  IF NoCalcFlag%(32) () -1 THEN INVAL!(32) = .36*OUTVAL!(26) + 1.07*OUTVAL!(27) + .11*OUTVAL!(
28) + .27*OUTVAL!(31)
   IF inval!(32) ( 1!THEN inval!(32) = 1!
   IF inval!(32) ) 100!THEN inval!(32) = 100!

CALCMAIN:

CALC111:
   LET LL = 35 :LET UL = 35 :GOSUB INTERP
  'aaaaa CALCULATE Box 36 *****
  IF NoCalcFlag%(36) () -1 THEN INVAL!(36) = (.021)*OUTVAL!(9)*OUTVAL!(35)
CALC112:
   LL = 23 :UL = 23 :GOSUB INTERP
   inval!(39) = inval!(35) :LET LL = 39 :LET UL = 39 :GOSUB INTERP
  '***** CALCULATE Box 40 *****
 IF NoCalcFlag%(40) () -1 THEN  INVAL!(40) = .021*OUTVAL!(23)*OUTVAL!(39)   'assume 112 same
form as 111
  LET LL = 36:LET UL = 36 :GOSUB INTERP
  LET LL = 40:LET UL = 40 :GOSUB INTERP
   '***** CALCULATE BOX 51  (node 11, delay)
  IF NoCalcFlag%(51) = -1 GOTO N51NoCalc     'DONE
    IF OUTVAL!(36) ) OUTVAL!(40) THEN OV!= OUTVAL!(36) ELSE OV!= OUTVAL!(40)
     IF OV!( .4 THEN INVAL!(51) = 10!
     IF OV!=) .4 THEN INVAL!(51) = 14.4 -11*OV!
     IF OV!) .7 THEN INVAL!(51) = 7.4 - OV!
     LET LL = 51 :LET UL = 51 :GOSUB Interp
  N51NoCalc:
  '***** CALCULATE BOX 59  (node 1,% warned in time)
 IF NoCalcFlag%(59) = -1 GOTO N59NoCalc
     Comm Delay = 1 : WarnDelay=OUTVAL!(51) + Comm Delay
     IF WarnDelay (= 7 THEN INVAL!(59) = 99
     IF WarnDelay ) 7  THEN INVAL!(59) = 99 - 49*(WarnDelay-7)/30
     IF WarnDelay ) 37 THEN INVAL!(59) = 50
```

```
 N59NoCalc:
CALC232:
  LL = 32:UL = 32 :GOSUB INTERP
  inval!(73) = inval!(35) :LL = 73 :UL = 73 :GOSUB INTERP
  IF NoCalcFlag%(74) () -1 THEN INVAL!(74) = (100/56)*OUTVAL!(32)*OUTVAL!(73)
 '**********BOX 66  (node 22,EAM delay)******
 'IF NoCalcFlag%(66)=-1 GOTO N66NoCalc
 ' IF OUTVAL!(64) ( .4 THEN INVAL!(66) = 28
 'IF OUTVAL!(64) =) .4 THEN INVAL!(66) = 22.1 - 6*OUTVAL!(64)
 ' IF OUTVAL!(64) ) .7 THEN INVAL!(66) = 18.3
 N66NoCalc:

  LL = 74:UL = 74 :GOSUB INTERP
  IF NoCalcFlag%(85)()-1 THEN INVAL!(85)=52.5 + 50*OUTVAL!(74)   'DONE
CALC2:
  LL = 85:UL = 85 :GOSUB INTERP
  IF NoCalcFlag%(86)()-1 THEN INVAL!(86) = 100*OUTVAL!(85)
   'EAM effectiveness is set equal to EAM correctness - input from  62 and 66 w as not available from
 data
CALC0:
  LL = 59:UL = 59 :GOSUB INTERP
  LL = 86:UL = 86 :GOSUB INTERP
  OUTVAL!(87)=1   'Force M anagem ent is set to 100% for this calculation
  IF NoCalcFlag%(88) () -1 THEN INVAL!(88) = 100*OUTVAL!(59)*OUTVAL!(86)*OUTVAL!(87)
CALCDONE:
  RETURN
'**********************end node calculations*****************

'*******************interpolation m odule*******************

INTERP:
  FOR J=LL TO UL

    IF RV!(J,2)=0 OR INVAL!(J)=RV!(J,1) THEN OUTVAL!(J) = SC!(J,1):GOTO NextInterp
    IF RV!(J,2) ) RV!(J,1) THEN InterpFlag=1 ELSE InterpFlag = -1
    IF  InterpFlag*INVAL!(J) (= InterpFLAG*RV!(J,1) THEN OUTVAL!(J) = SC!(J,1):GOTO NextInterp
    FOR K = 2 TO 6
      IF RV!(J,K) = 0  OR  RV!(J,K)=RV!(J,K-1) THEN OUTVAL!(J) = SC!(J,K-1):GOTO NextInterp
      IF  InterpFlag*INVAL!(J) (= InterpFlag*RV!(J,K) THEN OUTVAL!(J) = SC!(J,K-1) + (SC!(J,K)-SC
!(J,K-1))*(INVAL!(J)-RV!(J,K-1))/(RV!(J,K)-RV!(J,K-1)):GOTO NextInterp
    NEXT K
 NextInterp:
   NEXT J
   RETURN

'*******************end interpolation m odule*************************
```

```
'*****************HOUSEKEEPING*********************************

HouseKeeping:        'Startup routines

DEFINT A-Z
DIM INVAL!(88),OUTVAL!(88),BN(88),FF$(88),NN$(88),NA$(88),UM$(88)
DIM P(88,4),C(88,5),S(88,5),SC!(88,6),RV!(88,6),INVAL1!(88)
DIM REC(20):DIM PP(2500):DIM NoCalcFlag%(88)
XX = 88:ParentNode = XX:SelectedNode = ParentNode:CHANGEFLAG = 0
DataSetRecordSize = 730
pnam1$ = STRING$(26,"") :pnam2$ = STRING$(26,"")
GT$ = "  )  ":LT$ = "  (  "
TimeToGo = 42      'set this value so that counter is 0 when deffile loads
BDXWIDE=95
BDXHIGH=53
SPACING=125
PICTURE ON
  CALL PENSIZE(1,1)
  CALL MOVETO(0,0)
  CALL LINE(0,BDXHIGH)
  CALL LINE(BDXWIDE,0)
  CALL LINE(0,-BDXHIGH)
  CALL LINE(-BDXWIDE,0)
PICTURE OFF
BDX1$=PICTURE$
PICTURE ON
  CALL PENSIZE(2,2)
  CALL MOVETO(0,0)
  CALL LINE(0,BDXHIGH)
  CALL LINE(BDXWIDE,0)
  CALL LINE(0,-BDXHIGH)
  CALL LINE(-BDXWIDE,0)
PICTURE OFF
BDX2$=PICTURE$

  CLS
  WINDOW 1,,(105,80)-(405,300),4
  CALL TEXTSIZE(20)
  CALL TEXTFACE(1)

  PRINT
  PRINT "      VANGUARD"
  PRINT " ANALYSTS SUPPORT"
  PRINT "         TOOL"
  CALL TEXTSIZE(14):PRINT
  PRINT " Now Loading Tree Definitions ":PRINT
  PRINT "      PLEASE WAIT "
```

```
    CALL TEXTSIZE(12)
    CALL MOVETO(25,200)
    ON TIMER(1) GOSUB TimeTable: TIMER ON
  GOSUB LOADDEF
  TIMER OFF
  WINDOW CLOSE 1
  GOSUB CreateMenu
  GOSUB FirstScreen
RETURN

TimeTable:     'Provides countdown while the definition file is loading
    CALL MOVETO(25,200):TimeToGo = TimeToGo - 1
    PRINT "time to go is ";TimeToGo;" seconds"
    RETURN

LOADDEF:     'Load the definition file DEFFILE into memory
  OPEN "R",1,"DEFFILE"
  FIELD 1,2 AS A$,1 AS AA$,6 AS B$,30 AS C$,1 AS D$,8 AS E$,10 AS F$,10 AS G$,24 AS H$
,24 AS I$,12 AS REST$
  FOR I = 1 TO XX
    GET 1,I
    BN(I) = CVI(A$):FF$(I) = AA$:NN$(I) = B$:NA$(I) = C$:UM$(I) = D$
    FOR J = 1 TO 4: K = (J*2)-1:P(I,J) = CVI(MID$(E$,K,2)):NEXT J
    FOR J = 1 TO 5:K = (J*2)-1:C(I,J) = CVI(MID$(F$,K,2)):NEXT J
    FOR J = 1 TO 5:K = (J*2)-1:S(I,J) = CVI(MID$(G$,K,2)):NEXT J
    FOR J = 1 TO 6:K = (J*4)-3:SC!(I,J)= CVS(MID$(H$,K,4)):NEXT J
    FOR J = 1 TO 6:K = (J*4)-3:RV!(I,J)= CVS(MID$(I$,K,4)):NEXT J
  NEXT I
  CLOSE

   OPEN "R",2,"DATAFILE",730
  FIELD 2,26 AS DA$,704 AS DB$
  NumberOfDataSets=LOF(2)/730
  MaxPages% = 1+INT((NumberOfDataSets-1)/10)
  NUM$ = STR$(NumberOfDataSets)
  NumberEntered=NumberOfDataSets

  OPEN "R",3,"INDEXFILE",6
  FIELD 3,6 AS IX$

  OPEN "R",#4,"VAST Disk 2:TEXTFILE",900
  FIELD #4,900 AS TxtA$

' Note that Box 9 (Node 1111) has children box 4, box 5, box 6 and box 8.
' The model for this node differs depending on whether box 7 (time)
' is equal to 1 (10 mins) or 2 (20 mins). The following arrays hold two set's
'of scale values for use in calculating box 9.
```

```
DIM S4'(2,4) :DATA 59.86,75.77,82.15,86.47,13.94,42.17,63.60,102.87
DIM S5'(2,4) :DATA 36.42,78.70,86.24,104.51,21.28,53.13,58.57,90.46
DIM S6'(2,4) :DATA 42.13,80.00,118.95,0.0,16.35,21.82,126.90,0.0
DIM S8'(2,4) :DATA 20.07,91.13,97.34,104.6,13.9,16.02,63.89,82.54

  'load the two sets of scale values for calculating box 9.
  FOR i=1 TO 2 :FOR j=1 TO 4 :READ S4'(i,j):NEXT j :NEXT i
  FOR i=1 TO 2 :FOR j=1 TO 4 :READ S5'(i,j):NEXT j :NEXT i
  FOR i=1 TO 2 :FOR j=1 TO 4 :READ S6'(i,j):NEXT j :NEXT i
  FOR i=1 TO 2 :FOR j=1 TO 4 :READ S8'(i,j):NEXT j :NEXT i

  BlankStrings = STRINGS(450," ")

'The following loads the box numbers of the nodes that actually enter
'into the calculation,either as a primitive or a calculated value.
DIM RealBox(37)
DATA 1,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20
DATA 21,22,23,26,27,28,31,32,35,36,39,40,51,59,73,74,85,86,88
FOR i=1 TO 37 :READ RealBox(i):NEXT i

  RETURN
'eeeeeeeeeeeeeeeeeeeeeeend Startup Routineseeeeeeeeeeeeeeeeeeeeee

CreateMenu:
  MENU 1,0,1,"Help"
  MENU 1,1,1,"Information"
  MENU 2,0,1,"Data"
  MENU 2,1,1,"Enter New Capabilities"
  MENU 2,2,1,"Retrieve Existing Data"
  MENU 2,3,1,"Show Data Sets"
  MENU 3,0,1,"Special Functions"
  MENU 3,1,1,"Sensitivity Analysis"
  MENU 3,2,1,"Compare Data Sets"
  MENU 3,3,1,"Set Minimum Range Values"
  MENU 3,4,1,"Print screen"
  MENU 4,0,1,"EXIT"
  MENU 4,1,1,"End VAST Session
  MENU 5,0,0,"                    "
  ON MENU GOSUB menucheck          'menu is not turned on here
RETURN

menucheck:     'closed loops for permanently active menus
  menunumber=MENU(0):menuitem = MENU(1)
  IF menunumber=3 AND menuitem =4 THEN GOSUB screencopy
  IF menunumber =4 AND menuitem =1 THEN GOTO exitmenu
RETURN
```

```
FirstScreen:
  WINDOW 2,,(105,100)-(405,300),2
  CLS :CALL TEXTSIZE(20):CALL TEXTFACE(1)
  PRINT
  PRINT "       VANGUARD"
  PRINT "  ANALYSTS SUPPORT"
  PRINT "          TOOL"
  PRINT:PRINT "   (Pull Down a Menu)"
RETURN
```

Program Name: Modify DEFILE

Language: BASIC

Machine: Apple MacIntosh

Purpose: This program is a utility to support the VAST program. VAST reads a definition file which contains all data relevant to the hierarchical mission structure embedded in VAST. This program allows the user to modify data about any node in the hierarchy.

```
REM Modify DEFFILE
REM  This program enters node data into DEFFILE.
DEFINT A-Z
cleer$ = STRING$(120," ")
OPEN "R",1,"DEFFILE"
FIELD 1, 2 AS A$, 1 AS AA$, 6 AS B$, 30 AS C$, 1 AS D$, 8 AS E$, 10 AS F$, 10 AS G$, 24 AS H$,
 24 AS I$, 12 AS DU$
CLS
CLEARALL:
FF$ = "": NN$ = "": NA$ = "": UM$ = "": P$ = "": CH$ = "": SC$ = "": RV$ = "": S$ = ""
FOR I = 1 TO 6
P(I) = 0: C(I) = 0: S(I) = 0: SC(I) = 0: RV(I) = 0
NEXT I

WINDOW 1,,(1,20)-(490,340),2

CLS
PRINT "1  ENTER DATA FOR ALL NODES"
PRINT: PRINT: PRINT "2  EDIT DATA FOR A SPECIFIC NODE"
PRINT: PRINT: PRINT "3  EXIT PROGRAM"
PRINT: PRINT: INPUT "ENTER THE NUMBER OF YOUR CHOICE";CHOICE
IF CHOICE < 1 OR CHOICE > 3 THEN GOTO CLEARALL
ON CHOICE GOSUB ENTERALL, EDITONE, ENDIT
CLS: GOTO CLEARALL

ENDIT:
   CLS
   CLOSE
   WINDOW CLOSE 1
   END

ENTERALL:
   INPUT "ENTER THE BOX NUMBER OF THIS NODE OR 0 TO END ",BN
   IF BN = 0 THEN CLOSE: RETURN
   INPUT "ENTER THE FUNCTION CODE OF THE NODE ",FF$
   INPUT "ENTER THE NODE NUMBER ",NN$
   INPUT "ENTER THE NODE NAME ",NA$
   INPUT "ENTER THE NODE UNIT OF MEASURE ",UM$
   FOR I = 1 TO 4
   PRINT "ENTER THE BOX NUMBER OF PARENT "I;
   INPUT P(I)
   NEXT I
   FOR I = 1 TO 5
   PRINT "ENTER THE BOX NUMBER OF CHILD NUMBER "I;
   INPUT C(I)
   NEXT I
   FOR I = 1 TO 5
```

```
  PRINT "ENTER THE BOX NUMBER OF SIBLING NUMBER "I;
  INPUT S(I)
  NEXT I
  FOR I = 1 TO 6
  PRINT "ENTER THE RANGE VALUE FOR LEVEL "I "OF THIS NODE";
  INPUT RVK(I)
  NEXT I
  FOR I = 1 TO 6
  PRINT "ENTER THE SCALE VALUE FOR LEVEL "I  "OF THIS NODE ";
  INPUT  SCK(I)
  NEXT I
PUTIT:
  LSET A$ = MKI$ (BN)
  LSET AA$ = FF$
  LSET B$ = NN$
  LSET C$ = NA$
  LSET D$  = UM$
  FOR I = 1 TO 4
  P$ = P$ + MKI$(P(I))
  NEXT I
  LSET E$ = P$
  FOR I = 1 TO 5
  CH$ = CH$ + MKI$(C(I))
  NEXT I
  LSET F$ = CH$
  FOR I = 1 TO 5
  S$ = S$ + MKI$(S(I))
  NEXT I
  LSET G$ = S$
  FOR I = 1 TO 6
  SC$ = SC$ + MKS$(SCK(I))
  NEXT I
  LSET H$ = SC$
  FOR I = 1 TO 6
  RV$ = RV$+ MKS$(RVK(I))
  NEXT I
  LSET I$ = RV$
  PUT 1,BN
  CLS
  RETURN      :REM  goto clearall changed to return  12 may

EDITONE:
  CLS
  INPUT "ENTER THE BOX NUMBER OF THE NODE TO EDIT OR 0 TO END";BOX
  IF BOX = 0 THEN RETURN
  IF BOX < 1 OR BOX > 88 THEN PRINT "INVALID BOX NUMBER - TRY AGAIN": FOR TL = 1 TO 1000: NE
XT TL: GOTO EDITONE
```

```
   GET 1,BOX
   BN = CVI(A$): FF$ = AA$: NN$ = B$: NA$ = C$: UM$ = D$
   FOR J = 1 TO 4: K = (J*2)-1: P(J) = CVI(MID$(E$,K,2)): NEXT J
   FOR J = 1 TO 5: K = (J*2)-1: C(J) = CVI(MID$(F$,K,2)): NEXT J
   FOR J = 1 TO 5: K = (J*2)-1: S(J) = CVI(MID$(G$,K,2)): NEXT J
   FOR J = 1 TO 6: K = (J*4)-3: SCI(J) = CVS(MID$(H$,K,4)): NEXT J
   FOR J = 1 TO 6: K = (J*4)-3: RVI(J) = CVS(MID$(I$,K,4)): NEXT J
   CLS
showit:
   CALL TEXTFACE (1): PRINT " 0  NODE NUMBER ";: CALL TEXTFACE (0): PRINT NN$;TAB(50);"this
 is BOX ";BOX
   CALL TEXTFACE (1): PRINT "1  NODE NAME ";: CALL TEXTFACE (0): PRINT NA$
   CALL TEXTFACE (1): PRINT "2  NODE FUNCTION ";: CALL TEXTFACE (0): PRINT FF$
   CALL TEXTFACE (1): PRINT "3  UNIT OF MEASURE ";: CALL TEXTFACE (0): PRINT UM$
   CALL TEXTFACE(1): PRINT "4  PARENTS ": CALL TEXTFACE(0)
   FOR I = 1 TO 4: PRINT P(I),: NEXT I: PRINT
   CALL TEXTFACE(1): PRINT "5  CHILDREN ": CALL TEXTFACE(0)
   FOR I = 1 TO 5: PRINT C(I),: NEXT I: PRINT
   CALL TEXTFACE(1): PRINT "6  SIBLINGS ": CALL TEXTFACE(0)
   FOR I = 1 TO 5: PRINT S(I),: NEXT I: PRINT
   CALL TEXTFACE(1): PRINT "7  RANGE VALUES ": CALL TEXTFACE(0)
   FOR I = 1 TO 6: PRINT RVI(I),: NEXT I: PRINT
   CALL TEXTFACE(1): PRINT "8  SCALE VALUES ": CALL TEXTFACE(0)
   FOR I = 1 TO 6: PRINT SCI(I),: NEXT I: PRINT
   PRINT: PRINT
   vp = CSRLIN: hp = POS(0)
   PRINT "ENTER THE NUMBER ASSOCIATED WITH THE FIELD YOU WANT TO CHANGE."
   PRINT "ENTER -1 TO ESCAPE"  :REM  exit option inserted   12 may

WHICHON:
   INPUT WHICHONE
   IF WHICHONE <0 THEN RETURN       :REM  exit option inserted   12 may
   IF WHICHONE <0 OR WHICHONE> 8 THEN PRINT"INVALID CHOICE - TRY AGAIN" :FOR TL = 1 TO 1000
0: NEXT TL: GOTO WHICHON
   LOCATE vp,hp: PRINT cleer$
   LOCATE vp,hp
   IF whichone = 0 THEN INPUT "Enter the new node number ";nn$
   IF whichone = 1 THEN INPUT "Enter the new name of the node ";na$
   IF whichone = 2 THEN INPUT "Enter the new function code"; ff$
   IF whichone = 3 THEN INPUT "Enter the new unit of measure", um$
   IF whichone = 4 THEN FOR I = 1 TO 4: PRINT "Enter the number of the box for parent "i;: INPUT p
(I): NEXT I
   IF whichone = 5 THEN FOR I = 1 TO 5: PRINT "Enter the number of the box for child ";: INPUT c(i)
: NEXT i
   IF whichone = 6 THEN FOR I = 1 TO 5: PRINT "Enter the number of the box for sibling ";: INPUT s(
I): NEXT i
   IF whichone = 7 THEN FOR i = 1 TO 6. PRINT "Enter the number of the box for range value " INPU
```

```
T rv%(i): NEXT i
   IF whichone = 8 THEN FOR i = 1 TO 6: PRINT "Enter the number of the box for scale value ";: INPU
T sc%(i): NEXT i

checkok:

   INPUT "Is the data for this node correct ",yn$
   IF yn$ = "y" OR yn$ = "Y" THEN  BN = BOX : GOTO PUTIT
   CLS
   GOTO showit
```

Program Name: Load Text

Language: BASIC

Machine: Apple MacIntosh

Purpose: This program is a utility which supports the VAST program. The VAST program allows the user to view information about any node in the tree. That information is loaded into a file using this program.

