AD A111105

LEVEL

DTIC FILE COPY

DTIC
ELECTE
FEB 19 1982
S D
A

This document has been approved for public release and sale; its distribution is unlimited.

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Copy available to DTIC does not permit fully legible reproduction

82 02 18 121

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AFIT/GOR/AA/81D-1

MADAM:
MULTIPLE-ATTRIBUTE DECISION ANALYSIS MODEL

VOLUME II

THESIS

AFIT/GOR/AA/81D-1 Wayne A. Stimpson
2Lt USAFR

DTIC
ELECTE
FEB 1 9 1982
S D
A

This document has been approved for public release and sale; its distribution is unlimited.

AFIT/GOR/AA/81D-1

Thesis

MADAM:

MULTIPLE-ATTRIBUTE DECISION ANALYSIS MODEL

Volume II

by

Wayne A. Stimpson
2Lt USAFR

Prepared in partial
fulfillment of the
requirements for a
Masters Degree

December 1981

School of Engineering
Air Force Institute of Technology
Wright-Patterson Air Force Base
Ohio

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
A 23

DTIC
COPY
INSPECTED
2

Volume II

Table of Contents

List of Figures and Tables iii

Abstract iv

I. Introduction. 1

II. Data Structures 2

III. Sensitivity Analysis. 7

IV. Hierarchy Manipulation. 9
- Subroutines NODIN and FIND. 9
- Subroutines PRETOT, PRENEX, and NEXT. 11
- Subroutine SPAN 13
- Subroutine CALC 13

V. Variable List 15
- Variables Used. 15
- Pseudo-variables. 18
- Others. 18
- Cross Reference Map 19

VI Program Structure 20

Program Source Code. 23

List of Figures and Tables

		Page
Table 1	Pseudo-array Variables	3
Figure 1	Data Fields for Node Record.	4
Table 2	Record Content	4
Figure 2	The Search Process (FIND).	10
Figure 3	Logic Structure for NEXT	12
Figure 4	Logic Flow of CALC	14

Abstract

The complex multifaceted decision situations present today suggest the need for a timely, automated too. A decision-maker is for ed into comparing alternative actions or systems over an entire set of different measures of merit. This effort is an on-line, real-time, compu ter-based decision aid designed to assist the decision-maker in clarifying preferences in a complex decision environment. It is applicable to problems which may be represented by a hierarchy of objectives to be satisfied. The program is MADAM: Multiple-Attribute Decision Analysis Model, and it is written in FORTRAN V and is implemented on the CYBER 175 system. MADAM is designed to aid the decision-maker as he or she progresses through problem formulation, parameterization, sensitvity analyses, and a decision, including storage of all data and rationales. Deterministic problems are analyzed through Multi-Attribute Utility Theory concepts and an additive value function is utilized for sensitivity analysis. Pairwise preferential independence is tested between attributes. The sensitivity analyses include a cumulative weight analysis, a relative weight analysis, and an attribute level analysis. The analyses may be conducted by fixing an objective to be considered and conducting the analysis across the alternative systems or actions, or conversely by fixing the alternative to be considered and conducting the analysis across a desired set of objectives.

The work is divided into two volumes. Volume I is a theoretical presentation and includes a user's manual. It requires no programming expertise and may be used independently of Volume II. Volume II is a programming manual including the source code. It may not be used independently of Volume I.

I INTRODUCTION

This manual is designed to supplement the information contained in Volume I of this thesis. The foundamental intention to this volume is to provide decision analysts with an elucidation of the details of MADAM in order to better understand the program and, hence, facilitate its implementation. Future analysts may also have the desire to modify or extend the scope of problems suitable for analysis with MADAM, and the following information will be very useful.

MADAM is written in FORTRAN V and implemented on the CYBER 175 system. An effort was made to keep the code as machine independent as possible, and for those situations where this was not possible, an explicit discussion concerning modifications pre-requisite to transporting MADAM IS provided. All references to an example problem will refer to the example illustrated in Chapter V of Volume I.

II DATA STRUCTURES

MADAM utilizes a random access file for storing and manipulating data. Throughout the program are pairs of subroutines and functions which are responsible for setting and retrieving data. This is possible because FORTRAN V will not recognize an array variable until it is determined that no function by that name exists. Thus, for example, MADAM uses the function ARAY (I, J) to represent a pseudo-array variable. A complete list of the subroutine function pairs which represent psuedo-array variables is given in Table 1. Each record of the random access file is 100 words in length. This 100 word record is divided up in to data fields as shown in Figure 1 for the first 500 records. The remaining records have data fields as shown in Table 2.

> NOTE***The current ability to access both real and integer (character) data from a record is dependent on the fact that the CYBER 175 system uses one word for both types of information. In order to implement MADAM on a system which uses two words for real data and one word for integer data, it would be necessary to modify the psuedo-array variables so that the desired words will be accessed. This will require manipulation of the record length and storage (retrieval) subscripts.

Each data file (tree) constructed will require about 300 PRU for permanent storage.

As illustrated in Figure 1, each node record contains information about the position of the node in the objective hierarchy plus any node-

Psuedo-array Variable	Associated Subroutine	Associated Function	Information Generated by Variable
ARAY (I,J)	ASET (I,J,V)	ARAY (I,J)	Relative and Cumulative Weight of Node
ATT (I,J)	ATTVAL (I,J,V)	ATT (I,J)	Best/Worst Levels of Attributes
ATT1 (I)	ATTLBL (I,ATTRIB)	ATT1 (I)	Attribute Names
CRAY (I)	CSET (I)	CRAY (I) (Subroutine)	Comments or Rationale
IRAY (I,J)	ISET (I,J,IM)	IRAY (I,J)	Node Position Information
LABEL (I)	LSET (I,LABELT)	LABEL (I)	Attribute Name at Node 1
OBJECT (I,OBJCTV)	OSET (I,OBJCTV)	OBJECT (I,OBJCTV) (Subroutine)	Objective at Node 1
PARAM (I,J)	PSET (I,J,V)	PARAM (I,J)	Regression Parameters (Value Functions)
SYSLBL (I)	SYSSET (I,SYSLBL)	SYSLBL (I)	Alternative System Names
VRAY (I,J)	VSET (I,J,V)	VRAY (I,J)	System Values at Node 1

Table 1 Pseudo-array Variables

Data	Word(s)	Variable(s)
Node Reference Number Digit	1	IRAY (I,1)
First Descendant	2	IRAY (I,2)
First Crosslink	3	IRAY (I,3)
Backlink	4	IRAY (I,4)
Relative Weight	5	ARAY (I,1)
Cumulative Weight	6	ARAY (I,2)
Associated Attribute	7	LABEL (I)
System Values	8 to (NSYS+7)	VRAY (I,J)
Comments (Rationale)	(NSYS+7) to 64	-none-
Objective	65 to 80	OBJCTV (1) to OBJCTV (16

Figure 1 Data Fields for Node Record

Information	Record No.
Node Record	1 to 500
Number of Nodes (NNODES)	501
Number of Attributes (NATT)	502
Number of Systems (NSYS)	503
Null Record	504
Regression Intercepts	505
Regression Slopes	506
Sum of Squared Errors	507
Regression Form Indicators	508
System Names	509
Attribute Names	510
Null Record	511
Worst Attribute Levels	512
Best Attribute Levels	513

Table 2 Record Content

specific data. The positional information is provided by the first four words in the record. These words contain the node reference number digit, the first descendant, the first crosslink, and the backlink respectively. The node reference number digit indicates where this node is on its span. For instance, in the sample problem, the node reference number digit of the node with the objective: to maximize aircraft survivability, is two. The first descendant pointer contains a number of the first child of the given node (if any). Absence of a first child (IRAY (I,2) $\leq$ 0) implies that the given node is a data node. The first crosslink pointer contains a number of the immediate siblings of the given node (if any). The backlink pointer contains the number of the node which has this given node as a first descendant or a first crosslink.

The remaining words in a node record provide the data for identifying the node and incorporating it into the decision analysis. The fifth word of the record (ARRAY (1,1)) contains the weight of the given node relative to its siblings. These relative weights are normalized to a unity sum across the span. The sixth word of a node record (ARRAY (I,2)) contains the cumulative weight of a given node. It is automatically calculated by MADAM during the "folding-back" of the objective hierarchy (see subroutine CALC), and it is a function of the node's relative weight and the cumulative weight of its parent. The seventh word in a node record (LABEL (I)) contains the attribute label to be associated with that node. It is relevant only for data nodes, and is left a null field for all nodes interior to the tree.

NOTE***All character data is designed around the CYBER's 10 letters/word format. In order to implement MADAM on a system with a different

storage format, it will be necessary to scale all input data or data structures to incorporate a different number of characters per word.

The next set of words in a node record (VRAY (I,J)) hold the values of the alternative systems relative to a given node. These values are either calculated by MADAM (for an interior node) or indirectly input by the user (for a data node). Only as many words as needed are used for this information based on the number of systems. Those words between the system values and the 64th word are used to hold comments or rationale concerning either the relative weighting of the children of a given node or the input attribute levels of the alternative systems. Thus, the amount of space for these comments is inversely proportional to the number of alternative systems. Also, because of this structure, the number of alternative systems is limited to 57. Finally, the objective associated with the given node is contained in the 64th to the 80th words.

III SENSITIVITY ANALYSIS

The sensitivity analyses are controlled by the routines in the primary overlay (4,0). The types of analysis handled by the program are: relative weight analysis (RSA), cumulative weight analysis (CSA), attribute level analysis (LSA), and a system analysis (SSA) based on one of the first three types. The first three types of analysis involve determining a fixed node and conducting the appropriate analysis for all systems at that node. The last type of analysis (actually three subtypes SSA/RSA, SSA/CSA, SSA/LSA) involves fixing the alternative system and then conducting the appropriate analysis over a set of nodes.

The entry point of the overlay, routine SENSTV is used to determine the main type of analysis (CSA, RSA, LSA, or SSA). For any of the first three types of sensitivity analysis, control is shifted to routine DETNOD which is used to fix the analysis to the desired node. Routine DETNOD also establishes the range over which the factor (cumulative weight, relative weight, or attribute level) is to be considered. For the last type of analysis (SSA), control shifts to routine SYSSEN which in turn calls upon the routines, SENVAL, SENTYP, and NODSET to establish the fixed system, the factor to be considered, and the set of nodes over which the analysis is to be conducted.

Regardless of the type of analysis to be performed, once the initialization process (as described in the preceding paragraph), has been completed, control shifts to the routine COMPUT. This routine utilizes the equations presented in Chapter IV (of Volume I) to generate a matrix of overall values. Each row of this matrix corresponds to a different level of the factor under consideration. For the CSA, RSA, and LSA, each column

corresponds to a different system. For the SSA, each column represents a different node.

After the overall values have been calculated, the routines GRAPH and TABDIS are used to display the information in a graphical or tabular manner respectively (see Figures 4.1 through 4.4 in Volume I). At the conclusions of the desired type of sensitivity analysis, control is turned over to the main option selection.

IV HIERARCHY MANIPULATION

Subroutines NODIN and FIND

These subroutines are used to allow the user to enter the tree structure at a desired node. If the desired node is not an element of the tree, FIND will identify that node which is closest to the input node. All nodes are referenced by their node reference number (NRN). Except for option ***MOD***, the input of a non-validated NRN will be interpreted as an indication that the user desires to exit from the option. Program control will be return to the calling portion of MADAM. For option ***MOD*** an invalide NRN is interpreted as a desire to introduce a new node having the input NRN, so the appropriate branches (and dummy connecting nodes) are constructed so as to create a valid environment for the input node.

Subroutine NODIN. This routine allows the user to input a desired NRN. If anything but a null string (space followed by <CR>) is entered, this routine will call on FIND to establish the existence of a node corresponding to the input NRN.

Subroutine FIND. This routine implements a modified breadth-first search of the tree to search for a node corresponding to the input NRN. With each digit of the NRN representing a level, FIND starts with the first digit and attempts to locate a node with the desired NRN digit by crossing the span at that level. This process is repeated for all remaining levels (digits) of the input NRN. If a match cannot be established at a given level, this indicates that the desired node would be an added crosslink at that level. If a match is established, the next digit is brought up for consideration. If no more downlink path is available (a dat node has been reached) the routine is terminated.

Figure 2 illustrates an example of this search process.

Input NRN: 1.3.2

Iteration	Result	Remarks
1	1	First digit is satisfied
2	1 1	No Match
3	1 2	No Match
4	1 3	Second digit is satisfied
5	1 3 1	No Match
6	1 3 2	All digits satisfied so exit

Figure 2 The Search Process (FIND)

Subroutines PRENEX, PRETOT, and NEXT

These routines drive a depth-first search of the tree structure. Subroutines PRENEX and PRETOT are used to initialize the search process at a desired node or at the root node respectively. Once all variables have been initialized, subroutine NEXT performs a depth-first traversal, advancing one node each time it is called.

Subroutine PRENEX. This routine elicits a NRN from the user by invoking NODIN. If a valid NRN is entered, the traversal will begin at the node corresponding to the input NRN. All descendants of this input node will be encountered in the resulting transversal.

Subroutine PRETOT. This routine initialized the variables required by NEXT so as to run a transversal of the entire tree structure. It is usually invoked just prior to a potential application of PRENEX, thus allowing the user to gracefully opt a tour of the entire hierarchical structure.

Subroutine NEXT. This routine tours the hierarchical structure beginning at the node specified by either PRETOT or PRENEX. Since a check is made as to the validity of the input node before any processing is done, it is possible to exit the current option by entering a null string during option PRENEX. The logic implemented in NEXT is illustrated in Figure 3.

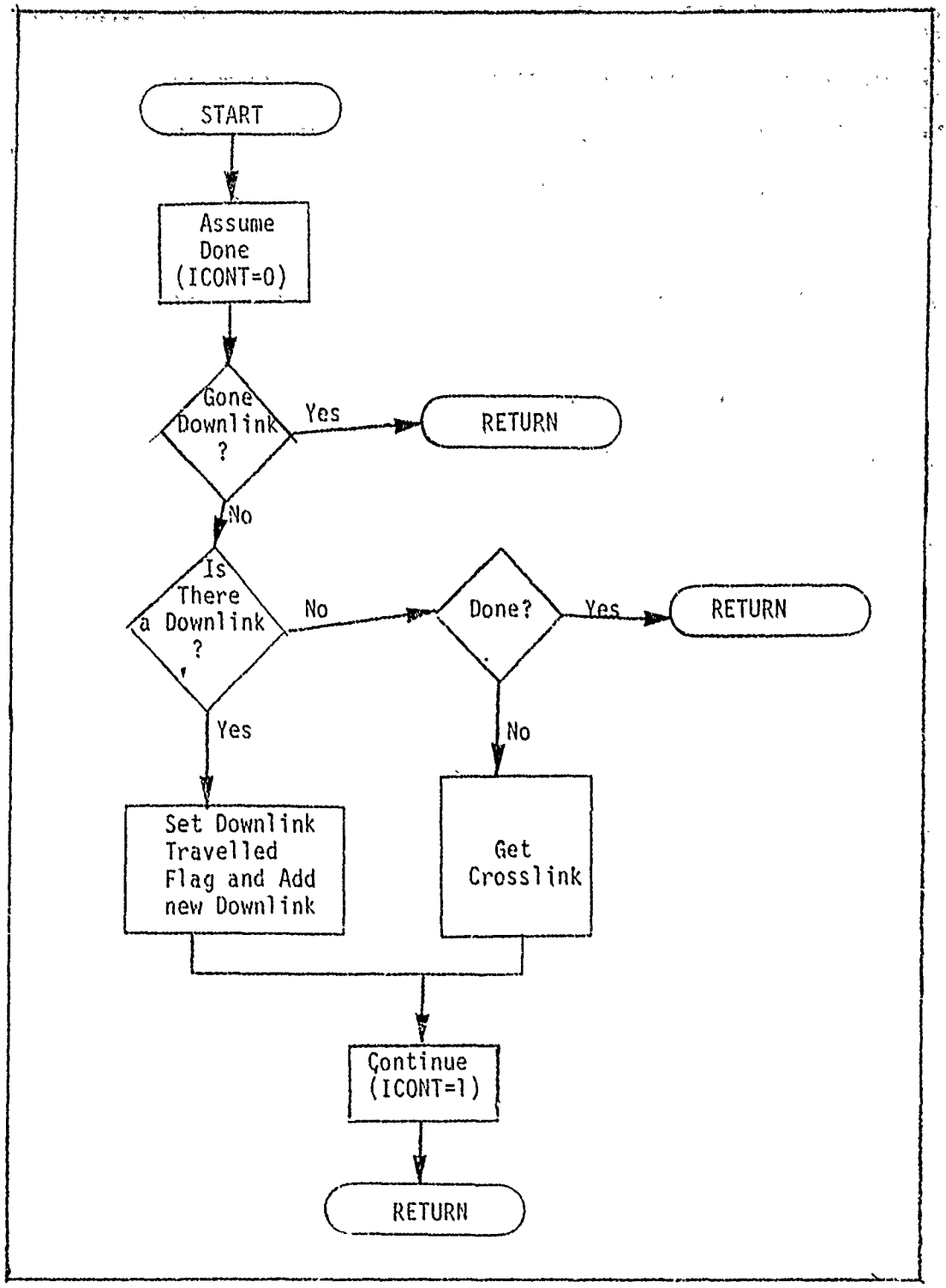

Figure 3 Logic Flow for NEXT

Subroutine SPAN

This routine allows for the input of the objective hierarchy structure by sequentially providing nodes in a depth-first fashion. The user inputs immediate descendants of the current node. Subroutine CHECK is used to establish a satisfactory set of children for the current node. Once a satisfactory set is established the first descendant of the current node becomes the new parent node. When a data node is reached (no descendants are input), the crosslink to the current node becomes the new parent node. If no more siblings exist, the routine will back up one level and establish the first crosslink as the new parent node (see Figure in the main thesis text).

Subroutine CALC

This routine calculates intermediate values (collapses the tree) by implementing a modified depth-first search. The tree is examined by each path from the root node to a data node, and the cumulative weights are calculated. System values based on the attributes are then computed for each node. The logic flow of CALC is provided in Figure 4. The cumulative weight of each node is given by:

CUMWT (NODE) = CUMWT(PARENT)* RELWT(NODE)

The system value at a particular node are the products of the system value at a data node and the cumulative weight of the upper nodes for each system.

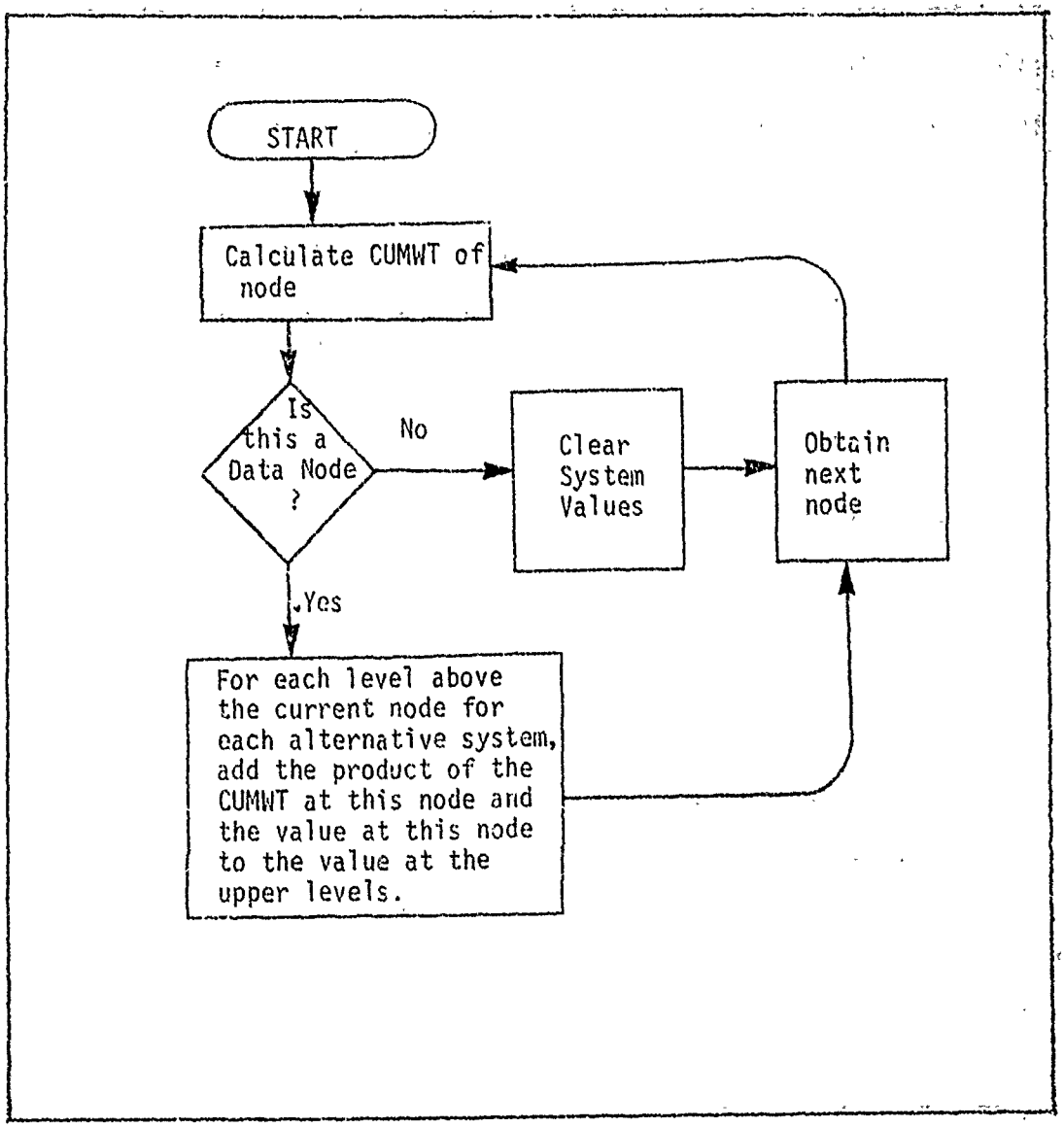

Figure 4 Logic Flow of CALC

V. Variable List

Variables Used

The following is a list of variables used in the program. This list includes only important variables and does not include scratch variables, do-loop indices, and dummy arguments. Psuedo-array variables are in the next section.

AIN - vector of relative weights

ARRAY - used to read and write to the data file. It holds one record worth of information

ATTMAX - maximum level of attribute

ATTMIN - minimum level of attribute

ATTNAM - name of attribute being perturbed during sensitivity analysis

ATTR1 - name of first attribute

ATTR2 - name of second attribute

B0 - intercept of linear regression (linear form)

B0EXP - intercept of linear regression (exponential form)

B0LN - intercept of linear regression (logarithmic form)

B0ROOT - intercept of linear regression (square-root form)

B0SQR - intercept of linear regression (squared form)

B1 - slope coefficient of linear regression (linear form)

B1EXP - slope coefficient of linear regression (exponential form)

B1LN - slope coefficient of linear regression (logarithmic form)

B1ROOT - slope coefficient of linear regression (square-root form)

B1SQR - slope coefficient of linear regression (squared form)

BAND - boundaries of indifference curve window

CMD - main option selection

DEL1 - change in first attribute

DEL2 - change in second attribute

ERRFLG - counts the number of consecutive errors in user entry during main option selection

FACLVL - level of attribute

FINDST - set of IFIND for SSA

ICHECK - flag if using between-node check
0 - not using check
1 - using check

ICONT - flags whether or not to continue processing
0 - stop
1 - continue

IDATA - flags whether or not node is data node
0 - interior node
1 - data node

IFADD - flags relationship of current node to input node. Determines how to add th input branch to the existing branch.
2 - IFIND is parent to input node
3 - IFIND is sibling to input node

IFIND - points to the last node of the branch contained in INNRN

INDERR - flag for PPI
0 - PPI hold
1 - PPI fails

INDX - NRN digit for IFIND

INDXST - set of INDX for SSA

IQUIT - flag to continue processing
0 - continue
1 - stop

IRRAY - used to read and write to the data file. It holds one record worth of information.

ISAVD - flags whether current information is saved on data file in case of abort
0 - not saved
1 - saved

ISIN - record number of record currently in memory (in ARRAY or IRRAY)

ISYS - number of systems being considered during sensitivity analysis

ITOL - tolerance during PPI testing

ITOTL - flags type of tree traversal
0 - not a total tree traversal
1 - traverse all descendants

KIDSET - set of NCHILD for nodes during SSA

LEVEL - vector containing branch to node IFIND
(*,1) - location of record containing node *
(*,2) - flags whether or not visited by PRETOT-PRENEX-NEXT
(*,3) - used in copying

LINE - one line of data for plot

LVL - length of LEVEL vector

MATRIX - pairwise-comparison matrix

MAX1 - maximum level of first attribute

MAX2 - maximum level of second attribute

MIN1 - minimum level of first attribute

MIN2 - minimum level of second attribute

MTYPE - type of SSA to be done

NATT - number of attributes

NCHILD - number of children for a node

NCROSS - number of siblings on a span

NDEEP - depth of tree (maximum number of levels)

NDIFF - number of digits in INNRN not matched by routine FIND

NLOUD - number of allowed nodes plus four

NLVLS - length of INNRN

NNODES - number in tree plus one (master)

NSYS - number of alternative systems

NTAPE - defines which data files is in use

NTREE - maximum number of simultaneous data files

OBJCTV - objective associated with a particular node

SENS - type of sensitivity analysis

SSE - sum of squared error (linear form)

SSEEXP - sum of squared error (exponential form)

SSELN - sum of squared error (logarithmic form)

SSERT - sum of squared error (square-root form)

SSESQR - sum of squared error (squared form)

TOLER - tolerance for PPI testing

TOP - objective of parent node during between-node check

USER - user identifier

VAL - attribute levels which define data points for the individual value function

WHOLD - matrix of overall value of systems during sensitivity analysis

WMAX - maximum level of factor during sensitivity analysis

WMIN - minimum level of factor during sensitivity analysis

Psuedo-arry Variables

The following are functions which are treated as if they were array variables (see Table 1).

ARAY(I,J)	relative and cumulative weight of node
ATT(I,J)	best/worst levels of attributes
ATT1(I)	attribute names
CRAY(I)	comments or rationale (subroutine)
IRAY(I,J)	node position information
LABEL(I)	name of attribute associated with node I
OBJECT(I,OBJCTV)	objective at node I
PARAM(I,J)	regression parameters

SYSLBL(I)	alternative system names
VRAY(I,J)	system values at node I

Cross Reference Map

The listing contains a partial load map, and combined with the comments within each program unit, a corss reference map of where each variable is used or modified may be generated.

VI. Program Structure

MADAM is divided up into five overlays (one main, four primary) which reduce the required field length for loading to about 60K. These overlays are machine dependent, but they may be removed by deleting the overlay statements, changing the program statement following each overlay statement to a subroutine statement (except for PROGRAM MADAM), and changing all overlay calls to subroutine calls to the subroutine making entry into the described overlay. Each primary overlay is designed to operate independently of the other primary overlays. Only the main overlay and one primary overlay are in the user's field length at one time, thus considerably reducing the required core memory allocation. The following is a list of routines in MADAM in the order in which they appear in the source code.

```
OVERLAY (XFILE,0,0) - main overlay
PROGRAM MADAM
     FUNCTION ARAY(I,J)
     SUBROUTINE ASET(I,J,V)
     FUNCTION ATT(I,J)
     FUNCTION ATT1(I)
     SUBROUTINE ATTLBL(I,ATTRIB)
     SUBROUTINE ATTSET(I,ATTRIB)
     SUBROUTINE ATTVAL(I,J,V)
     SUBROUTINE COPYR
     SUBROUTINE CRAY(I)
     SUBROUTINE CSET(I)
     SUBROUTINE FIND
     SUBROUTINE HELP
     SUBROUTINE INTRO
     FUNCTION IRAY(I,J)
     SUBROUTINE ISET(I,S,M)
     FUNCTION LABEL(I)
     SUBROUTINE LSET(I,LABELT)
     SUBROUTINE NEXT
     SUBROUTINE NODIN
     SUBROUTINE OBJECT(I,OBJCTV)
     SUBROUTINE OSET(I,OBJCTV)
     FUNCTION PARAM(I,J)
```

```
      SUBROUTINE PRENEX
      SUBROUTINE PRETOT
      SUBROUTINE PSET(I,J,V)
      FUNCTION SYSLBL(I)
      SUBROUTINE SYSSET(I,SYSLBL)
      SUBROUTINE TLOAD
      SUBROUTINE TSAVE
      FUNCTION VALU(X,LABEL)
      FUNCTION VRAY(I,J)
      SUBROUTINE VSET(I,K,V)
      FUNCTION XLEVEL(X,LABEL)

OVERLAY (XFILE,1,0) - primary overlay
PROGRAM DUMMY
      SUBROUTINE ADDSYS
      SUBROUTINE ASK(I, X1,X2,XMID)
      SUBROUTINE CHECK
      SUBROUTINE DELSYS
      SUBROUTINE GRAFIX
      SUBROUTINE INITT
      SUBROUTINE MASTER
      SUBROUTINE MODIFY
      SUBROUTINE NEW
      SUBROUTINE PICTUR(INDEX)
      SUBROUTINE PPI
      SUBROUTINE PRUNE
      SUBROUTINE RDATT
      SUBROUTINE RDSYSL
      SUBROUTINE RDTTL
      SUBROUTINE REGRS(INDEX)
      SUBROUTINE SPAN
      SUBROUTINE STAT
      SUBROUTINE TAPER
      SUBROUTINE VALUE

OVERLAY (XFILE,2,0) - primary overlay
PROGRAM WVLOAD
      SUBROUTINE CALL
      SUBROUTINE DVIDE(N,A,D)
      SUBROUTINE GMVEC(NCROSS,IFIND,AIN)
      SUBROUTINE RDV
      SUBROUTINE RDWT
      FUNCTION SUM(N,A)
      SUBROUTINE NVLODI(OPT)

OVERLAY (XFILE,3,0) - primary overlay
PROGRAM NUM
      SUBROUTINE DISPLA
      SUBROUTINE DSPLOT
      SUBROUTINE NEWPG
      SUBROUTINE NUMREV
```

```
OVERLAY (XFILE,4,0) - primary overlay
PROGRAM SENSTV
     SUBROUTINE COMPUT
     SUBROUTINE DETNOD
     SUBROUTINE HEADER(I,OBJCTV,ANSWER)
     SUBROUTINE SENVAL
     SUBROUTINE TABDIS
```

SOURCE CODE OF MADAM

Partial Load Map

```
      OVERLAY(XFILE,0,0)
      PROGRAM MADAM
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS PROGRAM IS THE MAIN PROGRAM OF MADAM.  IT IS USED TO   *
C*   CONTROL THE MAIN OPTION SELECTION.                          *
C*                                                               *
C*                                                               *
C*   CALLED BY:  (NONE)                                          *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  CMD,ERRFLG                                *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR / MATT
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/ONE/CMD
      COMMON/P/USER
      COMMON/RRAY1/ARRAY(88)
      COMMON/RRAYE/ISIN,NLOUD
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*3 CMD,USER*10
      INTEGER ERRFLG
      CALL INTRO
      NTAPE=1
      NSYS=2
      NDEEP=0
      CMD='SEL'
      CALL TLOAD
1     ERRFLG=0
      WRITE(*,'(5(/),1X)')
      PRINT*,USER,',YOUR OPTIONS ARE:'
      PRINT*,'ATT COP DIS DON MOD NEW NUM PRU REU SEL'
      PRINT*,'SEN SPA STA SYS TTL WUC'
      PRINT*,'***NOTE: IF YOU NEED AN EXPLANATION, ',USER
      PRINT*,'TYPE "HELP" ***'
      PRINT*,' '
      PRINT*,'WHAT IS YOUR CHOICE, ',USER,'?'
      READ(*,'(A3)')CMD
      IF(CMD.EQ.'ATT')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
7 CHARACTER SYMBOL _OVERLAY IS NON- ANSI
      ELSEIF(CMD.EQ.'COP')THEN
      ERRFLG=0
```

```
      CALL COPYR
      ELSEIF(CMD.EQ.'DIS')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',3,0)
      ELSEIF(CMD.EQ.'HEL')THEN
      ERRFLG=0
      CALL HELP
      ELSEIF(CMD.EQ.'MOD')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'NEW')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'NUM')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',3,0)
      ELSEIF(CMD.EQ.'PRU')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'REV')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',3,0)
      ELSEIF(CMD.EQ.'SEL')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'SEN')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',4,0)
      ELSEIF(CMD.EQ.'SPA')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'STA')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'SYS')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'TTL')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',1,0)
      ELSEIF(CMD.EQ.'UVC')THEN
      ERRFLG=0
      CALL OVERLAY('XFILE',2,0)
      ELSE
      ERRFLG=ERRFLG+1
      IF(CMD.NE.'DON')PRINT*,CMD,' IS NOT AN ALLOWED ENTRY. ',USER,'.'
      ENDIF
      IF((ERRFLG.GE.3).AND.(CMD.NE.'DON'))THEN
      PRINT*,'TO MINIMIZE ENTRY ERRORS, AN EXPLANATION WILL BE PROVIDED'
      CALL HELP
      ERRFLG=0
      ENDIF
      IF(CMD.NE.'DON')GO TO 1
      CALL TSAVE
      STOP
      END
```

```
--VARIABLE MAP--(LO-A)
 NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ARRAY       0B  /RRAY1/                     REAL           50      LEVEL       0B  /LEVEL/                     INTEGER        00
 CMD         0D  /ONE/                       CHARE3                  LVL         03  /LEVEL/                     INTEGER
 ERRFLG      0B  /SGABUEE/ SAV               INTEGER                 NATT        0E  /ATTN/                      INTEGER
 ICONT       0E  /NEX/                       INTEGER                 NDEEP       02  /C/                         INTEGER
 IDATA       0D  /NEX/                       INTEGER                 NDIFF       0B  /LEVEL/                     INTEGER
 IFADD       0R  /LEVEL/                     INTEGER                 NCLOUD      6B  /RRAY2/                     INTEGER
 IFIND       02  /LEVEL/                     INTEGER                 NLVLS       0B  /LEVEL/                     INTEGER
 IMAPN       05  /LEVEL/                     INTEGER        20       NNODES      02  /C/                         INTEGER
 ISROD       0R  /CNTRL/                     INTEGER                 NSYS        0B  /SYS1/                      INTEGER
 ISTN        0B  /RRAY2/                     INTEGER                 NTAPE       6B  /CNTRL/                     INTEGER
 ITOTL       0E  /NEX/                       INTEGER                 USER        03  /P/                         CHARE10

--PROCEDURES--(LO-P)
 NAME------TYPE--------ARGS------CLASS-----        NAME------TYPE--------ARGS------CLASS-----

 COPYR                  0       SUBROUTINE         OVERLAY                 3       SUBROUTINE
 HELP                   0       SUBROUTINE         TLOAD                   0       SUBROUTINE
 INTRO                  0       SUBROUTINE         TSAVE                   0       SUBROUTINE

- STATEMENT LABELS--(LO-A)
 LABEL-ADDRESS-----PROPERTIES----DEF

     1      0B                     42

--ENTRY POINTS--(LO-A)
 NAME---ADDRESS--ARGS---

 RADAR       0B      0

--STATISTICS--

 PROGRAM-UNIT LENGTH                 02 =  0
 CM LABELLED COMMON LENGTH         2648 =  180
 CM STORAGE USED                 606002 =  24960
 COMPILE TIME                   0.109 SECONDS

      1  ANSI      ERROR IN  RADAR
?
```

```
 1              FUNCTION ARAY(I,J)
 2        C
 3        C
 4        C*****************************************************************
 5        C*                                                               *
 6        C*                                                               *
 7        C*   THIS FUNCTION IS USED TO CIRULATE AN ARRAY VARIABLE         *
 8        C*   ARAY(I,J).  IT ACCESSES THE CUMULATIVE AND RELATIVE         *
 9        C*   WEIGHTS OF A NODE.  ARAY(I,1) IS THE NORMALIZED RELATIVE    *
10        C*   WEIGHT OF NODE I.  ARAY(I,2) IS THE CUMULATIVE WEIGHT       *
11        C*   OF NODE I.  THE VALUES ARE SET BY THE SUBROUTINE ASET(I,J,V).*
12        C*                                                               *
13        C*                                                               *
14        C*   CALLED BY:  COPYR,CALC,DISPLA,NUMREV,COMPUT,DETNOD          *
15        C*                                                               *
16        C*                                                               *
17        C*   VARIABLES                                                   *
18        C*          USED:  I,J,NTAPE                                     *
19        C*                                                               *
20        C*          MODIFIED:  ARRAY,ISIN,K                              *
21        C*                                                               *
22        C*                                                               *
23        C*****************************************************************
24        C
25        C
26              COMMON/CNTRL/ISAVD,NTAPE
27              COMMON/RRAY1/ARRAY(80)
28              COMMON/RRAY2/ISIN,NLOUD
29              SAVE
30              IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
31              ARAY=ARRAY(J+4)
32              ISIN=I
33              RETURN
34        99    PRINT*,' U ERROR IN ARAY'
35              RETURN
36              END
```

FUNCTION ARAY

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ARAY        00      2

--STATISTICS--

  PROGRAM-UNIT LENGTH                 598 =  42
  CM LABELLED COMMON LENGTH          1268 =  86
  CM STORAGE USED                  69600 = 24060
  COMPILE TIME                     0.030 SECONDS

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ARAY        00  /SBAEVSE/ SAV              REAL                    J          2  DUMMY-ARG                INTEGER
  ARRAY       00  /RRAY1/                    REAL          80        K          10 /SBASVSE/ SAV            INTEGER
  I           1   DUMMY-ARG                  INTEGER                 NLOUD      10 /RRAY2/                  INTEGER
  ISAVD       02  /CNTRL/                    INTEGER                 NTAPE      10 /CNTRL/                  INTEGER
  ISIN        08  /RRAY2/                    INTEGER

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS------PROPERTIES----DEF

    99     278                      34
```

```
      SUBROUTINE ASET(I,J,V)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS USED TO SET THE VALUES OF THE PSUEDO-ARRAY  *
C*   VARIABLE ARAY(I,J).  ASET(X,1,V) SETS THE RELATIVE WEIGHT   *
C*   OF NODE X TO V.  ASET(X,2,V) SETS THE CUMULATIVE WEIGHT     *
C*   OF NODE X TO V.  THE VALUES ARE ACCESSED VIA FUNCTION       *
C*   ARAY(I,J).                                                  *
C*                                                               *
C*                                                               *
C*   CALLED BY:  MODIFY,SPAN,CALC,RDWT                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,J,NTAPE,V                                   *
C*                                                               *
C*          MODIFIED:  ARRAY,ISIN,K                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/ARAY1/ARRAY(80)
      COMMON/ARAY2/ISIN,KLOUD
      IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,20)
      ARRAY(J+4)=V
      WRITE(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,20)
      ISIN=I
      RETURN
99    PRINT*,'I/O ERROR IN ASET'
      RETURN
      END
```

SUBROUTINE ASET

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARG$---
  ASET        5B      3

--STATISTICS--
  PROGRAM-UNIT LENGTH                 67B =  55
  CM LABELLED COMMON LENGTH          124B =  84
  CM STORAGE USED                   66600B = 24960
  COMPILE TIME                     0.039 SECONDS

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ARRAY       60  /ARAY1/                    REAL           80
  I            1  DUMMY-ARG                  INTEGER
  ISAVD       0B  /CNTRL/                    INTEGER
  ISIN        6B  /ARAY2/                    INTEGER
  J            2  DUMMY-ARG                  INTEGER
  K          64B                             INTEGER
  KLOUD       1B  /ARAY2/                    INTEGER
  NTAPE       1B  /CNTRL/                    INTEGER
  V            3  DUMMY-ARG                  REAL

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
    99     34B                     34
```

```
      FUNCTION ATT(I,J)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS FUNCTION IS USED TO SIMULATE THE ARRAY VARIABLE        *
C*   ATT(I,J).  IT ACCESSES THE WORST AND BEST LEVELS OF THE     *
C*   ATTRIBUTES.  ATT(X,1) IS THE WORST LEVEL OF ATTRIBUTE X.    *
C*   ATT(X,2) IS THE BEST LEVEL OF ATTRIBUTE X.  VALUES ARE      *
C*   SET BY SUBROUTINE ATTVAL(I,J,V).                            *
C*                                                               *
C*                                                               *
C*   CALLED BY: PICTUR,PPI,VALUE,RDV,DETMOD                      *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,J,NTAPE                                     *
C*                                                               *
C*          MODIFIED:  ARRAY,ISIN,K                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/ARRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      K=NLOUD+6+J
      IF(K.NE.ISIN)READ(NTAPE,REC=K,ERR=99)(ARRAY(L),L=1,20)
      ATT=ARRAY(I)
      ISIN=K
      RETURN
99    PRINT*,'I/O ERROR IN ATT'
      RETURN
      END
```

```
FUNCTION ATT

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ATT          6B      2

--STATISTICS--

  PROGRAM-UNIT LENGTH                 55B =  45
  CM LABELLED COMMON LENGTH          127B =  87
  CM STORAGE USED                  60600B = 24960
  COMPILE TIME                   0.037 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ARRAY        0B  /RRAY1/                    REAL          80       J          7   DUMMY-ARG                 INTEGER
  ATT          0B  /SBASUSE/ SAV              REAL                   K          1B  /SBASUSE/ SAV             INTEGER
  I            1   DUMMY-ARG                  INTEGER                L          2B  /SBASUSE/ SAV             INTEGER
  ISAVD        0B  /CNTRL/                    INTEGER                NLOUD      1B  /RRAY2/                   INTEGER
  ISIN         0B  /RRAY2/                    INTEGER                NTAPE      1B  /CNTRL/                   INTEGER

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     99     32B                     36
```

```
      SUBROUTINE ATTLBL(I,ATTRIB)
C                                                                       *
C                                                                       *
C*********************************************************************
C*                                                                     *
C*   THIS ROUTINE SETS THE VALUES OF THE PSEUDO-ARRAY                  *
C*   VARIABLE ATT1(I).  ATTLBL(I,ATTRIB) SETS THE NAME OF              *
C*   ATTRIBUTE I TO ATTRIB.  VALUES ARE ACCESSED VIA FUNCTION          *
C*   ATT1(I).                                                          *
C*                                                                     *
C*   CALLED BY:  ATTSET                                                *
C*                                                                     *
C*   VARIABLES                                                         *
C*         USED:  I,NTAPE                                              *
C*                                                                     *
C*         MODIFIED:  ATTRIB,IRRAY,ISIN,J                              *
C*                                                                     *
C*********************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/IRRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      CHARACTER*10 ATTRIB,IRRAY
      J=NLOUD+5
      IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(IRRAY(K),K=1,80)
      ATTRIB=ATTRIB//'
      IRRAY(I)=ATTRIB
      WRITE(NTAPE,REC=J,ERR=99)(IRRAY(K),K=1,80)
      ISIN=J
      RETURN
99    PRINT*,'I/O ERROR IN ATTLBL'
      RETURN
      END
```

```
SUBROUTINE ATTLBL

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
                                    37
  99     548

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---
  ATTLBL     58     2

--STATISTICS--
                                  1308 =  88
  PROGRAM-UNIT LENGTH             1278 =  87
  CM LABELLED COMMON LENGTH      60600B =  24960
  CM STORAGE USED                0.046 SECONDS
  COMPILE TIME
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ATTRIB     2   DUMMY-ARG                    CHAR*10
  I          1   DUMMY-ARG                    INTEGER
  IRRAY      0B  /RRAY1/                      CHAR*10      80
  ISAVD      0B  /CNTRL/                      INTEGER
  ISIN       0B  /RRAY2/                      INTEGER
```

```
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  J          1B  /SBABUSE/ SAV                INTEGER
  K          2B  /SBAGUSE/ SAV                INTEGER
  NLOUD      1B  /RRAY2/                      INTEGER
  NTAPE      1B  /CNTRL/                      INTEGER
```


PRECEDING PAGE BLANK-NOT FILMED

```
 1                SUBROUTINE ATTSET(I,ATTRIB)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS ROUTINE ELICITS THE BEST AND WORST LEVELS              *
 8          C*   OF ATTRIBUTE ATTRIB.                                        *
 9          C*                                                               *
10          C*                                                               *
11          C*   CALLED BY:  RDATT                                           *
12          C*                                                               *
13          C*                                                               *
14          C*   VARIABLES                                                   *
15          C*          USED:  LEVEL,ATTRIB,I,USER                           *
16          C*                                                               *
17          C*          MODIFIED:  ATTMIN,ATTMAX                             *
18          C*                                                               *
19          C*                                                               *
20          C*****************************************************************
21          C
22          C
23                COMMON/ATTR/NATT
24                COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
25                COMMON/P/USER
26                SAVE
27                CHARACTER*10 ATTRIB,ATT1,USER
28                CALL LSET(LEVEL(LVL,1),ATTRIB)
29                CALL ATTLBL(I,ATTRIB)
30                PRINT*,'WHAT IS THE WORST ACCEPTABLE '
31                PRINT*,'LEVEL (REAL NUMBER) OF ',ATTRIB
32                PRINT*,'     ?'
33                READ(*,'(F10.0)')ATTMIN
34                CALL ATTVAL(I,1,ATTMIN)
35                PRINT*,'THE LEVEL STORED WAS ',ATTMIN
36                PRINT*,' '
37                PRINT*,'WHAT IS THE BEST (REALISTICALLY)'
38                PRINT*,'LEVEL (REAL NUMBER) OF ',ATTRIB,', ',USER,'?'
39                PRINT*,'     ?'
40                READ(*,'(F10.0)')ATTMAX
41                CALL ATTVAL(I,2,ATTMAX)
42                PRINT*,'THE LEVEL STORED WAS ',ATTMAX
43                RETURN
44                END
```

```
SUBROUTINE ATTSET

--PROCEDURES--(LO*A)
 -NAME-------TYPE---------ARGS------CLASS-----

  ATTLBL                    2       SUBROUTINE
  ATTVAL                    3       SUBROUTINE
  LSET                      2       SUBROUTINE

--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  ATTSET      5B      2

--STATISTICS--

  PROGRAM-UNIT LENGTH                 232B =  154
  CM LABELLED COMMON LENGTH           133B =   91
  CM STORAGE USED                   60600B = 24960
  COMPILE TIME                      0.060 SECONDS
```

```
--VARIABLE MAP--(LO*A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ATTMAX      3B  /SBABUSE/ SAV              REAL
 ATTMIN      2B  /SBABUSE/ SAV              REAL
 ATTRIB      2   DUMMY-ARG                  CHAR*10
 ATT1        1B  /SBABUSE/ UNUSED/SAV       CHAR*10
 I           1   DUMMY-ARG                  INTEGER
 IFADD      27B  /LEVEL/                    INTEGER
 IFIND      25B  /LEVEL/                    INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 INRRN       1B  /LEVEL/                    INTEGER        20
 L[illegible]   31B  /LEVEL/                INTEGER        60
 L[illegible]   23B  /LEVEL/                INTEGER
 NATT        0B  /ATTR/                     INTEGER
 NDIFF      26B  /LEVEL/                    INTEGER
 NLVLS       0B  /LEVEL/                    INTEGER
 USER        0B  /P/                        CHAR*10
```

```
      SUBROUTINE ATTVAL(I,J,U)
      COMMON/CNTRL/ISAUD,NTAPE
      COMMON/RRAY1/ARRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      K=NLOUD+6+J
      IF(K.NE.ISIN)READ(NTAPE,REC=K,ERR=99)(ARRAY(L),L=1,80)
      ARRAY(I)=U
      WRITE(NTAPE,REC=K,ERR=99)(ARRAY(L),L=1,80)
      ISIN=K
      RETURN
   99 PRINT*,'I/O ERROR IN ATTVAL'
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ARRAY	0B	/RRAY1/		REAL	80
I	1	DUMMY-ARG		INTEGER	
ISAUD	0B	/CNTRL/		INTEGER	
ISIN	6B	/RRAY2/		INTEGER	
J	2	DUMMY-ARG		INTEGER	

-NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
K	1B	/$$ASVSE/	SAV	INTEGER	
L	2B	/$$ASVSE/	SAV	INTEGER	
NLOUD	1B	/RRAY2/		INTEGER	
NTAPE	1B	/CNTRL/		INTEGER	
U	3	DUMMY-ARG		REAL	

--STATEMENT LABELS--(LO=A)

-LABEL	-ADDRESS	-PROPERTIES	-DEF
99	37B		13

--ENTRY POINTS--(LO=A)

-NAME	---ADDRESS	--ARGS---
ATTVAL	52	3

--STATISTICS--