```
'
' This Program enters definition data into text file
'
GOSUB Housekeeping
GOSUB GetChoice
ON Choice GOSUB ENTERNEW, EditOne, EnterDefaults, PrintAll
GOSUB EndOfJob
'
'
'
Housekeeping:
DEFINT A-Z
OPEN "R", #1,"VAST Disk 1:DEFFILE"
FIELD #1, 2 AS A$, 1 AS B$, 6 AS C$, 30 AS D$, 89 AS E$
OPEN "R", #2,"VAST Disk 2:TEXTFILE",900
FIELD #2, 900 AS F$
CLS
RETURN
'
'
'
GetChoice:
PRINT "1 ENTER DATA FOR A SPECIFIC NODE"
PRINT: PRINT "2 EDIT DATA FOR A SPECIFIC NODE"
PRINT: PRINT "3 Enter All Defaults"
PRINT: PRINT "4 Print all NonDefault Entries"
PRINT: PRINT: INPUT "ENTER THE NUMBER OF YOUR CHOICE",CHOICE
'IF CHOICE <> 1 AND CHOICE <> 2 AND CHOICE <> 3 AND CHOICE <> 4 THEN GOTO GetChoice
IF CHOICE <> 1 AND CHOICE <> 2 AND CHOICE <> 4 THEN GOTO GetChoice
RETURN


ENTERNEW:
  WHILE 1 = 1
   CLS
   INPUT "ENTER THE BOX NUMBER OF THE NODE TO Enter OR X TO END";BOX$
   IF BOX$ = "X" THEN RETURN
   J = VAL(BOX$)
   IF J < 1 OR J > 88 THEN PRINT "INVALID BOX NUMBER - TRY AGAIN": FOR TL = 1 TO 1000: NEXT T
L: GOTO EDITONE
   FF$ = STRING$(900," ")
   GET #1,J
   Nodenum$ = C$
   Nodename$ = D$
    PRINT "THE NODE NUMBER Is",NodeNum$
    PRINT " THE NODE NAME Is ",NodeName$
```

```
   PRINT "Enter 12 lines of definition."
  FOR I = 1 TO 12
     Strt = (I - 1)*75 + 1
      PRINT STR$(I); : INPUT " ",text$
     IF LEN(text$) > 75 THEN lngth = 75 ELSE lngth = LEN(text$)
     MID$(FF$,strt,lngth) = text$
  NEXT I
  LSET F$ = FF$
  PUT #2,J
  J = J + 1
  WEND
RETURN

EditOne:
  WHILE 1 = 1
  CLS
  INPUT "ENTER THE BOX NUMBER OF THE NODE TO EDIT OR X TO END";BOX$
  IF BOX$ = "X"  THEN RETURN
  J = VAL(BOX$)
  IF J < 1 OR J > 88 THEN PRINT "INVALID BOX NUMBER - TRY AGAIN": FOR TL = 1 TO 1000 NEXT T
L: GOTO EDITONE
   GET #1,J
  Nodenum$ = C$
  Nodename$ = D$
  CALL MOVETO (10,30)
  PRINT "THE NODE NUMBER  is ";NodeNum$:
  PRINT " THE NODE NAME Is ";NodeName$
  GET #2,J
  FF$ = F$
  FOR I = 1 TO 12
      Strt = (I - 1)*75 + 1
     PRINT STR$(I);" ";MID$(FF$,strt,75)
  NEXT I
GetLine:
  PRINT
  INPUT "What line to change (full stop (.) to quit)";Num$
  IF Num$ = "." THEN GOTO CheckIt
  LInenum = VAL(Num$)
   Strt = (LInenum - 1)*75 + 1
   INPUT "Enter new text"; text$
  IF LEN(text$) > 75 THEN lngth = 75 ELSE lngth = LEN(text$)
  MID$(FF$,strt,lngth) = text$
CheckIt:
  CALL MOVETO (10,30)
  PRINT "THE NODE NUMBER  is",NodeNum$
   PRINT " THE NODE NAME is ",NodeName$
  FOR I = 1 TO 12
```

```
      Strt = (I - 1)*75 + 1
      PRINT STR$(I);" ";MID$(FF$,strt,75)
    NEXT I
    INPUT "Is this Correct (Y/N)"; yesno$
    IF yesno$ = "n" THEN GOTO getline
    LSET F$ = FF$
    PUT #2,j
    WEND
RETURN
.
.
.
EnterDefaults:
    FOR I = 1 TO 88
      FF$ = STRING$(900," ")
      GET #1, I
      text$ = "Node Name:  " + D$
      MID$ (FF$, 1,75) = Text$
      text$ = "The definition of box "+ STR$(I) + " has not been entered."
      MID$ (FF$, 76,75) = Text$
      text$ = "The value of this node does not impact the final  "
      MID$ (FF$, 151,75) = Text$
      text$ = "result at Node 0.  Intermediate results may or "
      MID$ (FF$, 226,75) = Text$
      text$ = "may not reflect actual relationships derived "
      MID$ (FF$,301,75) = Text$
      text$ = " from data collection."
      MID$ (FF$,376,75) = Text$
      MID$(FF$,451,450) = STRING$(450," ")
      LSET F$ = FF$
      PUT #2, I
    NEXT I
RETURN

PrintAll:
    Btext$ = STRING$(75," ")
    FOR J = 1 TO 88
    GET #1,J : GET #2,J
      Nodenum$ = C$
      Nodename$ = D$
      LPRINT : LPRINT
      LPRINT "THE NODE NUMBER  is ",NodeNum$
      LPRINT " THE NODE NAME is ",NodeName$
      FF$ = F$
      IF MID$(FF$,76,21) = "The definition of box" THEN GOTO NexJ
      FOR I = 1 TO 12
        Strt = (i-1)*75 +1
```

```
        text$ = MID$(FF$,Strt,75)
        IF text$ = Btext$ THEN GOTO nexI
        LPRINT text$
      nexI: NEXT I
    NexJ: NEXT J
RETURN

.
.
.

EndOfJob:
   CLS
   CLOSE
  ' SYSTEM
```

Program Name: Prepross

Language: FORTRAN

Machine: VAX 11/750

Purpose: This program is a utility to support the data reduction effort. It reads data from questionnaires, computes average responses, and produces a file that the program STF can read.

```
      DIMENSION LEV(5),IO(5),IQ(7),DAT(6,6,6,6,6),CNT(6,6,6,6,6)
      DIMENSION DAT1(25),DAT2(80),DATN(300)
      CHARACTER*4 IC
      CHARACTER*6 INDD,IND1(25),IND2(125),INDN(300)
      CHARACTER*20 IFIL,PFIL
      CHARACTER*40 ITIT
C       OPEN(UNIT=12,NAME='OUT.DAT',TYPE='NEW')
      WRITE(5,900)
  900 FORMAT(1X,'Enter FileName: ',$)
      READ(5,901)NC,IFIL
  901 FORMAT(Q,A)
      WRITE(5,902)
  902 FORMAT(1X,'Enter 40 char. Title: ',$)
      READ(5,901)NT,ITIT
      OPEN(UNIT=10,NAME=IFIL,TYPE='OLD')
      PFIL=IFIL(1:NC) // '.STF'
      OPEN(UNIT=11,NAME=PFIL,TYPE='NEW')
      WRITE(11,909)ITIT
  909 FORMAT(A40)
      READ(10,903)NF
  903 FORMAT(I)
      WRITE(11,904)NF
  904 FORMAT(5I4)
      READ(10,905)(LEV(I),I=1,NF)
  905 FORMAT(5I1)
      WRITE(11,904)(LEV(I),I=1,NF)
   10 CONTINUE
      READ(10,906,END=20)IQ,IR,IC,IC
  906 FORMAT(13I1,A)
      XC=WHAT(IC)
      I=IQ(I)+1
      J=IQ(2)+I
      K=IQ(3)+1
      L=IQ(4)+1
      M=IQ(5)+1
      DAT(I,J,K,L,M)=DAT(I,J,K,L,M)+XC
      CNT(I,J,K,L,M)=CNT(I,J,K,L,M)+1.0
      GO TO 10
   20 CONTINUE
      CLOSE(UNIT=10)
      DO 30 I=1,6
      DO 30 J=1,6
      DO 30 K=1,6
      DO 30 L=1,6
      DO 30 M=1,6
      IF(CNT(I,J,K,L,M).EQ.0.0)GO TO 30
      DAT(I,J,K,L,M)=DAT(I,J,K,L,M)/CNT(I,J,K,L,M)
   30 CONTINUE
      DO 40 M=1,6
      DO 40 L=1,6
      DO 40 K=1,6
      DO 40 J=1,6
      DO 40 I=1,6
      IF(CNT(I,J,K,L,M).EQ.0.0)GO TO 40
      IK=0
      IF(I.GT.1)IK=IK+1
      IF(J.GT.1)IK=IK+1
      IF(K.GT.1)IK=IK+1
      IF(L.GT.1)IK=IK+1
      IF(M.GT.1)IK=IK+1
```

```
      INDX=(I-1)*10000+(J-1)*1000+(K-1)*100+(L-1)*10+(M-1)
      ENCODE(6,910,INDO)INDX
  910 FORMAT(I5,'0')
      DO 45 II=1,6
      IF(INDO(II:II).EQ.' ')INDO(II:II)='0'
   45 CONTINUE
      GO TO (50,60,70,70,70),IK
   50 CONTINUE
      N1=N1+1
      DAT1(N1)=DAT(I,J,K,L,M)
      IND1(N1)=INDO
      GO TO 40
   60 CONTINUE
      N2=N2+1
      DAT2(N2)=DAT(I,J,K,L,M)
      IND2(N2)=INDO
      GO TO 40
   70 CONTINUE
      NN=NN+1
      DATN(NN)=DAT(I,J,K,L,M)
      INDN(NN)=INDO
   40 CONTINUE
      WRITE(11,908)N1
  908 FORMAT(I4)
      DO 90 I=1,N1
      WRITE(11,907)IND1(I),DAT1(I)
  907 FORMAT(A6,F8.3)
   90 CONTINUE
      WRITE(11,908)N2
      DO 100 I=1,N2
      WRITE(11,907)IND2(I),DAT2(I)
  100 CONTINUE
      WRITE(11,908)NN
      DO 110 I=1,NN
      WRITE(11,907)INDN(I),DATN(I)
  110 CONTINUE
      CLOSE(UNIT=11)
      CALL EXIT
      END
      FUNCTION WHAT(IC)
      CHARACTER*4 IC
      LOGICAL*1 SWITCH
      INTEGER*2 PLACE
      REAL*4 NUM,RINTM
      SWITCH = .FALSE.
      NUM=0.0
      DO 1 I=1,4
      IF(IC(I:I).EQ.'.')GO TO 3
      IF(IC(I:I).EQ.' ')GO TO 2
      DECODE(1,900,IC(I:I))INTM
  900 FORMAT(I1)
      RINTM = INTM
      IF (SWITCH.EQ..TRUE.) THEN
         NUM = NUM + RINTM*(10.0**PLACE)
         PLACE = PLACE - 1
      ELSE
         NUM=NUM*10+RINTM
      END IF
    1 CONTINUE
    3 SWITCH = .TRUE.




      PLACE = -1
      GO TO 1
    2 CONTINUE
      WHAT=NUM
      RETURN
      END
```

Program Name: STF

Language: FORTRAN

Machine: VAX 11/750

Purpose: This program does the data reduction. The user can specify the form of the model and initial scale values; the program produces the optimal set of weights and calculates the CHI-SQUARE value.

STF calls the following subroutines;

SDM - Select Data Module
SSV - Set Scale Values
SRV - Select Regression Variables
RUN - Compute Optimal Chi-Square
RES - Display Results

```
C
C
C     SUBJECTIVE TRANSFER FUNCTION -- STF -- MAIN ROUTINE
C
C           CALLS:  SMO - SELECT DATA MODEL
C                   SSV - SET SCALE VALUES
C                   SRV - SELECT REGRESSION VARIABLES
C                   RUN - RUN AN ITERATION
C                   RES - DISPLAY RESULTS
C
C
      COMMON /UNITS/ NVID,NKID,LUN1,IOC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(9,5),ISC(8,5),
     X             IC1(30,6),IC2(250,6),IC(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSZE
C
      CHARACTER*1  ANS,FACTR
      CHARACTER*40 INFIL
C
      DATA MF,ML,LUN1/8,5,10/
      DATA FACTR/'A','B','C','D','E','F','O','R'/
C
C     INITIALIZE
C
      NOFLG=0
      N2FLG=0
      NNFLG=0
      NE=0
      N1=0
      N2=0
      NN=0
      NV=0
      NW=0
      NF=0
      DO 12 I=1,MF
      LEV(I)=0
   12 CONTINUE
C
C     INITIALIZE "SCREEN MANAGEMENT GUIDELINES"
C
      CALL SMG$CREATE_PASTEBOARD(NPID)
      CALL SMG$CREATE_VIRTUAL_DISPLAY(24,80,NVID)
      CALL SMG$PASTE_VIRTUAL_DISPLAY(NVID,NPID,1,1)
      CALL SMG$CREATE_VIRTUAL_KEYBOARD(NKID)
C
C     DISPLAY MAIN MENU
C
   10 CONTINUE
      CALL SMG$ERASE_DISPLAY(NVID)
      CALL SMG$PUT_CHARS(NVID,'********  Main  Menu  ********',2,25)
      CALL SMG$PUT_CHARS(NVID,'1  Select Data Model',10,25)
      CALL SMG$PUT_CHARS(NVID,'2  Set Scale Values',11,25)
      CALL SMG$PUT_CHARS(NVID,'3  Select Regression Variables',12,25)
      CALL SMG$PUT_CHARS(NVID,'4  Run an Iteration',13,25)
      CALL SMG$PUT_CHARS(NVID,'5  Display Results',14,25)
      CALL SMG$PUT_CHARS(NVID,'6  Exit',15,25)
C
```

```
C      PROMPT FOR MENU CHOICE
C
   15 CONTINUE
      CALL SMG$PUT_CHARS(MVID,'Enter Choice(1-6): ',18,25)
      CALL SMG$READ_STRING(MKID,ANS,,1,,,NC,,MVID)
C
C     <CR> ?  ---  EXIT
C
      IF(NC.EQ.0)GO TO 100
C
C     CHOICE = 6 ?" --  EXIT
C
      IF(ANS.EQ.'6')GO TO 100
C
C     CHOICE = 1 ?" --  SELECT DATA MODEL
C
      IF(ANS.NE.'1')GO TO 20
      CALL SDM
      GO TO 70
C
C     CHOICE = 2 ?" --  SET SCALE VALUES
C
   20 CONTINUE
      IF(ANS.NE.'2')GO TO 30
      CALL SSV
      GO TO 70
C
C     CHOICE = 3 ?  --  SELECT REGRESSION VARIABLES
C
   30 CONTINUE
      IF(ANS.NE.'3')GO TO 40
      CALL SRV
      GO TO 70
C
C     CHOICE = 4 ?  --  RUN AN ITERATION(S)
C
   40 CONTINUE
      IF(ANS.NE.'4')GO TO 50
      CALL RUN
      GO TO 55
C
C     CHOICE = 5 ?  -  DISPLAY RESULTS
C
   50 CONTINUE
      IF(ANS.NE.'5')GO TO 60
   55 CONTINUE
      CALL RES
      GO TO 70
C
C     BAD CHOICE - SEND MESSAGE AND PROMPT FOR ANOTHER CHOICE
C
   60 CONTINUE
      CALL SMG$ERASE_LINE(MVID,18,25)
      CALL BEEP
      GO TO 15
   70 CONTINUE
      GO TO 10
C
C     CLOSE OUTPUT FILE, IF OPENED  ---  EXIT FROM PROGRAM
C
```

```
  100 CONTINUE
      IF(IOC.EQ.0)GO TO 110
      CLOSE(UNIT=LUN1)
  110 CONTINUE
      CALL SMG$ERASE_DISPLAY(NVID)
      CALL SMG$SET_CURSOR_ABS(NVID,1,1)
      CALL EXIT
      END
      SUBROUTINE BEEP
C
C     RINGS THE BELL ON THE TERMINAL
C
      COMMON /UNITS/ NVID,NKID,LUN1,IOC,INFIL
      CHARACTER*40 INFIL
      CALL SMG$RING_BELL(NVID)
      RETURN
      END
```

```
      SUBROUTINE SDM
C
C
C     SELECT DATA MODEL
C
C         LETS THE USER:
C               1) SELECT THE INPUT DATA FILE
C               2) CHOOSE A 2-WAY OR N-WAY ANALYSIS
C               3) INITIALIZES FOR OTHER ROUTINES
C
C
      COMMON /UNITS/ NVIO,NKIO,LUNI,IGC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ N0FLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             ID1(30,6),ID2(250,6),ID(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAC,MSTEPS,SSIE
C
      DIMENSION JS(2)
C
      CHARACTER*1 ANS,FACTR
      CHARACTER*2 DUM(6),IN1*4,DUM3*12
      CHARACTER*40 INFIL,ITIT,IN*60
C
      CALL SMG$ERASE_DISPLAY(NVIO)
C
C     PROMPT FOR INPUT DATA FILENAME
C
   10 CONTINUE
      CALL SMG$PLT_CHARS(NVIO,'Enter FileName  -  <CR> to Exit: ',23,10)
      CALL SMG$READ_STRING(NKIO,INFIL,,16,,,,NC,,NVIO)
C
C     <CR> ?  --  EXIT
C
      IF(NC.EQ.0)GO TO 200
      CALL SMG$ERASE_LINE(NVIO,20,1)
C
C     OPEN INPUT DATA FILE
C
      OPEN(UNIT=LUN1,NAME=INFIL,TYPE='OLD',ERR=20)
      GO TO 30
C
C     ERROR TRAP - SEND MESSAGE AND PROMPT AGAIN FOR FILENAME
C
   20 CONTINUE
      CALL BEEP
      CALL SMG$ERASE_LINE(NVIO,23,1)
      CALL SMG$PLT_CHARS(NVIO,'FileName not found',20,10)
      GO TO 10
C
C     READ:     1) TITLE    2) #FACTORS   3) LEVEL FOR EACH FACTOR
C
   30 CONTINUE
      CALL SMG$ERASE_LINE(NVIO,23,1)
      CALL SMG$PLT_CHARS(NVIO,'[ Reading Data File ]',23,10)
      READ(LUN1,500)ITIT
      READ(LUN1,501)NF
      READ(LUN1,501)(LEV(I),I=1,NF)
      LEV(7)=1
```

```
      LEV(8)=1
      MFI=MF-2
C
C     READ:  1) #1-WAY DATA POINTS   2) THE 1-WAY DATA POINTS
C
      READ(LUN1,901)N1
      DO 50 I=1,N1
      READ(LUN1,903)(ID1(I,J),J=1,MFI),DAT1(I)
   50 CONTINUE
C
C     READ:  2) #2-WAY DATA POINTS   2) THE 2-WAY DATA POINTS
C
      READ(LUN1,901)N2
      DO 60 I=1,N2
      READ(LUN1,903)(ID2(I,J),J=1,MFI),DAT2(I)
   60 CONTINUE
C
C     READ:  1) #N-WAY DATA POINTS   2) THE N-WAY DATA POINTS
C
      READ(LUN1,901)NN
      DO 70 I=1,NN
      READ(LUN1,903)(ID(I,J),J=1,MFI),DAT(I)
   70 CONTINUE
      CLOSE(UNIT=LUN1)
C
C     DISPLAY THE INPUT DATA FILE PARAMETERS
C
      CALL SMG$ERASE_LINE(NVID,23,1)
      CALL SMG$PUT_CHARS(NVIC,'File:',1,10)
      CALL SMG$PUT_CHARS(NVIC,'Title:',2,10)
      CALL SMG$PUT_CHARS(NVIC,'#Pctrs:',5,10)
      CALL SMG$PUT_CHARS(NVIC,'#1-Way:',6,10)
      CALL SMG$PUT_CHARS(NVIC,'#2-Way:',7,10)
      CALL SMG$PUT_CHARS(NVIC,'# -Way:',8,10)
      CALL SMG$PUT_CHARS(NVIC,INFIL,1,20)
      CALL SMG$PUT_CHARS(NVIC,ITIT,2,20)
      ENCODE(1,902,IN)NF
      CALL SMG$PUT_CHARS(NVIC,IN(1:1),5,20)
      CALL SMG$PUT_CHARS(NVIC,IN(1:1),8,11)
      DO 40 I=1,NF
      ENCODE(2,904,DUM(I))LEV(I)
   40 CONTINUE
      DUMA=DUM(1)//DUM(2)//DUM(3)//DUM(4)//DUM(5)//DUM(6)
      IB=NF*2
      IN='(Levels:'//DUMA(1:IB)//')'
      IB=IB+9
      CALL SMG$PUT_CHARS(NVIC,IN(1:IB),5,26)
      ENCODE(2,904,IN)N1
      IB=1
      IF(IN(1:1).EQ.' ')IB=2
      CALL SMG$PUT_CHARS(NVID,IN(IB:2),6,20)
      ENCODE(3,905,IN)N2
      IB=1
      IF(IN(1:1).EQ.' ')IB=2
      IF(IN(2:2).EQ.' ')IB=3
      CALL SMG$PUT_CHARS(NVID,IN(IB:3),7,20)
      ENCODE(3,905,IN)NN
      IB=1
      IF(IN(1:1).EQ.' ')IB=2
      IF(IN(2:2).EQ.' ')IB=3
```