```
PROGRAM-UNIT LENGTH              70B =  56
CM LABELLED COMMON LENGTH       127B =  87
CM STORAGE USED                 606000B = 24960
COMPILE TIME                    0.037 SECONDS
```

```
      SUBROUTINE COPYR
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS USED TO COPY THE CHILDREN OF ONE NODE       *
C*   TO ANOTHER NODE.  THE OBJECTIVE AND WEIGHTS ARE             *
C*   TRANSFERRED.                                                *
C*                                                               *
C*                                                               *
C*   CALLED BY:  MADAM                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ICONT,IDATA,IFADD,IFIND,NDIFF,NLULS           *
C*                                                               *
C*          MODIFIED:  IFINDT,LEVEL,NNODES                       *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLULS,INRRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/ARRAY1/IRRAY(20)
      COMMON/RRAY2/ISIM,NLOUD
      SAVE
      CHARACTER*10 OBJCTV(10)
      PRINT*,' '
      PRINT*,'ENTER NODE TO BE COPIED TO.'
      CALL NODIN
      IF(NLULS.LE.0)RETURN
      IF(NDIFF.GT.0)THEN
      PRINT*,'NODE MUST EXIST BEFORE COPYING IS ALLOWED.'
      RETURN
      ENDIF
      IFINDT=IFIND
      PRINT*,'ENTER NODE TO BE COPIED.'
      CALL PRENEX
      IF(ICONT.EQ.0)RETURN
      IF(NDIFF.GT.0)THEN
      PRINT*,'NODE TO BE COPIED DOES NOT EXIST.'
      RETURN
      ENDIF
      LEVEL(LUL,3)=IFINDT
      IF(IDATA.NE.0)THEN
      PRINT*,'THE NODE ENTERED IS A DATA NODE,'
      PRINT*,'AND CANNOT BE COPIED FROM.'
      RETURN
      ENDIF
1     CALL NEXT
      IF(ICONT.EQ.0)THEN
      PRINT*,'COPYING IS COMPLETED.'
      RETURN
```

```
58                ENDIF
59                IFINDT=LEVEL(LUL,3)
60                IF(IFADD.EQ.2)IFINDT=LEVEL(LUL-1,3)
61                NNODES=NNODES+1
62                CALL ISET(IFINDT,IFADD,NNODES)
63                CALL OBJECT(IFIND,OBJCTU)
64                CALL OSET(NNODES,OBJCTU)
65                CALL ISET(NNODES,1,IRAY(LEVEL(LUL,1),1))
66                CALL ISET(NNODES,2,-1)
67                CALL ISET(NNODES,3,-1)
68                CALL ISET(NNODES,4,IFINDT)
69                CALL ASET(NNODES,1,ARAY(IFIND,1))
70                CALL ASET(NNODES,2,ARAY(IFIND,2))
71                LEVEL(LUL,3)=NNODES
72                GO TO 1
73                END
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ICONT      0B  /NEX/                          INTEGER
 IDATA      1B  /NEX/                          INTEGER
 IFADD     27B  /LEVEL/                        INTEGER
 IFIND     25B  /LEVEL/                        INTEGER
 IFINDT    21B  /SSA$USE/ SAU                  INTEGER
 INARM      1B  /LEVEL/                        INTEGER          20
 IRRAY      0B  /RRAY1/                        INTEGER          80
 ISAUD      0B  /CNTRL/                        INTEGER
 ISIN       0B  /RRAY2/                        INTEGER
 ITOTL      2B  /NEX/                          INTEGER
 INTEGER
```

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 LEVEL     31B  /LEVEL/                        INTEGER          60
 LUL       30B  /LEVEL/                        INTEGER
 NDEEP      1B  /C/                            INTEGER
 NDIFF     26B  /LEVEL/                        INTEGER
 NLOUD      1B  /RRAY2/                        INTEGER
 NLULS      0B  /LEVEL/                        INTEGER
 NNODES     0B  /C/                            INTEGER
 NTAPE      1B  /CNTRL/                        INTEGER
 OBJCTU     1B  /S$A$USE/ SAU                  CHAR*10          16
```

```
--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 ARAY       REAL          2      FUNCTION
 ASET                     3      SUBROUTINE
 IRAY       INTEGER       2      FUNCTION
 ISET                     3      SUBROUTINE
 NEXT                     0      SUBROUTINE
```

```
-NAME------TYPE--------ARGS------CLASS-----

 NODIN                    0      SUBROUTINE
 OBJECT                   2      SUBROUTINE
 OSET                     2      SUBROUTINE
 PRENEX                   0      SUBROUTINE
```

```
--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    1     63B                      54
```

SUBROUTINE COPYR

```
--STATISTICS--

PROGRAM-UNIT LENGTH          3225 =  21C
CM LABELLED COMMON LENGTH    330B =  192
CM STORAGE USED              6C66CB =  24969
COMPILE TIME                 0.10B SECONDS
```

```
--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 COPYR      5B      0
```

```
      SUBROUTINE CRAY(I)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS USED TO ACCESS COMMENTS (OTHER THAN THE     *
C*   OBJECTIVE) ASSOCIATED WITH A NODE.  THESE COMMENTS WILL     *
C*   BE A TITLE FOR THE DATA STRUCTURE FOR CRAY(1).  THEY WILL   *
C*   BE RATIONALE FOR THE RELATIVE WEIGHTS OF THE CHILDREN FOR   *
C*   AN INTERIOR (NON-DATA) NODE.  THEY WILL BE COMMENTS         *
C*   CONCERNING ATTRIBUTE LEVELS FOR A DATA NODE.  THESE         *
C*   COMMENTS ARE ENTERED VIA SUBROUTINE CSET(I).                *
C*                                                               *
C*                                                               *
C*   CALLED BY:  NEWPG,NUMREV                                    *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,MSYS,NTAPE                                  *
C*                                                               *
C*          MODIFIED:  ARRAY,ISIN,ISTART,ISTOP,J,K               *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/ARRAY1/ARRAY(80)
      COMMON/ARRAY2/ISIN,NLOUD
      COMMON/SYS1/MSYS
      SAVE
      CHARACTER*10 ARRAY
      ISTART=6+MSYS
      IF(ISIN.NE.I)READ(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
1     IF(ARRAY(ISTART).EQ.' ')THEN
      ISIN=I
      RETURN
      ENDIF
      ISTOP=ISTART+7
      IF(ISTOP.GT.64)THEN
      ISIN=I
      RETURN
      ENDIF
      WRITE(*,'(1X,8A10)')(ARRAY(J),J=ISTART,ISTOP)
      ISTART=ISTART+8
      GO TO 1
99    PRINT*,'I/O ERROR IN CRAY'
      RETURN
      END
```

```
      SUBROUTINE CRAY

--VARIABLE MAP--(LO-A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
                                            CHAR*10        80
 ARRAY        0B   /RRAY1/
 I            1    DUMMY-ARG                INTEGER
 ISAUD        6B   /CNTRL/                  INTEGER
 ISIN         4B   /RRAY2/                  INTEGER
 ISTART       1B   /SSASUSE/ SAU            INTEGER
 ISTOP        3B   /SSASUSE/ SAU            INTEGER

--STATEMENT LABELS--(LO-A)
-LABEL-ADDRESS-----PROPERTIES----DEF
                                 36
    1     22B                    43
   99     73B

--ENTRY POINTS--(LO-A)
-NAME---ADDRESS--ARGS---

 CRAY        5B     1

--STATISTICS--

  PROGRAM-UNIT LENGTH              137B =  95
  CM LABELLED COMMON LENGTH        132B =  90
  CM STORAGE USED                60680B = 24960
  COMPILE TIME                   0.061 SECONDS
```

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
                                            INTEGER
 J            4B   /SSASUSE/ SAU            INTEGER
 K            2B   /SSASUSE/ SAU            INTEGER
 NLOUD        1B   /RRAY2/                  INTEGER
 NSYS         6B   /SYS1/                   INTEGER
 NTAPE        1B   /CNTRL/
```

```
      SUBROUTINE CSET(I)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE SETS THE COMMENTS ACCESSED BY SUBROUTINE       *
C*   CRAY(I).  SEE CRAY(I) FOR A DESCRIPTION OF THEM.            *
C*                                                               *
C*                                                               *
C*   CALLED BY:  RDTTL,RDU,RDUT                                  *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,NSYS,NTAPE                                  *
C*                                                               *
C*          MODIFIED:  ARRAY,ISIN,ISTART,ISTOP,J,K               *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/ARRAY1/ARRAY(80)
      COMMON/ARRAY2/ISIN,NLOUD
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 ARRAY
      ISTART=8+NSYS
      IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
1     ISTOP=ISTART+7
      IF(ISTOP.GT.64)THEN
      WRITE(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
      ISIN=I
      RETURN
      ENDIF
      PRINT*,'?'
      READ(*,'($A16)')(ARRAY(J),J=ISTART,ISTOP)
      IF(ARRAY(ISTART).EQ.' ')THEN
      WRITE(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
      ISIN=I
      RETURN
      ENDIF
      ISTART=ISTART+8
      GO TO 1
99    PRINT*,'I/O ERROR IN CSET'
      RETURN
      END
```

```
        SUBROUTINE CSET

--VARIABLE MAP--(LO-A)
 -NAME----ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ARRAY        0B   /RRAY1/                    CHARS10        20
  I            1    DUMMY-ARG                  INTEGER
  ISAUD        0B   /CNTRL/                    INTEGER
  ISIN         0B   /RRAY2/                    INTEGER
  ISTART       1B   /SEABUSE/ SAU              INTEGER
  ISTOP        3B   /SEABUSE/ SAU              INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  J            4B   /SEABUSE/ SAU              INTEGER
  K            2B   /SEABUSE/ SAU              INTEGER
  NLOUD        1B   /RRAY2/                    INTEGER
  NSYS         0B   /SYS1/                     INTEGER
  NTIME        1B   /CNTRL/                    INTEGER

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1     22B                     31
    99    111B                     46

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  CSET        5B     1

--STATISTICS--

  PROGRAM-UNIT LENGTH            170B =  120
  CM LABELLED COMMON LENGTH      132B =   90
  CM STORAGE USED              606000 =  24960
  COMPILE TIME                0.070 SECONDS
```

```
      SUBROUTINE FIND
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE FINDS THE NODE ASSOCIATED WITH THE NRN HELD    *
C*   IN THE VARIABLE INNRN.  IF THE NODE DOES NOT EXIST, THE     *
C*   IFADD AND NDIFF FLAGS ARE SET, AND THE FOUND NODE IS THE    *
C*   LAST EXISTING NODE ON THE BRANCH THAT WOULD LEAD TO THE     *
C*   NODE ASSOCIATED WITH THE NRN STORED IN INNRN.               *
C*                                                               *
C*                                                               *
C*   CALLED BY:  NODIN                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  INNRN,NLVLS                                   *
C*                                                               *
C*          MODIFIED:  IDATA,IFADD,IFIND,LEVEL,LVL,NDIFF         *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      IFIND=1
      NDIFF=0
      IFADD=2
      DO 10 J=1,NLVLS
1     IF(INNRN(J).EQ.IRAY(IFIND,1))THEN
      LVL=J
      LEVEL(LVL,1)=IFIND
      LEVEL(LVL,2)=0
      IF(J.EQ.NLVLS)THEN
      GO TO 999
      ELSE
      IF(IRAY(IFIND,2).LE.0)THEN
      NDIFF=NLVLS-J
      GO TO 999
      ELSE
      LEVEL(LVL,2)=1
      IFIND=IRAY(IFIND,2)
      ENDIF
      ENDIF
      ELSE
      IF(IRAY(IFIND,3).GT.0)THEN
      IFIND=IRAY(IFIND,3)
      GO TO 1
      ELSE
      IFADD=3
      NDIFF=NLVLS-J+1
      GO TO 999
      ENDIF
      ENDIF
```

```
      SUBROUTINE FIND

 58          10    CONTINUE
 59          999   IDATA=0
 60                IF(IRAY(IFIND,2).LE.0)IDATA=1
 61                RETURN
 62                END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ICONT       0B  /NEX/                      INTEGER
  IDATA       1B  /NEX/                      INTEGER
  IFADD      27B  /LEVEL/                    INTEGER
  IFIND      25B  /LEVEL/                    INTEGER
  INNRM       1B  /LEVEL/                    INTEGER        20
  ITOTL       2B  /NEX/                      INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  J           1B  /S$ASV2E/ SAV              INTEGER
  LEVEL      31B  /LEVEL/                    INTEGER        60
  LVL        30B  /LEVEL/                    INTEGER
  NDIFF      26B  /LEVEL/                    INTEGER
  NLVLS       0B  /LEVEL/                    INTEGER
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  IRAY      INTEGER        2      FUNCTION
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1      22B                     33
    10   INACTIVE   DO-TERM         58
   999     110B                     59
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FIND        5B      0
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH              142B =  98
  CM LABELLED COMMON LENGTH        132B =  90
  CM STORAGE USED                 606420 = 24960
  COMPILE TIME                    0.067 SECONDS
```

```
      SUBROUTINE HELP
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE PROVIDES A BRIEF DESCRIPTION OF THE MAIN       *
C*   OPTIONS.  IT IS CALLED DIRECTLY IF OPTION ***HEL*** IS      *
C*   INPUT.  IT IS AUTOMATICALLY INVOKED BY MADAM IF THE         *
C*   USER INPUTS THREE SUCCESSIVE UNACCEPTABLE ENTRIES TO        *
C*   MADAM.                                                      *
C*                                                               *
C*                                                               *
C*   CALLED BY:  MADAM                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  USER                                          *
C*                                                               *
C*          MODIFIED:  CH                                        *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/P/USER
      SAVE
      CHARACTER CH,USER*10
      PRINT*,'THE FOLLOWING IS A BRIEF EXPLANATION OF THE OPTIONS. FOR'
      PRINT*,'MORE DETAIL, SEE THE USER''S MANUAL, ',USER,'.'
      PRINT*,' '
      PRINT*,'*** ATT *** ATTRIBUTE LABEL ENTRY'
      PRINT*,' '
      PRINT*,'*** COP *** COPIES ONE NODE TO ANOTHER'
      PRINT*,' '
      PRINT*,'*** DIS *** DISPLAY OF ONE NODE'
      PRINT*,' '
      PRINT*,'*** DON *** DONE WITH WORK, SAVE ALL FILES'
      PRINT*,' '
      PRINT*,'*** HEL *** WILL REPEAT THIS INFORMATION'
      PRINT*,' '
      PRINT*,'*** MOD *** MODIFIES EXISTING TREE, NODE BY NODE'
      PRINT*,' '
      PRINT*,'*** NEW *** NEW TREE BUILDING DRIVER'
      PRINT*,' '
      PRINT*,'*** NUM *** SAME AS NEW, BUT WITH WEIGHTS AND VALUES'
      PRINT*,' '
      PRINT*,'      (PRESS ANY LETTER TO CONTINUE)'
      PRINT*,'      ?'
      READ(2,'(A1)')CH
      PRINT*,'*** PRU *** PRUNES THE TREE NODES'
      PRINT*,' '
      PRINT*,'*** REV *** REVIEW PRINT OF TREE'
      PRINT*,' '
      PRINT*,'*** SEL *** SELECT TREE FILE (STORED DATA)'
      PRINT*,' '
      PRINT*,'*** SEN *** CONDUCT SENSITIVITY ANALYSIS'
```

SUBROUTINE HELP

```
58                PRINT*,' '
59                PRINT*,'*** SPA *** ADDS DOWNLINKS TO EXISTING NODES'
60                PRINT*,' '
61                PRINT*,'*** STA *** PROVIDES TREE STATISTICS'
62                PRINT*,' '
63                PRINT*,'*** SYS *** INPUT ALTERNATIVE SYSTEMS'
64                PRINT*,' '
65                PRINT*,'*** TTL *** DATA FILE TITLE ENTRY'
66                PRINT*,' '
67                PRINT*,'*** WVC *** LOADS WEIGHTS AND VALUES, DOES CALCULATIONS'
68                PRINT*,' '
69                PRINT*,'     (PRESS ANY LETTER TO CONTINUE)'
70                PRINT*,'     ?'
71                READ(*,'(A1)')CH
72                RETURN
73                END
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  CH          1B  /SBASUSE/ SAV              CHAR*1
  USER        0B  /?/                        CHAR*10

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  HELP        SB     0

--STATISTICS--

  PROGRAM-UNIT LENGTH            421B =  273
  CM LABELLED COMMON LENGTH        3B =    3
  CM STORAGE USED              60600B =  24960
  COMPILE TIME                  0.109 SECONDS
```

```
      SUBROUTINE INTRO
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ESTABLISHES AN IDENTIFICATION OF THE USER      *
C*   FOR EMBELLISHING SEVERAL PROMPTS.                           *
C*                                                               *
C*                                                               *
C*   CALLED BY:  MADAM                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  USER                                      *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/P/USER
      SAVE
      CHARACTER*10 USER
      PRINT*,'WHAT IS YOUR NAME, PLEASE?'
      READ(*,'(A10)')USER
      PRINT*,'THANK YOU, ',USER,'.  WE WILL NOW BEGIN THE '
      PRINT*,'DECISION ANALYSIS.'
      RETURN
      END
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  USER        6B  /P/                        CHAR*10

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  INTRO       5B     0

--STATISTICS--

  PROGRAM-UNIT LENGTH              62B =  50
  CM LABELLED COMMON LENGTH         2B =  2
  CM STORAGE USED                 006003 =  24550
  COMPILE TIME                    0.020 SECONDS
```

```
      FUNCTION IRAY(I,J)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS FUNCTION IS USED TO SIMULATE AN ARRAY VARIABLE         *
C*   IRAY(I,J).  IRAY(*,1) CONTAINS THE MSN DIGIT ASSOCIATED     *
C*   WITH NODE *.  IRAY(*,2) POINTS TO THE FIRST DESCENDANT      *
C*   OF NODE *.  IRAY(*,3) POINTS TO THE FIRST CROSSLINK OF      *
C*   NODE *.  IRAY(*,4) POINTS TO THAT NODE WHICH HAS NODE *     *
C*   AS ITS FIRST DESCENDANT OR CROSSLINK.  THESE VALUES ARE     *
C*   SET BY THE ROUTINE ISET(I,J,IN).                            *
C*                                                               *
C*                                                               *
C*   CALLED BY:  COPYR,FIND,NEXT,PREMEX,PRETOT,PRUNE,GMVEC,      *
C*               RDWT,DISPLA,DSPLOT,COMPUT,DETMOD,NODSET         *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,J,NTAPE                                     *
C*                                                               *
C*          MODIFIED:  IRRAY,ISIN,K                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/IRRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(IRRAY(K),K=1,80)
      IRAY=IRRAY(J)
      ISIN=I
      RETURN
99    PRINT*,'I/O ERROR IN IRAY'
      RETURN
      END
```

```
                    FUNCTION IRAY

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
    99      86B                     37.

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---
  IRAY        6B      2

--STATISTICS--
  PROGRAM-UNIT LENGTH              51B =  41
  CM LABELLED COMMON LENGTH       126B =  86
  CM STORAGE USED                 60696B = 24960
  COMPILE TIME                    0.02_ SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  I          1   DUMMY-ARG                   INTEGER
  IRAY       6B  /SSAMUSE/ SAU               INTEGER
  IRRAY      6B  /RRAY1/                     INTEGER        80
  ISAVD      0B  /CNTRL/                     INTEGER
  ISIN       CB  /RRAY2/                     INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  J          2   DUMMY-ARG                   INTEGER
  K          1B  /SSAMUSE/ SAU               INTEGER
  NLOUD      1B  /RRAY2/                     INTEGER
  NTAPE      1B  /CNTRL/                     INTEGER
```

```
      1                SUBROUTINE ISET(I,J,IM)
      2          C
      3          C
      4          C*****************************************************************
      5          C*                                                               *
      6          C*                                                               *
      7          C*   THIS ROUTINE IS USED TO SET THE VALUES OF THE PSUEDO-       *
      8          C*   ARRAY VARIABLE IRAY(I,J).  SEE FUNCTION IRAY(I,J) FOR       *
      9          C*   A DESCRIPTION OF THE VALUES.                                *
     10          C*                                                               *
     11          C*                                                               *
     12          C*   CALLED BY:  MASTER,MODIFY,PRUNE,SPAN                        *
     13          C*                                                               *
     14          C*                                                               *
     15          C*   VARIABLES                                                   *
     16          C*          USED:  I,IM,J,NTAPE                                  *
     17          C*                                                               *
     18          C*          MODIFIED:  IRRAY,ISIN,K                              *
     19          C*                                                               *
     20          C*                                                               *
     21          C*****************************************************************
     22          C
     23          C
     24                COMMON/CNTRL/ISAVD,NTAPE
     25                COMMON/RRAY1/IRRAY(80)
     26                COMMON/RRAY2/ISIN,NLOUD
     27                SAVE
     28                IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(IRRAY(K),K=1,80)
     29                IRRAY(J)=IM
     30                WRITE(NTAPE,REC=I,ERR=99)(IRRAY(K),K=1,80)
     31                ISIN=I
     32                RETURN
     33          99    PRINT*,'I/O ERROR IN ISET'
     34                RETURN
     35                END
```

```
                                                    SUBROUTINE ISET

                                          --ENTRY POINTS--(LO=A)
                                           -NAME---ADDRESS--ARGS---

                                            ISET        5B      3

                                          --STATISTICS--

                                            PROGRAM-UNIT LENGTH              64B =  52
                                            CM LABELLED COMMON LENGTH       12CB =  86
                                            CM STORAGE USED                 60600B =  24960
                                            COMPILE TIME                    0.039 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I          1    DUMMY-ARG                  INTEGER
  IM         3    DUMMY-ARG                  INTEGER
  IRRAY      0B   /RRAY1/                    INTEGER        80
  ISAVD      0B   /CNTRL/                    INTEGER
  ISIN       0B   /RRAY2/                    INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  J          2    DUMMY-ARG                  INTEGER
  K          1B   /SBASVSE/ SAV              INTEGER
  NLOUD      1B   /RRAY2/                    INTEGER
  NTAPE      1B   /CNTRL/                    INTEGER

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    99     33B                    33
```

```
 1                FUNCTION LABEL(I)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS FUNCTION ACCESSES THE NAME OF THE ATTRIBUTE           *
 8          C*   ASSOCIATED WITH NODE I.  IT SIMULATES AN ARRAY VARIABLE     *
 9          C*   LABEL(I).  THE VALUES ARE SET BY ROUTINE LSET(I,LABELT).    *
10          C*                                                               *
11          C*                                                               *
12          C*   CALLED BY:  RDU,COMPUT,DETNOD                               *
13          C*                                                               *
14          C*                                                               *
15          C*   VARIABLES                                                   *
16          C*          USED:  I,NTAPE                                       *
17          C*                                                               *
18          C*          MODIFIED:  IRRAY,ISIN,K                              *
19          C*                                                               *
20          C*                                                               *
21          C*****************************************************************
22          C
23          C
24                COMMON/CNTRL/ISAVD,NTAPE
25                COMMON/RRAY1/IRRAY(80)
26                COMMON/RRAY2/ISIN,NLOUD
27                SAVE
28                CHARACTER*10 LABEL,IRRAY
29                IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(IRRAY(K),K=1,20)
30                LABEL=IRRAY(7)
31                ISIN=I
32                RETURN
33          99    PRINT*,'I/O ERROR IN LABEL'
34                RETURN
35                END
```

FUNCTION LABEL

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---
  LABEL       58      1

--STATISTICS--
  PROGRAM-UNIT LENGTH              60D =  48
  CM LABELLED COMMON LENGTH       12BD =  96
  CM STORAGE USED                 606E0D = 24960
  COMPILE TIME                    0.035 SECONDS

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  I          2   DUMMY-ARG                   INTEGER
  IRRAY      0B  /RRAY1/                     CHAR*10        80
  ISAVD      0B  /CNTRL/                     INTEGER
  ISIN       0B  /RRAY2/                     INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  K          1B  /S$AGUSE/ SAV               INTEGER
  LABEL      0B                              CHAR*10
  NLOUD      1B  /RRAY2/                     INTEGER
  NTAPE      1B  /CNTRL/                     INTEGER

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
    99     26B                     23
```

```
      SUBROUTINE LSET(I,LABELT)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE SETS THE VALUES OF THE PSUEDO-ARRAY            *
C*   VARIABLE LABEL(I).  LSET(I,LABELT) SETS THE NAME OF         *
C*   THE ATTRIBUTE ASSOCIATED WITH NODE I TO LABELT.             *
C*   THESE VALUES ARE ACCESSED VIA FUNCTION LABEL(I).            *
C*                                                               *
C*                                                               *
C*   CALLED BY:  ATTSET,MASTER                                   *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,LABELT,NTAPE                                *
C*                                                               *
C*          MODIFIED:  IRRAY,ISIN,J                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/IRRAY1/IRRAY(80)
      COMMON/IRRAY2/ISIN,NLOUD
      SAVE
      CHARACTER*10 LABELT,IRRAY
      IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(IRRAY(J),J=1,80)
      IRRAY(7)=LABELT
      WRITE(NTAPE,REC=I,ERR=99)(IRRAY(J),J=1,80)
      ISIN=I
      RETURN
99    PRINT*,'I/O ERROR IN LSET'
      RETURN
      END
```

```
SUBROUTINE LSET

--ENTRY POINTS--(LO=A)
 -NAME----ADDRESS--ARGS---

 LSET        58      8

--STATISTICS--

  PROGRAM-UNIT LENGTH                 103B =  67
  CM LABELLED COMMON LENGTH           126B =  86
  CM STORAGE USED                   60600B = 24960
  COMPILE TIME                      0.042 SECONDS
```

```
-VARIABLE MAP--(LO=A)
-NAME----ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 I          1   DUMMY-ARG                  INTEGER
 IRRAY      0B  /RRAY1/                    CHAR*10          80
 ISAVD      0B  /CNTRL/                    INTEGER
 ISIN       0B  /RRAY2/                    INTEGER

 J          1B  /SAVE/ SAV                 INTEGER
 LABELT     2   DUMMY-ARG                  CHAR*10
 NLOUD      1B  /RRAY2/                    INTEGER
 NTAPE      1B  /CNTRL/                    INTEGER
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES-----DEF

   99     36B                       36
```

```
      SUBROUTINE NEXT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS USED TO PERFORM A NODE-BY-NODE DEPTH-       *
C*   FIRST SEARCH OF THE TREE.  A NEW NODE IS ACCESSED EACH      *
C*   TIME THAT THIS ROUTINE IS CALLED.                           *
C*                                                               *
C*                                                               *
C*   CALLED BY:  COPYR,CHECK,DELSYS,GRAFIX,RDATT,SPAN,CALC,      *
C*               UVLOD1,NUMREV                                   *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  NLULS                                         *
C*                                                               *
C*          MODIFIED:  ICONT,IDATA,IFADD,IFIND,INNRN,LEVEL,      *
C*                     LUL                                       *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      ICONT=0
      IDATA=0
      IF(LUL.GT.0)THEN
      IF((LEVEL(LUL,2).NE.1).AND.(IRAY(LEVEL(LUL,1),2).GT.0))THEN
      LEVEL(LUL,2)=1
      LEVEL(LUL+1,1)=IRAY(LEVEL(LUL,1),2)
      LEVEL(LUL+1,2)=0
      LUL=LUL+1
      IFIND=LEVEL(LUL,1)
      ICONT=1
      IFADD=2
      GO TO 99
      ENDIF
1     IF((LUL.EQ.1).AND.(NLULS.NE.0))GO TO 99
      IF(IRAY(IFIND,3).GT.0)THEN
      LEVEL(LUL,1)=IRAY(IFIND,3)
      LEVEL(LUL,2)=0
      IFIND=LEVEL(LUL,1)
      ICONT=1
      IFADD=3
      GO TO 99
      ENDIF
      LUL=LUL-1
      IF(LUL.GT.0)THEN
      IFIND=LEVEL(LUL,1)
      GO TO 1
      ENDIF
99    IF(IRAY(IFIND,2).LE.0)IDATA=1
      IF((LUL.GT.0).AND.(LUL.LT.21))INNRN(LUL)=IRAY(IFIND,1)
```

```
      SUBROUTINE NEXT

   58                ENDIF
   59                RETURN
   60                END
```

```
--VARIABLE MAP--(LO*A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ICONT       6B  /NEX/                      INTEGER
  IDATA       1B  /NEX/                      INTEGER
  IFADD      67B  /LEVEL/                    INTEGER
  IFIND      25B  /LEVEL/                    INTEGER
  INNAM       1B  /LEVEL/                    INTEGER        20

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ITOTL       2B  /NEX/                      INTEGER
  LEVEL      31B  /LEVEL/                    INTEGER        60
  LVL        30B  /LEVEL/                    INTEGER
  NDIFF      26B  /LEVEL/                    INTEGER
  NLVLS       0B  /LEVEL/                    INTEGER
```

```
--PROCEDURES--(LO*A)
 -NAME------TYPE--------ARGS-- ----CLASS-----

  IRAY      INTEGER        2      FUNCTION
```

```
--STATEMENT LABELS--(LO*A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1     54B                      42
    99    113B                      56
```

```
--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  NEXT        5B     0
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH              156B =  110
  CM LABELLED COMMON LENGTH        131B =   89
  CM STORAGE USED                 60600B =  24960
  COMPILE TIME                   0.003 SECONDS
```

```
      SUBROUTINE NODIN
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE ELICITS A NRN VECTOR.  THE ROUTINE FIND        *
C*   IS INVOKED TO SEARCH FOR THE NODE ASSOCIATED WITH THE       *
C*   INPUT NRN VECTOR.                                           *
C*                                                               *
C*   CALLED BY:  COPYR,PRENEX,MODIFY,PRUNE,NUM                   *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  ICONT,INNRN,NLULS                         *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      PRINT*,' '
      PRINT*,'ENTER...NRN?'
      ICONT=0
      NLULS=0
1     NLULS=NLULS+1
      READ*,INNRN(NLULS)
      IF(INNRN(NLULS).GT.0)GO TO 1
      NLULS=NLULS-1
      IF(INNRN(1).GT.0)CALL FIND
      RETURN
      END
```

```
                    SUBROUTINE NODIN

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
      1     16B                     31

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---
  NODIN      5B      0

--STATISTICS--
  PROGRAM-UNIT LENGTH                 64B =  52
  CM LABELLED COMMON LENGTH          131B =  89
  CM STORAGE USED                   606000B = 24960
  COMPILE TIME                      0.041 SECONDS

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ICONT      0B  /NEX/                        INTEGER
  IDATA      1B  /NEX/                        INTEGER
  IFADD     27B  /LEVEL/                      INTEGER
  IFIND     25B  /LEVEL/                      INTEGER
  INNRN      1B  /LEVEL/                      INTEGER        20

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ITOTL      2B  /NEX/                        INTEGER
  LEVEL     31B  /LEVEL/                      INTEGER        60
  LUL       30B  /LEVEL/                      INTEGER
  NDIFF     26B  /LEVEL/                      INTEGER
  NLULS      0B  /LEVEL/                      INTEGER

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----
  FIND                    0       SUBROUTINE
```

```
 1                SUBROUTINE OBJECT(I,OBJCTV)
 2          C
 3          C
 4          C*****************************************************
 5          C*                                                   *
 6          C*                                                   *
 7          C*   THIS ROUTINE ACCESSES THE OBJECTIVE ASSOCIATED WITH A *
 8          C*   PARTICULAR NODE.  OBJECT(*,OBJCTV) STORES THE OBJECTIVE*
 9          C*   OF NODE * IN THE VARIABLE OBJCTV.  THE OBJECTIVES ARE *
10          C*   SET VIA THE SUBROUTINE OSET(I,OBJCTV).              *
11          C*                                                       *
12          C*                                                       *
13          C*   CALLED BY:  COPYR,GRAFIX,PRUNE,RDATT,SPAN,GMVEC,RDV,  *
14          C*               RDUT,DISPLA,MXXREV,DETNOD,GRAPH,HEADER,  *
15          C*               NODSET                                  *
16          C*                                                       *
17          C*                                                       *
18          C*   VARIABLES                                           *
19          C*         USED:  I,NTAPE                                *
20          C*                                                       *
21          C*         MODIFIED:  ISIN,J,OBJCTV                      *
22          C*                                                       *
23          C*                                                       *
24          C*****************************************************
25          C
26          C
27                COMMON/CNTRL/ISAVD,NTAPE
28                COMMON/RRAY1/IRRAY(80)
29                COMMON/RRAY2/ISIN,NLOUD
30                SAVE
31                CHARACTER*10 OBJCTV(16),IRRAY
32                IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(IRRAY(J),J=1,80)
33                DO 10 J=1,16
34          10    OBJCTV(J)=IRRAY(J+64)
35                ISIN=I
36                RETURN
37          99    PRINT*,'I/O ERROR IN OBJECT'
38                RETURN
39                END
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        34
     99     448                    37

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  OBJECT      58     2

--STATISTICS--

  PROGRAM-UNIT LENGTH              108B =  70
  CM LABELLED COMMON LENGTH        126B =  86
  CM STORAGE USED                 606002 = 24960
  COMPILE TIME                    0.043 SECONDS
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

I          1    DUMMY-ARG                  INTEGER
IRRAY      0B   /RRAY1/                    CHAR*10        80
ISAVD      0B   /CNTRL/                    INTEGER
ISIN       0B   /RRAY2/                    INTEGER
J          1B   /SSASUBE/ SAV              INTEGER
NLOUD      1B   /RRAY2/                    INTEGER
NTAPE      1B   /CNTRL/                    INTEGER
OBJCTV     2    DUMMY-ARG                  CHAR*10        16
```

```
      SUBROUTINE OSET(I,OBJCTV)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE SETS THE OBJECTIVE TO BE ASSOCIATED WITH A     *
C*   PARTICULAR NODE.  OSET(*,OBJCTV) SETS THE OBJECTIVE OF      *
C*   NODE * TO BE OBJCTV.  THESE OBJECTIVES ARE ACCESSED BY      *
C*   SUBROUTINE OBJECT(I,OBJCTV).                                *
C*                                                               *
C*                                                               *
C*   CALLED BY:  COPYR,MODIFY,SPAN                               *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  I,NTAPE,OBJCTV                                *
C*                                                               *
C*          MODIFIED:  IRRAY,ISIN,J                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/IRRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      CHARACTER*10 OBJCTV(16),IRRAY
      IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(IRRAY(J),J=1,80)
      DO 10 J=1,16
10    IRRAY(J+64)=OBJCTV(J)
      WRITE(NTAPE,REC=I,ERR=99)(IRRAY(J),J=1,80)
      ISIN=I
      RETURN
99    PRINT*,'I/O ERROR IN OSET'
      RETURN
      END
```

SUBROUTINE OSET

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  OSET        5B      2

--STATISTICS--

  PROGRAM-UNIT LENGTH              1818 =  81
  CM LABELLED COMMON LENGTH        1208 =  96
  CM STORAGE USED                 606608 =  24960
  COMPILE TIME                    0.044 SECONDS

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I          1   DUMMY-ARG                  INTEGER
  IRRAY      6B  /RRAY1/                    CHAR*10        80
  ISAVD      0B  /CNTRL/                    INTEGER
  ISIN       0B  /RRAY2/                    INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  J          1B  /SBASUBE/ SAV              INTEGER
  NLOUD      1B  /RRAY2/                    INTEGER
  NTAPE      1B  /CNTRL/                    INTEGER
  OBJCTV     2   DUMMY-ARG                  CHAR*10        16

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        32
    99     52B                    38
```

```
 1                FUNCTION PARAM(I,J)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS FUNCTION SIMULATES A PSUEDO-ARRAY VARIABLE            *
 8          C*   PARAM(I,J).  PARAM(*,1) IS THE INTERCEPT (B0) OF THE        *
 9          C*   LINEAR REGRESSION REPRESENTING THE INDIVIDUAL VALUE         *
10          C*   FUNCTION O: ATTRIBUTE *.  PARAM(*,2)  IS THE SLOPE          *
11          C*   COEFFICIENT (B1).  PARAM(*,3) IS THE SUM OF SQUARED         *
12          C*   ERROR.  PARAM(*,4) IS THE INDEX DENOTING THE                *
13          C*   TRANSFORMATION ON THE INDEPENDENT VARIABLE.  VALUES         *
14          C*   ARE SET BY THE SUBROUTINE PSET(I,J,V).                      *
15          C*                                                               *
16          C*                                                               *
17          C*   CALLED BY:  VALU,XLEVEL,PICTUR,VALUE                        *
18          C*                                                               *
19          C*                                                               *
20          C*   VARIABLES                                                   *
21          C*          USED:  I,NLOUD,NTAPE                                 *
22          C*                                                               *
23          C*          MODIFIED:  ARRAY,ISIN,J,JJ,K                         *
24          C*                                                               *
25          C*                                                               *
26          C*****************************************************************
27          C
28          C
29                COMMON/CNTRL/ISAVD,NTAPE
30                COMMON/PRAY1/ARRAY(80)
31                COMMON/PRAY2/ISIN,NLOUD
32                SAVE
33                JJ=J
34                J=NLOUD+J-1
35                IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(ARRAY(K),K=1,80)
36                PARAM=ARRAY(I)
37                ISIN=J
38                J=JJ
39                RETURN
40          99    PRINT*,'I/O ERROR IN PARAM'
41                RETURN
42                END
```

```
                FUNCTION PARAM

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
     99      358                    40

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS--
  PARAM       60      2

--STATISTICS--
  PROGRAM-UNIT LENGTH              608 =  48
  CM LABELLED COMMON LENGTH       1278 =  87
  CM STORAGE USED                 606608 =  24880
  COMPILE TIME                    0.037 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ARRAY       08  /RRAY1/                    REAL            80
  I           1   DUMMY-ARG                  INTEGER
  ISAVD       08  /CNTRL/                    INTEGER
  ISIN        08  /RRAY2/                    INTEGER
  J           1   DUMMY-ARG                  INTEGER
  JJ          18  /SBASUBE/ SAV              INTEGER
  K           28  /SBASUBE/ SAV              INTEGER
  NLOUD       18  /RRAY2/                    INTEGER
  NTAPE       18  /CNTRL/                    INTEGER
  PARAM       08  /SBASUBE/ SAV              REAL
```

```
      SUBROUTINE PRENEX
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE INITIALIZES THE PROGRAM TO BEGIN A             *
C*   DEPTH-FIRST SEARCH, USING SUBROUTINE NEXT, FROM A           *
C*   DESIRED NODE.  ANALOGOUS TO PRETOT.                         *
C*                                                               *
C*                                                               *
C*   CALLED BY:  COPYR,SPAN,WJLOD1,MUR,DETNOD,NODSET             *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  NDIFF,NLVLS,NNODES                            *
C*                                                               *
C*          MODIFIED:  ICONT,IDATA,IFIND,INNRN,ITOTL,LEVEL,      *
C*                     LVL                                       *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      IF(NNODES.GT.1)THEN
      ICONT=0
      ITOTL=0
      CALL NODIN
      IF(NLVLS.GT.0)THEN
      IF(NDIFF.LE.0)THEN
      LEVEL(1,1)=IFIND
      LEVEL(1,2)=0
      LVL=1
      IFIND=LEVEL(1,1)
      INNRN(1)=IRAY(IFIND,1)
      IF(IRAY(IFIND,2).LE.0)IDATA=1
      ICONT=1
      ENDIF
      ENDIF
      ELSE
      PRINT*,'NO TREE IN ARRAY'
      ICONT=0
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE PRENEX

--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ICONT       0B   /NEX/                      INTEGER
  IDATA       1B   /NEX/                      INTEGER
  ITADD      27B   /LEVEL/                    INTEGER
  IFIND      25B   /LEVEL/                    INTEGER          20
  INMRN       1B   /LEVEL/                    INTEGER
  ITOTL       2B   /NEX/

--PROCEDURES--(LO-A)
 -NAME------TYPE---------ARGS------CLASS-----

  IRAY      INTEGER          2      FUNCTION
  NODIM                      0      SUBROUTINE

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  PRENEX      5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH                103B =  67
  CM LABELLED COMMON LENGTH          133B =  91
  CM STORAGE USED                  606000B = 24960
  COMPILE TIME                     0.049 SECONDS
```

```
-NAME---ADDRESS--BLOCK-----PROPERTIES--------TYPE---------SIZE

 LEVEL      31B   /LEVEL/                     INTEGER          50
 LVL        30B   /LEVEL/                     INTEGER
 MDEEP       1B   /C/                         INTEGER
 MDIFF     26B   /LEVEL/                      INTEGER
 MLVLS      0B   /LEVEL/                      INTEGER
 MMODES     6B   /C/                          INTEGER
```

```
      SUBROUTINE PRETOT
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE INITIALIZES THE PROGRAM TO BEGIN A DEPTH-      *
C*   FIRST SEARCH, USING SUBROUTINE NEXT, FROM THE ROOT          *
C*   NODE.  ANALOGOUS TO PREMEX.                                 *
C*                                                               *
C*   CALLED BY:  DELSYS,RDATT,RDSYSL,SPAN,CALC,UVLOD1,NUMREV,    *
C*               SENSTU                                          *
C*                                                               *
C*   VARIABLES                                                   *
C*         USED:  NNODES                                         *
C*                                                               *
C*         MODIFIED:  ICONT,IDATA,IFIND,INNRN,ITOTL,LEVEL,       *
C*                    LVL,NDIFF,NLVLS                            *
C*                                                               *
C*****************************************************************
C
      COMMON/C/NNODES,NDEEP
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      ICONT=0
      IDATA=0
      LEVEL(1,1)=2
      LEVEL(1,2)=0
      LVL=1
      NLVLS=0
      NDIFF=1
      IFIND=LEVEL(1,1)
      IF(IRAY(IFIND,2).LE.0)IDATA=1
      IF(NNODES.GT.1)ICONT=1
      INNRN(1)=IRAY(IFIND,1)
      ITOTL=1
      RETURN
      END
```

SUBROUTINE PRETOT

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
IRAY	INTEGER	2	FUNCTION

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
PRETOT	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	60B	48
CM LABELLED COMMON LENGTH	133B	91
CM STORAGE USED	606008	24960
COMPILE TIME	0.045 SECONDS	

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ICONT	6B	/NEX/		INTEGER	
IDATA	1B	/NEX/		INTEGER	
IFADD	87B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20
ITOTL	2B	/NEX/		INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	32B	/LEVEL/		INTEGER	
NDEEP	1B	/C/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	6B	/LEVEL/		INTEGER	
NNODES	6B	/C/		INTEGER	

```
      1                 SUBROUTINE PSET(I,J,V)
      2           C
      3           C
      4           C*****************************************************
      5           C*                                                   *
      6           C*                                                   *
      7           C*   THIS ROUTINE SETS THE VALUES OF THE PSUEDO-ARRAY *
      8           C*   VARIABLE PARAM(I,J).  PSET(I,J,V) SETS THE J-TH  *
      9           C*   PARAMETER ASSOCIATED WITH ATTRIBUTE I TO V.  SEE *
     10           C*   PARAM(I,J) FOR AN EXPLANATION OF THE PARAMETERS. *
     11           C*   THE VALUES ARE ACCESSED VIA FUNCTION PARAM(I,J). *
     12           C*                                                   *
     13           C*                                                   *
     14           C*   CALLED BY:  REQRS                               *
     15           C*                                                   *
     16           C*                                                   *
     17           C*   VARIABLES                                       *
     18           C*         USED:  NLOUD,NTAPE,V                      *
     19           C*                                                   *
     20           C*         MODIFIED:  ARRAY,ISIN,J,JJ,K              *
     21           C*                                                   *
     22           C*                                                   *
     23           C*****************************************************
     24           C
     25           C
     26                 COMMON/CNTRL/ISAUD,NTAPE
     27                 COMMON/RRAY1/ARRAY(80)
     28                 COMMON/RRAY2/ISIN,NLOUD
     29                 SAVE
     30                 JJ=J
     31                 J=NLOUD+J-1
     32                 IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(ARRAY(K),K=1,80)
     33                 ARRAY(I)=V
     34                 WRITE(NTAPE,REC=J,ERR=99)(ARRAY(K),K=1,80)
     35                 ISIN=J
     36                 J=JJ
     37                 RETURN
     38           99    PRINT*,'I/O ERROR IN PSET'
     39                 RETURN
     40                 END
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PSET        5B      3

--STATISTICS--

  PROGRAM-UNIT LENGTH                 733 =  59
  CM LABELLED COMMON LENGTH          1270 =  87
  CM STORAGE USED                  60B03B =  24963
  COMPILE TIME                     0.044 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ARRAY       0B  /RRAY1/                    REAL            50
  I           1   DUMMY-ARG                  INTEGER
  ISAUD       0B  /CNTRL/                    INTEGER
  ISIN        0B  /RRAY2/                    INTEGER
  J           B   DUMMY-ARG                  INTEGER
  JJ          1B  /SSASUSE/ SAV              INTEGER
  K           2B  /SSASUSE/ SAV              INTEGER
  NLOUD       1B  /RRAY2/                    INTEGER
  NTAPE       1B  /CNTRL/                    INTEGER
  V           3   DUMMY-ARG                  REAL
```

```
      SUBROUTINE PSET        74/74    OPT=0          FTN 5.1+528        11/23/81  19.35.01        PAGE     8
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    99     42B                     32
```

```
      1                 FUNCTION SYSLBL(I)
      2          C
      3          C
      4          C*****************************************************************
      5          C*                                                               *
      6          C*                                                               *
      7          C*    THIS FUNCTION SIMULATES AN ARRAY VARIABLE SYSLBL(I).       *
      8          C*    IT ACCESSES THE NAMES OF THE ALTERNATIVE SYSTEMS.          *
      9          C*    VALUES ARE SET BY THE SUBROUTINE SYSSET(I,SYSLBL).         *
     10          C*                                                               *
     11          C*                                                               *
     12          C*    CALLED BY:  DELSYS,NEW,RDU,DISPLA,DSPLOT,NUMREV,GRAPH,     *
     13          C*                SENVAL,TABDIS                                  *
     14          C*                                                               *
     15          C*                                                               *
     16          C*    VARIABLES                                                  *
     17          C*          USED:  NLOUD,NTAPE                                   *
     18          C*                                                               *
     19          C*          MODIFIED:  IRRAY,ISIN,J,K                            *
     20          C*                                                               *
     21          C*                                                               *
     22          C*****************************************************************
     23          C
     24          C
     25                COMMON/CNTRL/ISAVD,NTAPE
     26                COMMON/RRAY1/IRRAY(80)
     27                COMMON/RRAY2/ISIN,NLOUD
     28                SAVE
     29                CHARACTER*10 SYSLBL,IRRAY
     30                J=NLOUD+4
     31                IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(IRRAY(K),K=1,80)
     32                SYSLBL=IRRAY(I)
     33                ISIN=J
     34                RETURN
     35          99    PRINT*,'I/O ERROR IN SYSLBL'
     36                RETURN
     37                END
```