```
      CALL SMG$PLT_CHARS(NVID,IN(19:3),8,20)
      CALL SMG$ERASE_LINE(NVID,23,1)
C
C     DO A 2-WAY ANALYSIS ?
C
      CALL SMG$PLT_CHARS(NVID,'Perform a 2-Way Analysis (Y/N)? ',23,10)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 180
C
C     PACK Y AND NE FOR A 2-WAY ANALYSIS - SET FLAGS
C
      DO 90 I=1,N2
      Y(I)=DAT2(I)
   90 CONTINUE
      NE=N2
      N2FLG=1
      NNFLG=0
      GO TO 190
C
C     DO A N-WAY ANALYSIS ?
C
  180 CONTINUE
      ENCODE(1,902,ANS)NF
      IN='Perform a '//ANS//'-Way Analysis (Y/N)?'
      CALL SMG$PLT_CHARS(NVID,IN(1:32),23,10)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 200
C
C     PACK Y AND NE FOR A N-WAY ANALYSIS - SET FLAGS
C
      DO 100 I=1,NN
      Y(I)=DAT(I)
C
C     SET PARAMETERS - INITIALIZE ARRAYS, ETC. FOR THE OTHER ROUTINES
C
  100 CONTINUE
      NE=NN
      NNFLG=1
      N2FLG=0
  190 CONTINUE
      NOFLG=1
      NV=0
      NW=NF+NF*(NF-1)/2+NF*(NF-1)*(NF-2)/6+2
      DO 195 I=1,MF
      DO 195 J=1,ML
      S(I,J)=0.0
      T(I,J)=0.0
      ISC(I,J)=0.0
  195 CONTINUE
      DO 205 I=1,27
      ISV(I)=0
      W(I)=0.0
  205 CONTINUE
      CHISQ=0.0
      PCHISQ=0.0
C
C     EXIT TO MAIN MENU
C
```

```
200 CONTINUE
    CALL SMGSERASE_DISPLAY(KVIC)
    RETURN
900 FORMAT(A)
901 FORMAT(6I4)
902 FORMAT(I1)
903 FORMAT(6I1,F8.0)
904 FORMAT(I2)
905 FORMAT(I3)
    END
```

```
      SUBROUTINE SRV
C
C
C     SELECT REGRESSION VARIABLES
C
C     LETS THE USER:
C           DESIGN THE MATHEMATICAL STF MODEL BY
C           SELECTING REGRESSION VARIABLES.
C
C
      COMMON /UNITS/ NVID,NKID,LUN1,ICC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             IC1(30,6),ID2(250,6),ID(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CMIS2,PCMISQ
      COMMON /RUNRES/ NITER,NSTEPS,N8AD,MSTEPS,SSIE
C
      CHARACTER*1 ANS,FACTR
      CHARACTER*3 YN(2)
      CHARACTER*14 VARFM(27)
      CHARACTER*40 INFIL
      CHARACTER*60 IN
      CHARACTER*120 IN1
C
      DATA YN/' NO','YES'/
      DATA IRON,IROFF/15,0/
C
      CALL SMG$ERASE_DISPLAY(NVIC)
C
C     WAS DATA MODEL SELECTED ?
C
      IF(NOFLG.EC.0)GO TO 90
C
C     DISPLAY ALL POSSIBLE REGRESSION VARIABLES TO CHOOSE FROM
C
      CALL SMG$DRAW_LINE(NVID,2,5,2,74)
      CALL SMG$DRAW_LINE(NVID,18,5,18,74)
      CALL SMG$DRAW_LINE(NVID,22,5,22,74)
      CALL SMG$DRAW_LINE(NVID,2,39,18,39)
      CALL SMG$DRAW_LINE(NVID,2,40,18,40)
      IN='Var#  Variable Form   Selected'
      IN1=IN(1:30)//'           '//IN(1:30)
      CALL SMG$PLT_CHARS(NVID,IN1(1:70),1,5)
      DO 80 I=1,NW
      ENCODE(3,900,IN)I
      IDX=ISV(I)+1
      IL=I+2
      IC1=5
      IC2=29
      IF(I.LE.14)GO TO 70
      IL=IL-14
      IC1=45
      IC2=69
   70 CONTINUE
      CALL SMG$PLT_CHARS(NVID,IN(1:3),IL,IC1)
      CALL SMG$PLT_CHARS(NVID,YN(IDX),IL,IC2)
   80 CONTINUE
C
C     ENCODE ALL 1-WAY COMBINATIONS
```

```
C
      IV=0
      DO 100 I=1,NF
      IV=IV+1
      ENCODE(14,503,VARFM(IV))FACTR(I)
  100 CONTINUE
      IF(NF.LT.2)GO TO 130
C
C     ENCODE ALL 2-WAY COMBINATIONS
C
      DO 110 I=1,NF-1
      DO 110 J=I+1,NF
      IV=IV+1
      ENCODE(14,504,VARFM(IV))FACTR(I),FACTR(J)
  110 CONTINUE
      IF(NF.LT.3)GO TO 130
C
C     ENCODE ALL 3-WAY COMBINATIONS
C
      DO 120 I=1,NF-2
      DO 120 J=I+1,NF-1
      DO 120 K=J+1,NF
      IV=IV+1
      ENCODE(14,505,VARFM(IV))FACTR(I),FACTR(J),FACTR(K)
  120 CONTINUE
C
C     ENCODE INITIAL IMPRESSION - S(0)
C
  130 CONTINUE
      IV=IV+1
      ENCODE(14,506,VARFM(IV))
C
C     ENCODE RANGE VARIABLE - S(R)
C
      IV=IV+1
      ENCODE(14,507,VARFM(IV))
C
C     DISPLAY ALL VARIABLE FORMS
C
      DO 140 I=1,NW
      IL=I+2
      IC=11
      IF(I.LE.14)GO TO 145
      IL=IL-14
      IC=51
  145 CONTINUE
      CALL SMG$PLT_CHARS(NVIC,VARFM(I),IL,IC)
  140 CONTINUE
C
C     DISPLAY #VARIABLES SELECTED, VARIABLES SELECTED
C
      CALL DISPCV
   60 CONTINUE
      CALL SMG$ERASE_LINE(NVIC,23,1)
      IN='Use SPACE BAR to move Line Cursor  -  '
      IN1=IN(1:38)//'<CR> to (Un)Select a Variable'
      CALL SMG$PLT_CHARS(NVIC,IN1(1:67),23,5)
      IPT=1
      IL=IPT+2
      IC=6
```

```
      CALL SMG$CHANGE_RENDITION(NVIO,IL,IC,1,26,IRON,0)
C
C
C     BEGIN LOOP FOR SELECTING VARIABLES  --  READ CHOICE
C
C          SPACE = NO CHANGE        <CR> = CHANGE
C
C
   10 CONTINUE
      CALL SMG$READ_STRING(NKIO,ANS,,1,,,,NC,,NVIO)
      CALL SMG$CHANGE_RENDITION(NVIO,IL,IC,1,26,IROFF,0)
      IF(NC.EQ.1)GO TO 40
C
C     <CR>  --  CHANGE A "YES" TO "NO" OR A "NO" TO "YES"  --  UPDATE COU
C
      IF(ISV(IPT).EQ.0)NV=NV+1
      IF(ISV(IPT).EQ.1)NV=NV-1
      ISV(IPT)=1-ISV(IPT)
      IOX=ISV(IPT)+1
      CALL SMG$PLT_CHARS(NVIO,YN(IOX),IL,IC+23)
C
C     DISPLAY #VARIABLES SELECTED, VARIABLES SELECTED
C
      CALL DISPCV
C
C     SPACE BAR  --  NO CHANGE  -- LOOP TO NEXT VARIABLE
C
   40 CONTINUE
      IPT=IPT+1
      IF(IPT.GT.NW)GO TO 30
      IL=IPT+2
      IC=6
      IF(IPT.LE.14)GO TO 20
      IL=IL-14
      IC=46
   20 CONTINUE
      CALL SMG$CHANGE_RENDITION(NVIO,IL,IC,1,26,IRON,0)
      GO TO 10
C
C     END OF LOOP  --  CONFIRM SELECTION
C
   30 CONTINUE
      CALL SMG$ERASE_LINE(NVIO,23,1)
      CALL SMG$PLT_CHARS(NVIO,'Is this Correct (Y/N)? ',23,5)
      CALL SMG$READ_STRING(NKIO,ANS,,1,,,,NC,,NVIO)
      IF(NC.EQ.0)GO TO 50
      IF(ANS.NE.'N'.AND.ANS.NE.'n')GO TO 50
      GO TO 60
C
C     DATA MODEL NOT SELECTED - SEND MESSAGE AND EXIT
C
   90 CONTINUE
      CALL BEEP
      IN='No Data Model Selected'
      CALL SMG$PLT_CHARS(NVIO,IN(1:22),10,30)
      IN='See Option 1 in Main Menu'
      CALL SMG$PLT_CHARS(NVIO,IN(1:25),12,30)
      CALL SMG$SET_CURSOR_ABS(NVIO,16,30)
      CALL SMG$READ_STRING(NKIO,ANS,
     X     'Press any KEY to Return',1,,,,NC,,NVIO)
```

```
C
C     EXIT TO MAIN MENU
C
   50 CONTINUE
      RETURN
  900 FORMAT(I3)
  903 FORMAT('S(',A1,')           ')
  904 FORMAT('S(',A1,')*S(',A1,')      ')
  905 FORMAT('S(',A1,')*S(',A1,')*S(',A1,')')
  906 FORMAT('S(C)           ')
  907 FORMAT('S(R)           ')
      END
      SUBROUTINE DISPCV
C
C     DISPLAYS THE #VARIABLES SELECTED AND WHICH VARIABLES WERE SELECTED
C
      COMMON /UNITS/ NVIO,NKIO,LUN1,IOC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             ID1(30,6),ID2(250,6),ID(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27)
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSIZE
C
      CHARACTER*1 FACTR
      CHARACTER*40 ,INFIL,IN
C
      DO 40 I=19,21
      CALL SMG$ERASE_LINE(NVIO,I,1)
   40 CONTINUE
C
C     ENCODE AND DISPLAY #VARIABLES SELECTED
C
      ENCODE(22,900,IN)NV
      CALL SMG$PUT_CHARS(NVIO,IN(1:22),15,5)
      CALL SMG$PUT_CHARS(NVIO,'Selected Variables:',20,5)
      IF(NV.EQ.0)GO TO 30
      NS=0
      DO 20 I=1,NW
      IF(ISV(I).EQ.0)GO TO 20
      NS=NS+1
      IL=20
      IC=(NS-1)*3+24
      IF(NS.LE.14)GO TO 10
      IL=21
      IC=(NS-15)*3+24
   10 CONTINUE
C
C     ENCODE AND DISPLAY VARIABLE#'S SELECTED
C
      ENCODE(3,901,IN)I
      CALL SMG$PUT_CHARS(NVIO,IN(1:3),IL,IC)
   20 CONTINUE
C
C     RETURN TO "SRV"
C
   30 CONTINUE
      RETURN
  300 FORMAT('Current #Variables:',I3)
  901 FORMAT(I3)




      END
```

```
      SUBROUTINE RES
C
C
C     DISPLAYS THE RESULTS OF THE ITERATION(S) AND LINE SEARCH
C
C        SCREEN 1:  DISPLAYS INITIAL AND NEW SCALE VALUES
C        SCREEN 2:  DISPLAYS WEIGHTS
C
C
      COMMON /UNITS/ NVID,NKID,LUN1,IOC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ OAT1(30),OAT2(250),OAT(300),S(8,5),ISC(8,5),
     X             IC1(30,6),IO2(250,6),IO(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSIZE
C
      CHARACTER*1 ANS,FACTR
      CHARACTER*12 BLK,IN2
      CHARACTER*14 VARFM(27)
      CHARACTER*40 INFIL
      CHARACTER*80 IN
      CHARACTER*120 IN1
C
      DATA BLK/'            '/
C
      CALL SMG$ERASE_DISPLAY(NVIC)
C
C     WAS DATA MODEL SELECTED ?
C
      IF(NOFLG.EQ.0)GO TO 200
C
C     DISPLAY SCALE VALUES AND "TEST" SCALE VALUES
C
      DO 10 I=1,5
      ENCODE(12,908,IN2)I
      J=(I-1)*12+12
      CALL SMG$PLT_CHARS(NVID,IN2,1,J)
   10 CONTINUE
      DO 20 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      K=LEV(IK)
      NCH=K*12+7
      ENCODE(NCH,909,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIC,IN1(1:67),IL,5)
      ENCODE(NCH,910,IN)(T(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIC,IN1(1:67),IL+1,5)
   20 CONTINUE
C
C     DISPLAY THE PARAMETERS FROM THE ITERATION(S)
C
      CALL SMG$PLT_CHARS(NVID,'*Iterations:',20,5)
      CALL SMG$PLT_CHARS(NVIC,'*Steps:',20,27)
      CALL SMG$PLT_CHARS(NVIC,'(Max=  )',20,38)
      CALL SMG$PLT_CHARS(NVIC,'Step Size:',20,54)
      CALL SMG$PLT_CHARS(NVID,'Current CHISQ:',21,11)
```

```
      CALL SMG$PLT_CHARS(NVID,'Previous CHISQ:',21,45)
      ENCODE(2,902,IN2)NITER
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,17)
      IF(SSIZE.LT.1.0)GO TO 60
      ENCODE(2,902,IN2)NSTEPS
      GO TO 90
   60 CONTINUE
      ENCODE(2,902,IN2)NBAD
   90 CONTINUE
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,34)
      ENCODE(2,902,IN2)MSTEPS
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,43)
      ENCODE(12,911,IN2)SSIZE
      CALL SMG$PLT_CHARS(NVID,IN2,20,64)
      ENCODE(10,912,IN2)CHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,25)
      ENCODE(10,912,IN2)PCHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,60)
      CALL SMG$DRAW_LINE(NVID,19,5,19,75)
      CALL SMG$DRAW_LINE(NVID,22,5,22,75)
C
C     SET THE SCALE VALUES TO THE "TEST" SCALE VALUES ?
C
      IN='Set Scale Values to Test Results (Y/N)? '
      CALL SMG$PLT_CHARS(NVID,IN(1:40),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
      IYES=0
C
C     <CR> ?" --  DONT SET THE SCALE VALUES
C
      IF(NC.EQ.0)GO TO 50
C
C     ANSWER IS NOT YES  --  DONT SET SCALE VALUES
C
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 50
C
C
C     ANSWER IS YES  --  SET THE SCALE VALUES TO THE "TEST" SCALE VALUES
C
C     REDISPLAY THE NEW SCALE VALUES AND ZERO OUT THE "TEST" SCALE VALUES
C
C
      IYES=1
      DO 30 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      K=LEV(IK)
      IL=(I-1)*3+2
      NCH=K*12+7
      DO 40 J=1,K
      S(IK,J)=T(IK,J)
      T(IK,J)=0.0
   40 CONTINUE
      ENCODE(NCH,909,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN1(1:67),IL,5)
      ENCODE(NCH,910,IN)(T(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN1(1:67),IL+1,5)
```

```
   30 CONTINUE
C
C     PAUSE FOR USER TO READ SCREEN 1
C
   50 CONTINUE
      IN='***      Press any KEY for Next Screen      ***'
      CALL SMG$PUT_CHARS(NVIO,IN(1:45),23,17)
      CALL SMG$READ_STRING(NKIO,ANS,,1,,,,NC,,NVIO)
      CALL SMG$ERASE_DISPLAY(NVIO)
C
C     BEGIN SCREEN 2 DISPLAY OF WEIGHTS AND VARIABLE FORMS
C
      CALL SMG$DRAW_LINE(NVIO,2,5,2,76)
      CALL SMG$DRAW_LINE(NVIO,18,5,18,76)
      CALL SMG$DRAW_LINE(NVIO,22,5,22,76)
      CALL SMG$DRAW_LINE(NVIO,2,40,18,40)
      CALL SMG$DRAW_LINE(NVIO,2,41,18,41)
      IN='Var#  ** Weight **  Variable Form '
      IN1=IN(1:34)//'     '//IN(1:34)
      CALL SMG$PUT_CHARS(NVIO,IN1(1:72),1,5)
C
C     ENCODE AND DISPLAY WEIGHTS
C
      IV=0
      DO 80 I=1,NW
      ENCODE(3,900,IN)I
      IN1='          '
      IF(ISV(I).EQ.0)GO TO 65
      IV=IV+1
      ENCODE(12,901,IN1)W(IV)
   65 CONTINUE
      IL=I+2
      IC1=5
      IC2=11
      IF(I.LE.14)GO TO 70
      IL=IL-14
      IC1=43
      IC2=49
   70 CONTINUE
      CALL SMG$PUT_CHARS(NVIO,IN(1:3),IL,IC1)
      CALL SMG$PUT_CHARS(NVIO,IN1(1:12),IL,IC2)
   80 CONTINUE
C
C     ENCODE 1-WAY COMBINATIONS OF VARIABLE FORM
C
      IV=0
      DO 100 I=1,NF
      IV=IV+1
      ENCODE(14,903,VARFM(IV))FACTR(I)
  100 CONTINUE
      IF(NF.LT.2)GO TO 130
C
C     ENCODE 2-WAY COMBINATIONS OF VARIABLE FORM
C
      DO 110 I=1,NF-1
      DO 110 J=I+1,NF
      IV=IV+1
      ENCODE(14,904,VARFM(IV))FACTR(I),FACTR(J)
  110 CONTINUE
      IF(NF.LT.3)GO TO 130
```

```
C
C     ENCODE 3-WAY COMBINATIONS OF VARIABLE FORM
C
      DO 120 I=1,NF-2
      DO 120 J=I+1,NF-1
      DO 120 K=J+1,NF
      IV=IV+1
      ENCODE(14,905,VARFM(IV))FACTR(I),FACTR(J),FACTR(K)
  120 CONTINUE
C
C     ENCODE INITIAL IMPRESSION - S(0)
C
  130 CONTINUE
      IV=IV+1
      ENCODE(14,906,VARFM(IV))
C
C     ENCODE RANGE VARIABLE - S(R)
C
      IV=IV+1
      ENCODE(14,907,VARFM(IV))
C
C     DISPLAY THE VARIABLE FORMS JUST ENCODED
C
      DO 140 I=1,NW
      IL=I+2
      IC=25
      IF(I.LE.14)GO TO 145
      IL=IL-14
      IC=63
  145 CONTINUE
      CALL SMG$PLT_CHARS(NVID,VARFM(I),IL,IC)
  140 CONTINUE
C
C     DISPLAY PARAMETERS FROM ITERATION(S)
C
      CALL SMG$PLT_CHARS(NVID,'#Iterations:',20,5)
      CALL SMG$PLT_CHARS(NVID,'#Steps:',20,27)
      CALL SMG$PUT_CHARS(NVID,'(Max=  )',20,38)
      CALL SMG$PLT_CHARS(NVID,'Step Size:',20,54)
      CALL SMG$PLT_CHARS(NVID,'Current CHISQ:',21,11)
      CALL SMG$PLT_CHARS(NVID,'Previous CHISQ:',21,45)
      ENCODE(2,902,IN2)NITER
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,17)
      IF(SSIZE.LT.1.0)GO TO 160
      ENCODE(2,902,IN2)NSTEPS
      GO TO 190
  160 CONTINUE
      ENCODE(2,902,IN2)NBAD
  190 CONTINUE
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,34)
      ENCODE(2,902,IN2)MSTEPS
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,43)
      ENCODE(12,911,IN2)SSIZE
      CALL SMG$PLT_CHARS(NVID,IN2,20,64)
      ENCODE(10,912,IN2)CHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,25)
      ENCODE(10,912,IN2)PCHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,60)
C
C     HAS THE SCALE VALUES BEEN SET TO THE "TEST" SCALE VALUES ?
```

```
C
      IF(IYES.EQ.1)GO TO 250
      IN='Set Scale Values to Test Results (Y/N)? '
      CALL SMG$PLT_CHARS(NVID,IN(1:40),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
C
C     <CR> ?  --  DONT SET THE SCALE VALUES TO THE TEST SCALES
C
      IF(NC.EQ.0)GO TO 240
C
C     ANSWER IS NO ?" --  DONT SET THE SCALE VALUES
C
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 240
C
C     ANSWER IS YES  --  SET SCALE VALUES TO "TEST" SCALE VALUES
C
      DO 230 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      K=LEV(IK)
      DO 230 J=1,K
      S(IK,J)=T(IK,J)
      T(IK,J)=0.0
  230 CONTINUE
      GO TO 250
C
C
C     DONT SET THE SCALE VALUES TO THE "TEST" SCALE VALUES
C
C     RESET THE VALUE OF CHISQ AND REDISPLAY IT
C
C
  240 CONTINUE
      CHISQ=PCHISQ
      ENCODE(10,912,IN2)CHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,25)
C
C     SAVE THE RESULTS ON A FILE ?
C
  250 CONTINUE
      IN='Save Results in a File (Y/N)? '
      CALL SMG$PLT_CHARS(NVID,IN(1:28),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
C
C     <CR> ?  --  DONT SAVE THE RESULTS
C
      IF(NC.EQ.0)GO TO 210
C
C     ANSWER IS NO  --  DONT SAVE THE RESULTS
C
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 210
C
C     ANSWER IS YES  --  SAVE THE RESULTS
C
C     IS THE OUTPUT FILE ALREADY OPENED ?
C
      IF(IOC.EQ.1)GO TO 260
C
```