```
                                                        FUNCTION SYSLBL

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SYSLBL      58      1

--STATISTICS--

  PROGRAM-UNIT LENGTH                    72B =   58
  CM LABELLED COMMON LENGTH              127B =   87
  CM STORAGE USED                      000003 =  24860
  COMPILE TIME                         0.042 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I          2   DUMMY-ARG                  INTEGER
  IRRAY      0B  /RRAY1/                    CHAR*10        80
  ISAVD      6B  /CNTRL/                    INTEGER
  ISIN       0B  /RRAY2/                    INTEGER
  J          1B  /SSASUBE/ SAV              INTEGER
  K          2B  /SSASUBE/ SAV              INTEGER
  NLOUD      1B  /RRAY2/                    INTEGER
  NTAPE      1B  /CNTRL/                    INTEGER
  SYSLBL     0B                             CHAR*10
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     99     30B                    35
```

```
      SUBROUTINE SYSSET(I,SYSLBL)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE SETS THE VALUES OF THE PSUEDO-ARRAY            *
C*   VARIABLE SYSLBL(I).  THESE ARE THE LABELS OF THE            *
C*   ALTERNATIVE SYSTEMS.  THESE VALUES ARE ACCESSED VIA         *
C*   FUNCTION SYSLBL(I).                                         *
C*                                                               *
C*                                                               *
C*   CALLED BY:  ADDSYS,DELSYS,NEW                               *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*           USED:  NLOUD,NTAPE,SYSLBL                           *
C*                                                               *
C*           MODIFIED:  IRRAY,ISIN,J,K                           *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/IRRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      CHARACTER*10 SYSLBL,IRRAY
      J=NLOUD-4
      IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(IRRAY(K),K=1,80)
      SYSLBL=SYSLBL//'
      IRRAY(I)=SYSLBL
      WRITE(NTAPE,REC=J,ERR=99)(IRRAY(K),K=1,80)
      ISIN=J
      RETURN
99    PRINT*,'I/O ERROR IN SYSSET'
      RETURN
      END
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  $YSSET      58      2

--STATISTICS--

  PROGRAM-UNIT LENGTH                 1305 =   83
  CM LABELLED COMMON LENGTH           1875 =   87
  CM STORAGE USED                   006008 =  24960
  COMPILE TIME                      0.049 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I          1   DUMMY-ARG                   INTEGER
  IRRAY     05   /RRAY1/                     CHAR*10        80
  ISAVD     08   /CNTRL/                     INTEGER
  ISIN      09   /RRAY2/                     INTEGER
  J         1D   /SAVEUSE/ SAV               INTEGER
        SUBROUTINE SYSSET     74/74    OPT=0

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  K         2B   /SAVEUSE/ SAV               INTEGER
  NLOUD     1B   /RRAY2/                     INTEGER
  NTAPE     1B   /CNTRL/                     INTEGER
  SYSLBL    8    DUMMY-ARG                   CHAR*10
                FTN 5.1+322        11/23/81  15.35.01        PAGE     2
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    99     54D                     37
```

```
      SUBROUTINE TLOAD
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE LOADS OR INITIALIZES A DATA FILE.              *
C*                                                               *
C*   CALLED BY:  RADAM,INITT,TAPER                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  NTAPE                                         *
C*                                                               *
C*          MODIFIED:  CH,I,IRRAY,ISIN,ISTOP,J,NATT,NLOUD,       *
C*                     NNODES,NSYS                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/IRRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER CH
      NLOUD=505
      ISIN=0
      OPEN(NTAPE,ERR=99,ACCESS='DIRECT',RECL=100)
1     PRINT*,' '
      PRINT*,'OPENING FILE NUMBER ',NTAPE
      PRINT*,'IS THIS DATA NEW (N) OR STORED (S)?'
      READ(*,'(A1)')CH
      IF((CH.NE.'N').AND.(CH.NE.'S'))THEN
      PRINT*,'PLEASE ENTER 'N' OR 'S' '
      GO TO 1
      ENDIF
      IF(CH.EQ.'N')THEN
      DO 10 I=1,80
10    IRRAY(I)=0
      ISTOP=NLOUD+10
      DO 20 I=1,ISTOP
20    WRITE(NTAPE,REC=I,ERR=99)(IRRAY(J),J=1,80)
      NNODES=0
      ELSE
      I=NLOUD-4
      READ(NTAPE,REC=I,ERR=99)NNODES
      I=NLOUD-3
      READ(NTAPE,REC=I,ERR=99)NATT
      I=NLOUD-2
      READ(NTAPE,REC=I,ERR=99)NSYS
      ENDIF
      RETURN
99    PRINT*,'I/O ERROR IN TLOAD'
```

```
      SUBROUTINE TLOAD

   58                 RETURN
   60                 END

--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 CH          15  /$$ABUSE/ SAV          CHAR*1
 I           25  /$$ABUSE/ SAV          INTEGER
 IBRAY       08  /RRAY1/                INTEGER        80
 IGAUD       06  /CNTRL/                INTEGER
 ISIN        03  /RRAY2/                INTEGER
 ISTOP       32  /$$ABUSE/ SAV          INTEGER
 J           43  /$$ABUSE/ SAV          INTEGER

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1      158                       33
    10  INACTIVE   DO-TERM           43
    20  INACTIVE   DO-TERM           46
    98     1228                      57

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  TLOAD       58      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              2478 =  167
  CM LABELLED COMMON LENGTH        1358 =   93
  CM STORAGE USED                 60S006 =  24006
  COMPILE TIME                   0.009 SECONDS
```

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 NATT        52  /$$ABUSE/ SAV          INTEGER
 NDEEP       12  /C/                    INTEGER
 NLOUD       13  /RRAY2/                INTEGER
 NNODES      08  /C/                    INTEGER
 NSYS        03  /SYS1/                 INTEGER
 NTAPE       12  /CNTRL/                INTEGER
```

```
      SUBROUTINE TSAVE
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE CONVERTS THE CURRENT DATA FILE TO A LOCAL      *
C*   FILE, STORING ALL INFORMATION.                              *
C*                                                               *
C*                                                               *
C*   CALLED BY:  MADAM,INITT,TOPER                               *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*           USED:  NATT,NLOUD,NNODES,NSYS,NTAPE                 *
C*                                                               *
C*           MODIFIED:  I,ISAVD                                  *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/IRRAY(90)
      COMMON/RRAY2/ISIN,NLOUD
      COMMON/SYS1/NSYS
      SAVE
      I=NLOUD-4
      WRITE(NTAPE,REC=I,ERR=99)NNODES
      I=NLOUD-3
      WRITE(NTAPE,REC=I,ERR=99)NATT
      I=NLOUD-2
      WRITE(NTAPE,REC=I,ERR=99)NSYS
      CLOSE(NTAPE,ERR=99)
      ISAVD=1
      RETURN
99    PRINT*,'I/O ERROR IN TSAVE'
      RETURN
      END
```

```
SUBROUTINE TSAVE

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    99     338                     38

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  TSAVE       58      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              738 =  59
  CM LABELLED COMMON LENGTH       1328 =  90
  CM STORAGE USED                606008 =  24960
  COMPILE TIME                   0.048 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I          18  /SBASUBE/ SAV              INTEGER
  IRRAY      03  /RRAY1/                    INTEGER          90
  ISAVD      08  /CNTRL/                    INTEGER
  ISIN       08  /RRAY2/                    INTEGER
  NATT       28  /SBASUBE/ UNL/SAV          INTEGER

  NDEEP      18  /C/                        INTEGER
  NLOUD      18  /RRAY2/                    INTEGER
  NNODES     08  /C/                        INTEGER
  NSYS       08  /SYS1/                     INTEGER
  NTAPE      18  /CNTRL/                    INTEGER
```

```
 1                FUNCTION VALU(X,LABEL)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS FUNCTION RETURNS THE VALUE OF A PARTICULAR            *
 8          C*   ATTRIBUTE LEVEL (X) FOR AN ATTRIBUTE (LABEL) USING          *
 9          C*   THE INDIVIDUAL VALUE FUNCTION.                              *
10          C*                                                               *
11          C*                                                               *
12          C*   CALLED BY:  RDU,COMPUT                                      *
13          C*                                                               *
14          C*                                                               *
15          C*   VARIABLES                                                   *
16          C*          USED:  LABEL,NATT,X                                  *
17          C*                                                               *
18          C*          MODIFIED:  VALU                                      *
19          C*                                                               *
20          C*                                                               *
21          C*****************************************************************
22          C
23          C
24                COMMON/ATTR/NATT
25                SAVE
26                CHARACTER*10 LABEL,ATT1
27                VALU=0.0
28                DO 10 I=1,NATT
29                IF(LABEL.EQ.ATT1(I))THEN
30                IF(PARAM(I,4).EQ.1.0)X=X**2
31                IF(PARAM(I,4).EQ.-1.0)X=X**.5
32                IF(PARAM(I,4).EQ.2.0)X=EXP(X)
33                IF(PARAM(I,4).EQ.-2.0)X=ALOG(X)
34                VALU=PARAM(I,1)+PARAM(I,2)*X
35                GO TO 99
36                ENDIF
37          10    CONTINUE
38          99    RETURN
39                END
```

FUNCTION VALU

```
--PROCEDURES--(LO=A)
 -NAME------TYPE---------ARGS------CLASS-----

  ALOG      REAL           1       INTRINSIC
  ATT1      CHAR*10        1       FUNCTION
  EXP       GENERIC        1       INTRINSIC
  PARAM     REAL           2       FUNCTION

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        37
     99     752                    38

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  VALU        6B     2

--STATISTICS--

  PROGRAM-UNIT LENGTH              143B =  99
  CM LABELLED COMMON LENGTH         3B =  3
  CM STORAGE USED                 606000B =  24960
  COMPILE TIME                   0.054 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I          1B   /SBASUBE/ SAV              INTEGER
  LABEL      2    DUMMY-ARG                  CHAR*10
  NATT       6B   /ATTR/                     INTEGER
  VALU       0B   /SBASUBE/ SAV              REAL
  X          1    DUMMY-ARG                  REAL
```

```
      FUNCTION VRAY(I,J)
C
C*****************************************************************
C                                                                 *
C*                                                               *
C*   THIS FUNCTION SIMULATES THE ARRAY VARIABLE VRAY(I,J).       *
C*   VRAY(I,J) IS THE VALUE OF SYSTEM J AT NODE I.  THESE        *
C*   VALUES ARE SET BY VSET(I,J,V).                              *
C*                                                               *
C*   CALLED BY:  DELSYS,CALC,RDU,DISPLA,DSPLOT,NUMREV,COMPUT,    *
C*               SENVAL                                          *
C*                                                               *
C*   VARIABLES
C*         USED:  I,J,NTAPE                                      *
C*                                                               *
C*         MODIFIED:  ARRAY,IPULL,ISIN,K                         *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/RRAY1/ARRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      SAVE
      IPULL=7+J
      IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
      VRAY=ARRAY(IPULL)
      ISIN=I
      RETURN
99    PRINT*,'I/O ERROR IN VRAY'
      RETURN
      END
```

```
                    FUNCTION VRAY

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---
  VRAY        5B      2

--STATISTICS--
  PROGRAM-UNIT LENGTH                    54B =  44
  CM LABELLED COMMON LENGTH             127B =  87
  CM STORAGE USED                      60600B = 24960
  COMPILE TIME                         0.037 SECONDS
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ARRAY      0B   /RRAY1/                      REAL          80
  I          1    DUMMY-ARG                   INTEGER
  IPULL      1B   /SSABUSE/ SAV               INTEGER
  ISAVD      0B   /CNTRL/                     INTEGER
  ISIN       0B   /RRAY2/                     INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  J          2    DUMMY-ARG                   INTEGER
  K          2B   /SSABUSE/ SAV               INTEGER
  NLOUD      1B   /RRAY2/                     INTEGER
  NTAPE      1B   /CNTRL/                     INTEGER
  VRAY       0B   /SSABUSE/ SAV               REAL

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
   99     31B                      34
```

```
 1                SUBROUTINE USET(I,J,U)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS ROUTINE SETS THE VALUES OF THE PSUEDO-ARRAY            *
 8          C*   VARIABLE VRAY(I,J).  USET(I,J,U) SETS THE VALUE OF          *
 9          C*   SYSTEM J AT NODE I TO U.  THESE VALUES AREE ACCESSED        *
10          C*   VIA FUNCTION VRAY(I,J).                                     *
11          C*                                                               *
12          C*                                                               *
13          C*   CALLED BY:  DELSYS,RDU,CALC                                 *
14          C*                                                               *
15          C*                                                               *
16          C*   VARIABLES                                                   *
17          C*          USED:  I,J,NTAPE,U                                   *
18          C*                                                               *
19          C*          MODIFIED:  ARRAY,IPUT,ISIN,K                         *
20          C*                                                               *
21          C*                                                               *
22          C*****************************************************************
23          C
24          C
25                COMMON/CNTRL/ISAVD,NTAPE
26                COMMON/RRAY1/ARRAY(80)
27                COMMON/RRAY2/ISIN,NLOUD
28                SAVE
29                IPUT=7+J
30                IF(I.NE.ISIN)READ(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
31                ARRAY(IPUT)=U
32                WRITE(NTAPE,REC=I,ERR=99)(ARRAY(K),K=1,80)
33                ISIN=I
34                RETURN
35          99    PRINT*,'I/O ERROR IN USET'
36                RETURN
37                END
```

```
                SUBROUTINE USET

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---
  USET        5B      3

--STATISTICS--
  PROGRAM-UNIT LENGTH                 078 =  55
  CM LABELLED COMMON LENGTH          127B =  87
  CM STORAGE USED                  80S00B = 34880
  COMPILE TIME                     0.041 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
  ARRAY       0B  /RRAY1/                    REAL           80
  I           1   DUMMY-ARG                  INTEGER
  IPUT        1B  /S$ABUBE/ SAV              INTEGER
  ISAVD       0B  /CNTRL/                    INTEGER
  ISIN        0B  /RRAY2/                    INTEGER
  J           2   DUMMY-ARG                  INTEGER
  K           2B  /S$ABUGE/ SAV              INTEGER
  NLOUD       1B  /RRAY2/                    INTEGER
  NTAPE       1B  /CNTRL/                    INTEGER
  U           3   DUMMY-ARG                  REAL

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
   99     36B                     36
```

```
      FUNCTION XLEVEL(X,LABEL)
C
C*****************************************************************
C                                                                *
C                                                                *
C*  THIS FUNCTION RETURNS THE ATTRIBUTE LEVEL OF AN ATTRIBUTE     *
C*  (LABEL) FOR PARTICULAR VALUE (X) BASED ON THE INDIVIDUAL      *
C*  VALUE FUNCTION.                                               *
C*                                                                *
C*  CALLED BY:  RDV,SENVAL                                        *
C*                                                                *
C*  VARIABLES
C*          USED:  LABEL,NATT,X                                   *
C*                                                                *
C*          MODIFIED:  (NONE)                                     *
C*                                                                *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      SAVE
      CHARACTER*10 LABEL,ATT1
      XLEVEL=0.0
      DO 10 I=1,NATT
      IF(LABEL.EQ.ATT1(I))THEN
      XLEVEL=(X-PARAM(I,1))/PARAM(I,2)
      IF(PARAM(I,4).EQ.1.0)XLEVEL=XLEVEL**.5
      IF(PARAM(I,4).EQ.-1.0)XLEVEL=XLEVEL**2
      IF(PARAM(I,4).EQ.2.0)XLEVEL=ALOG(XLEVEL)
      IF(PARAM(I,4).EQ.-2.0)XLEVEL=EXP(XLEVEL)
      GO TO 99
      ENDIF
   10 CONTINUE
   99 RETURN
      END
```

FUNCTION XLEVEL

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

I          1B   /SSASUBC/ SAV          INTEGER
LABEL      2    DUMMY-ARG              CHAR*10
NATT       C8   /ATTR/                 INTEGER
X          1    DUMMY-ARG              REAL
XLEVEL     9B   /SSASUBC/ SAV          REAL
```

```
--PROCEDURES--(LO=A)
-NAME-------TYPE---------ARGS------CLASS-----

ALOG       REAL          1          INTRINSIC
ATT1       CHAR*10       1          FUNCTION
EXP        GENERIC       1          INTRINSIC
PARAM      REAL          2          FUNCTION

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

   10   INACTIVE    DO-TERM        37
   99     76B                      38

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

XLEVEL     6B     2

--STATISTICS--

PROGRAM-UNIT LENGTH              145B =  101
CM LABELLED COMMON LENGTH         3B =    3
CM STORAGE USED                 6B6000B =  24960
COMPILE TIME                    0.658 SECONDS
```

```
      OVERLAY(XFILE,1,0)
      PROGRAM DUMMY
C
C
C*****************************************************************
C*                                                               *
C*   THIS PROGRAM PROVIDES THE ENTRY POINT FOR                   *
C*   OVERLAY(1,0).                                               *
C*                                                               *
C*                                                               *
C*   CALLED BY:  (NONE)                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  CMD                                           *
C*                                                               *
C*          MODIFIED:  (NONE)                                    *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/MATT
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INARN(25),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/ONE/CMD
      COMMON/P/USER
      COMMON/RRAY1/ARRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      COMMON/SYS1/NSYS
      CHARACTER*3 CMD
      IF(CMD.EQ.'ATT')THEN
      CALL RDATT
      ELSEIF(CMD.EQ.'NEW')THEN
      CALL TAPER
      ELSEIF(CMD.EQ.'SEL')THEN
      CALL TAPER
      ELSEIF(CMD.EQ.'SPA')THEN
      CALL SPAN
      ELSEIF(CMD.EQ.'TTL')THEN
      CALL RDTTL
      ELSEIF(CMD.EQ.'MOD')THEN
      CALL MODIFY
      ELSEIF(CMD.EQ.'PRU')THEN
      CALL PRUNE
      ELSEIF(CMD.EQ.'STA')THEN
      CALL STAT
      ELSEIF(CMD.EQ.'SYS')THEN
      CALL RDSYSL
      ENDIF
      RETURN
*
RETURN IN MAIN PROGRAM -- ACTS AS END
      END
```

```
        PROGRAM DUMMY

--VARIABLE MAP--(LO-A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
 ARRAY        0B   /RRAY1/                   REAL                 80
 CMD          0B   /ONE/                     CHAR*3
 ICONT        0B   /NEX/                     INTEGER
 IDATA        0B   /NEX/                     INTEGER
 IFADD        0B   /LEVEL/                   INTEGER
 IFIND        0B   /LEVEL/                   INTEGER              20
 INNRM        0B   /LEVEL/                   INTEGER
 ISAVD        0B   /CNTRL/                   INTEGER
 ISIN         0B   /RRAY2/                   INTEGER
 ITOTL        0B   /NEX/                     INTEGER              60
 LEVEL        0B   /LEVEL/

 -NAME---ADDRESS--BLOCK-----PROPERTIES--------TYPE---------SIZE
  LVL         0B   /LEVEL/                   INTEGER
  NATT        0B   /ATTR/                    INTEGER
  NDEEP       0B   /C/                       INTEGER
  NDIFF       0B   /LEVEL/                   INTEGER
  NLOUD       0B   /RRAY2/                   INTEGER
  NLVLS       0B   /LEVEL/                   INTEGER
  NNODES      0B   /C/                       INTEGER
  NSYS        0B   /SYS1/                    INTEGER
  NTAPE       0B   /CNTRL/                   INTEGER
  USER        0B   /P/                       REAL

--PROCEDURES--(LO-A)
-NAME------TYPE--------ARGS------CLASS-----
 MODIFY                 0       SUBROUTINE
 PRUNE                  0       SUBROUTINE
 RDATT                  0       SUBROUTINE
 RDSYSL                 0       SUBROUTINE

 -NAME------TYPE--------ARGS------CLASS-----
  RDTTL                  0       SUBROUTINE
  SPAN                   0       SUBROUTINE
  STAT                   0       SUBROUTINE
  TAPER                  0       SUBROUTINE

--ENTRY POINTS--(LO-A)
-NAME---ADDRESS--ARGS---
 DUMMY      0D      0

--STATISTICS--
  PROGRAM-UNIT LENGTH                0B =  0
  CM LABELLED COMMON LENGTH       262B =  178
  CM STORAGE USED                 60600B =  24960
  COMPILE TIME                    0.046 SECONDS

      1  ANSI      ERROR IN  DUMMY
```

```
      SUBROUTINE ADDSYS
C
C*****************************************************************
C                                                                 *
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ALLOWS THE EXTENSION OF AN EXISTING SET        *
C*   OF ALTERNATIVE SYSTEMS.                                     *
C*                                                               *
C*                                                               *
C*   CALLED BY:  RDSYSL                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  ANSWER,ISAVD,NSYS                         *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 SYSLBL,ANSWER
      PRINT*,' '
      IF(NSYS.EQ.80)THEN
      PRINT*,'MAXIMUM NUMBER OF SYSTEMS EXCEEDED.'
      ELSE
      NSYS=NSYS+1
      WRITE(*,'(5(/),1X)')
      PRINT*,'ADDING SYSTEM'
      PRINT*,'LABEL?'
      READ(*,'(A10)')ANSWER
      CALL SYSSET(NSYS,ANSWER)
      PRINT*,'USE *** WWC *** FOR ENTERING VALUES'
      PRINT*,'AND RECALCULATING TREE (IF NECESSARY).'
      ISAVD=0
      ENDIF
      RETURN
      END
```

```
--VARIABLE MAP--(LO-A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

ANSWER      2B   /$$ASUBE/ SAV          CHAR*10
ISAVD       3B   /$$ASUBE/ SAV          INTEGER
NSYS        0B   /$YS1/                 INTEGER
SYSLBL      1B   /$$ASUBE/ UNUSED/SAV   CHAR*10
```

```
      SUBROUTINE ADDSYS

--PROCEDURES--(LO-A)
-NAME-------TYPE---------ARGS-------CLASS-----

 SYSSET                    2        SUBROUTINE

--ENTRY POINTS--(LO-A)
-NAME---ADDRESS--ARGS---

 ADDSYS      5B      0

--STATISTICS--

 PROGRAM-UNIT LENGTH                  131B =  89
 CM LABELLED COMMON LENGTH             5B =  5
 CM STORAGE USED                     60600B = 24960
 COMPILE TIME                        0.644 SECONDS
```

```
 1              SUBROUTINE ASK(I,X1,X2,XMID)
 2        C
 3        C
 4        C*****************************************************************
 5        C*                                                               *
 6        C*                                                               *
 7        C*   THIS ROUTINE ASKS THE QUESTIONS WHICH ELICIT THE            *
 8        C*   DATA POINTS FROM WHICH THE ESTIMATED INDIVIDUAL             *
 9        C*   VALUE FUNCTION MAY BE ESTABLISHED.                          *
10        C*                                                               *
11        C*                                                               *
12        C*   CALLED BY:  VALUE                                           *
13        C*                                                               *
14        C*                                                               *
15        C*   VARIABLES                                                   *
16        C*         USED:  X1,X2,XMAX,XMIN                                *
17        C*                                                               *
18        C*         MODIFIED:  XMID                                       *
19        C*                                                               *
20        C*                                                               *
21        C*****************************************************************
22        C
23        C
24              SAVE
25              CHARACTER ATT1*10
26        1     PRINT*,' '
27              PRINT*,'WHAT  LEVEL OF ',ATT1(I)
28              PRINT*,'WOULD BE SUCH THAT YOU WOULD FEEL THE SAME'
29              PRINT*,'AMOUNT OF CHANGE IN SATISFACTION IN MOVING'
30              PRINT*,'FROM ',X1,' TO IT, AS FROM THAT LEVEL TO ',X2,'?'
31              PRINT*,'      ?'
32              READ(*,'(F10.0)')XMID
33              XMIN=AMIN1(X1,X2)
34              XMAX=AMAX1(X1,X2)
35              IF((XMID.LT.XMIN).OR.(XMID.GT.XMAX))THEN
36              PRINT*,XMID,' IS OUTSIDE THE RANGE OF ',X1,' TO ',X2
37              GO TO 1
38              ENDIF
39              RETURN
40              END
```