```
908 FORMAT('      Level ',I1)
909 FORMAT('S(',A1,')    ',5F12.3)
910 FORMAT('Test    ',5F12.3)
911 FORMAT(E12.6)
912 FORMAT(F10.4)
913 FORMAT(1X,'FileName:',A40)
914 FORMAT(1X,'#1-Ways:',I2,5X,'#2-Ways:',I3,5X,'#',I1,'-Ways:'
   X    I3/1X,'#Factors:',I1,3X,'Levels:',6I2)
915 FORMAT(1X,'#Experiments:',I3,5X,'#Variables:',I2,'  of  ',I
916 FORMAT(1X,I3,23X,A14)
917 FORMAT(1X,I3,4X,E16.8,3X,A14)
918 FORMAT(1X,5F14.4)
919 FORMAT(/1X,'CHISQ:',E16.8////)
920 FORMAT(/1X,'Scale Values')
921 FORMAT(/1X,'Var#   ***  Weight  ***   Variable Form ')
    END
```

```
      SUBROUTINE SSV
C
C
C     SET SCALE VALUES
C
C     LETS THE USER:
C           1) SET SCALE VALUSE
C           2) SET CONSTRAINTS
C           3) CHANGE A SCALE VALUE
C           4) CHANGE A CONSTRAINT
C
C
      COMMON /UNITS/ NVIO,NKIO,LUN1,IOC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             IO1(30,6),IO2(250,6),IO(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSZE
C
      DIMENSION CNT(8,5),LEVEL(5)
C
      CHARACTER*1 FACTR,ANS,LEVEL
      CHARACTER*12 BLK,IN1
      CHARACTER*40 INFIL
      CHARACTER*80 IN
      CHARACTER*120 IN2
C
      DATA BLK/'            '/
      DATA LEVEL/'1','2','3','4','5'/
C
      CALL SMG$ERASE_DISPLAY(NVID)
C
C     WAS A DATA MODEL SELECTED ?
C
      IF(NOFLG.EQ.0)GO TO 490
C
C     DISPLAY CURRENT SCALE VALUES AND THEIR CONSTRAINTS
C
      DO 10 I=1,5
      ENCODE(12,900,IN1)I
      J=(I-1)*12+12
      CALL SMG$PLT_CHARS(NVID,IN1,1,J)
   10 CONTINUE
      CALL SMG$PLT_CHARS(NVID,'*Scale*',1,5)
      DO 20 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      K=LEV(IK)
      NCH=K*12+7
      ENCODE(NCH,901,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN2=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN2(1:67),IL,5)
      ENCODE(NCH,902,IN)(ISC(IK,J),J=1,K)
      IN2=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN2(1:67),IL+1,5)
   20 CONTINUE
C
C     DISPLAY SUB-MENU
```

```
C     NO  --  OPEN THE FILE AND SET FLAG
C
      OPEN(UNIT=LUN1,NAME='RESULTS.STF',TYPE='NEW')
      IOC=1
C
C     WRITE THE RESULTS
C
  260 CONTINUE
      WRITE(LUN1,913)INFIL
      WRITE(LUN1,914)N1,N2,NF,NN,NF,(LEV(I),I=1,NF)
      WRITE(LUN1,915)NE,NV,NW
      WRITE(LUN1,921)
      IV=0
      DO 270 I=1,NW
      IF(ISV(I).EQ.1)GO TO 275
      WRITE(LUN1,916)I,VARFM(I)
      GO TO 270
  275 CONTINUE
      IV=IV+1
      WRITE(LUN1,917)I,W(IV),VARFM(I)
  270 CONTINUE
      WRITE(LUN1,920)
      DO 280 I=1,MF
      WRITE(LUN1,918)(S(I,J),J=1,ML)
  280 CONTINUE
      WRITE(LUN1,919)CHISQ
      GO TO 210
C
C     NO DATA MODEL SELECTED - SEND MESSAGE AND EXIT
C
  200 CONTINUE
      CALL BEEP
      IN='No Data Model Selected'
      CALL SMG$PLT_CHARS(NVID,IN(1:22),10,30)
      IN='See Option 1 in Main Menu'
      CALL SMG$PLT_CHARS(NVID,IN(1:25),12,30)
      CALL SMG$SET_CURSOR_ABS(NVID,16,30)
      CALL SMG$READ_STRING(NKID,ANS,
     X    'Press any KEY to Return',1,,,,,NC,,NVID)
      GO TO 220
C
C     PAUSE FOR USER TO READ SCREEN 2
C
  210 CONTINUE
      IN='***      Press any KEY to Return      ***'
      CALL SMG$PLT_CHARS(NVID,IN(1:39),23,20)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,,NC,,NVID)
C
C     RETURN TO MAIN MENU
C
  220 CONTINUE
      RETURN
  900 FORMAT(I3)
  901 FORMAT(E12.6)
  902 FORMAT(I2)
  903 FORMAT('S(',A1,')          ')
  904 FORMAT('S(',A1,')*S(',A1,')      ')
  905 FORMAT('S(',A1,')*S(',A1,')*S(',A1,')')
  906 FORMAT('S(C)             ')
  907 FORMAT('S(R)             ')
```

```
C
   30 CONTINUE
      IN='1=Scale Defaults          3=Constraint Defaults     '
      IN2=IN(1:50)//'5=Return to Main Menu'
      CALL SMG$PLT_CHARS(NVIC,IN2(1:71),20,5)
      IN='2=Change a Scale          4=Change a Constraint'
      CALL SMG$PLT_CHARS(NVIC,IN(1:48),21,5)
      CALL SMG$DRAW_LINE(NVID,19,5,19,75)
      CALL SMG$DRAW_LINE(NVIC,22,5,22,75)
C
C     PROMPT FOR SUB-MENU CHOICE
C
   35 CONTINUE
      CALL SMG$PLT_CHARS(NVIC,'Enter Choice (1-5): ',23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
C
C     <CR> ?  -- EXIT
C
      IF(NC.EQ.0)GO TO 500
C
C     CHOICE = 5 ?  --  EXIT
C
      IF(ANS.EQ.'5')GO TO 500
C
C     CHOICE = 1 ?  --  SET SCALE VALUES TO DEFAULTS
C
      IF(ANS.NE.'1')GO TO 40
C
C     ERASE PREVIOUS SUB-MENU  --  DISPLAY NEW SUB-MENU
C
      DO 100 I=20,23
      IF(I.EQ.22)GO TO 100
      CALL SMG$ERASE_LINE(NVID,I,1)
  100 CONTINUE
      IN='1=Means          3=log(Means)    '
      CALL SMG$PLT_CHARS(NVIC,IN(1:30),20,5)
      IN='2=1-Ways         4=log(1-Ways)   '
      CALL SMG$PLT_CHARS(NVIC,IN(1:30),21,5)
C
C     PROMPT FOR NEW SUB-MENU CHOICE (OF DEFAULTS)
C
  105 CONTINUE
      CALL SMG$PLT_CHARS(NVIC,'Enter Choice - <CR> to"Exit: ',23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
C
C     <CR> ?" --  GO BACK TO PREVIOUS SUB-MENU
C
      IF(NC.EQ.0)GO TO 70
C
C     CHOICE = 1 OR 3 ?" --  DEFAULT SCALE VALUES TO MEANS
C
      IF(ANS.NE.'1'.AND.ANS.NE.'3')GO TO 110
      DO 150 I=1,MF
      DO 150 J=1,ML
      S(I,J)=0.0
      CNT(I,J)=0.0
  150 CONTINUE
C
C     CALCULATE SUMS - KEEP COUNT
C
```

```
      DO 155 II=1,NE
      DO 155 I=1,NF
      IF(N2FLG.EQ.1)GO TO 152
C
C     N-WAY ANALYSIS
C
      J=IO(II,I)
      IF(J.EQ.0)GO TO 155
      S(I,J)=S(I,J)+CAT(II)
      CNT(I,J)=CNT(I,J)+1.0
      GO TO 155
C
C     2-WAY ANALYSIS
C
  152 CONTINUE
      J=IO2(II,I)
      IF(J.EQ.0)GO TO 155
      S(I,J)=S(I,J)+CAT2(II)
      CNT(I,J)=CNT(I,J)+1.0
  155 CONTINUE
C
C     CALCULATE MEANS
C
      DO 160 I=1,NF
      K=LEV(I)
      DO 160 J=1,K
      IF(CNT(I,J).EQ.0.0)GO TO 160
      S(I,J)=S(I,J)/CNT(I,J)
      IF(ANS.EQ.'1')GO TO 160
      S(I,J)=ALOG10(S(I,J))
  160 CONTINUE
      GO TO 130
C
C     CHOICE = 2 OR 4 ?  --  DEFAULT SCALE VALUES TO 1-WAYS
C
  110 CONTINUE
      IF(ANS.NE.'2'.AND.ANS.NE.'4')GO TO 125
      DO 115 I=1,MF
      DO 115 J=1,ML
      S(I,J)=0.0
      CNT(I,J)=0.0
  115 CONTINUE
C
C     CALCULATE SUMS (IF ANY) - KEEP COUNT
C
      DO 120 II=1,N1
      DO 120 I=1,NF
      J=IO1(II,I)
      IF(J.EQ.0)GO TO 120
      S(I,J)=S(I,J)+CAT1(II)
      CNT(I,J)=CNT(I,J)+1.0
  120 CONTINUE
C
C     CALCULATE MEANS OF 1-WAYS
C
      DO 122 I=1,NF
      K=LEV(I)
      DO 122 J=1,K
      IF(CNT(I,J).EQ.0.0)GO TO 122
      S(I,J)=S(I,J)/CNT(I,J)
```

```
      IF(ANS.EQ.'2')GO TO 122
      S(I,J)=ALOG10(S(I,J))
  122 CONTINUE
      GO TO 130
C
C     BAD ANSWER - SEND MESSAGE AND ASK AGAIN
C
  125 CONTINUE
      CALL BEEP
      CALL SMG$ERASE_LINE(NVID,23,1)
      GO TO 105
C
C     COMPUTE INITIAL IMPRESSION DEFAULT  --  S(0)
C
  130 CONTINUE
      SMIN=1.0E+10
      DO 135 I=1,NF
      K=LEV(I)
      DO 135 J=1,K
      IF(S(I,J).LT.SMIN)SMIN=S(I,J)
  135 CONTINUE
      S(7,1)=SMIN
C
C     REDISPLAY SCALE VALUES  --  RETURN TO PREVIOUS SUB-MENU
C
      DO 140 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      K=LEV(IK)
      NCH=K*12+7
      ENCODE(NCH,901,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN2=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN2(1:67),IL,5)
  140 CONTINUE
      GO TO 70
C
C     CHOICE = 2 ?  --  CHANGE A SCALE VALUE
C
   40 CONTINUE
      IF(ANS.NE.'2')GO TO 50
C
C     ERASE PREVIOUS SUB-MENU
C
      DO 210 I=20,23
      IF(I.EQ.22)GO TO 210
      CALL SMG$ERASE_LINE(NVID,I,1)
  210 CONTINUE
      IN='Scale A, Level 2:      CODE=A2          '
      IN2=IN(1:3E)//'Scale C, Level 3:  CODE=C3   '
      CALL SMG$PLT_CHARS(NVID,IN2(1:66),20,5)
      IN='Scale B, all Levels:  CODE=B            '
      IN2=IN(1:3E)//'Scale D:           CODE=D'
      CALL SMG$PLT_CHARS(NVID,IN2(1:66),21,5)
C
C     PROMPT FOR "CODE"
C
  220 CONTINUE
      CALL SMG$PLT_CHARS(NVID,'Enter CODE - <CR> to Exit: ',23,5)
      CALL SMG$READ_STRING(NKID,IN1,,3,,,,NC,,NVID)
```

```
C
C     <CR> ?  --  GO BACK TO PREVIOUS MENU
C
      IF(NC.EQ.0)GO TO 70
C
C     IS "CODE" SYNTAX CORRECT ?
C
      IF(NC.GT.2)GO TO 270
      DO 230 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      IF(IN1(1:1).EQ.FACTR(IK))GO TO 240
  230 CONTINUE
      GO TO 270
  240 CONTINUE
      K=LEV(IK)
      IF(NC.EQ.1)GO TO 280
      DO 250 J=1,K
      JK=J
      IF(IN1(2:2).EQ.LEVEL(J))GO TO 280
  250 CONTINUE
C
C     "CODE" SYNTAX IS BAD  --  SEND MESSAGE AND PROMPT AGAIN
C
  270 CONTINUE
      CALL BEEP
      CALL SMG$ERASE_LINE(NVIO,23,1)
      GO TO 220
C
C     "CODE" SYNTAX IS CORRECT  --  PROMPT FOR A NEW SCALE VALUE
C
  280 CONTINUE
      CALL SMG$PLT_CHARS(NVIO,'Enter New Scale Value: ',23,40)
      CALL SMG$READ_STRING(NKIO,IN1,,8,,,,MC,,NVIO)
C
C     <CR> ?  --  GO BACK TO PREVIOUS MENU
C
      IF(MC.EQ.0)GO TO 70
C
C     DECODE NEW SCALE VALUE
C
      DO 275 I=1,MC
      IF(IN1(I:I).EQ.'.')GO TO 285
  275 CONTINUE
      DECODE(MC,504,IN1(1:MC))IOFLT
      OFLT=IOFLT
      GO TO 295
  285 CONTINUE
      DECODE(MC,503,IN1(1:MC))OFLT
  295 CONTINUE
      IF(NC.EQ.2)GO TO 300
C
C     SET SCALE TO NEW VALUE
C
      DO 290 J=1,K
      S(IK,J)=OFLT
  290 CONTINUE
      GO TO 310
  300 CONTINUE
```

```
      S(IK,JK)=DFLT
  310 CONTINUE
C
C     REDISPLAY SCALE VALUES  --  LOOP BACK FOR ANOTHER "CODE"
C
      NCH=K*12+7
      ENCODE(NCH,901,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN2=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIC,IN2(1:67),IL,5)
      CALL SMG$ERASE_LINE(NVIO,23,1)
      GO TO 220
C
C     CHOICE = 3 ?  --  SET CONSTRAINTS TO  DEFALLTS
C
   50 CONTINUE
      IF(ANS.NE.'3')GO TO 60
C
C     ERASE PREVIOUS SUB-MENU  --  DISPLAY NEW SUB-MENU
C
      DO 165 I=20,23
      IF(I.EQ.22)GO TO 165
      CALL SMG$ERASE_LINE(NVIO,I,1)
  165 CONTINUE
      IN='0  -  WILL NOT Constrain the Scale Value'
      CALL SMG$PLT_CHARS(NVIO,IN(1:40),20,5)
      IN='1  -  WILL Constrain the Scale Value'
      CALL SMG$PLT_CHARS(NVIC,IN(1:36),21,5)
      IN='Enter Default for all Constraints (0 or 1): '
C
C     PROMPT FOR DEFAULT VALUE OF CONSTRAINTS
C
  170 CONTINUE
      CALL SMG$PLT_CHARS(NVIC,IN(1:44),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVIC)
C
C     <CR> ?  --  GO BACK TO PREVIOUS SUB-MENU
C
      IF(NC.EQ.0)GO TO 70
C
C     IS DEFAULT VALUE VALID (0 OR 1) ?
C
      IF(ANS.EQ.'0'.OR.ANS.EQ.'1')GO TO 180
C
C     BAD ANSWER - SEND MESSAGE AND ASK AGAIN
C
      CALL BEEP
      CALL SMG$ERASE_LINE(NVIO,23,1)
      GO TO 170
C
C     SET CONSTRAINTS TO DEFAULT VALUE  --  REDISPLAY CONSTRAINTS
C
  180 CONTINUE
      IDFLT=0
      IF(ANS.EQ.'1')IDFLT=1
      DO 190 I=1,MF
      DO 190 J=1,ML
      ISC(I,J)=IDFLT
  190 CONTINUE
      DO 200 I=1,NF+1
      IK=I
```

```
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      K=LEV(IK)
      NCH=K*12+7
      ENCODE(NCH,902,IN)(ISC(IK,J),J=1,K)
      IN2=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIC,IN2(1:67),IL+1,5)
  200 CONTINUE
      GO TO 70
C
C     CHOICE = 4 ?" --  CHANGE A CONSTRAINT
C
   60 CONTINUE
      IF(ANS.NE.'4')GO TO 90
C
C     ERASE PREVIOUS SUB-MENU  --  PROMPT FOR "CODE"
C
      DO 315 I=20,23
      IF(I.EQ.22)GO TO 315
      CALL SMG$ERASE_LINE(NVIC,I,1)
  315 CONTINUE
      IN='Constr. A, Level 2:     CODE=A2          '
      IN2=IN(1:40)//'Constr. C, Level 3:  CODE=C3   '
      CALL SMG$PLT_CHARS(NVIC,IN2(1:70),20,5)
      IN='Constr. B, all Levels:  CODE=B           '
      IN2=IN(1:40)//'Constr. 0:           CODE=0    '
      CALL SMG$PLT_CHARS(NVIC,IN2(1:70),21,5)
C
C     PROMPT FOR "CODE"
C
  320 CONTINUE
      CALL SMG$PLT_CHARS(NVIO,'Enter CODE - <CR> to Exit: ',23,5)
      CALL SMG$READ_STRING(NKIO,IN1,,3,,,,NC,,NVIO)
C
C     <CR> ?  --  GO BACK TO PREVIOUS SUB-MENU
C
      IF(NC.EQ.0)GO TO 70
C
C     IS "CODE" SYNTAX CORRECT ?
C
      IF(NC.GT.2)GO TO 370
      DO 330 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      IF(IN1(1:1).EQ.FACTR(IK))GO TO 340
  330 CONTINUE
      GO TO 370
  340 CONTINUE
      K=LEV(IK)
      IF(NC.EQ.1)GO TO 380
      DO 350 J=1,K
      JK=J
      IF(IN1(2:2).EQ.LEVEL(J))GO TO 375
  350 CONTINUE
C
C     "CODE" SYNTAX IS BAD  --  SEND MESSAGE AND ASK AGAIN
C
  370 CONTINUE
      CALL BEEP
```

```
      CALL SMG$ERASE_LINE(NVIO,23,1)
      GO TO 320
C
C     "CODE" SYNTAX IS CORRECT  --  FLIP FLOP CONSTRAINT VALUE
  375 CONTINUE
      ISC(IK,JK)=1-ISC(IK,JK)
      GO TO 400
  380 CONTINUE
      DO 390 J=1,K
      ISC(IK,J)=1-ISC(IK,J)
  390 CONTINUE
C
C     REDISPLAY CONSTRAINTS  -  LOOP BACK FOR ANOTHER "CODE"
C
  400 CONTINUE
      NCH=K*12+7
      ENCODE(NCH,902,IN)(ISC(IK,J),J=1,K)
      IN2=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIO,IN2(1:67),IL+1,5)
      CALL SMG$ERASE_LINE(NVIO,23,1)
      GO TO 320
C
C     LOOP BACK TO PREVIOUS SUB-MENU
C
   70 CONTINUE
      DO 80 I=20,23
      IF(I.EQ.22)GO TO 80
      CALL SMG$ERASE_LINE(NVIO,I,1)
   80 CONTINUE
      GO TO 30
C
C     BAD ANSWER - SEND MESSAGE AND PROMPT AGAIN
C
   90 CONTINUE
      CALL BEEP
      CALL SMG$ERASE_LINE(NVIO,23,1)
      GO TO 35
C
C     DATA MODEL WAS NOT SELECTED - SEND MESSAGE AND EXIT
C
  490 CONTINUE
      CALL BEEP
      IN='No Data Model Selected'
      CALL SMG$PLT_CHARS(NVIO,IN(1:22),10,30)
      IN='See Option 1 in Main Menu'
      CALL SMG$PLT_CHARS(NVIO,IN(1:25),12,30)
      CALL SMG$SET_CURSOR_ABS(NVIO,16,30)
      CALL SMG$READ_STRING(NKIO,ANS,
     X     'Press any KEY to Return',1,,,,NC,,NVIO)
C
C     EXIT TO MAIN MENU
C
  500 CONTINUE
      RETURN
  900 FORMAT('     Level ',I1)
  901 FORMAT('S(',A1,')   ',5F12.2)
  902 FORMAT('Constr.',5I12)
  903 FORMAT(F)
  904 FORMAT(I)
      END
```

```
      SUBROUTINE RUN
C
C
C     THIS ROUTINE PERFORMS THE ITERATIONS FOR FINDING THE OPTIMAL
C     CHISQ FOR THE MODEL SELECTED.
C
C     THE STEPS INCLUDE:
C         1) ENTER #ITERATIONS (DEFAULT=1)
C         2) ENTER MAX #STEPS (DEFAULT=20)
C         3) COMPUTE C MATRIX (USING INITIAL SCALE VALUES)
C         4) COMPUTE INITIAL CHISQ AND WEIGHTS
C         5) COMPUTE GRADIENTS (FOR EVERY LEVEL OF EACH FACTOR)
C         6) COMPUTE "TEST" SCALE VALUES
C         7) COMPUTE NEW C MATRIX (USING "TEST" SCALE VALUES)
C         8) COMPUTE NEW CHISQ AND WEIGHTS
C         9) COMPARE NEW CHISQ WITH PREVIOUS CHISQ
C        10) ADJUST STEPSIZE - GO TO STEP 6
C
C
C     THIS ROUTINE CALLS 3 SUBROUTINES
C         1) COMPC - COMPUTES C MATRIX
C         2) COMPW - COMPUTES WEIGHTS AND CHISQ
C         3) COMPG - COMPUTES THE GRADIENT
C
C     COMPW CALLS MLINEQ - MATRIX INVERSION FROM ALPHATECH LIBRARY
C     COMPG CALLS COMPCD - COMPUTES THE PARTIAL DERIVATIVE OF C MATRIC
C
C
      COMMON /UNITS/ NVID,NKID,LUN1,IOC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NDFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             IC1(30,6),IO2(250,6),IO(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAC,MSTEPS,SSIZE
C
      DIMENSION GRAD(8,5),TEM(8,5)
C
      CHARACTER*1 ANS,FACTR
      CHARACTER*40 INFIL
      CHARACTER*80 IN
C
      DATA MXSTEPS,MXITER/20,20/
C
C      OPEN(UNIT=11,NAME='OUT.DAT',TYPE='NEW')
      CALL SMG$ERASE_DISPLAY(NVID)
C
C     WAS DATA MODEL SELECTED ?
C
      IF(NDFLG.EQ.0)GO TO 410
C
C     WERE ANY REGRESSION VARIABLES SELECTED
C
      IF(NV.EQ.0)GO TO 420
C
C     PROMPT FOR #ITERATIONS (DEFAULT=1)
C
      NITER=1
      CALL SMG$PLT_CHARS(NVID,'Enter #Iterations: ',10,30)
      CALL SMG$READ_STRING(NKID,IN,,3,,,,NC,,NVID)
```

```
C
C      <CR>  --  SET TO DEFAULT
C
       IF(NC.EQ.0)GO TO 60
       DECODE(NC,900,IN(1:NC))NITER
       IF(NITER.GT.MXITER)NITER=MXITER
C
C      PROMPT FOR MAX #STEPS (DEFAULT=20)
C
   60  CONTINUE
       MSTEPS=MXSTEPS
       CALL SMG$PUT_CHARS(NVID,'Enter Max #Steps: ',12,30)
       CALL SMG$READ_STRING(NKID,IN,,3,,,,NC,,NVID)
C
C      <CR>  --  SET TO DEFAULT
C
       IF(NC.EQ.0)GO TO 70
       DECODE(NC,900,IN(1:NC))MSTEPS
       IF(MSTEPS.GT.MXSTEPS)MSTEPS=MXSTEPS
C
C      SAVE CHISQ AND SCALE VALUES
C
   70  CONTINUE
       CALL SMG$ERASE_DISPLAY(NVID)
       PCHISQ=CHISQ
       DO 80 I=1,MF
       DO 80 J=1,ML
       TEM(I,J)=S(I,J)
       T(I,J)=0.0
   80  CONTINUE
C
C      DRAW DISPLAY TO SHOW USER (DURING ITERATIONS)
C           1) ITERATION
C           2) ELAPSED TIME
C           3) PREVIOUS CHISQ
C           4) CURRENT CHISQ
C
       CALL SMG$DRAW_LINE(NVID,8,24,8,57)
       CALL SMG$DRAW_LINE(NVID,11,24,11,57)
       CALL SMG$DRAW_LINE(NVID,14,24,14,57)
       CALL SMG$DRAW_LINE(NVID,8,24,14,24)
       CALL SMG$DRAW_LINE(NVID,8,57,14,57)
       CALL SMG$PUT_CHARS(NVID,'Iteration:     of',9,28)
       CALL SMG$PUT_CHARS(NVID,'Elapsed Time:',10,28)
       CALL SMG$PUT_CHARS(NVID,'Previous CHISQ:',12,28)
       CALL SMG$PUT_CHARS(NVID,'Current CHISQ:',13,28)
       CALL SMG$PUT_CHARS(NVID,' 0',9,38)
       ENCODE(2,901,IN)NITER
       CALL SMG$PUT_CHARS(NVID,IN(1:2),9,46)
       CALL SMG$PUT_CHARS(NVID,' 0:00',10,43)
       ENCODE(12,902,IN)PCHISQ
       CALL SMG$PUT_CHARS(NVID,IN(1:12),12,43)
       CURRJ=0.0
       ENCODE(12,902,IN)CURRJ
       CALL SMG$PUT_CHARS(NVID,IN(1:12),13,42)
       CALL LIB$DAY(ION,,NTICK)
C
C
C      BEGIN LOOP TO DO AN ITERATION
C
```

```
C      UPDATE ITERATION# ON DISPLAY
C
C
       DO 200 ITER=1,NITER
       ENCODE(2,9C1,IN)ITER
       CALL SMG$PLT_CHARS(NVIC,IN(1:2),9,38)
       ITERS=ITER
C
C      COMPUTE C MATRIX (USING CURRENT SCALE VALUES)
C
       CALL CCMPC(0)
C
C      CHECK C MATRIX FOR A COLUMN OF ZERCES
C
       DO 130 J=1,NV
       IZS=J
       SUM=0.0
       DO 120 I=1,NE
       SUM=SUM+C(I,J)
  120  CONTINUE
C
C      IS COLUMN ALL ZERDES ?
C
       IF(SUM.EQ.0.0)GC TO 400
  130  CONTINUE
C
C      COMPUTE THE WEIGHTS AND CURRENT CHISC
C
       CALL CCMPW(CURRJ)
C
C      COMPUTE THE GRADIENTS
C
       CALL CCMPG(GRAD)
C
C      INITIALIZE FOR LINE SEARCH
C
       NSTEPS=1
       NBAD=0
       SSZE=1.0
C
C
C      BEGIN LOOP ON LINE SEARCH
C
C
  100  CONTINUE
       DO 10 I=1,MF
       DO 10 J=1,ML
C
C      IS VARIABLE CONSTRAINED ?  --  IF YES, SET GRADIENT TO ZERC
C
       IF(ISC(I,J).EQ.1)GRAD(I,J)=0.0
C
C      COMPUTE "TEST" SCALE VALUES
C
       T(I,J)=S(I,J)+GRAD(I,J)*SSZE
   10  CONTINUE
C
C      COMPUTE NEW C MATRIX (USING "TEST" SCALE VALUES)
C
       CALL CCMPC(1)
```

```
C
C     COMPUTE NEW WEIGHTS AND NEW CHISQ
C
      CALL COMPW(XNEWJ)
C
C     COMPARE CURRENT CHISQ WITH NEW CHISQ
C
      IF(XNEWJ.LT.CURRJ)GO TO 20
C
C     NEW CHISQ NOT BETTER THAN CURRENT CHISQ
C     IF #STEPS > 1; WE HAVE FOUND OPTIUM CHISQ
C
      IF(NSTEPS.GT.1)GO TO 30
C
C     CUT STEP SIZE IN HALF  --  CHECK MAX #STEPS
C
      SSZE=SSZE/2.0
      NBAD=NBAD+1
      IF(NBAD.LE.MSTEPS)GO TO 100
      GO TO 50
C
C     NEW CHISQ BETTER THAN CURRENT CHISQ
C     DOUBLE STEP SIZE  --  UPDATE CURRENT CHISQ  --  CHECK MAX #STEPS
C
   20 CONTINUE
      SSZE=SSZE*2.0
      CURRJ=XNEWJ
      NSTEPS=NSTEPS+1
      IF(NSTEPS.LE.MSTEPS)GO TO 100
      GO TO 50
C
C     OPTIMUM CHISQ FOUND  --  BACKTRACK 1 STEP TO OPTIMUM
C     "TEST" SCALE VALUES AND CORRESPONDING WEIGHTS
C
   30 CONTINUE
      SSZE=SSZE/2.0
      DO 40 I=1,NF
      K=LEV(I)
      DO 40 J=1,K
      T(I,J)=S(I,J)+GRAD(I,J)*SSZE
   40 CONTINUE
      CALL COMPC(1)
      CALL COMPW(CURRJ)
C
C     SET SCALE VALUES TO OPTIMUM "TEST" SCALE VALUES FOUND
C
   50 CONTINUE
      DO 170 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      K=LEV(IK)
      DO 170 J=1,K
      S(IK,J)=T(IK,J)
      T(IK,J)=0.0
  170 CONTINUE
C
C     UPDATE ELAPSED TIME AND CURRENT CHISQ ON SCREEN DISPLAY
C
      CALL LIB$DAY(IDN,,ITICK)
      IET=(ITICK-NTICK)/100
```

```
      IM=IET/60
      IS=IET-IM*60
      ENCODE(5,903,IN)IM,IS
      CALL SMG$PLT_CHARS(NVID,IN(1:5),10,43)
      ENCODE(12,902,IN)CURRJ
      CALL SMG$PLT_CHARS(NVID,IN(1:12),12,42)
C
C     GO ANOTHER ITERATION
C
  200 CONTINUE
C
C
C     END OF ITERATIONS  --  ADJUST COUNTS
C
C
      NITER=ITERS
      IF(NBAC.GT.MSTEPS)NBAC=MSTEPS
      IF(NSTEPS.GT.MSTEPS)NSTEPS=MSTEPS
C
C     RESTORE SCALE VALUES TO INITIAL CONTENTS
C     SET "TEST" SCALE VALUES TO OPTIMUM VALUES OF LAST ITERATION
C     SET CURRENT CHISQ TO OPTIMUM VALUE OF LAST ITERATION
C
  210 CONTINUE
      CHISQ=CURRJ
      DO 110 I=1,MF
      DO 110 J=1,ML
      T(I,J)=S(I,J)
      S(I,J)=TEM(I,J)
  110 CONTINUE
      GO TO 430
C
C     A COLUMN OF ZEROES WAS FOUND IN THE C MATRIX
C     THIS WILL CAUSE THE MATRIX INVERSION PROGRAM TO BLOW UP
C     SEND MESSAGE AND EXIT
C
  400 CONTINUE
      CALL SMG$ERASE_DISPLAY(NVID)
      CALL BEEP
      NITER=ITERS-1
      IN='The data for a variable is all zeroes'
      CALL SMG$PLT_CHARS(NVID,IN(1:37),10,20)
      IN='Please change the data or the variable'
      CALL SMG$PLT_CHARS(NVID,IN(1:38),12,20)
      IN='Iteration:                  Variable:'
      CALL SMG$PLT_CHARS(NVID,IN(1:35),14,20)
      ENCODE(2,901,IN)ITERS
      CALL SMG$PLT_CHARS(NVID,IN(1:2),14,30)
      ENCODE(2,901,IN)IZS
      CALL SMG$PLT_CHARS(NVID,IN(1:2),14,55)
      CALL SMG$SET_CURSOR_ABS(NVID,20,27)
      CALL SMG$READ_STRING(NKID,ANS,
     X    'Press any KEY to Return',1,,,,KC,,NVID)
      GO TO 210
C
C     NO DATA MODEL SELECTED - SEND MESSAGE AND EXIT
C
  410 CONTINUE
      CALL BEEP
      IN='No Data Model Selected'
```

```
      CALL SMG$PLT_CHARS(NVID,IN(1:22),10,30)
      IN='See Option 1 in Main Menu'
      CALL SMG$PLT_CHARS(NVID,IN(1:25),12,30)
      CALL SMG$SET_CURSOR_ABS(NVID,16,30)
      CALL SMG$READ_STRING(NKID,ANS,
     X    'Press any KEY to Return',1,,,,AC,,NVID)
      GO TO 430
C
C     NO REGRESSION VARIABLES SELECTED - SEND MESSAGE AND EXIT
C
  420 CONTINUE
      CALL BEEP
      IN='No Regression Variables Selected'
      CALL SMG$PLT_CHARS(NVID,IN(1:32),10,30)
      IN='See Option 3 in Main Menu'
      CALL SMG$PLT_CHARS(NVID,IN(1:25),12,30)
      CALL SMG$SET_CURSOR_ABS(NVID,16,30)
      CALL SMG$READ_STRING(NKID,ANS,
     X    'Press any KEY to Return',1,,,,AC,,NVID)
C
C     EXIT TO MAIN MENU
C
  430 CONTINUE
      RETURN
  900 FORMAT(I)
  901 FORMAT(I2)
  902 FORMAT(E12.6)
  903 FORMAT(I2,':',I2)
      END
      SUBROUTINE COMPC(IFT)
C
C
C     COMPUTES C MATRIX BASED ON THE VARIABLES SELECTED AND THE
C     SCALE VALUES.
C            IFT=0  --  USE SCALE VALUES TO COMPUTE C
C            IFT=1  --  USE "TEST" SCALE VALUES TO COMPUTE C
C
C            N2FLG=1,NNFLG=0  --  DOING A 2-WAY ANALYSIS
C            N2FLG=0,NNFLG=1  --  DOING A N-WAY ANALYSIS
C
C
      COMMON /UNITS/ NVID,NKID,LUN1,IOC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             IO1(30,6),IO2(250,6),IO(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSZE
C
      DIMENSION SP(6)
C
      CHARACTER*1 FACTR
      CHARACTER*40 INFIL
C
      DO 200 II=1,NE
C
C     INITIALIZE C MATRIX - MAX AND MIN
C
      DO 160 I=1,NV
      C(II,I)=0.0
```

```
  160 CONTINUE
      RMAX=0.0
      RMIN=1.0E+10
C
C     FIND THE SCALE (OR TEST) VALUES FOR THE VARIOUS FACTORS
C     AT THE LEVEL INDICATED BY THE EXPERIMENT (II)
C
      DO 190 I=1,NF
      SP(I)=0.0
C
C     GET THE LEVEL (IX) FOR THIS FACTOR (I) FOR EXPERIMENT (II)
C
      IX=ID(II,I)
      IF(N2FLG.EQ.1)IX=ID2(II,I)
      IF(IX.EQ.0)GO TO 190
      IF(IFT.EQ.1)GO TO 170
C
C     SET TO THE SCALE VALUE FOR FACTOR "I", LEVEL "IX"
C     CALCULATE MAX AND MIN
C
      SP(I)=S(I,IX)
      IF(S(I,IX).GT.RMAX)RMAX=S(I,IX)
      IF(S(I,IX).LT.RMIN)RMIN=S(I,IX)
      GO TO 190
C
C     SET TO THE "TEST" SCALE VALUE FOR FACTOR "I", LEVEL "IX"
C     CALCULATE MAX AND MIN
C
  170 CONTINUE
      SP(I)=T(I,IX)
      IF(T(I,IX).GT.RMAX)RMAX=T(I,IX)
      IF(T(I,IX).LT.RMIN)RMIN=T(I,IX)
  190 CONTINUE
C
C
C     GENERATE THE C MATRIX
C
C
      IW=0
      IV=0
C
C     1-WAY COMBINATIONS
C
      DO 100 I=1,NF
      IV=IV+1
C
C     WAS THIS VARIABLE SELECTED ?
C
      IF(ISV(IV).EQ.0)GO TO 100
      IW=IW+1
      C(II,IW)=SP(I)
  100 CONTINUE
      IF(NF.LT.2)GO TO 130
C
C     2-WAY COMBINATIONS
C
      DO 110 I=1,NF-1
      DO 110 J=I+1,NF
      IV=IV+1
C
```

```
C       WAS THIS VARIABLE SELECTED ?
C
        IF(ISV(IV).EQ.0)GO TO 110
        IW=IW+1
        C(II,IW)=SP(I)*SP(J)
  110 CONTINUE
        IF(NF.LT.3)GO TO 130
C
C       3-WAY COMBINATIONS
C
        DO 120 I=1,NF-2
        DO 120 J=I+1,NF-1
        DO 120 K=J+1,NF
        IV=IV+1
C
C       WAS THIS VARIABLE SELECTED ?
C
        IF(ISV(IV).EQ.0)GO TO 120
        IW=IW+1
        C(II,IW)=SP(I)*SP(J)*SP(K)
  120 CONTINUE
C
C       INITIAL IMPRESSION -- S(0)
C
  130 CONTINUE
        IV=IV+1
C
C       WAS THIS VARIABLE SELECTED ?
C
        IF(ISV(IV).EQ.0)GO TO 140
        IW=IW+1
        C(II,IW)=S(7,1)
C
C       RANGE VARIABLE -- S(R)
C
  140 CONTINUE
        IV=IV+1
C
C       WAS THIS VARIABLE SELECTED ?
C
        IF(ISV(IV).EQ.0)GO TO 200
        IW=IW+1
        C(II,IW)=RMAX-RMIN
  200 CONTINUE
C
C       RETURN TO "RUN"
C
        RETURN
        END
        SUBROUTINE COMPW(XJAY)
C
C
C       THIS ROUTINE CALCULATES THE WEIGHTS AND CHISQ GIVEN THE
C       C MATRIX AND THE Y MATRIX (DEPENDENT VARIABLE)
C
C
        COMMON /UNITS/ NVID,NKID,LUN1,ICC,INFIL
        COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
        COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
        COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
```

```
     x              ID1(30,6),ID2(250,6),ID(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAC,MSTEPS,SSIZE
C
      DOUBLE PRECISION CTC(27,27),CTY(27),WORK(27),COND
      DIMENSION YMCW(250),IPVT(27)
C
      CHARACTER*1 FACTR
      CHARACTER*40 INFIL
C
C     START REGRESSION  --  COMPUTE:  (C transpose * C)
C
      DO 10 I=1,NV
      DO 10 J=1,NV
      CTC(I,J)=0.0
   10 CONTINUE
      DO 20 I=I,NV
      DO 20 J=I,NE
      DO 20 K=1,NV
      CTC(I,K)=CTC(I,K)+C(J,I)*C(J,K)
   20 CONTINUE
C
C     COMPUTE:  (C transpose * Y)
C
      DO 30 I=1,NV
      CTY(I)=0.0
   30 CONTINUE
      DO 40 I=1,NV
      DO 40 J=1,NE
      CTY(I)=CTY(I)+C(J,I)*Y(J)
   40 CONTINUE
C
C     CALL MATRIX INVERSION  --  ANS IN CTY
C
      CALL MLINEQ(27,27,NV,1,CTC,CTY,COND,IPVT,WORK)
C
C     UPDATE WEIGHTS
C
      DO 50 I=1,NV
      W(I)=CTY(I)
   50 CONTINUE
C
C     COMPUTE:  (C * W)
C
      DO 70 I=1,NE
      YMCW(I)=0.0
   70 CONTINUE
      DO 80 I=1,NE
      DO 80 J=I,NV
      YMCW(I)=YMCW(I)+C(I,J)*W(J)
   80 CONTINUE
C
C     COMPUTE:  (Y - C * W)
C
      DO 90 I=1,NE
      YMCW(I)=Y(I)-YMCW(I)
   90 CONTINUE
C
C     COMPUTE CHISQ:  (Y - C * W)transpose * (Y - C * W)
C
```

```
      XJAY=0.0
      DO 100 J=1,NE
      XJAY=XJAY+YMCW(J)*YMCW(J)
  100 CONTINUE
C
C     RETURN TO "RUN"
C
      RETURN
      END
      SUBROUTINE COMPG(GRAD)
C
C
C     COMPUTES THE GRADIENTS
C
C
      COMMON /UNITS/ NVID,NKID,LUN1,ICC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             ID1(30,6),ID2(250,6),ID(300,6),T(8,5)
      COMMON /RECR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSZE
C
      DIMENSION GRAD(8,5),DUM1(27),DUM4(27,27),DUM5(27,27),DUM6(27)
      DIMENSION CC(250,27)
C
      CHARACTER*1 FACTR
      CHARACTER*40 INFIL
C
      DO 40 I=1,MF
      DO 40 J=1,ML
      GRAD(I,J)=0.0
   40 CONTINUE
C
C     COMPUTE THE GRADIENT FOR EVERY LEVEL OF EACH FACTOR
C     AND FOR THE INITIAL IMPRESSION - S(0)
C
      DO 100 II=1,NF+1
      IK=II
      IF(II.EQ.(NF+1))IK=7
      KK=LEV(IK)
      DO 100 JJ=1,KK
C
C     CALCULATE PARTIAL DERIVATIVES OF C MATRIX FOR FACTOR IK, LEVEL JJ
C
      CALL COMPCC(CC,IK,JJ)
C
C     COMPUTE:  (Y transpose * partial C)
C
      DO 70 I=1,NV
      DUM1(I)=0.0
   70 CONTINUE
      DO 80 J=1,NE
      DO 80 K=1,NV
      DUM1(K)=DUM1(K)+Y(J)*CC(J,K)
   80 CONTINUE
C
C     COMPUTE: 2 * (Y transpose * partial C) * W
C
      DUMA=0.0
```