```
                    SUBROUTINE ASK

--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----

  AMAX1      REAL          7       INTRINSIC
  AMIN1      REAL          ?       INTRINSIC
  ATT1       CHAR*10       1       FUNCTION

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      1      7B                     26

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  ASK         5B     4

--STATISTICS--

  PROGRAM-UNIT LENGTH              1578 =  111
  CM LABELLED COMMON LENGTH          3B =  3
  CM STORAGE USED                 60600B =  24960
  COMPILE TIME                    0.053 SECONDS
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 I          1    DUMMY-ARG                  INTEGER
 XMAX       2B   /S$A$USE/ SAV              REAL
 XMID       4    DUMMY-ARG                  REAL

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 XMIN       1B   /S$A$USE/ SAV              REAL
 X1         2    DUMMY-ARG                  REAL
 X2         3    DUMMY-ARG                  REAL
```

```
      SUBROUTINE CHECK
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS USED TO CONDUCT THE BETWEEN NODE            *
C*   CHECK WHICH VALIDATES THE SET OF CHILDREN FOR THE           *
C*   CURRENT PARENT NODE.                                        *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SPAN                                            *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  CONTRL,NNODES,USER                            *
C*                                                               *
C*          MODIFIED:  CH,ICHECK                                 *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ONE/CMD
      COMMON/LEVEL/NLVLS,IMNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/CHILD/CONTRL
      COMMON/C/NNODES,NDEEP
      COMMON/P/USER
      COMMON/CHCK/ICHECK
      SAVE
      CHARACTER*10 LABEL,USER,CH*1,CMD*3
      INTEGER CONTRL
1     IF((NNODES.GT.2).AND.(CONTRL.GT.1))THEN
      ICHECK=0
      CALL GRAFIX
2     PRINT*,' '
      PRINT*,USER,', DO THE SUBOBJECTIVES ADDRESS ALL FACETS'
      PRINT*,'OF THE PARENT OBJECTIVE?  (Y/N)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'ONLY "Y" AND "N" ARE LEGITIMATE ENTRIES, ',USER
      GO TO 2
      ENDIF
      IF(CH.NE.'Y')THEN
      ICHECK=1
      CALL MODIFY
      GO TO 1
      ELSE
3     PRINT*,' '
      PRINT*,'IS THERE ANY OVERLAP BETWEEN THE COVERAGES OF'
      PRINT*,'THE SUBOBJECTIVES, ',USER,'?  (Y/N)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'PLEASE ENTER "Y" OR "N", ',USER
      GO TO 3
```

SUBROUTINE CHECK

```
 58                 ENDIF
 59                 IF(CH.NE.'N')THEN
 60                 ICHECK=1
 61                 CALL PRUNE
 62                 GO TO 1
 63                 ELSE
 64           4     PRINT*,' '
 65                 PRINT*,USER,', ARE ALL THE SUBOBJECTIVES OPERATIONALLY '
 66                 PRINT*,'MEANINGFUL TO YOU?  (Y/N)'
 67                 PRINT*,'      ?'
 68                 READ(*,'(A1)')CH
 69                 IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
 70                 PRINT*,'ONLY 'Y' OR 'N' IS ALLOWED, ',USER
 71                 GO TO 4
 72                 ENDIF
 73                 IF(CH.NE.'Y')THEN
 74                 ICHECK=1
 75                 CALL MODIFY
 76                 GO TO 1
 77                 ELSE
 78           5     PRINT*,' '
 79                 PRINT*,'COULD ANY OF THE SUBOBJECTIVES BE IGNORED WITHOUT'
 80                 PRINT*,'SIGNIFICANTLY IMPACTING YOUR PREFERENCES, ',USER,'? (Y/N)
 81                 PRINT*,'      ?'
 82                 READ(*,'(A1)')CH
 83                 IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
 84                 PRINT*,'PLEASE ENTER 'Y' OR 'N', ',USER
 85                 GO TO 5
 86                 ENDIF
 87                 IF(CH.NE.'N')THEN
 88                 ICHECK=1
 89                 CALL PRUNE
 90                 GO TO 1
 91                 ENDIF
 92                 ENDIF
 93                 ENDIF
 94                 ENDIF
 95                 ENDIF
 96                 ICHECK=0
 97                 CALL STAT
 98                 CALL NEXT
 99                 RETURN
100                 END
```

SUBROUTINE CHECK

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
GRAFIX		0	SUBROUTINE
MODIFY		0	SUBROUTINE
NEXT		0	SUBROUTINE
PRUNE		0	SUBROUTINE
STAT		0	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
1	7B		33
2	20B		36
3	56B		50
4	114B		64
5	152B		78

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
CHECK	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	440B =	288
CM LABELLED COMMON LENGTH	12CB =	84
CM STORAGE USED	60600B =	24960
COMPILE TIME	0.157 SECONDS	

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CH	2B	/$$A*USE/ SAV		CHAR*1	
CMD	6B	/ONE/		CHAR*3	
CONTRL	0B	/CHILD/		INTEGER	
ICHECK	6B	/CHCK/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
INNRN	1B	/LEVEL/		INTEGER	
LABEL	1B	/$$ABUSE/ UNUSED/SAV		CHAR*10	20
LEVEL	31B	/LEVEL/		INTEGER	60
LUL	36B	/LEVEL/		INTEGER	
MDEEP	1B	/C/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLULS	0B	/LEVEL/		INTEGER	
NNODES	0B	/C/		INTEGER	
USER	0B	/P/		CHAR*10	

```
      SUBROUTINE DELSYS
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS USED TO DELETE AN ALTERNATIVE SYSTEM        *
C*   FROM AN EXISTING SET OF ALTERNATIVES.                       *
C*                                                               *
C*                                                               *
C*   CALLED BY:  RDSYSL                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  IFIND,NNODES,NSYS                             *
C*                                                               *
C*          MODIFIED:  ANSWER,I,ISTR,J,L,NSYS,TSYS               *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 SYSLBL,TSYS(80),ANSWER,ISTR
      WRITE(*,'(5(/),1X)')
      PRINT*,'CURRENT SYSTEMS...'
      DO 10 I=1,NSYS
      TSYS(I)=SYSLBL(I)
      PRINT*,SYSLBL(I)
 10   CONTINUE
      PRINT*,' '
      PRINT*,'ENTER THE SYSTEM TO BE DELETED.'
      PRINT*,'    ?'
      READ(*,'(A10)')ANSWER
      ANSWER=ANSWER//'          '
      ISTR=' '
      DO 20 I=1,NSYS
      IF(TSYS(I).EQ.ANSWER)THEN
      ISTR=ANSWER
      L=I
      ENDIF
 20   CONTINUE
      IF(ISTR.EQ.' ')THEN
      PRINT*,'SYSTEM ',ANSWER,' NOT FOUND'
      ELSE
      DO 30 I=L,79
 30   CALL SYSSET(I,TSYS(I+1))
      CALL PRSTOT
      DO 40 I=1,NNODES
      DO 40 J=L,79
      CALL USET(IFIND,J,URAY(IFIND,J+1))
      CALL NEXT
 40   CONTINUE
      NSYS=NSYS-1
      ENDIF
```

```
      SUBROUTINE DELSYS

   58                 RETURN
   59                 END

--VARIABLE MAP--(LO*A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ANSWER     121B  /SSASUSE/ SAV          CHAR*10
  I          123B  /SSASUSE/ SAV          INTEGER
  IFIND      126B  /SSASUSE/ SAV          INTEGER
  ISTR       122B  /SSASUSE/ SAV          CHAR*10
  J          125B  /SSASUSE/ SAV          INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  L          124B  /SSASUSE/ SAV          INTEGER
  NDEEP        1B  /C/                    INTEGER
  NNODES      0B   /C/                    INTEGER
  NSYS        0B   /SYS1/                 INTEGER
  TSYS        1B   /SSASUSE/ SAV          CHAR*10        80

--PROCEDURES--(LO*A)
 -NAME------TYPE--------ARGS------CLASS-----

  NEXT                    0      SUBROUTINE
  PRETOT                  0      SUBROUTINE
  SYSLBL     CHAR*10      1      FUNCTION

 -NAME------TYPE--------ARGS------CLASS-----

  SYSSET                  2      SUBROUTINE
  URAY       REAL         2      FUNCTION
  USET                    3      SUBROUTINE

--STATEMENT LABELS--(LO*A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM      32
     20  INACTIVE   DO-TERM      44
     30  INACTIVE   DO-TERM      49
     40  INACTIVE   DO-TERM      55

--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  DELSYS      5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              326B =  214
  CM LABELLED COMMON LENGTH        132B =   90
  CM STORAGE USED                 60600B = 24960
  COMPILE TIME                   0.097 SECONDS
```

```
      SUBROUTINE GRAFIX
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE DRAWS A WIRING DIAGRAM OF THE CURRENT          *
C*   PARENT NODE AND ITS SPAN OF CHILDREN.                       *
C*                                                               *
C*                                                               *
C*   CALLED BY:  CHECK                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  CH,CONTRL,I,II,ICOUNT,INDEX,INNRN,        *
C*                     J,K,L,LEVEL,LUL,NCHILD,OBJCTV,TINNRN,     *
C*                     TLUL,TOP,TSTORE                           *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLUL5,INNRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
      COMMON/CHILD/CONTRL
      SAVE
      CHARACTER CH,OBJCTV(16)*10,F1*17,F2*29,F3*31,F4*20,L*42,TOP(16)*10
      INTEGER CONTRL,TSTORE(20,3),TLUL,TINNRN(20)
      DATA F1,F2/'(1X,1H0,4A10,1H0)','(1X,T21,1H0,T30,1H0,4A10,1H0)'/
      DATA L/'[illegible]'/
      DATA F3/'(1X,T21,1H[illegible],4A10,1H0)'/
      DATA F4/'(1X,T21,1H0,T30,A42)'/
      DO 5 J=1,3
      DO 5 I=1,20
5     TSTORE(I,J)=LEVEL(I,J)
      TLUL=LUL
      NCHILD=CONTRL
      ICOUNT=0
1     WRITE(*,'(5(/),1X)')
      ICOUNT=ICOUNT+1
      IF(ICOUNT.EQ.1)THEN
      CALL OBJECT(LEVEL(LUL,1),OBJCTV)
      DO 6 I=1,16
6     TOP(I)=OBJCTV(I)
      DO 8 I=1,20
8     TINNRN(I)=INNRN(I)
      ELSE
      DO 7 I=1,16
7     OBJCTV(I)=TOP(I)
      ENDIF
      WRITE(*,'(80(1X,I3))')(TINNRN(I),I=1,LUL)
      PRINT*,L
      II=1
      DO 10 I=1,4
      WRITE(*,FMT=F1)(OBJCTV(J),J=II,II+3)
      II=II+4
```

```
      SUBROUTINE GRAFIX

   58          10     CONTINUE
   59                 PRINT*,L
   60                 IF(CONTRL.GE.3)THEN
   61                 INDEX=3
   62                 ELSE
   63                 INDEX=CONTRL
   64                 ENDIF
   65                 DO 30 K=1,INDEX
   66                 CALL NEXT
   67                 CALL OBJECT(LEVEL(LVL,1),OBJCTV)
   68                 WRITE(*,'(1X,T21,1H0,T30,20(I3,1X))')(INNRN(I),I=1,LVL)
   69                 WRITE(*,FMT=F4)L
   70                 II=1
   71                 DO 20 I=1,4
   72                 IF(I.NE.3)THEN
   73                 WRITE(*,FMT=F2)(OBJCTV(J),J=II,II+3)
   74                 ELSE
   75                 WRITE(*,FMT=F3)(OBJCTV(J),J=II,II+3)
   76                 ENDIF
   77                 II=II+4
   78          20     CONTINUE
   79                 WRITE(*,FMT=F4)L
   80          30     CONTINUE
   81                 CONTRL=CONTRL-3
   82                 IF(CONTRL.GT.0)THEN
   83                 PRINT*,'(PRESS ANY LETTER TO SEE REMAINING SUBOBJECTIVES)'
   84                 PRINT*,'      ?'
   85                 READ(*,'(A1)')CH
   86                 GO TO 1
   87                 ENDIF
   88                 DO 40 J=1,3
   89                 DO 40 I=1,10
   90          40     LEVEL(I,J)=TSTORE(I,J)
   91                 DO 50 I=1,20
   92          50     INNRN(I)=TINNRN(I)
   93                 LVL=TLVL
   94                 CONTRL=NCHILD
   95                 RETURN
   96                 END
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
CH	1B	/S$ASVSE/	SAV	CHAR*1	
CONTRL	0B	/CHILD/		INTEGER	
F1	22B	/S$ASVSE/	SAV	CHAR*17	
F2	24B	/S$ASVSE/	SAV	CHAR*29	
F3	27B	/S$ASVSE/	SAV	CHAR*31	
F4	33B	/S$ASVSE/	SAV	CHAR*29	
I	204B	/S$ASVSE/	SAV	INTEGER	
ICOUNT	206B	/S$ASVSE/	SAV	INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
II	207B	/S$ASVSE/	SAV	INTEGER	
INDEX	210B	/S$ASVSE/	SAV	INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
J	203B	/S$ASVSE/	SAV	INTEGER	
K	211B	/S$ASVSE/	SAV	INTEGER	
L	75B	/S$ASVSE/	SAV	CHAR*48	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NCHILD	205B	/S$ASVSE/	SAV	INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
OBJCTV	2B	/S$ASVSE/	SAV	CHAR*10	16
TINNRN	157B	/S$ASVSE/	SAV	INTEGER	20
TLVL	156B	/S$ASVSE/	SAV	INTEGER	
TOP	42B	/S$ASVSE/	SAV	CHAR*10	16
TSTORE	62B	/S$ASVSE/	SAV	INTEGER	30

```
        SUBROUTINE GRAFIX

--PROCEDURES--(LO=A)
 -NAME------TYPE---------ARGS------CLASS-----

  NEXT                        0       SUBROUTINE
  OBJECT                      2       SUBROUTINE

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      1     45B                          40
      5  INACTIVE    DO-TERM             36
      6  INACTIVE    DO-TERM             45
      7  INACTIVE    DO-TERM             50

 -LABEL-ADDRESS-----PROPERTIES----DEF

      8  INACTIVE    DO-TERM             47
     10  INACTIVE    DO-TERM             58
     20  INACTIVE    DO-TERM             78

 -LABEL-ADDRESS-----PROPERTIES----DEF

     30  INACTIVE    DO-TERM             80
     40  INACTIVE    DO-TERM             90
     50  INACTIVE    DO-TERM             92

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  GRAFIX      5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              5552 =  365
  CM LABELLED COMMON LENGTH        3400 =  224
  CM STORAGE USED                 606000 =  24960
  COMPILE TIME                   0.204 SECONDS
```

```
      SUBROUTINE INITT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE INITIALIZES THE CONSTRUCTION OF A NEW          *
C*   DATA STRUCTURE.                                             *
C*                                                               *
C*                                                               *
C*   CALLED BY:  TAPER                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*           USED:  (NONE)                                       *
C*                                                               *
C*           MODIFIED:  (NONE)                                   *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      CALL RDSYSL
      CALL RDTTL
      CALL MASTER
      CALL SPAN
      CALL TSAVE
      CALL TLOAD
      RETURN
      END
```

--PROCEDURES--(LO-A)

NAME	TYPE	ARGS	CLASS
MASTER		0	SUBROUTINE
RDSYSL		0	SUBROUTINE
RDTTL		0	SUBROUTINE
SPAN		0	SUBROUTINE
TLOAD		0	SUBROUTINE
TSAVE		0	SUBROUTINE

--ENTRY POINTS--(LO-A)

NAME	ADDRESS	ARGS
INITT	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	33D = 27
OF STORAGE USED	606008 = 24969
COMPILE TIME	0.022 SECONDS

```
      SUBROUTINE MASTER
C
C
C*****************************************************************
C                                                                 *
C                                                                 *
C   THIS ROUTINE SETS UP A DUMMY ROOT NODE FROM                   *
C   WHICH THE TREE IS TO BE BUILT.                                *
C                                                                 *
C                                                                 *
C   CALLED BY:  INITT                                             *
C                                                                 *
C                                                                 *
C   VARIABLES                                                     *
C           USED:  (NONE)                                         *
C                                                                 *
C           MODIFIED:  NNODES                                     *
C                                                                 *
C                                                                 *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      SAVE
      CALL LSET(1,'MASTER')
      CALL ISET(1,1,0)
      CALL ISET(1,3,-1)
      CALL ISET(1,4,0)
      NNODES=1
      RETURN
      END
```

SUBROUTINE MASTER

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE--STATISTICS--
                                                            PROGRAM-UNIT LENGTH          54B =  44
 NDEEP       1B  /C/                          INTEGER       CM LABELLED COMMON LENGTH     3B =  3
 NNODES      0B  /C/                          INTEGER       CM STORAGE USED          606000B = 24960
                                                            COMPILE TIME             0.026 SECONDS

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 ISET                     3      SUBROUTINE
 LSET                     2      SUBROUTINE

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 MASTER      5C     0
```

```
      SUBROUTINE MODIFY
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ALLOWS THE INTRODUCTION OF A NEW NODE TO       *
C*   THE EXISTING TREE STRUCTURE, OR TO CHANGE THE OBJECTIVE     *
C*   ASSOCIATED WITH AN EXISTING NODE.                           *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DUMMY,CHECK                                     *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ICHECK,NDIFF                                  *
C*                                                               *
C*          MODIFIED:  ANSWER,CONTRL,I,IFADD,IFIND,INNRN,        *
C*                     IQUIT,ISAVD,J,JSTART,LEVEL,LVL,NLVLS,     *
C*                     NNODES,TINNRN,TLVL,TNLVLS,TSTORE          *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/C/NNODES,NDEEP
      COMMON/CHILD/CONTRL
      COMMON/CHCK/ICHECK
      SAVE
      CHARACTER*10 ANSWER(16)
      REAL TSTORE(20,3)
      INTEGER TLVL,TINNRN(20),TNLVLS,CONTRL
      IQUIT=0
      TNLVLS=NLVLS
      TLVL=LVL
      DO 6 I=1,20
      DO 5 J=1,3
5     TSTORE(I,J)=LEVEL(I,J)
6     TINNRN(I)=INNRN(I)
      PRINT*,' '
      PRINT*,'YOU ARE NOW MODIFYING THE TREE.'
1     CALL NODIN
      IF(NLVLS.LT.1)THEN
      IQUIT=1
      ELSE
      ISAVD=0
      PRINT*,'ENTER YOUR SUBOBJECTIVE'
      PRINT*,'?'
      READ(*,'(8A10)')(ANSWER(I),I=1,8)
      IF((ANSWER(1).EQ.'DONE').OR.(ANSWER(1).EQ.' '))THEN
      IQUIT=1
      ELSE
      PRINT*,'PLEASE CONTINUE'
      PRINT*,'?'
      READ(*,'(8A10)')(ANSWER(I),I=9,16)
      IF(NDIFF.EQ.0)THEN
```

```
      SUBROUTINE MODIFY

 58                 CALL OSET(IFIND,ANSWER)
 59                 ELSE
 60                 IF(ICHECK.EQ.1)CONTRL=CONTRL+1
 61                 JSTART=NLVLS-NDIFF+1
 62                 DO 10 J=JSTART,NLVLS
 63                 NNODES=NNODES+1
 64                 CALL ISET(IFIND,IFADD,NNODES)
 65                 CALL OSET(NNODES,' ')
 66                 CALL ISET(NNODES,1,INNRN(J))
 67                 CALL ISET(NNODES,2,-1)
 68                 CALL ISET(NNODES,3,-1)
 69                 CALL ISET(NNODES,4,IFIND)
 70                 CALL ASET(NNODES,1,0.0)
 71                 CALL ASET(NNODES,2,0.0)
 72                 IFIND=NNODES
 73                 IFADD=2
 74           10    CONTINUE
 75                 CALL OSET(NNODES,ANSWER)
 76                 ENDIF
 77                 ENDIF
 78                 ENDIF
 79                 IF(IQUIT.NE.1)GO TO 1
 80                 NLVLS=TNLVLS
 81                 LVL=TLVL
 82                 DO 21 I=1,20
 83                 DO 20 J=1,3
 84           20    LEVEL(I,J)=TSTORE(I,J)
 85           21    INNRN(I)=TINNRN(I)
 86                 RETURN
 87                 END
```

```
                    SUBROUTINE MODIFY

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
      1     47B                     43
      5  INACTIVE   DO-TERM         39
      6  INACTIVE   DO-TERM         40
 -LABEL-ADDRESS-----PROPERTIES----DEF
     10  INACTIVE   DO-TERM         74
     20  INACTIVE   DO-TERM         84
     21  INACTIVE   DO-TERM         85

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  MODIFY      53      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              411B =  265
  CM LABELLED COMMON LENGTH        301B =  193
  CM STORAGE USED                 60600B =  24960
  COMPILE TIME                   0.144 SECONDS
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ANSWER	1B	/SAVSE/	SAV	CHAR*10	16
CONTRL	0B	/CHILD/		INTEGER	
I	144B	/SAVSE/	SAV	INTEGER	
ICHECK	0B	/CHCK/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20
IQUIT	143B	/SAVSE/	SAV	INTEGER	
ISAVD	146B	/SAVSE/	SAV	INTEGER	
J	145B	/SAVSE/	SAV	INTEGER	
JSTART	147B	/SAVSE/	SAV	INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NDEEP	1B	/C/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
NNODES	0B	/C/		INTEGER	
TINNRN	116B	/SAVSE/	SAV	INTEGER	20
TLVL	115B	/SAVSE/	SAV	INTEGER	
TNLVLS	142B	/SAVSE/	SAV	INTEGER	
TSTORE	21B	/SAVSE/	SAV	REAL	60