```
      DO 90 J=1,NV
      DUMA=DUMA+DUM1(J)*W(J)
   90 CONTINUE
      DUMA=DUMA*2.0
C
C     COMPUTE:  (C transpose * partial C)
C
      DO 150 I=1,NV
      DO 150 J=1,NV
      DUM4(I,J)=0.0
  150 CONTINUE
      DO 160 I=1,NV
      DO 160 J=1,NE
      DO 160 K=1,NV
      DUM4(I,K)=DUM4(I,K)+C(J,I)*CD(J,K)
  160 CONTINUE
C
C     COMPUTE:  (C transpose * partial C) +
C               (C transpose * partial C)transpose
C
      DO 230 I=1,NV
      DO 230 J=1,NV
      DUM5(I,J)=0.0
  230 CONTINUE
      DO 240 I=1,NV
      DO 240 J=1,NV
      DUM5(I,J)=DUM4(I,J)+DUM4(J,I)
  240 CONTINUE
C
C     COMPUTE:  W * [ (C t * D C) + (C t * D C)t ]
C
      DO 250 I=1,NV
      DUM6(I)=0.0
  250 CONTINUE
      DO 210 J=1,27
      DO 210 K=1,27
      DUM6(K)=DUM6(K)+W(J)*DUM5(J,K)
  210 CONTINUE
C
C     COMPUTE: W * [ (C t * D C) + (C t * D C)t ] * W t
C
      DUMB=0.0
      DO 220 J=1,27
      DUMB=DUMB+DUM6(J)*W(J)
  220 CONTINUE
C
C     COMPUTE GRADIENT FOR FACTOR IK, LEVEL JJ
C
      GRAD(IK,JJ)=DUMA-DUMB
  100 CONTINUE
C
C     RETURN TO "RUN"
C
      RETURN
      END
      SUBROUTINE COMPCD(CD,I1,J1)
C
C
C     CALCULATES THE PARTIAL DERIVATIVES OF THE C MATRIX
C     WITH RESPECT TO FACTOR "I1", LEVEL "J1"
```

```
C
C
      COMMON /UNITS/ NVIO,NKID,LUN1,ICC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NR,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ NOFLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             IO1(30,6),IO2(250,6),IO(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAD,MSTEPS,SSIZE
C
      DIMENSION CO(250,27),SP(6)
C
      CHARACTER*1 FACTR
      CHARACTER*40 INFIL
C
      DO 200 II=1,NE
      DO 160 I=1,NV
      CO(II,I)=0.0
  160 CONTINUE
      RRMAX=0.0
      RMAX=0.0
      RRMIN=1.0E+10
      RMIN=1.0E+10
      XMASK=0.0
C
C     FIND THE SCALE VALUES FOR THE VARIOUS FACTORS AT THE LEVELS
C     INDICATED BY THE EXPERIMENT (II)
C
      DO 190 I=1,NF
      SP(I)=0.0
C
C     GET THE LEVEL "IX" FOR FACTOR "I"
C
      IX=IO(II,I)
      IF(N2FLG.EQ.1)IX=IO2(II,I)
C
C     IS THIS THE FACTOR AND LEVEL OF THE PARTIAL DERIVATIVE ?
C
      IF(I.EQ.II.AND.IX.EQ.J1)GO TO 180
C
C     NO  --  SET PARTIAL DERIVATIVE TO SCALE VALUE FOR FACTOR "II",
C             LEVEL "IX"  --  COMPUTE MAX AND MIN
C
      SP(I)=S(I,IX)
      IF(S(I,IX).GT.RMAX)RMAX=S(I,IX)
      IF(S(I,IX).LT.RMIN)RMIN=S(I,IX)
      GO TO 190
C
C     YES  --  SET PARTIAL DERIVATIVE TO 1.0 FOR FACTOR "II",
C     LEVEL "IX"  --  COMPUTE MAX AND MIN
C
  180 CONTINUE
      SP(I)=1.0
      XMASK=1.0
      IF(S(I,IX).GT.RRMAX)RRMAX=S(I,IX)
      IF(S(I,IX).LT.RRMIN)RRMIN=S(I,IX)
  190 CONTINUE
C
C
C     GENERATE THE PARTIAL DERIVATIVE OF THE C MATRIX
```

```
C
C
      IW=0
      IV=0
C
C     1-WAY COMBINATIONS
C
      DO 100 I=1,NF
      IV=IV+1
C
C     IS THIS VARIABLE SELECTED ?
C
      IF(ISV(IV).EQ.0)GO TO 100
      IW=IW+1
      CO(II,IW)=SP(I)*XMASK
  100 CONTINUE
      IF(NF.LT.2)GO TO 130
C
C     2-WAY COMBINATIONS
C
      DO 110 I=1,NF-1
      DO 110 J=I+1,NF
      IV=IV+1
C
C     IS THIS VARIABLE SELECTED ?
C
      IF(ISV(IV).EQ.0)GO TO 110
      IW=IW+1
      CO(II,IW)=SP(I)*SP(J)*XMASK
  110 CONTINUE
      IF(NF.LT.3)GO TO 130
C
C     3-WAY COMBINATIONS
C
      DO 120 I=1,NF-2
      DO 120 J=I+1,NF-1
      DO 120 K=J+1,NF
      IV=IV+1
C
C     IS THIS VARIABLE SELECTED ?
C
      IF(ISV(IV).EQ.0)GO TO 120
      IW=IW+1
      CO(II,IW)=SP(I)*SP(J)*SP(K)*XMASK
  120 CONTINUE
C
C     INITIAL IMPRESSION - S(0)
C
  130 CONTINUE
      IV=IV+1
C
C     IS THIS VARIABLE SELECTED ?
C         IF YES --  SET THE DERIVATIVE TO 1.0
C         IF NO  --  SET THE DERIVATIVE TO 0.0
C
      IF(ISV(IV).EQ.0)GO TO 140
      IW=IW+1
      DER=0.0
      IF(I1.EQ.7)DER=1.0
      CO(II,IW)=DER
```

```
C
C     RANGE VARIABLE - S(R)
C
  140 CONTINUE
      IV=IV+1
C
C     WAS THIS VARIABLE SELECTED ?
C         IF NO    --  SET THE DERIVATIVE TO 0.0
C         IF YES   --  SET DERIVATIVE TO -1.0 OR 1.0 IF MIN
C                      OR MAX WAS TRUE MIN OR TRUE MAX
C
      IF(ISV(IV).EQ.0)GO TO 200
      IW=IW+1
      R1=0.0
      R2=0.0
      IF(RRMAX.GE.RMAX)R1=1.0
      IF(RRMIN.LE.RMIN)R2=1.0
      CO(II,IW)=R1-R2
  200 CONTINUE
C
C     RETURN TO "COMPG"
C
      RETURN
      END
```

```
      SUBROUTINE RES
C
C
C     DISPLAYS THE RESULTS OF THE ITERATION(S) AND LINE SEARCH
C
C        SCREEN 1:  DISPLAYS INITIAL AND NEW SCALE VALUES
C        SCREEN 2:  DISPLAYS WEIGHTS
C
C
      COMMON /UNITS/ NVIO,NKIO,LUN1,ICC,INFIL
      COMMON /PARAM/ NE,N1,N2,NN,NV,NW,MF,ML,NF,LEV(8),FACTR(8)
      COMMON /FLAGS/ N0FLG,N2FLG,NNFLG
      COMMON /RAW/ DAT1(30),DAT2(250),DAT(300),S(8,5),ISC(8,5),
     X             ID1(30,6),ID2(250,6),ID(300,6),T(8,5)
      COMMON /REGR/ Y(250),C(250,27),W(27),ISV(27),CHISQ,PCHISQ
      COMMON /RUNRES/ NITER,NSTEPS,NBAC,MSTEPS,SSIZE
C
      CHARACTER*1 ANS,FACTR
      CHARACTER*12 BLK,IN2
      CHARACTER*14 VAREM(27)
      CHARACTER*40 INFIL
      CHARACTER*80 IN
      CHARACTER*120 IN1
C
      DATA BLK/'           '/
C
      CALL SMG$ERASE_DISPLAY(NVIO)
C
C     WAS DATA MODEL SELECTED ?
C
      IF(N0FLG.EQ.0)GO TO 200
C
C     DISPLAY SCALE VALUES AND "TEST" SCALE VALUES
C
      DO 10 I=1,5
      ENCODE(12,908,IN2)I
      J=(I-1)*12+12
      CALL SMG$PLT_CHARS(NVIO,IN2,1,J)
   10 CONTINUE
      DO 20 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      IL=(I-1)*3+2
      K=LEV(IK)
      NCH=K*12+7
      ENCODE(NCH,909,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIO,IN1(1:67),IL,5)
      ENCODE(NCH,910,IN)(T(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVIO,IN1(1:67),IL+1,5)
   20 CONTINUE
C
C     DISPLAY THE PARAMETERS FROM THE ITERATION(S)
C
      CALL SMG$PLT_CHARS(NVIO,'#Iterations:',20,5)
      CALL SMG$PLT_CHARS(NVIO,'#Steps:',20,27)
      CALL SMG$PLT_CHARS(NVIO,'(Max=  )',20,38)
      CALL SMG$PLT_CHARS(NVIO,'Step Size:',20,54)
      CALL SMG$PLT_CHARS(NVIO,'Current CHISQ:',21,11)
```

```
      CALL SMG$PLT_CHARS(NVID,'Previous CHISQ:',21,45)
      ENCODE(2,902,IN2)NITER
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,17)
      IF(SSZE.LT.1.0)GO TO 60
      ENCODE(2,902,IN2)NSTEPS
      GO TO 90
   60 CONTINUE
      ENCODE(2,902,IN2)NBAD
   90 CONTINUE
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,34)
      ENCODE(2,902,IN2)MSTEPS
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,43)
      ENCODE(12,911,IN2)SSZE
      CALL SMG$PLT_CHARS(NVID,IN2,20,64)
      ENCODE(10,912,IN2)CHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,25)
      ENCODE(10,912,IN2)PCHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,60)
      CALL SMG$DRAW_LINE(NVID,19,5,19,75)
      CALL SMG$DRAW_LINE(NVID,22,5,22,75)
C
C     SET THE SCALE VALUES TO THE "TEST" SCALE VALUES ?
C
      IN='Set Scale Values to Test Results (Y/N)? '
      CALL SMG$PLT_CHARS(NVID,IN(1:40),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
      IYES=0
C
C     <CR> ?  --  DONT SET THE SCALE VALLES
C
      IF(NC.EQ.0)GO TO 50
C
C     ANSWER IS NOT YES  --  DONT SET SCALE VALUES
C
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 50
C
C
C     ANSWER IS YES  --  SET THE SCALE VALUES TO THE "TEST" SCALE VALLES
C
C     REDISPLAY THE NEW SCALE VALUES AND ZERO OUT THE "TEST" SCALE VALLES
C
C
      IYES=1
      DO 30 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      K=LEV(IK)
      IL=(I-1)*3+2
      NCH=K*12+7
      DO 40 J=1,K
      S(IK,J)=T(IK,J)
      T(IK,J)=0.0
   40 CONTINUE
      ENCODE(NCH,909,IN)FACTR(IK),(S(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN1(1:67),IL,5)
      ENCODE(NCH,910,IN)(T(IK,J),J=1,K)
      IN1=IN(1:NCH)//BLK//BLK//BLK//BLK
      CALL SMG$PLT_CHARS(NVID,IN1(1:67),IL+1,5)
```

```
   30 CONTINUE
C
C     PAUSE FOR USER TO READ SCREEN 1
C
   50 CONTINUE
      IN='***      Press any KEY for Next Screen      ***'
      CALL SMG$PLT_CHARS(NVID,IN(1:45),23,17)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,,NC,,NVID)
      CALL SMG$ERASE_DISPLAY(NVID)
C
C     BEGIN SCREEN 2 DISPLAY OF WEIGHTS AND VARIABLE FORMS
C
      CALL SMG$DRAW_LINE(NVID,2,5,2,76)
      CALL SMG$DRAW_LINE(NVID,18,5,18,76)
      CALL SMG$DRAW_LINE(NVID,22,5,22,76)
      CALL SMG$DRAW_LINE(NVID,2,40,18,40)
      CALL SMG$DRAW_LINE(NVID,2,41,18,41)
      IN='Var#  ** Weight **  Variable Form '
      IN1=IN(1:34)//'     '//IN(1:34)
      CALL SMG$PLT_CHARS(NVID,IN1(1:72),1,5)
C
C     ENCODE AND DISPLAY WEIGHTS
C
      IV=0
      DO 80 I=1,NW
      ENCODE(3,900,IN)I
      IN1='                '
      IF(ISV(I).EQ.0)GO TO 65
      IV=IV+1
      ENCODE(12,901,IN1)W(IV)
   65 CONTINUE
      IL=I+2
      ICI=5
      IC2=11
      IF(I.LE.14)GO TO 70
      IL=IL-14
      ICI=43
      IC2=49
   70 CONTINUE
      CALL SMG$PLT_CHARS(NVID,IN(1:3),IL,ICI)
      CALL SMG$PLT_CHARS(NVID,IN1(1:12),IL,IC2)
   80 CONTINUE
C
C     ENCODE 1-WAY COMBINATIONS OF VARIABLE FORM
C
      IV=0
      DO 100 I=1,NF
      IV=IV+1
      ENCODE(14,903,VARFM(IV))FACTR(I)
  100 CONTINUE
      IF(NF.LT.2)GO TO 130
C
C     ENCODE 2-WAY COMBINATIONS OF VARIABLE FORM
C
      DO 110 I=1,NF-1
      DO 110 J=I+1,NF
      IV=IV+1
      ENCODE(14,904,VARFM(IV))FACTR(I),FACTR(J)
  110 CONTINUE
      IF(NF.LT.3)GO TO 130
```

```
C
C     ENCODE 3-WAY COMBINATIONS OF VARIABLE FORM
C
      DO 120 I=1,NF-2
      DO 120 J=I+1,NF-1
      DO 120 K=J+1,NF
      IV=IV+1
      ENCODE(14,505,VARFM(IV))FACTR(I),FACTR(J),FACTR(K)
  120 CONTINUE
C
C     ENCODE INITIAL IMPRESSION - S(0)
C
  130 CONTINUE
      IV=IV+1
      ENCODE(14,506,VARFM(IV))
C
C     ENCODE RANGE VARIABLE - S(R)
C
      IV=IV+1
      ENCODE(14,507,VARFM(IV))
C
C     DISPLAY THE VARIABLE FORMS JUST ENCODED
C
      DO 140 I=1,NW
      IL=I+2
      IC=25
      IF(I.LE.14)GO TO 145
      IL=IL-14
      IC=63
  145 CONTINUE
      CALL SMG$PLT_CHARS(NVID,VARFM(I),IL,IC)
  140 CONTINUE
C
C     DISPLAY PARAMETERS FROM ITERATION(S)
C
      CALL SMG$PLT_CHARS(NVID,'*Iterations:',20,5)
      CALL SMG$PLT_CHARS(NVID,'*Steps:',20,27)
      CALL SMG$PLT_CHARS(NVID,'(Max=  )',20,38)
      CALL SMG$PLT_CHARS(NVID,'Step Size:',20,54)
      CALL SMG$PLT_CHARS(NVID,'Current CHISQ:',21,11)
      CALL SMG$PLT_CHARS(NVID,'Previous CHISQ:',21,45)
      ENCODE(2,902,IN2)NITER
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,17)
      IF(SSZE.LT.1.0)GO TO 160
      ENCODE(2,902,IN2)NSTEPS
      GO TO 190
  160 CONTINUE
      ENCODE(2,902,IN2)NBAD
  190 CONTINUE
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,34)
      ENCODE(2,902,IN2)MSTEPS
      CALL SMG$PLT_CHARS(NVID,IN2(1:2),20,43)
      ENCODE(12,911,IN2)SSZE
      CALL SMG$PLT_CHARS(NVID,IN2,20,64)
      ENCODE(10,912,IN2)CHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,25)
      ENCODE(10,912,IN2)PCHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,60)
C
C     HAS THE SCALE VALUES BEEN SET TO THE "TEST" SCALE VALUES ?
```

```
C
      IF(IYES.EQ.1)GO TO 250
      IN='Set Scale Values to Test Results (Y/N)? '
      CALL SMG$PLT_CHARS(NVID,IN(1:40),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
C
C     <CR> ?" --  DONT SET THE SCALE VALUES TO THE TEST SCALES
C
      IF(NC.EQ.0)GO TO 240
C
C     ANSWER IS NO ?  --  DONT SET THE SCALE VALUES
C
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 240
C
C     ANSWER IS YES  --  SET SCALE VALUES TO "TEST" SCALE VALUES
C
      DO 230 I=1,NF+1
      IK=I
      IF(I.GT.NF)IK=7
      K=LEV(IK)
      DO 230 J=1,K
      S(IK,J)=T(IK,J)
      T(IK,J)=0.0
  230 CONTINUE
      GO TO 250
C
C
C     DONT SET THE SCALE VALUES TO THE "TEST" SCALE VALUES
C
C     RESET THE VALUE OF CHISQ AND REDISPLAY IT
C
C
  240 CONTINUE
      CHISQ=PCHISQ
      ENCODE(10,912,IN2)CHISQ
      CALL SMG$PLT_CHARS(NVID,IN2(1:10),21,25)
C
C     SAVE THE RESULTS ON A FILE ?
C
  250 CONTINUE
      IN='Save Results in a File (Y/N)? '
      CALL SMG$PLT_CHARS(NVID,IN(1:28),23,5)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
      CALL SMG$ERASE_LINE(NVID,23,1)
C
C     <CR> ?  --  DONT SAVE THE RESULTS
C
      IF(NC.EQ.0)GO TO 210
C
C     ANSWER IS NO  --  DONT SAVE THE RESULTS
C
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')GO TO 210
C
C     ANSWER IS YES  --  SAVE THE RESULTS
C
C     IS THE OUTPUT FILE ALREADY OPENED ?
C
      IF(IDO.EQ.1)GO TO 260
C
```

```
C     NO  --  OPEN THE FILE AND SET FLAG
C
      OPEN(UNIT=LUN1,NAME='RESULTS.STF',TYPE='NEW')
      IOC=1
C
C     WRITE THE RESULTS
C
  260 CONTINUE
      WRITE(LUN1,913)INFIL
      WRITE(LUN1,914)N1,N2,NF,NN,NF,(LEV(I),I=1,NF)
      WRITE(LUN1,915)NE,NV,NW
      WRITE(LUN1,921)
      IV=0
      DO 270 I=1,NW
      IF(ISV(I).EQ.1)GO TO 275
      WRITE(LUN1,916)I,VAREM(1)
      GO TO 270
  275 CONTINUE
      IV=IV+1
      WRITE(LUN1,917)I,W(IV),VAREM(I)
  270 CONTINUE
      WRITE(LUN1,920)
      DO 280 I=1,NF
      WRITE(LUN1,918)(S(I,J),J=1,NL)
  280 CONTINUE
      WRITE(LUN1,919)CHISQ
      GO TO 210
C
C     NO DATA MODEL SELECTED - SEND MESSAGE AND EXIT
C
  200 CONTINUE
      CALL BEEP
      IN='No Data Model Selected'
      CALL SMG$PLT_CHARS(NVID,IN(1:22),10,30)
      IN='See Option 1 in Main Menu'
      CALL SMG$PLT_CHARS(NVID,IN(1:25),12,30)
      CALL SMG$SET_CURSOR_ABS(NVID,16,30)
      CALL SMG$READ_STRING(NKID,ANS,
     X    'Press any KEY to Return',1,,,,NC,,NVID)
      GO TO 220
C
C     PAUSE FOR USER TO READ SCREEN 2
C
  210 CONTINUE
      IN='***      Press any KEY to Return      ***'
      CALL SMG$PLT_CHARS(NVID,IN(1:39),23,20)
      CALL SMG$READ_STRING(NKID,ANS,,1,,,,NC,,NVID)
C
C     RETURN TO MAIN MENU
C
  220 CONTINUE
      RETURN
  900 FORMAT(I3)
  901 FORMAT(E12.6)
  902 FORMAT(I2)
  903 FORMAT('S(',A1,')              ')
  904 FORMAT('S(',A1,')*S(',A1,')      ')
  905 FORMAT('S(',A1,')*S(',A1,')*S(',A1,')')
  906 FORMAT('S(C)             ')
  907 FORMAT('S(R)             ')
```

```
908 FORMAT('     Level ',I1)
909 FORMAT('S(',A1,')   ',5F12.?)
910 FORMAT('Test    ',5F12.3)
911 FORMAT(E12.6)
912 FORMAT(F10.4)
913 FORMAT(1X,'FileName:',A40)
914 FORMAT(1X,'#1-Ways:',I2,5X,'#2-Ways:',I3,5X,'#',I1,'-Ways:',
   X    I3/1X,'#Factors:',I1,3X,'Levels:',6I2)
915 FORMAT(1X,'#Experiments:',I3,5X,'#Variables:',I2,'  of  ',I2)
916 FORMAT(1X,I3,23X,A14)
917 FORMAT(1X,I3,4X,E16.8,3X,A14)
918 FORMAT(1X,5F14.4)
919 FORMAT(/1X,'CHISQ:',E16.8////)
920 FORMAT(/1X,'Scale Values')
921 FORMAT(/1X,'Var#   ***  Weight  ***   Variable Form ')
    END
```

Program Name: STF10

Language: BASIC

Machine: Compaq Personal Computer (IBM compatible)

Purpose: This program is a utility that supports the data collection effort. Data required to print questionnaires is entered using this program.