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ASET		3	SUBROUTINE
ISET		3	SUBROUTINE
MODIF		0	SUBROUTINE
OSET		2	SUBROUTINE

```
 1                SUBROUTINE NEW
 2          C
 3          C
 4          C*****************************************************
 5          C*                                                   *
 6          C*                                                   *
 7          C*   THIS ROUTINE ALLOWS THE GENERATION OF AN INITIAL *
 8          C*   SET OF ALTERNATIVE SYSTEMS.                      *
 9          C*                                                   *
10          C*                                                   *
11          C*   CALLED BY:  RDSYSL                              *
12          C*                                                   *
13          C*                                                   *
14          C*   VARIABLES                                       *
15          C*          USED:  (NONE)                            *
16          C*                                                   *
17          C*          MODIFIED:  ANSWER,ISAVD,NSYS             *
18          C*                                                   *
19          C*                                                   *
20          C*****************************************************
21          C
22          C
23                COMMON/SYS1/NSYS
24                SAVE
25                CHARACTER*10 SYSLBL,ANSWER
26                WRITE(*,'(5(/),1X)')
27          11    PRINT*,' '
28                NSYS=1
29          1     PRINT*,' '
30                PRINT*,'ENTER...SYSTEM ',NSYS,' LABEL'
31                PRINT*,'(10 LETTERS OR LESS)'
32                PRINT*,'    ?'
33                READ(*,'(A10)')ANSWER
34                CALL SYSSET(NSYS,ANSWER,)
35                IF((SYSLBL(NSYS).NE.'DONE').AND.(SYSLBL(NSYS).NE.' '))THEN
36                NSYS=NSYS+1
37                GO TO 1
38                ENDIF
39                NSYS=NSYS-1
40                IF(NSYS.LT.1)THEN
41                PRINT*,'YOU MUST ENTER AT LEAST ONE SYSTEM'
42                GO TO 11
43                ENDIF
44                ISAVD=0
45                RETURN
46                END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ANSWER      1B  /S$ASUSE/ SAV          CHAR*10
  ISAVD       2B  /S$ACUSE/ SAV          INTEGER
  NSYS        0B  /SYS1/                 INTEGER
```

```
                    SUBROUTINE NEW

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  SYSLBL     CHAR*10        1       FUNCTION
  SYSSET                    B       SUBROUTINE

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1      15B                     29
    11      11B                     27

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  NEW        5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              163B =  115
  CM LABELLED COMMON LENGTH          4B =    4
  CM STORAGE USED                 60600B =  24960
  COMPILE TIME                    0.056 SECONDS
```

```
      SUBROUTINE PICTUR(INDEX)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES A PLOT OF THE INDIVIDUAL VALUE       *
C*   FUNCTION OF A PARTICULAR ATTRIBUTE (INDEX).                 *
C*                                                               *
C*                                                               *
C*   CALLED BY:  VALUE                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  INDEX                                         *
C*                                                               *
C*          MODIFIED:  FACLVL,I,ISUB,J,L,L2,LINE,TLVL,XLVL,      *
C*                     XVAL                                      *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/MATT
      SAVE
      CHARACTER LINE(0:40),F1*22,F2*14,L*41,L2*41,ATT1*10
      DATA F1/'(1X,F7.2,5X,2H-+,41A1)'/
      DATA F2/'(14X,1H+,41A1)'/
      DATA L/'+---------+---------+---------+---------+'/
      DATA L2/'0.0       .25        .5        .75       1.0'/
      FACLVL=1.0
      DO 10 I=1,20
      DO 20 J=0,40
20    LINE(J)=' '
      XLVL=(ATT(INDEX,2)-ATT(INDEX,1))*FACLVL+ATT(INDEX,1)
      IF(PARAM(INDEX,4).EQ.0.0)TLVL=XLVL
      IF(PARAM(INDEX,4).EQ.1.0)TLVL=XLVL**2
      IF(PARAM(INDEX,4).EQ.-1.0)TLVL=XLVL**.5
      IF(PARAM(INDEX,4).EQ.-2.0)TLVL=ALOG(XLVL)
      IF(PARAM(INDEX,4).EQ.2.0)TLVL=EXP(XLVL)
      XVAL=PARAM(INDEX,1)+PARAM(INDEX,2)*TLVL
      IF(XVAL.GT.1.0)XVAL=1.0
      IF(XVAL.LT.0.0)XVAL=0.0
      ISUB=INT(XVAL*40.0)
      LINE(ISUB)='*'
      IF((I.EQ.1).OR.(I.EQ.5).OR.(I.EQ.10).OR.(I.EQ.15).OR.
     *(I.EQ.20))THEN
      WRITE(*,FMT=F1)XLVL,(LINE(JJ),JJ=0,40)
      ELSE
      WRITE(*,FMT=F2)(LINE(JJ),JJ=0,40)
      ENDIF
      FACLVL=FACLVL-.05
10    CONTINUE
      WRITE(*,'(15X,A41)')L
      WRITE(*,'(15X,A41)')L2
      PRINT*,'                    VALUE'
      PRINT*,' '
```

```
      SUBROUTINE PICTUR

   58                 PRINT*,' '
   59                 PRINT*,'VALUE FUNCTION FOR ',ATT1(INDEX)
   60                 RETURN
   61                 END
```

```
--VARIABLE MAP--(LO*A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  FACLUL      25B  /S$A$USE/ SAV          REAL
  F1           6B  /S$A$USE/ SAV          CHAR*22
  F2          11B  /S$A$USE/ SAV          CHAR*14
  I           26B  /S$A$USE/ SAV          INTEGER
  INDEX        1   DUMMY-ARG              INTEGER
  ISUB        33B  /S$A$USE/ SAV          INTEGER
  J           27B  /S$A$USE/ SAV          INTEGER
  JJ          34B  /S$A$USE/ SAV          INTEGER
```

```
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  L           13B  /S$A$USE/ SAV          CHAR*41
  LINE         1B  /S$A$USE/ SAV          CHAR*1          41
  L2          20B  /S$A$USE/ SAV          CHAR*41
  NATT         5B  /ATTR/                 INTEGER
  TLUL        31B  /S$A$USE/ SAV          REAL
  XLUL        30B  /S$A$USE/ SAV          REAL
  XUAL        32B  /S$A$USE/ SAV          REAL
```

```
--PROCEDURES--(LO*A)
 -NAME------TYPE--------ARGS------CLASS-----

  ALOG       REAL          1      INTRINSIC
  ATT        REAL          2      FUNCTION
  ATT1       CHAR*10       1      FUNCTION
```

```
 -NAME------TYPE--------ARGS------CLASS-----

  EXP        GENERIC       1      INTRINSIC
  INT        GENERIC       1      INTRINSIC
  PARAM      REAL          2      FUNCTION
```

```
--STATEMENT LABELS--(LO*A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        53
    20  INACTIVE   DO-TERM        34
```

```
--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  PICTUR      5B      1
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH              365B =  245
  CM LABELLED COMMON LENGTH         36B =   30
  CM STORAGE USED                60600B = 24960
  COMPILE TIME                   0.136 SECONDS
```

```
      SUBROUTINE PPI
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE CONDUCTS A TEST OF PAIRWISE PREFERENTIAL       *
C*   INDEPENDENCE BETWEEN THE ATTRIBUTES.                        *
C*                                                               *
C*                                                               *
C*   CALLED BY:  RDATT                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  NATT                                          *
C*                                                               *
C*          MODIFIED:  ATTR1,ATTR2,BAND,CH,D1,DEL1,DEL2,I,       *
C*                     INDERR,ITOL,J,K,MAX1,MAX2,MIN1,MIN2,      *
C*                     TEMP,TEMP1,TEMP2,TOLER,TOTHER             *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/P/USER
      SAVE
      CHARACTER*10 ATTR1,ATTR2,TOTHER,ATT1,USER,CH*1
      REAL BAND(2,2),MAX1,MIN1,MAX2,MIN2
      INDERR=0
      PRINT*,' '
      PRINT*,'AT WHAT TOLERANCE DO YOU WANT TO CHECK YOUR'
      PRINT*,'RESPONSES, ',USER,' (PLUS OR MINUS X PERCENT)?'
      PRINT*,'      X=?'
      READ(*,'(I2)')ITOL
      TOLER=FLOAT(ITOL)/100.0
      PRINT*,'WE ARE WORKING AT PLUS OR MINUS ',ITOL,' PERCENT.'
      DO 10 I=1,NATT-1
      ATTR1=ATT1(I)
      MAX1=AMAX1(ATT(I,1),ATT(I,2))
      MIN1=AMIN1(ATT(I,1),ATT(I,2))
      DEL1=(MAX1-MIN1)/10.0
      DO 20 J=I+1,NATT
      ATTR2=ATT1(J)
      MAX2=AMAX1(ATT(J,1),ATT(J,2))
      MIN2=AMIN1(ATT(J,1),ATT(J,2))
      DEL2=(MAX2-MIN2)/10.0
      PRINT*,' '
      PRINT*,'SUPPOSE THAT THE FOLLOWING'
      PRINT*,'ATTRIBUTES ARE AT THESE LEVELS:'
      DO 30 K=1,NATT
      IF((K.EQ.I).OR.(K.EQ.J))GO TO 30
      TOTHER=ATT1(K)
      TEMP=(ATT(K,2)-ATT(K,1))*.25+ATT(K,1)
      PRINT*,TOTHER,'=',TEMP
   30 CONTINUE
      PRINT*,'THAT IS AT THE 25 PERCENT LEVEL'
```

SUBROUTINE PPI

```
1       PRINT*,' '
        TEMP1=MAX1-5.0*DEL1
        D1=TEMP1*TOLER
        TEMP2=MAX2-5.0*DEL2
        PRINT*,'NOW SUPPOSE THAT YOU HAVE THE INITIAL CONDITIONS:'
        PRINT*,ATTR1,'=',TEMP1,' AND ',ATTR2,'=',TEMP2
        PRINT*,' '
        TEMP2=TEMP2-DEL2*3.0
        PRINT*,'IMAGINE THAT ',ATTR2,' IS CHANGED TO ',TEMP2
        PRINT*,'WHAT LEVEL OF ',ATTR1,' WOULD KEEP YOU AS SATISFIED'
        PRINT*,'AS YOU WERE UNDER THE INITIAL CONDITIONS?'
        PRINT*,'(REMEMBER THAT ALL OTHER ATTRIBUTES ARE AT'
        PRINT*,' THE 25 PERCENT LEVEL)'
        PRINT*,'  ?'
        READ(*,'(F10.0)')TEMP1
        IF((TEMP1.LT.MIN1).OR.(TEMP1.GT.MAX1))THEN
        PRINT*,'THE INPUT LEVEL OF ',TEMP1
        PRINT*,'IS OUTSIDE THE GIVEN RANGE OF ',ATT(I,1),' TO ',ATT(I,2)
        GO TO 1
        ENDIF
        BAND(1,1)=TEMP1+D1
        BAND(1,2)=TEMP1-D1
        TEMP1=MAX1-5.0*DEL1
        TEMP2=MAX2-5.0*DEL2
2       PRINT*,' '
        PRINT*,'SUPPOSE THAT YOU ARE STARTING AT'
        PRINT*,ATTR1,'=',TEMP1,' AND ',ATTR2,'=',TEMP2
        PRINT*,' '
        TEMP2=TEMP2+3.0*DEL2
        PRINT*,'IMAGINE THAT ',TEMP2,' IN ',ATTR2,' IS ACHIEVED.'
        PRINT*,'TO WHAT LEVEL WOULD YOU CHANGE ',ATTR1,', IN ORDER TO'
        PRINT*,'REMAIN AS SATISFIED AS YOU WERE INITIALLY?'
        PRINT*,'(REMEMBER THAT ALL OTHER ATTRIBUTES ARE AT THE '
        PRINT*,' 25 PERCENT LEVEL)'
        PRINT*,'  ?'
        READ(*,'(F10.0)')TEMP1
        IF((TEMP1.LT.MIN1).OR.(TEMP1.GT.MAX1))THEN
        PRINT*,'THE INPUT LEVEL OF ',TEMP1
        PRINT*,'IS OUTSIDE THE GIVEN RANGE OF ',ATT(I,1),' TO ',ATT(I,2)
        GO TO 2
        ENDIF
        BAND(2,1)=TEMP1+D1
        BAND(2,2)=TEMP1-D1
        PRINT*,' '
        PRINT*,USER,', SUPPOSE NOW THAT THE FOLLOWING ATTRIBUTES'
        PRINT*,'ARE SHIFTED TO THESE LEVELS:'
        DO 40,K=1,NATT
        IF((K.EQ.I).OR.(K.EQ.J))GO TO 40
        TOTHER=ATT1(K)
        TEMP=(ATT(K,2)-ATT(K,1))*.75+ATT(K,1)
        PRINT*,TOTHER,'=',TEMP
40      CONTINUE
        PRINT*,'THAT IS AT THE 75 PERCENT LEVEL'
3       PRINT*,' '
        TEMP1=MAX1-5.0*DEL1
        TEMP2=MAX2-5.0*DEL2
        PRINT*,'SUPPOSE THAT YOU HAVE'
```

SUBROUTINE PPI

```
      PRINT*,ATTR1,'=',TEMP1,' AND ',ATTR2,'=',TEMP2
      PRINT*,' '
      TEMP2=TEMP2-3.0*DEL2
      PRINT*,'IMAGINE THAT THE LEVEL OF ',ATTR2
      PRINT*,'IS CHANGED TO ',TEMP2,'.'
      PRINT*,'WOULD THE LEVEL OF ',ATTR1,' NEEDED TO REMAIN'
      PRINT*,'AS SATISFIED AS AT THE INITIAL CONDITIONS'
      PRINT*,'LIE BETWEEN ',BAND(1,1),' AND ',BAND(1,2)
      PRINT*,' (Y/N)  ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'ONLY "Y" OR "N" IS ALLOWED, ',USER
      GO TO 3
      ENDIF
      IF(CH.EQ.'N')INDERR=1
4     PRINT*,' '
      TEMP1=MAX1-5.0*DEL1
      TEMP2=MAX2-5.0*DEL2
      PRINT*,'SUPPOSE THAT YOU HAVE THE INITIAL CONDITIONS'
      PRINT*,ATTR1,'=',TEMP1,' AND ',ATTR2,'=',TEMP2
      PRINT*,' '
      TEMP2=TEMP2+3.0*DEL2
      PRINT*,'IMAGINE THAT YOU MUST ACCEPT'
      PRINT*,'A LEVEL OF ',TEMP2,' IN ',ATTR2,'.'
      PRINT*,'WOULD THE LEVEL OF ',ATTR1
      PRINT*,'THAT YOU WOULD HAVE TO MOVE TO (IN ORDER TO BE AS'
      PRINT*,'SATISFIED AS UNDER THE INITIAL CONDITIONS) LIE'
      PRINT*,'BETWEEN ',BAND(2,1),' AND ',BAND(2,2)
      PRINT*,' (Y/N)  ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,USER,', YOU MUST ENTER "Y" OR "N"'
      GO TO 4
      ENDIF
      IF(CH.EQ.'N')INDERR=1
      IF(INDERR.EQ.0)THEN
      PRINT*,'THERE ARE NO INDEPENDENCE PROBLEMS'
      PRINT*,'WITH THE ATTRIBUTES TESTED SO FAR.'
5     PRINT*,'DO YOU WISH TO ASSUME PPI FOR THE'
      PRINT*,'REMAINING ATTRIBUTES? (Y/N)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
      PRINT*,'PLEASE ENTER "Y" OR "N" '
      GO TO 5
      ENDIF
      IF(CH.EQ.'N')THEN
6     PRINT*,'EVEN IF YOU DO NOT WISH TO ASSUME'
      PRINT*,'PPI AMONG THE REMAING ATTRIBUTES,'
      PRINT*,'DO YOU WANT TO STOP PPI TESTING? (Y/N)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
      PRINT*,'PLEASE ENTER "Y" OR "N" '
      GO TO 6
      ENDIF
      ENDIF
```

```
      SUBROUTINE PPI

                IF(CH.EQ.'Y')GO TO 7
                ELSE
                PRINT*,'THERE ARE INDEPENDENCE PROBLEMS'
                PRINT*,'AMONG THE ATTRIBUTES ALREADY TESTED.'
                PRINT*,'DO YOU WANT TO STOP PPI TESTING? (Y/N)'
        5       PRINT*,'      ?'
                READ(*,'(A1)')CH
                IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
                PRINT*,'PLEASE ENTER 'Y' OR 'N' '
                GO TO 8
                ENDIF
                IF(CH.EQ.'Y')GO TO 7
                ENDIF
        20      CONTINUE
        10      CONTINUE
        7       IF(INDERR.EQ.0)THEN
                CALL VALUE
                ELSE
                PRINT*,'THERE ARE INDEPENDENCE PROBLEMS'
                PRINT*,'AMONG THE ATTRIBUTES (PPI DOES'
                PRINT*,'NOT HOLD).  DO YOU WISH TO '
                PRINT*,'CONTINUE THE ANALYSIS WITH AN'
                PRINT*,'ADDITIVE VALUE FUNCTION? (Y/N)'
                PRINT*,'      ?'
                READ(*,'(A1)')CH
                IF(CH.EQ.'Y')CALL VALUE
                ENDIF
                RETURN
                END
```

--VARIABLE MAP--(LO*A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ATTR1	1B	/S$ASUSE/	SAV	CHAR*10	
ATTR2	2B	/S$ASUSE/	SAV	CHAR*10	
BAND	5B	/S$ASUSE/	SAV	REAL	4
CH	4B	/S$ASUSE/	SAV	CHAR*1	
DEL1	21B	/S$ASUSE/	SAV	REAL	
DEL2	23B	/S$ASUSE/	SAV	REAL	
D1	27B	/S$ASUSE/	SAV	REAL	
I	20B	/S$ASUSE/	SAV	INTEGER	
INDERR	15B	/S$ASUSE/	SAV	INTEGER	
ITOL	16B	/S$ASUSE/	SAV	INTEGER	
J	22B	/S$ASUSE/	SAV	INTEGER	
K	24B	/S$ASUSE/	SAV	INTEGER	
MAX1	11B	/S$ASUSE/	SAV	REAL	
MAX2	13B	/S$ASUSE/	SAV	REAL	
MIN1	12B	/S$ASUSE/	SAV	REAL	
MIN2	14B	/S$ASUSE/	SAV	REAL	
NATT	0B	/ATTR/		INTEGER	
TERP	25B	/S$ASUSE/	SAV	REAL	
TEMP1	26B	/S$ASUSE/	SAV	REAL	
TEMP2	30B	/S$ASUSE/	SAV	REAL	
TOLER	17B	/S$ASUSE/	SAV	REAL	
TOTHER	3B	/S$ASUSE/	SAV	CHAR*10	
USER	6B	/P/		CHAR*10	

--PROCEDURES--(LO*A)

NAME	TYPE	ARGS	CLASS
AMAX1	REAL	7	INTRINSIC
AMIN1	REAL	7	INTRINSIC
ATT	REAL	2	FUNCTION
ATT1	CHAR*10	1	FUNCTION
FLOAT	REAL	1	INTRINSIC
VALUE		0	SUBROUTINE

```
      SUBROUTINE PPI

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
     1    167B                      58
     2    254B                      82
     3    412B                     111
     4    472B                     130

 -LABEL-ADDRESS-----PROPERTIES----DEF
     5    563B                     153
     6    611B                     162
     7    706B                     187
     8    647B                     176

 -LABEL-ADDRESS-----PROPERTIES----DEF
    10  INACTIVE   DO-TERM         186
    20  INACTIVE   DO-TERM         185
    30    160B     DO-TERM          56
    40    403B     DO-TERM         169

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PPI        5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              2005B =  1029
  CM LABELLED COMMON LENGTH         33B =  27
  CM STORAGE USED                  64600B =  27008
  COMPILE TIME                     0.497 SECONDS
```

```
      SUBROUTINE PRUNE
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE DELETES EXISTING NODES FROM THE CURRENT        *
C*   TREE STRUCTURE.  THIS ROUTINE IS CALLED DIRECTLY THROUGH*
C*   OPTION *** PRU **.  IT IS CALLED AUTOMATICALLY FROM         *
C*   SUBROUTINE CHECK IF IT IS REQUIRED.                         *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DUMMY,CHECK                                     *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ICHECK,NDIFF                                  *
C*                                                               *
C*          MODIFIED:  CH,I,II,IDATA,IFIND,IFROM,INNRN,J,K,      *
C*                     LEVEL,LVL,NLVLS,OBJCTV,TINNRN,TLVL,       *
C*                     TNLVLS,TSTORE                             *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTYPE
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/CHILD/CONTRL
      COMMON/CHCK/ICHECK
      SAVE
      REAL TSTORE(20,3)
      INTEGER TLVL,TINNRN(20),TNLVLS,CONTRL
      CHARACTER CH,OBJCTV(16)*10
1     PRINT*,' '
      TNLVLS=NLVLS
      TLVL=LVL
      DO 12 I=1,20
      DO 11 J=1,3
11    TSTORE(I,J)=LEVEL(I,J)
12    TINNRN(I)=INNRN(I)
      PRINT*,'ENTER NODE TO BE REMOVED.'
      CALL NODIN
      IF(NLVLS.GT.0)THEN
      IF(NDIFF.NE.0)THEN
2     PRINT*,' '
      PRINT*,'NODE NOT FOUND.  DO YOU WISH TO TRY AGAIN?  (Y/N)'
      PRINT*,'    ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'PLEASE ENTER "Y" OR "N" '
      GO TO 2
      ENDIF
      IF(CH.EQ.'Y')THEN
      GO TO 1
      ELSE
```

SUBROUTINE PRUNE

```
      GO TO 999
      ENDIF
      ENDIF
3     PRINT*,' '
      IFROM=2
      PRINT*,'SELECT PRUNING OPTION:'
      PRINT*,'  N(ODE+DOWN      D(OWN ONLY      S(ELECT DOWN      E(XIT'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'N').AND.(CH.NE.'D').AND.(CH.NE.'S').AND.
     * (CH.NE.'E'))THEN
      PRINT*,'YOU MUST SELECT 'N','D','S', OR 'E' '
      GO TO 3
      ENDIF
      IF(CH.EQ.'E')GO TO 999
      IF(CH.EQ.'N')THEN
      IF(ICHECK.EQ.1)CONTRL=CONTRL-1
      LVL=LVL-1
      IFROM=IRAY(IFIND,4)
      IF(IRAY(IFROM,3).NE.IFIND)THEN
      CALL ISET(IFROM,2,IRAY(IFIND,3))
      IF(IRAY(IFROM,2).GT.0)CALL ISET(IRAY(IFROM,2),4,IFROM)
      GO TO 400
      ELSE
      CALL ISET(IFROM,3,IRAY(IFIND,3))
      IF(IRAY(IFROM,3).GT.0)CALL ISET(IRAY(IFROM,3),4,IFROM)
      GO TO 400
      ENDIF
      ELSEIF(CH.EQ.'D')THEN
      CALL ISET(IFIND,2,-1)
      GO TO 400
      ELSEIF(CH.EQ.'S')THEN
4     IFIND=IRAY(IFIND,IFROM)
      IF(IFIND.LE.0)GO TO 402
      CALL OBJECT(IFIND,OBJCTV)
5     II=1
      DO 10 J=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
10    CONTINUE
      PRINT*,' '
      PRINT*,'WHICH IS CURRENT DESCENDANT NUMBER ',IRAY(IFIND,1)
      PRINT*,'DO YOU WHICH TO ELIMINATE THIS DESCENDANT?  (Y/N)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'ONLY 'Y' OR 'N' IS ALLOWED'
      GO TO 5
      ENDIF
      IF(CH.EQ.'N')THEN
      IFROM=3
      GO TO 4
      ELSE
      CALL ISET(IRAY(IFIND,4),IFROM,IRAY(IFIND,3))
      IF(IRAY(IFIND,3).LE.0)GO TO 400
      CALL ISET(IRAY(IFIND,3),4,IRAY(IFIND,4))
      IFROM=3
```

SUBROUTINE PRUNE

```
115                 GO TO 4
116                 ENDIF
117                 ENDIF
118          400    IFIND=LEVEL(LVL,1)
119                 IDATA=0
120                 IF(IRAY(IFIND,2).LE.0)IDATA=1
121                 GO TO 1
122                 ENDIF
123          999    PRINT*, ' '
124                 NLVLS=TNLVLS
125                 LVL=TLVL
126                 DO 21 I=1,20
127                 DO 20 J=1,3
128          20     LEVEL(I,J)=TSTORE(I,J)
129          21     INNRN(I)=TINNRN(I)
130                 RETURN
131                 END
```

SUBROUTINE PRUNE

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
PRUNE	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	734B =	476
CM LABELLED COMMON LENGTH	305B =	197
CM STORAGE USED	60600B =	24960
COMPILE TIME	0.252 SECONDS	

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CH	123B	/SAU$E/	SAV	CHAR*1	
CONTRL	0B	/CHILD/		INTEGER	
I	144B	/SAU$E/	SAV	INTEGER	
ICHECK	0B	/CHCK/		INTEGER	
ICONT	0B	/MCX/		INTEGER	
IDATA	1B	/MEX/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
IFROM	146B	/SAU$E/	SAV	INTEGER	
I-	147B	/SAU$E/	SAV	INTEGER	20
INNRN	1B	/LEVEL/		INTEGER	
ISAVD	0B	/CNTRL/		INTEGER	
ITOTL	2B	/MEX/		INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
J	145B	/SAU$E/	SAV	INTEGER	
K	150B	/SAU$E/	SAV	INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
NTAPE	1B	/CNTRL/		INTEGER	
OBJCTU	124B	/SAU$E/	SAV	CHAR*10	16
TINNRN	76B	/SAU$E/	SAV	INTEGER	20
TLVL	75B	/SAU$E/	SAV	INTEGER	
TNLVLS	122B	/SAU$E/	SAV	INTEGER	
TSTORE	1B	/SAU$E/	SAV	REAL	60