```
10 '    THIS PROGRAM GETS DATA ABOUT A NODE IN A HIERARCHY AND
20 '    THE LEVELS OF THAT NODE, AND WRITES THE DATA TO A FILE
30 '
40 '    PROGRAM-ID.  STF-10.
50 '    DATE-WRITTEN.  FEBRUARY 1985.
60 '    INSTALLATION.  ALPHATECH, INC.
70 '    ACCOUNT.  J187-01.
80 '
90 GOSUB 160                                        'housekeeping
100 GOSUB 470                                       'display node screen, etc
110 GOSUB 1290                                       'write or rewrite file
120 GOSUB 1460                                      'end of job
130 '
140 '    Determine drive questionnaire file is/willbe on (and other things).
150 '
160 CLS : LOCATE 8,4 : PRINT "STF-10."
170 LOCATE 10,4 : PRINT "This program accepts data about a node,and writes that
data to a file."
180 LOCATE 12,4 : PRINT "Please enter drive that file is/will be  on."
190 LOCATE 12, 50 : INPUT " ", DRIVE$
200 IF NOT (DRIVE$ = "A" OR DRIVE$ = "a" OR DRIVE$ = "B" OR DRIVE$ = "b")
      THEN BEEP : PRINT "Drive must be a or b or A or B." : GOTO 190
210 OPTION BASE 1
220 DIM PROMPTS$(12)
230 DATA "NODE INFORMATION"
240 DATA "Node number:"
250 DATA "Node name:"
260 DATA "Short Form:"
270 DATA "Node Definition:"
280 DATA "Number of Factors:"
290 DATA "Measure of Node:"
300 DATA "Number of Levels:"
310 DATA "Level Definitions:"
320 DATA "Level"
330 DATA ":  Def'n:"
340 DATA "Short-Form:"
350 FOR I = 1 TO 12 : READ PROMPTS$(I) : NEXT I
360 DIM PLOCS(12,2)
370 DATA 1,30,2,8,3,10,4,9,6,4,9,2,10,4,11,3,13,2,15,2,15,9,19,7
380 FOR I = 1 TO 12 : FOR J = 1 TO 2 : READ PLOCS (I, J) : NEXT J : NEXT I
390 DIM RROWS(9)
400 DATA 3,4,5,6,7,8,9,10,11
410 FOR I = 1 TO 9 : READ RROWS(I) : NEXT I
420 S1$ = STRING$(55,"-") : S2$ = STRING$(10,"-") : S3$ = STRING$(35,"-")
430 DIM NODEFM$(2),NODEDEFN$(3), LEVELDEFN$(5,3), SHORTFM$(5,2)
440 K1$ = "quit"+ CHR$(13) : K2$ = "next"+CHR$(13)
450 KEY 1,K1$ : KEY 2,K2$
460 RETURN
```

```
470 '
480 '     Node Screen Subroutine
490 '
500 '      The following lines display the screen, with promots.
510 '
520 CLS
530 FOR I = 1 TO 12 : LOCATE PLOCS(I,1),PLOCS(I,2) : PRINT PROMPTS$ (I) :
    NEXT I
540 '
550 '      the following lines get data about the node
560 '
570 LOCATE 2,21 : INPUT " ",NODENUM$            'get number of node
580 IF NOT (LEN(NODENUM$) = 7)
       THEN BEEP : LOCATE 23,4 :
            PRINT "Node number must have 7 digits" : GOTO 570
590 GOSUB 1010                                  'get existing data
600 IF EXISTING = 0 THEN LOCATE 3,22 : PRINT S3$ : LOCATE 4,22 : PRINT S2$ :
       LOCATE 5,22 : PRINT S2$ : LOCATE 6,22 : PRINT S1$ : LOCATE 7,22 :
       PRINT S1$ : LOCATE 8,22 : PRINT S1$
610 FOR I = 1 TO 9
620      LOCATE RROWS(I),21 : LINE INPUT " ", STUFF$
630      IF LEN(STUFF$) = 0 THEN GOTO 780
640      IF STUFF$ = "quit" THEN GOTO 1000
650      IF STUFF$ = "next" THEN GOTO 790
655      IF STUFF$ = "." THEN STUFF$ = ""
660      ON I GOTO 670, 680, 690, 700, 710, 720, 730, 750, 760
670         NODENAME$ = STUFF$ : GOTO 780
680         NODEFM$(1) = STUFF$ : GOTO 780
690         NODEFM$(2) = STUFF$ : GOTO 780
700         NODEDEFN$(1) = STUFF$ : GOTO 780
710         NODEDEFN$(2) = STUFF$ : GOTO 780
720         NODEDEFN$(3) = STUFF$ : GOTO 780
730         NFACS = VAL(STUFF$)
740         IF (NFACS ) 5) THEN BEEP : LOCATE 23,10 :
               PRINT " number of factors must be not greater than 5." : GOTO 620
745         GOTO 780
750         NODEMSRE$ = STUFF$ : GOTO 780
760         NODELVLS = VAL(STUFF$)
770         IF (NODELVLS ) 5) THEN BEEP : LOCATE 23,10 :
            PRINT " number of levels must not be greater than 5." : GOTO 620
```

```
780 NEXT I
790 FOR I = 1 TO NODELVLS
800     LOCATE 15,7 : PRINT STR$(I)
810     IF EXISTING = 0 THEN LOCATE 15,20 : PRINT S1$ : LOCATE 16,20 :
           PRINT S1$ : LOCATE 17,20 : PRINT S1$ : LOCATE 19,20 : PRINT S2$ :
           LOCATE 20,20 : PRINT S2$ : GOTO 860
820     XROW = 15
830     FOR J = 1 TO 3 : LOCATE XROW,20 : PRINT LEVELDEFN$(I,J) : XROW = XROW + 1 :
 :           NEXT J
840     XROW = 19
850     FOR J = 1 TO 2 : LOCATE XROW,20 : PRINT SHORTFM$(I,J) : XROW = XROW + 1 :
           NEXT J
860     XROW = 15
870     FOR J = 1 TO 3
880         LOCATE XROW,19 : LINE INPUT " ",STUFF$ : XROW = XROW + 1
885         IF LEN(STUFF$) = 0 THEN GOTO 920
890         IF STUFF$ = "quit" THEN GOTO 1000
900         IF STUFF$ = "next" THEN GOTO 990
905         IF STUFF$ = "." THEN STUFF$ = ""
910         LEVELDEFN$(I,J) = STUFF$
920     NEXT J
930     XROW = 19
940     FOR J = 1 TO 2
950         LOCATE XROW,19 : LINE INPUT " ",STUFF$ : XROW = XROW + 1
951         IF LEN(STUFF$) = 0 THEN GOTO 980
955         IF STUFF$ = "quit" THEN GOTO 1000
960         IF STUFF$ = "next" THEN GOTO 990
965         IF STUFF$ = "." THEN STUFF$ = ""
970         SHORTFM$(I,J) = STUFF$
980     NEXT J
990 NEXT I
1000 RETURN
1010 '
1020 '  read data in existing file, if any
1030 '
1040 NODEFILE$ = DRIVE$+":N"+NODENUM$
1050 ON ERROR GOTO 1390
1060 OPEN NODEFILE$ FOR INPUT AS #1
1070 EXISTING = -1
1080 INPUT #1, NODENUM$, NODENAME$, NODEFM$(1), NODEFM$(2), NODEDEFN$(1),
                NODEDEFN$(2), NODEDEFN$(3), NFACS, NODEMSRE$, NODELVLS
1090 FOR I = 1 TO NODELVLS
1100     INPUT #1, LEVLNUM,LEVELDEFN$(I,1), LEVELDEFN$(I,2), LEVELDEFN$(I,3),
                   SHORTFM$(I,1), SHORTFM$(I,2)
```

```
1110 NEXT I
1120 CLOSE #1
1130 LOCATE 3,22 : PRINT NODENAME$                    'print node data
1140 LOCATE 4,22 : PRINT NODEFM$(1)
1150 LOCATE 5,22 : PRINT NODEFM$(2)
1160 LOCATE 6,22 : PRINT NODEDEFN$(1)
1170 LOCATE 7,22 : PRINT NODEDEFN$(2)
1180 LOCATE 8,22 : PRINT NODEDEFN$(3)
1190 LOCATE 9,21 : PRINT NFACS
1200 LOCATE 10,22 : PRINT NODEMSRE$
1210 LOCATE 11,21 : PRINT NODELVLS
1220 LOCATE 15,8 : PRINT "1"
1230 LOCATE 15,20 : PRINT LEVELDEFN$(1,1)
1240 LOCATE 16,20 : PRINT LEVELDEFN$(1,2)
1250 LOCATE 17,20 : PRINT LEVELDEFN$(1,3)
1260 LOCATE 19,20 : PRINT SHORTFM$(1,1)
1270 LOCATE 20,20 : PRINT SHORTFM$(1,2)
1275 LOCATE 22,10 : PRINT "Enter . (full stop) to delete a field."
1280 RETURN
1290 '
1300 ' write records and close files
1310 '
1320 OPEN NODEFILE$ FOR OUTPUT AS #1
1330 WRITE #1, NODENUM$, NODENAME$, NODEFM$(1), NODEFM$(2), NODEDEFN$(1),
               NODEDEFN$(2), NODEDEFN$(3), NFACS, NODEMSRE$, NODELVLS
1340 FOR I = 1 TO NODELVLS
1350     WRITE #1, I,LEVELDEFN$(I,1), LEVELDEFN$(I,2), LEVELDEFN$(I,3),
                   SHORTFM$(I,1), SHORTFM$(I,2)
1360 NEXT I
1370 CLOSE #1
1380 RETURN
1390 '
1400 '    error handling
1410 '
1420 IF ERL = 1060 AND ERR = 53 THEN
        EXISTING = 0 : RESUME 1280
1430 LOCATE 20,10: PRINT "erl is ", STR$(ERL), "err is ", STR$(ERR)
1440 LOCATE 21,10 : PRINT "END OF PROGRAM IN ERROR HANDLING"
1450 END
1460 '
1470 ' end of program
1480 '
1490 CLS
1500 LOCATE 10,10 : PRINT "SUCCESSFUL END OF PROGRAM"
1510 END
```

Program Name: STF30

Language: BASIC

Machine: Compaq Person Computer (IBM compatible)

Purpose: This program is a utility that supports the data collection effort. It prints the questionnaires required for data collection.

```
10 '     THIS PROGRAM READS FILES AND PRINTS A QUESTIONNAIRE.
30 '
40 '     PROGRAM-ID.   STF-30.
50 '     DATE-WRITTEN.    MARCH, 1985
60 '     INSTALLATION.    ALPHATECH, INC.
70 '     ACCOUNT.  J187-01.
80 '
90 CLEAR ,,8192
100 GOSUB 230                                          'Get node data
110 FOR I = 1 TO NFACS                                 'Get factor data
120      GOSUB 430
130      GOSUB 550
140 NEXT I
150 GOSUB 660                                         'Print one ways
160 GOSUB 1090                                        'print two ways
170 FOR I = 1 TO NFACS
180      GOSUB 1740                                   'reduce dimensions
190 NEXT I
200 IF NFACS = 3 THEN GOSUB 1880                      'print three-ways
210 IF NFACS = 4 THEN GOSUB 2020                      'print four-ways
220 GOSUB 2420                                        'End of job
230 '
240 'prompt for node number and read nodefile.
250 '
260 PCOUNT = 1 : ON ERROR GOTO 2340 : CLS
270 OPTION BASE 1
280 DIM NODEFM$(2),NODEDEFN$(3),FACNAME$(5),FACFM$(5,2),FACDEFN$(5,3),
      FACMSRE$(5),FACLEVELS(5),LEVELDEF$(5,5,3),SHORTFM$(5,5,2)
290 LOCATE 8,4 : PRINT "STF-30."
300 LOCATE 10,4 : PRINT "This program prints a questionnaire."
310 LOCATE 12,4 : PRINT "Please enter drive files are on."
320 LOCATE 14,4 : PRINT "Please enter (output) node number."
330 LOCATE 12,45 : INPUT " ",DRIVE$
340 IF NOT (DRIVE$ = "a" OR DRIVE$ = "b" OR DRIVE$ = "A" OR DRIVE$ = "B")
    THEN BEEP : LOCATE 20,10 : PRINT "Drive must be A or B or a or b." :
         GOTO 330
350 LOCATE 14,45 : INPUT " ",NODENUM$
360 IF NOT LEN(NODENUM$) = 7 THEN BEEP : LOCATE 20,10 :
    PRINT "Node number must have seven digits." : GOTO 350
370 NODEFILE$ = DRIVE$ + ":N" + NODENUM$
380 OPEN NODEFILE$ FOR INPUT AS #1
390 INPUT #1, NODENUM$, NODENAME$, NODEFM$(1), NODEFM$(2), NODEDEFN$(1),
              NODEDEFN$(2), NODEDEFN$(3), NFACS, NODEMSRE$
400 CLOSE #1
410 ASTERISK$ = " " + STRING$(78,"*")
420 RETURN
430 '
440 'generate factor number.
450 '
460 IF NODENUM$ = "0000000" THEN
       FACNUM$ = RIGHT$(STR$(I),1) + "000000" : RETURN
470 REST$ = NODENUM$
480 FOR J = 1 TO 7
490     DIGIT$ = LEFT$(REST$,1)
500     REST$ = RIGHT$(REST$,(7-J))
510     IF DIGIT$ = "0" THEN GOTO 530
520 NEXT J
530 FACNUM$ = LEFT$(NODENUM$,(J-1)) + RIGHT$(STR$(I),1)
                                    + RIGHT$(NODENUM$,(7-J))
540 RETURN
```

```
550 '
560 'read factor node file
570 '
580 FACFILE$ = DRIVE$ + ":N" + FACNUM$
590 OPEN FACFILE$ FOR INPUT AS #1
600 INPUT #1,FILLER$,FACNAME$(I),FACFM$(I,1),FACFM$(I,2),FACDEFN$(I,1),
          FACDEFN$(I,2),FACDEFN$(I,3), FILLER,FACMSRE$(I),FACLEVELS(I)
610 FOR J = 1 TO FACLEVELS(I)
620     INPUT #1,FILLER,LEVELDEF$(I,J,1),LEVELDEF$(I,J,2),LEVELDEF$(I,J,3),
                SHORTFM$(I,J,1),SHORTFM$(I,J,2)
630 NEXT J
640 CLOSE #1
650 RETURN
660 '
670 'print one-ways
680 '
690 LPRINT CHR$(27);CHR$(49) :
    LCOUNT = 0 : TWOWAYS = 0 : THREEWAY = 0 : FOURWAY = 0
700 HLINE1$ = STRING$(65," ") + "Q:" + NODENUM$
710 HLINE2$ = STRING$(65," ") + "r:  __________" : GOSUB 2270
720 PLINE$ = "In the following questions you are given a single piece of informa
tion." : GOSUB 2210
730 PLINE$ = "For each question please give your best estimate of " +
              NODENAME$  + "." : GOSUB 2210
740 GOSUB 2210 : PLINE$ =  "Please respond " + NODEMSRE$ : GOSUB 2210
760 FOR I = 1 TO NFACS
770    IF LCOUNT ) (65 - (11 + FACLEVELS(I)*5)) THEN GOSUB 2270
780   GOSUB 2210 : PLINE$ = "Consider levels of this factor: " :
      GOSUB 2210 : GOSUB 2210
790    FOR J = 1 TO 3
800        PLINE$ = STRING$(15," ") + FACDEFN$(I,J) : GOSUB 2210
810    NEXT J : GOSUB 2210
820    PLINE$ = STRING$(79," ")
830    MID$(PLINE$,10) = "LEVEL"
840    MID$(PLINE$,38) = "(Short Form)"
850    MID$(PLINE$,65) = "OUTPUT"
860    GOSUB 2210 : GOSUB 2210
870    GOSUB 930
880 NEXT I
890 RETURN
900 '
910 'print factor levels and response lines
920 '
930 FOR J = 1 TO FACLEVELS(I)
940     GOSUB 2210
950     FOR K = 1 TO 3
960         PLINE$ = STRING$(79," ")
970         MID$(PLINE$,1,40) = LEVELDEF$(I,J,K)
990         IF NOT (K = 1) THEN MID$(PLINE$,42) = SHORTFM$(I,J,(K-1))
1000          IF K = 3 THEN MID$(PLINE$,58) = " _________________"
1010          GOSUB 2210
1020     NEXT K
1030    GOSUB 2210
1040 NEXT J
1050 GOSUB 2210 : PLINE$ = ASTERISK$ : GOSUB 2210 : GOSUB 2210
1060 RETURN
```

```
1070 '
1080 ' print two ways (factor with most levels goes down the page)
1090 '
1100 TWOWAYS = +1 : NWAY$ = "two"
1110 K = 0
1120 '
1130 '
1140 '
1150 FOR I = (K+1) TO (NFACS-1)
1160    FOR J = (I+1) TO NFACS
1170       IF FACLEVELS(I) ) FACLEVELS(J) THEN II = I : JJ = J
                                          ELSE II = J : JJ = I
1180          GOSUB 1220
1190    NEXT J
1200 NEXT I
1210 RETURN
1220 '
1230 IF ((LCOUNT ) (65 - (10+FACLEVELS(II)*5))) OR (TWOWAYS = +1 ))
        THEN GOSUB 2270
1240 IF (TWOWAYS = +1) THEN GOSUB 2120
1250 GOSUB 1310                                    'print text headers across
1260 FOR RESPBLOCK = 1 TO FACLEVELS(II)
1270    GOSUB 1570                                 'print response block
1280 NEXT RESPBLOCK
1290 IF ((THREEWAY = +1) OR (FOURWAY = +1)) THEN GOSUB 2210 :
        PLINE$ = ASTERISK$ : GOSUB 2210
1300 RETURN
1310 '
1320 'print text headers across
1330 '
1340 GOSUB 2210 : GOSUB 2210
1350 IF (FOURWAY = +1) THEN PLINE$ = CHR$(31)+FACFM$(L,1)+" "+FACFM$(L,2)+
   CHR$(30)+" is FIXED at "+CHR$(31)+SHORTFM$(L,LL,1)+" "+SHORTFM$(L,LL,2)+
   CHR$(30)
1360 GOSUB 2210 : GOSUB 2210
1370 PLINE$ = STRING$(30," ") + FACNAME$(JJ) + ":" : GOSUB 2210
1380 PLINE$ = STRING$(30," ") + STRING$(LEN(FACNAME$(JJ)),"-") : GOSUB 2210
1390 PLINE$=STRING$(79," ")
1400 FOR ROW = 1 TO 2
1410    FOR COL = 1 TO FACLEVELS(JJ)
1420       MID$(PLINE$, (31+11*(COL-1)),10) = SHORTFM$(JJ,COL,ROW)
1430    NEXT COL
1440    GOSUB 2210
1450    PLINE$ = STRING$(79," ")
1460 NEXT ROW
1470 '
1480 FOR ROW = 1 TO 2
1490    IF ((THREEWAY = +1) OR (FOURWAY = +1))
           THEN MID$(PLINE$,1,10) = FACFM$(K,ROW)
1500    MID$(PLINE$,12,10) = FACFM$(II,ROW)
1510    GOSUB 2210
1520    PLINE$ = STRING$(79," ")
1530 NEXT ROW
1540 IF ((THREEWAY = +1) OR (FOURWAY = +1)) THEN MID$(PLINE$,1) = "------ ---
1550 MID$(PLINE$,12) = "----------" : GOSUB 2210
1560 RETURN
```

```
1570 '
1580 'print response block
1590 '
1600 GOSUB 2210
1650 FOR ROW = 1 TO 2
1660    PLINE$ = STRING$(79," ")
1670    IF ((THREEWAY = +1) OR (FOURWAY = +1))
            THEN MID$(PLINE$,1,10) = SHORTFM$(K,KK,ROW)
1680    MID$(PLINE$,12,10) = SHORTFM$(II,RESPBLOCK,ROW)
1700    IF ROW = 2 THEN FOR COL = 1 TO FACLEVELS(JJ) :
                    MID$(PLINE$,(30+11*(COL-1)),10) = "__________" : NEXT COL
1710    GOSUB 2210
1720 NEXT ROW
1725 GOSUB 2210
1730 RETURN
1740 '
1750 ' reduce dimensions : five factor levels to three
1760 '
1770 IF FACLEVELS(I) < 5 THEN RETURN
1780 FOR J1 = 3 TO 5 STEP 2 : GOSUB 1810 : NEXT J1
1790 FACLEVELS(I) = 3
1800 RETURN
1810 '
1820 'for 5 factor levels shift 3 to 2 and 5 to 3
1830 '
1840 IF J1 = 3 THEN J2 = 2 ELSE J2 = 3
1850 FOR K = 1 TO 3 : LEVELDEF$(I,J2,K) = LEVELDEF$(I,J1,K) : NEXT K
1860 FOR K = 1 TO 2 : SHORTFM$(I,J2,K) = SHORTFM$(I,J1,K) : NEXT K
1870 RETURN
1880 '
1890 ' print three-ways
1900 '
1910 TWOWAYS = 0 : THREEWAY = +1 : NWAY$ = "three" : GOSUB 2270 : GOSUB 2120
1920 L = 0
1930 '
1940 '
1950 '
1960 FOR K = (L+1) TO (NFACS-2)
1970    FOR KK = 1 TO FACLEVELS(K)
1980        GOSUB 1150
1990    NEXT KK
2000 NEXT K
2010 RETURN
2020 '
2030 ' print fourways
2040 '
2050 TWOWAYS = 0 : FOURWAY = +1 : NWAY$ = "four" : GOSUB 2270 : GOSUB 2120
2060 FOR L = 1 TO (NFACS-3)
2070     FOR LL = 1 TO FACLEVELS(L)
2080         GOSUB 1960
2090     NEXT LL
2100 NEXT L
2110 RETURN
```

```
2120 '
2130 ' print n-way heading
2140 '
2150 PLINE$ = "On these pages you are given " + NWAY$ +
             " pieces of information together."  : GOSUB 2210
2160 PLINE$ = "For each question please give your best estimate of " +
               NODENAME$ + "." : GOSUB 2210
2170 GOSUB 2210 : PLINE$ = "Please respond " + NODEMSRE$ : GOSUB 2210
2190 GOSUB 2210
2200 RETURN
2210 '
2220 'print one line
2230 '
2240 LPRINT PLINE$ : PLINE$ = " "
2250 LCOUNT = LCOUNT + 1
2260 RETURN
2270 '
2280 'page break
2290 '
2300 LPRINT CHR$(12) : LCOUNT = 5
2310 LPRINT HLINE1$ : PLINE$ = " " : LPRINT PLINE$ : LPRINT HLINE2$
2320 LPRINT PLINE$
2330 RETURN
2340 '
2350 ' error handling
2360 '
2370 IF (ERR = 53 AND LEN(FACFILE$) = 0) THEN LOCATE 20,4 :
     PRINT "File not found.  Please re-enter drive and node." : RESUME 330
2380 IF (ERR = 53 AND LEN(FACFILE$) ) 0) THEN LOCATE 20,4 :
     PRINT "Factor file "; FACFILE$; " not found.  Abandoning Process." :
     END
2390 LOCATE 20,10 : PRINT "erl is "; STR$(ERL); " err is "; STR$(ERR)
2400 LOCATE 21,10 : PRINT "end of program in error handling."
2410 END
2420 '
2430 ' end of program
2440 '
2450 CLS : LOCATE 10,10 : PRINT "Successful end of program."
2460 END
```

Program Name: STF40

Language: BASIC

Machine: Compaq Personal Computer (IBM compatible)

Purpose: This program is a utility that supports the data collection effort. Questionnaire responses can be entered into a file using this program.