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
IRAY	INTEGER	2	FUNCTION
ISET		3	SUBROUTINE
NODIN		6	SUBROUTINE
OBJECT		2	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
1	7B		26
2	54B		47
3	107B		61
4	234B		95
5	244B		93
10	INACTIVE	DO-TERM	97
11	INACTIVE	DO-TERM	41
12	INACTIVE	DO-TERM	42
20	INACTIVE	DO-TERM	128
21	INACTIVE	DO-TERM	129
400	376B		118
999	404B		123

```
      SUBROUTINE RDATT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ELICITS THE ATTRIBUTE SET WHICH IS             *
C*   ASSOCIATED WITH THE OBJECTIVE HIERARCHY.                    *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DUMTY,SPRM                                      *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ICONT,IDATA,LEVEL,LVL                         *
C*                                                               *
C*          MODIFIED:  ATTRIB,CH,II,.MATT,J,K,NATT,OBJCTV,*
C*                     TATT                                      *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/MEX/ICONT,IDATA,ITOTL
      COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/P/USER
      SAVE
      CHARACTER*10 ATTRIB,OBJCTV(16),ATT1,USER,CH*1,TATT(4)
99    CALL PRETOT
      NATT=0
1     IF(ICONT.NE.0)THEN
      IF(IDATA.EQ.1)THEN
      NATT=NATT+1
2     PRINT*,' '
      PRINT*,USER,', PLEASE INPUT AN ATTRIBUTE FOR
      PRINT*,'THE DATA NODE WITH THE OBJECTIVE:'
      CALL OBJECT(LEVEL(LVL,1),OBJCTV)
      II=1
      DO 20 J=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
20    CONTINUE
      PRINT*,'(10 LETTERS OR LESS)'
      PRINT*,'     ?'
      READ(*,'(A10)')ATTRIB
3     PRINT*,' '
      PRINT*,'IS THE ATTRIBUTE ',ATTRIB
      PRINT*,'SUCH THAT BY KNOWING ITS LEVEL,'
      PRINT*,'THE ATTAINMENT OF THE OBJECTIVE'
      PRINT*,'IS TOTALLY DETERMINED?  (Y/N)'
      PRINT*,'     ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'ONLY "Y" AND "N" ARE PERMITTED, ',USER
      GO TO 3
      ENDIF
```

SUBROUTINE RDATT

```
      IF(CH.EQ.'N')GO TO 2
4     PRINT*,' '
      PRINT*,'COULD THE ATTRIBUTE ',ATTRIB
      PRINT*,'BE CHANGED SO AS TO IMPROVE'
      PRINT*,'COMMUNICATING WHAT IS IMPLIED'
      PRINT*,'IN THE OBJECTIVE?  (Y/N)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,USER,', PLEASE ENTER 'Y' OR 'N'
      GO TO 4
      ENDIF
      IF(CH.EQ.'Y')GO TO 2
      CALL ATTSET(NATT,ATTRIB)
      ENDIF
      CALL NEXT
      GO TO 1
      ENDIF
      INATT=NATT/4
      DO 10 J=1,INATT
      II=1
      DO 11 K=J*4-3,J*4
      TATT(II)=ATTI(K)
11    II=II+1
10    WRITE(*,'(4(1X,A10))')(TATT(K),K=1,4)
      IF(INATT*4.LT.NATT)THEN
      II=1
      DO 12 K=INATT*4+1,NATT
      TATT(II)=ATTI(K)
12    II=II+1
      WRITE(*,'(4(1X,A10))')(TATT(K),K=1,II-1)
      ENDIF
      PRINT*,' '
      PRINT*,'THE ABOVE IS THE CURRENT SET OF ATTRIBUTES, ',USER,'.'
      PRINT*,'IF YOU SEE ANY WHICH ARE REDUNDANT, OR'
      PRINT*,'WHICH HAVE A DIRECT IMPACT ON ONE ANOTHER'
      PRINT*,'(E.G. WEIGHT AND THRUST),'
      PRINT*,'YOU SHOULD REFORM THE ATTRIBUTE SET TO'
      PRINT*,'REMOVE THESE PROBLEMS.'
5     PRINT*,' '
      PRINT*,'DO YOU WISH TO REFORM THE ATTRIBUTE SET, ',USER,'?'
      PRINT*,'(Y/N)   ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,'ONLY 'Y' OR 'N' IS ALLOWED, ',USER
      GO TO 5
      ENDIF
      IF(CH.EQ.'Y')GO TO 99
      PRINT*,' '
      PRINT*,'DO YOU WISH TO BYPASS INDEPENDENCE TESTING?'
      READ(*,'(A1)')CH
      IF(CH.NE.'Y')THEN
      CALL PPI
      ELSE
      CALL VALUE
      ENDIF
      RETURN
```

```
      SUBROUTINE RDATT

  116                 END

--VARIABLE MAP--(LO*A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
 ATTRIB      1B  /S$ABUSE/ SAV          CHAR*10
 CH         22B  /S$ABUSE/ SAV          CHAR*1
 ICONT       0B  /NEX/                  INTEGER
 IDATA       1B  /NEX/                  INTEGER
 IFADD      27B  /LEVEL/                INTEGER
 IFIND      25B  /LEVEL/                INTEGER
 II         27B  /S$ABUSE/ SAV          INTEGER
 INATT      32B  /S$ABUSE/ SAV          INTEGER
 INNRM       1B  /LEVEL/                INTEGER
 ITOTL       2B  /NEX/                  INTEGER          20

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
 J          30B  /S$ABUSE/ SAV          INTEGER
 K          31B  /S$ABUSE/ SAV          INTEGER
 LEVEL      31B  /LEVEL/                INTEGER          60
 LVL        30B  /LEVEL/                INTEGER
 MATT        0B  /ATTR/                 INTEGER
 NDIFF      26B  /LEVEL/                INTEGER
 NLVLS       3B  /LEVEL/                INTEGER
 OBJCTV      2B  /S$ABUSE/ SAV          CHAR*10          16
 TATT       23B  /S$ABUSE/ SAV          CHAR*10           4
 USER        0B  /P/                    CHAR*10

--PROCEDURES--(LO*A)
 -NAME------TYPE---------ARGS------CLASS-----
 ATTSET                   2      SUBROUTINE
 ATTI       CHAR*10       1      FUNCTION
 NEXT                     0      SUBROUTINE
 OBJECT                   2      SUBROUTINE

 -NAME------TYPE---------ARGS------CLASS-----
 PPI                      0      SUBROUTINE
 PRETOT                   0      SUBROUTINE
 VALUE                    0      SUBROUTINE

--STATEMENT LABELS--(LO*A)
 -LABEL-ADDRESS-----PROPERTIES----DEF
     1     13B                     32
     2     23B                     35
     3    106B                     47
     4    140B                     50

 -LABEL-ADDRESS-----PROPERTIES----DEF
     5     245B                    97
    10  INACTIVE   DO-TERM         82
    11  INACTIVE   DO-TERM         81

 -LABEL-ADDRESS-----PROPERTIES----DEF
    12  INACTIVE   DO-TERM         87
    28  INACTIVE   DO-TERM         43
    99      7B                     56

--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---
  RDATT      5B      0

--STATISTICS--
  PROGRAM-UNIT LENGTH              1023B =  531
  CM LABELLED COMMON LENGTH         165B =  117
  CM STORAGE USED                 62660B =  25904
  COMPILE TIME                    8.249 SECONDS
```

```
        SUBROUTINE RDSYSL

--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  CH        228B                                CHAR(1)

--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----

  ADDSYS                    0      SUBROUTINE
  DELSYS                    0      SUBROUTINE
  NEW                       0      SUBROUTINE
  PRETOT                    0      SUBROUTINE

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      1      7B                     24

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  RDSYSL      5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              241D = 161
  CM STORAGE USED                 60688D = 24960
  COMPILE TIME                   0.068 SECONDS
```


PRECEDING PAGE BLANK-

```
      SUBROUTINE RDTTL
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ELICITS AN IDENTIFICATION FOR THE DATA         *
C*   STRUCTURE.  IT MAY BE CALLED DIRECTLY VIA OPTION            *
C*   *** TTL ***.  IT IS AUTOMATICALLY INVOKED WHEN USING        *
C*   OPTION *** NEW ***.                                         *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DUMMY,INITT                                     *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  (NONE)                                    *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      PRINT*,' '
      PRINT*,'ENTER A TITLE FOR THIS DATA STRUCTURE...'
      CALL CSET(1)
      RETURN
      END
```

```
--PROCEDURES--(LO*A)
 -NAME------TYPE--------ARGS------CLASS-----

  CSET                      1       SUBROUTINE

--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  RDTTL       5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              43B =  35
  CM STORAGE USED                00060B =  24D0B
  COMPILE TIME                   0.02B SECONDS
```

```
      SUBROUTINE REGRS(INDEX)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE COMPUTES THE CLOSEST LEAST-SQUARES             *
C*   REGRESSION TO APPROXIMATE THE INDIVIDUAL VALUE FUNCTION.    *
C*   THE PARAMETERS GENERATED ARE STORED BY ROUTINE              *
C*   PSET(I,J,V), AND ACCESSED BY THE FUNCTION PARAM(I,J).       *
C*   SEE PARAM(I,J) FOR A DESCRIPTION OF THE FIRST THREE         *
C*   PARAMETERS.  THE FOURTH PARAMETER (PARAM(I,4)) IS AN        *
C*   INDICATOR OF THE TYPE OF TRANSFORMATION REQUIRED ON THE     *
C*   INDEPENDENT VARIABLE.  THE CODE IS AS FOLLOWS:              *
C*                                                               *
C*          MARKER                    TRANSFORMATION             *
C*          0.0                       X=X (LINEAR)               *
C*         -1.0                       X=X**.5 (SQUARE ROOT)      *
C*          1.0                       X=X**2 (SQUARE)            *
C*         -2.0                       X=ALOG(X) (LOGARITHIC)     *
C*          2.0                       X=EXP(X) (EXPONENTIAL)     *
C*                                                               *
C*   CALLED BY:  VALUE                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  VAL,INDEX                                     *
C*                                                               *
C*          MODIFIED:  B0,B1,B0EXP,B1EXP,B0LN,B1LN,B0ROOT,       *
C*                     B1ROOT,B0SQR,B1SQR,PROD1,PROD2,PROD3,     *
C*                     PROD4,PROD5,RTX,EXPX,LNX,SEXPX,SLNX,      *
C*                     SQRX,SRTX,SRTX2,SSE,SSEEXP,SSELN,SSERT,   *
C*                     SSESQR,SSQRX,SSQRX2,SVLU,SVLU2,SXLVL,     *
C*                     SXLVL2,VLU,VLUMIN,XLVL                    *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/VAL/VAL(5)
      SAVE
      REAL VLU(0:5),XLVL(5),RTX(5),SQRX(5),LNX(5),EXPX(5)
      VLU(0)=-.25
      DO 10 I=1,5
      VLU(I)=VLU(I-1)+.25
      XLVL(I)=VAL(I)
      IF(VAL(I).GT.0.0)THEN
      RTX(I)=VAL(I)**.5
      ELSE
      RTX(I)=0.0
      ENDIF
      SQRX(I)=VAL(I)***2
      IF(VAL(I).GT.0.0)THEN
      LNX(I)=ALOG(VAL(I))
      ELSE
      LNX(I)=0.0
```

SUBROUTINE RCORS

```
                ENDIF
                IF(VAL(I).LT.50.0)THEN
                EXPX(I)=EXP(VAL(I))
                ELSE
                EXPX(I)=1.1
                ENDIF
        10      CONTINUE
                SULU=0.0
                SXLUL=0.0
                SRTX=0.0
                SSQRX=0.0
                SLNX=0.0
                SEXPX=0.0
                PROD1=0.0
                PROD2=0.0
                PROD3=0.0
                PROD4=0.0
                PROD5=0.0
                SULU2=0.0
                SXLUL2=0.0
                SRTX2=0.0
                SSQRX2=0.0
                SLNX2=0.0
                SEXPX2=0.0
                DO 20 I=1,5
                PROD1=ULU(I)*XLUL(I)+PROD1
                PROD2=ULU(I)*RTX(I)+PROD2
                PROD3=ULU(I)*SQRX(I)+PROD3
                PROD4=ULU(I)*LNX(I)+PROD4
                PROD5=ULU(I)*EXPX(I)+PROD5
                SULU=SULU+ULU(I)
                SXLUL=SXLUL+XLUL(I)
                SRTX=SRTX+RTX(I)
                SSQRX=SSQRX+SQRX(I)
                SLNX=SLNX+LNX(I)
                SEXPX=SEXPX+EXPX(I)
                SULU2=SULU2+ULU(I)**2
                SXLUL2=SXLUL2+XLUL(I)**2
                SRTX2=SRTX2+RTX(I)**2
                SSQRX2=SSQRX2+SQRX(I)**2
                SLNX2=SLNX2+LNX(I)**2
                SEXPX2=SEXPX2+EXPX(I)**2
        20      CONTINUE
                B1=(5.0*PROD1-SULU*SXLUL)/(5.0*SXLUL2-SXLUL**2)
                B0=SULU/5.0-B1*(SXLUL/5.0)
                B1ROOT=(5.0*PROD2-SULU*SRTX)/(5.0*SRTX2-SRTX**2)
                B0ROOT=SULU/5.0-B1ROOT*(SRTX/5.0)
                B1SQR=(5.0*PROD3-SULU*SSQRX)/(5.0*SSQRX2-SSQRX**2)
                B0SQR=SULU/5.0-B1SQR*(SSQRX/5.0)
                B1LN=(5.0*PROD4-SULU*SLNX)/(5.0*SLNX2-SLNX**2)
                B0LN=SULU/5.0-B1LN*(SLNX/5.0)
                B1EXP=(5.0*PROD5-SULU*SEXPX)/(5.0*SEXPX2-SEXPX**2)
                B0EXP=SULU/5.0-B1EXP*(SEXPX/5.0)
                SSE=0.0
                SSEROOT=0.0
                SSESQR=0.0
                SSELN=0.0
```

```
      SUBROUTINE REGNS

      SSEEXP=0.0
      DO 30 I=1,6
      SSE=SSE+(VLU(I)-B0-B1*XLU(I))**2
      SSERT=SSERT+(VLU(I)-B0ROOT-B1ROOT*RTX(I))**2
      SSESQR=SSESQR+(VLU(I)-B0SQR-B1SQR*SQRX(I))**2
      SSELN=SSELN+(VLU(I)-B0LN-B1LN*LNX(I))**2
      SSEEXP=SSEEXP+(VLU(I)-B0EXP-B1EXP*EXPX(I))**2
   30 CONTINUE
      VLUMIN=AMIN1(SSE,SSERT,SSESQR,SSELN,SSEEXP)
      IF(VLUMIN.EQ.SSE)THEN
      CALL PSET(INDEX,1,B0)
      CALL PSET(INDEX,2,B1)
      CALL PSET(INDEX,3,SSE)
      CALL PSET(INDEX,4,0.0)
      ELSEIF(VLUMIN.EQ.SSERT)THEN
      CALL PSET(INDEX,1,B0ROOT)
      CALL PSET(INDEX,2,B1ROOT)
      CALL PSET(INDEX,3,SSERT)
      CALL PSET(INDEX,4,-1.0)
      ELSEIF(VLUMIN.EQ.SSESQR)THEN
      CALL PSET(INDEX,1,B0SQR)
      CALL PSET(INDEX,2,B1SQR)
      CALL PSET(INDEX,3,SSESQR)
      CALL PSET(INDEX,4,1.0)
      ELSEIF(VLUMIN.EQ.SSELN)THEN
      CALL PSET(INDEX,1,B0LN)
      CALL PSET(INDEX,2,B1LN)
      CALL PSET(INDEX,3,SSELN)
      CALL PSET(INDEX,4,-2.0)
      ELSEIF(VLUMIN.EQ.SSEEXP)THEN
      CALL PSET(INDEX,1,B0EXP)
      CALL PSET(INDEX,2,B1EXP)
      CALL PSET(INDEX,3,SSEEXP)
      CALL PSET(INDEX,4,2.0)
      ENDIF
      RETURN
      END
```

--VARIABLE MAP--(LO-A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
B0	638	/S8ABUSE/	SAV	REAL	
B0EXP	738	/S8ABUSE/	SAV	REAL	
B0LN	718	/S8ABUSE/	SAV	REAL	
B0ROOT	668	/S8ABUSE/	SAV	REAL	
B0SQR	678	/S8ABUSE/	SAV	REAL	
B1	628	/S8ABUSE/	SAV	REAL	
B1EXP	728	/S8ABUSE/	SAV	REAL	
B1LN	708	/S8ABUSE/	SAV	REAL	
B1ROOT	648	/S8ABUSE/	SAV	REAL	
B1SQR	668	/S8ABUSE/	SAV	REAL	
EXPX	338	/S8ABUSE/	SAV	REAL	6
I	408	/S8ABUSE/	SAV	INTEGER	
INDEX	1	DUMMY-ARG		INTEGER	
LNX	268	/S8ABUSE/	SAV	REAL	5
PROD1	478	/S8ABUSE/	SAV	REAL	
PROD2	508	/S8ABUSE/	SAV	REAL	
PROD3	518	/S8ABUSE/	SAV	REAL	
PROD4	528	/S8ABUSE/	SAV	REAL	
PROD5	538	/S8ABUSE/	SAV	REAL	
RTX	148	/S8ABUSE/	SAV	REAL	5
SEXPX	468	/S8ABUSE/	SAV	REAL	
SEXPX2	618	/S8ABUSE/	SAV	REAL	
SLNX	458	/S8ABUSE/	SAV	REAL	
SLNX2	668	/S8ABUSE/	SAV	REAL	
SQRX	218	/S8ABUSE/	SAV	REAL	5
SRTX	438	/S8ABUSE/	SAV	REAL	
SRTX2	568	/S8ABUSE/	SAV	REAL	
SSE	748	/S8ABUSE/	SAV	REAL	
SSEEXP	1008	/S8ABUSE/	SAV	REAL	
SSELN	778	/S8ABUSE/	SAV	REAL	

```
        SUBROUTINE REGRS       74/74    OPT=0                                    FTN 5.1+523          11/23/81  19.36.01        PAGE      4
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 SSERT       75B  /SSASUSE/ SAV              REAL                   SXLVL       42B  /SSASUSE/ SAV              REAL
 SSESQR      70B  /SSASUSE/ SAV              REAL                   SXLVL2      55B  /SSASUSE/ SAV              REAL
 SSGRX       44B  /SSASUSE/ SAV              REAL                   VAL         0B   /VAL/                     REAL          5
 SSWRXB      57B  /SSASUSE/ SAV              REAL                   VLU         1B   /SSASUSE/ SAV              REAL          6
 SULU        41B  /SSASUSE/ SAV              REAL                   VLUMIN      101B /SSASUSE/ SAV              REAL
 SULU2       54B  /SSASUSE/ SAV              REAL                   XLVL        7B   /SSASUSE/ SAV              REAL          5

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS----

  ALOG      REAL          1      INTRINSIC
  AMIN1     REAL          7      INTRINSIC
  EXP       GENERIC       1      INTRINSIC
  PSET                    3      SUBROUTINE

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM       64
     20  INACTIVE   DO-TERM      100
     30  INACTIVE   DO-TERM      122

 -ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  REGRS       5B     1

 -STATISTICS--

  PROGRAM-UNIT LENGTH              723B =  467
  CM LABELLED COMMON LENGTH        107B =  71
  CM STORAGE USED                 60600B =  24960
  COMPILE TIME                   0.306 SECONDS
```

```
      SUBROUTINE SPAN
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE CONDUCTS A NODE-BY-NODE CONSTRUCTION OF THE    *
C*   OBJECTIVE HIERARCHY.  STARTING AT THE ROOT (OR DESIRED)     *
C*   NODE, ALL CHILDREN OF THE NODE ARE GENERATED.  ONCE A       *
C*   SATISFACTORY SPAN OF CHILDREN IS ESTABLISHED, THE NEXT      *
C*   PARENT NODE IS CHOSEN IN A DEPTH-FIRST ORDER.               *
C*                                                               *
C*   CALLED BY:  DUMMY,INIT                                      *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  USER                                          *
C*                                                               *
C*          MODIFIED:  ANSWER,CH,CONTRL,ICONT,ICOUNT,IFADD,      *
C*                     IFIND,IFINDT,INARN,LEVEL,LVL,NDEEP,       *
C*                     NNODES,NNRN,TENSTR,U                      *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/LEVEL/NLVLS,INARN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,2)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/CHILD/CONTRL
      COMMON/P/USER
      SAVE
      CHARACTER*1* USER,LABEL,ANSWER(16),TENSTR,OBJCTV(16),CH*1,U*1
      INTEGER CONTRL
      CALL PRETOT
      TENSTR='          '
1     WRITE(*,'(5(/),1X)')
      PRINT*,'SPANNING NODES:     "A"=ALL     "S"=SELECT'
      PRINT*,'     ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'S').AND.(CH.NE.'A'))THEN
      PRINT*,'ONLY "A" OR "S" IS ALLOWED ',USER
      GO TO 1
      ENDIF
      IF(CH.EQ.'S')CALL PREMEX
      IFADD=2
      IF(ICONT.NE.1)THEN
2     WRITE(*,'(5(/),1X)')
      PRINT*,'DO YOU WISH TO BUILD A NEW TREE, ',USER,'?     (Y/N)'
      PRINT*,'     ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
      PRINT*,'PLEASE INPUT "Y" OR "N" , ',USER
      GO TO 2
      ENDIF
      PRINT*,' '
```

SUBROUTINE SNWK

```
      PRINT*,'DO YOU WISH TO BY-PASS THE BETWEEN NODE CHECK?'
      READ(*,'(A1)')CH
      IF(CH.EQ.'N')THEN
      ANSWER(1)='EXIT'
      ELSE
      LEVEL(1,1)=1
      LEVEL(1,2)=1
      NWRN(1)=0
      LVL=1
      IFIND=LEVEL(1,1)
      IFADD=3
      ICONT=1
      NLVLS=0
      ENDIF
      ENDIF
      ICOUNT=0
3     IF(ICONT.NE.0)THEN
      NWRN=0
      IFIRST=IFIND
      CONTRL=0
      PRINT*,'ADDING DOWNLINKS TO NODE:'
      WRITE(*,'(20I3,1X)')(NWRN(I),I=1,LVL)
      IF NNODES.GT.1)THEN
      CALL OBJECT(LEVEL(LVL,1),OBJCTV)
      II=1
      DO 20 I=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(J),J=II,II+3)
      II=II+4
20    CONTINUE
      ENDIF
4     PRINT*,' '
      ICOUNT=ICOUNT+1
      IF(ICOUNT.EQ.2)THEN
      ANSWER(1)=TENSTR
      ELSE
      PRINT*,USER,', WHAT IS THE NEXT SUBOBJECTIVE?'
      PRINT*,'(USE NO MORE THAN TWO 20 CHARACTER LINES)'
      PRINT*,'?'
      READ(*,'(8A10)')(ANSWER(I),I=1,8)
      ENDIF
      IF((ANSWER(1).NE.TENSTR).AND.(ANSWER(1).NE.'DONE').AND.
     *(ANSWER(1).NE.'EXIT'))THEN
      PRINT*,'PLEASE CONTINUE'
      PRINT*,'?'
      READ(*,'(8A10)')(ANSWER(I),I=1,16)
      ENDIF
      IF(ANSWER(1).EQ.'EXIT')RETURN
      IF((CONTRL.LE.0).AND.(ANSWER(1).NE.TENSTR).AND.
     *(ANSWER(1).NE.'DONE'))THEN
      CONTRL=CONTRL+1