```
10 '     THIS PROGRAM ACCEPTS DATA FROM A QUESTIONNAIRE AND WRITES IT TO A FILE.
20 '
30 '     PROGRAM-ID.  STF-40.
40 '     DATE-WRITTEN.    MARCH, 1985
50 '     INSTALLATION.    ALPHATECH, INC.
60 '     ACCOUNT.  J187-01.
70 '
80 CLEAR ,,8192
90 GOSUB 360                                        'Get node data
100 FOR I = 1 TO NFACS                               'Get factor data
110     GOSUB 460
120     GOSUB 580
130 NEXT I
140 GOSUB 740                                       'open output file
150 FOR RESP = 1 TO NUMRESP
160 GOSUB 850                                        'reset factor levels
170 GOSUB 1000                                       'do one ways
180 GOSUB 1530                                      'do two ways
190 FOR I = 1 TO NFACS
200      GOSUB 2390                                 'reduce dimensions
210 NEXT I
220 IF NFACS = 3 THEN GOSUB 2540                    'do three-ways
230 IF NFACS = 4 THEN GOSUB 2680                    'or do four-ways
240 NEXT RESP
250 GOSUB 2960                                      'End of job
260 '
270 'prompt for node number and read nodefile.
280 '
290 CLS : ON ERROR GOTO 2830 : OPTION BASE 1
300 LOCATE 8,4 : PRINT "STF-40."
310 LOCATE 10,4 : PRINT "This program accepts data from a questionnaire."
320 LOCATE 12,4 : PRINT "Please enter drive files are on."
330 LOCATE 14,4 : PRINT "Please enter questionnaire number."
340 LOCATE 16,4 : PRINT "Please enter total number of respondents."
350 LOCATE 12,45 : INPUT " ",DRIVE$
360 IF NOT (DRIVE$ = "a" OR DRIVE$ = "b" OR DRIVE$ = "A" OR DRIVE$ = "B")
    THEN BEEP : LOCATE 20,10 : PRINT "Drive must be A or B or a or b." :
         GOTO 350
370 LOCATE 14,45 : INPUT " ",NODENUM$
380 IF NOT LEN(NODENUM$) = 7 THEN BEEP : LOCATE 20,10 :
    PRINT "Node number must have seven digits." : GOTO 370
390 LOCATE 16,45 : INPUT " ",RESPNUM$ : NUMRESP = VAL(RESPNUM$)
400 IF NOT (NUMRESP ) 1 AND NUMRESP ( 6) THEN BEEP : LOCATE 20,10 :
       PRINT "Number of respondents must be between 2 and 6." : GOTO 390
410 NODEFILE$ = DRIVE$ + ":N" + NODENUM$
420 OPEN NODEFILE$ FOR INPUT AS #1
430 INPUT #1, NODENUM$, NODENAME$, NODEFM$(1), NODEFM$(2), NODEDEFN$(1),
              NODEDEFN$(2), NODEDEFN$(3), NFACS, NODEMSRE$
440 CLOSE #1
450 RETURN
```

```
460 '
470 'generate factor number.
480 '
490 IF NODENUM$ = "0000000" THEN
       FACNUM$ = RIGHT$(STR$(I),1) + "000000" : RETURN
500 REST$ = NODENUM$
510 FOR J = 1 TO 7
520     DIGIT$ = LEFT$(REST$,1)
530     REST$ = RIGHT$(REST$,(7-J))
540     IF DIGIT$ = "0" THEN GOTO 560
550 NEXT J
560 FACNUM$ = LEFT$(NODENUM$,(J-1)) + RIGHT$(STR$(I),1)
                                    + RIGHT$(NODENUM$,(7-J))
570 RETURN
580 '
590 'read factor mode file
600 '
610 FACFILE$ = DRIVE$ + ":N" + FACNUM$
620 OPEN FACFILE$ FOR INPUT AS #1
630 INPUT #1,FACNUM$(I),FACNAME$(I),FACFM$(I,1),FACFM$(I,2),FACDEFN$(I,1),
          FACDEFN$(I,2),FACDEFN$(I,3), FILLER,FACMSRE$(I),FACLEVELS(I)
640 FOR J = 1 TO FACLEVELS(I)
650     INPUT #1,FILLER,LEVELDEF$(I,J,1),LEVELDEF$(I,J,2),LEVELDEF$(I,J,3),
              SHORTFM$(I,J,1),SHORTFM$(I,J,2)
660     SAVELEVELS(I) = FACLEVELS(I) : LEVELNUM(I,J) = J
670     FOR K = 1 TO 3
680         SAVEDEF$(I,J,K) = LEVELDEF$(I,J,K)
690         IF K < 3 THEN SAVEFM$(I,J,K) = SHORTFM$(I,J,K)
700     NEXT K
710 NEXT J
720 CLOSE #1
730 RETURN
740 '
750 ' open output file
760 '
770 QFILE$ = DRIVE$+":Q"+NODENUM$
780 OPEN QFILE$ FOR OUTPUT AS #1
790 QNUM$=NODENUM$
800 PRINT #1,RIGHT$(STR$(NFACS),1)
810 STUFF$ = ""
820 FOR I = 1 TO NFACS : STUFF$ = STUFF$ + RIGHT$(STR$(FACLEVELS(I)),1) : NEXT
830 PRINT #1,STUFF$
840 RETURN
850 '
860 'reset levels of factors
870 '
880 FOR I = 1 TO NFACS
890     IF SAVELEVELS(I) < 5 THEN GOTO 980
900     FOR J = 1 TO 5
910     LEVELNUM(I,J) = J
920         FOR K = 1 TO 3
930             LEVELDEF$(I,J,K) = SAVEDEF$(I,J,K)
940             IF K < 3 THEN SHORTFM$(I,J,K) = SAVEFM$(I,J,K)
950         NEXT K
960     NEXT J
970     FACLEVELS(I) = SAVELEVELS(I)
980 NEXT I
990 RETURN
```

```
1000 '
1010 'do one-ways
1020 '
1030 THREEWAY = 0 : FOURWAY = 2
1040 FOR I = 1 TO NFACS
1050     XROW = 1 : CLS
1060     PLINE$ = STRING$(70," ") + "O:" + NODENUM$ : GOSUB 2780
1070     PLINE$ = STRING$(70," ") + "Reso:" + STR$(RESP) : GOSUB 2780
1080     GOSUB 2780
1090     PLINE$ = "Response is    " + NODENAME$ : GOSUB 2780
1100     PLINE$ = "Factor is " + FACNAME$(I) : GOSUB 2780 : GOSUB 2780
1110     PLINE$ = STRING$(79," ")
1120     MID$(PLINE$,10) = "LEVEL"
1130     MID$(PLINE$,36) = "(Short Form)"
1140     MID$(PLINE$,50) = "RESPONSE"
1150     GOSUB 2780
1160     GOSUB 1200
1170     GOSUB 1340
1180 NEXT I
1190 RETURN
1200 '
1210 'print factor levels and response lines
1220 '
1230 SAVEROW = CSRLIN
1240 FOR J = 1 TO FACLEVELS(I)
1250     FOR K = 1 TO 3
1260         PLINE$ = STRING$(79," ")
1270         MID$(PLINE$,5,30) = LEVELDEF$(I,J,K)
1280         IF NOT (K = 1) THEN MID$(PLINE$,36) = SHORTFM$(I,J,(K-1))
1290         IF K = 3 THEN MID$(PLINE$,50,20) = "__________________"
1300         GOSUB 2780
1310     NEXT K
1320 NEXT J
1330 RETURN
1340 '
1350 'get responses
1360 '
1370 XROW = SAVEROW
1380 FOR J = 1 TO FACLEVELS(I)
1390     FOR K = 1 TO 3
1400         IF NOT (K = 3) THEN GOSUB 2780 : GOTO 1460
1410         FOR QN = 1 TO 5 : Q(QN) = 0 : NEXT QN
1420         LOCATE CSRLIN,50 : INPUT STUFF$
1430         Q(I) = J
1440         QID$ = "" : FOR QN = 1 TO 5 : QID$ = QID$+RIGHT$(STR$(Q(QN)),1)
                                 : NEXT QN
1450       GOSUB 2840
1460     NEXT K
1470    GOSUB 2780
1480 NEXT J
1490 RETURN
```

```
1500 '
1510 ' print two ways (factor with most levels goes down the page)
1520 '
1530 X = 0 : TWOWAYS = +1
1540 '
1550 '
1560 '
1570 FOR I = (X+1) TO (NFACS-1)
1580    FOR J = (I+1) TO NFACS
1590       IF FACLEVELS(I) ) FACLEVELS(J) THEN II = I : JJ = J
                                          ELSE II = J : JJ = I
1600       GOSUB 1640
1610    NEXT J
1620 NEXT I
1630 RETURN
1640 '
1650 XROW = 3
1660 GOSUB 1790                                   'print text headers across
1670 SAVEROW = CSRLIN : RPTR = 0
1680 FOR RESPBLOCK = 1 TO FACLEVELS(II)
1690    GOSUB 2060                                'print response block
1700 NEXT RESPBLOCK
1710 IF TWOWAYS = +1 OR THREEWAY = +1 OR FOURWAYS = +1 THEN LOCATE 23,10 :
     PRINT "Enter . (full stop) to skip screen, .. (two stops) to skip line."
1720 XROW = SAVEROW : RPTR = 0
1730 FOR RESPBLOCK = 1 TO FACLEVELS(II)
1740    STUFF$ = ""
1750    GOSUB 2200                                'get responses
1760    IF STUFF$ = "." THEN GOTO 1780
1770 NEXT RESPBLOCK
1780 RETURN
1790 '
1800 'print text headers across
1810 '
1820 CLS
1830 IF (FOURWAY = +1) THEN PLINE$ = "In the following, " + FACNAME$(L) +
  " is fixed at " + SHORTFM$(L,LL,1) + " " + SHORTFM$(L,LL,2) : GOSUB 2780
1840 PLINE$ = STRING$(79," ") : MID$(PLINE$,25) = FACNAME$(JJ) : GOSUB 2780
1850 PLINE$ = STRING$(79," ") : MID$(PLINE$,25) = STRING$(LEN(FACNAME$(JJ)),"-")
             : GOSUB 2780
1860 PLINE$ = STRING$(79," ")
1870 FOR ROW = 1 TO 2
1880    FOR COL = 1 TO FACLEVELS(JJ)
1890       MID$(PLINE$,(25+11*(COL-1)),10) = SHORTFM$(JJ,COL,ROW)
1900    NEXT COL
1910    GOSUB 2780
1920    PLINE$ = STRING$(79," ")
1930 NEXT ROW
1940 '
1950 ' print left hand side column headings
1960 '
```

```
2460 '
2470 'for 5 factor levels shift 3 to 2 and 5 to 3
2480 '
2490 IF J1 = 3 THEN J2 = 2 ELSE J2 = 3
2500 LEVELNUM(I,J2) = LEVELNUM(I,J1)
2510 FOR K = 1 TO 3 : LEVELDEF$(I,J2,K) = LEVELDEF$(I,J1,K) : NEXT K
2520 FOR K = 1 TO 2 : SHORTFM$(I,J2,K) = SHORTFM$(I,J1,K) : NEXT K
2530 RETURN
2540 '
2550 ' print three-ways
2560 '
2570 TWOWAYS = 0 : THREEWAY = +1
2580 L = 0
2590 '
2600 '
2610 '
2620 FOR K = (L+1) TO (NFACS-2)
2630    FOR KK = 1 TO FACLEVELS(K)
2640         GOSUB 1570
2650    NEXT KK
2660 NEXT K
2670 RETURN
2680 '
2690 ' print fourways
2700 '
2710 TWOWAYS = 0 : FOURWAY = +1
2720 FOR L = 1 TO (NFACS-3)
2730     FOR LL = 1 TO FACLEVELS(L)
2740         GOSUB 2620
2750     NEXT LL
2760 NEXT L
2770 RETURN
2780 '
2790 'print one line on screen
2800 '
2810 LOCATE XROW,1 : PRINT PLINE$ : PLINE$ = ""
2820 XROW = XROW + 1
2830 RETURN
2840 '
2850 ' write one record to the file
2860 '
2870 PRINT #1, QNUM$; RIGHT$(STR$(RESP),1); QID$; STUFF$
2880 RETURN
2890 '
2900 ' error handling
2910 '
2920 IF (ERR = 53 AND ERL = 430) THEN LOCATE 20,4 :
        PRINT "Nodefile not found.  Please reenter data." : RESUME 350
2930 IF (ERR = 53 AND ERL = 620) THEN LOCATE 20,4 :
        PRINT "Factor file ";FACFILE$;" not found.  Abandoning process." : END
2940 LOCATE 20,10 : PRINT "erl is ";STR$(ERL);" err is ";STR$(ERR)
2950 LOCATE 21,10 : PRINT "end of program in error handling." : END
2960 '
2970 ' end of program
2980 '
2990 CLOSE #1
3000 CLS : LOCATE 10,10 : PRINT "Successful end of program."
3010 END
```

```
1970 FOR ROW = 1 TO 2
1980    IF ((THREEWAY = +1) OR (FOURWAY = +1))
           THEN MID$(PLINE$,1,10) = FACFM$(K,ROW)
1990    MID$(PLINE$,12,10) = FACFM$(II,ROW)
2000    GOSUB 2780
2010    PLINE$ = STRING$(79," ")
2020 NEXT ROW
2030 IF ((THREEWAY = +1) OR (FOURWAY = +1)) THEN MID$(PLINE$,1) = "----------"
2040 MID$(PLINE$,12) = "----------" : GOSUB 2780
2050 RETURN
2060 '
2070 'print response block
2080 '
2090 FOR ROW = 1 TO 2
2100    PLINE$ = STRING$(79," ")
2110    IF ((THREEWAY = +1) OR (FOURWAY = +1))
           THEN MID$(PLINE$,1,10) = SHORTFM$(K,KK,ROW)
2120    MID$(PLINE$,11,10) = SHORTFM$(II,RESPBLOCK,ROW)
2130    IF ROW = 1 THEN GOTO 2170
2140    FOR COL = 1 TO FACLEVELS(JJ)
2150        MID$(PLINE$,(24+11*(COL-1)),10) = "__________"
2160    NEXT COL
2170    GOSUB 2780
2180 NEXT ROW
2190 RETURN
2200 '
2210 'get responses
2220 '
2230 FOR ROW = 1 TO 2
2240    IF NOT (ROW = 2) THEN GOSUB 2780 : GOTO 2360
2250    FOR COL = 1 TO FACLEVELS(JJ)
2260        LOCATE XROW,(24+11*(COL-1)) : INPUT STUFF$
2270        IF STUFF$ = "" THEN BEEP : GOTO 2260
2280        IF (STUFF$ = "." OR STUFF$ = "..") THEN GOTO 2360
2290        FOR QN = 1 TO 5 : Q(QN) = 0 : NEXT QN
2300        Q(II) = LEVELNUM(II,RESPBLOCK) : Q(JJ) = LEVELNUM(JJ,COL)
2310        IF THREEWAY = +1 OR FOURWAY = +1 THEN Q(K) = LEVELNUM(K,KK)
2320        IF FOURWAY = +1 THEN Q(L) = LEVELNUM(L,LL)
2330        QID$ = "" : FOR QN = 1 TO 5 : QID$ = QID$+RIGHT$(STR$(Q(QN)),1)
                     : NEXT QN
2340        GOSUB 2840
2350    NEXT COL
2360 NEXT ROW
2370 GOSUB 2780
2380 RETURN
2390 '
2400 ' reduce dimensions : five factor levels to three
2410 '
2420 IF FACLEVELS(I) < 5 THEN RETURN
2430 FOR J1 = 3 TO 5 STEP 2 : GOSUB 2460 : NEXT J1
2440 FACLEVELS(I) = 3
2450 RETURN
```

REPORT DOCUMENTATION PAGE

EPORT SECURITY CLASSIFICATION nclassified	1b. RESTRICTIVE MARKINGS
ECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
ECLASSIFICATION/DOWNGRADING SCHEDULE	
RFORMING ORGANIZATION REPORT NUMBER(S) -10221U	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-86- 201 (I)
AME OF PERFORMING ORGANIZATION LPHATECH, Inc	6b. OFFICE SYMBOL (If applicable)
	7a. NAME OF MONITORING ORGANIZATION Hq Electronic Systems Division (XR)
DDRESS (City, State and ZIP Code) Burlington Executive Center 11 Middlesex Turnpike urlington, MA 01803	7b. ADDRESS (City, State and ZIP Code) Hanscom Air Force Base, MA 01731
AME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)
	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-84-D0016
ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
TITLE (Include Security Classification) C3I Analysis Tools for Development Planning (Volume 1)	
PERSONAL AUTHOR(S) P.A. Vail, G.H. Weissman, J.G. Wohl	
TYPE OF REPORT Final	13b. TIME COVERED FROM _____ TO _____
	14. DATE OF REPORT (Yr., Mo., Day) 1985 September 27
	15. PAGE COUNT 92
SUPPLEMENTARY NOTATION Refer to ESD-TR-86-201 (II) for Appendix F: Software Listings	

COSATI CODES		
IELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Vanguard Ballistic Missile Warning
NORAD Subject Transfer Function (STF) (Over)

ABSTRACT (Continue on reverse if necessary and identify by block number)

Subjective transfer function modeling method was developed for use in Command, Control Communications and Intelligence (C^3) systems planning and evaluation. Model is based on algrebraic relationship for each node of a tree structure which describes decision process in command and control systems for strategic and ballistic missile defense. Trees were found to be easy to construct, however, quantitative measures were difficult to define. Prototype program was developed for the Apple MacIntosh computer. Program does not cover all mission areas, but can be used for limited sensitivity analysis. Program was not found suitable for ranking proposed programs in order of importance.

0. DISTRIBUTION/AVAILABILITY OF ABSTRACT NCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL JAMES D. TAYLOR, Lt Col, USAF	22b. TELEPHONE NUMBER (Include Area Code) (617) 271-3116
	22c. OFFICE SYMBOL ESD/XRX

D FORM 1473, 83 APR EDITION OF 1 JAN OBSOLETE.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE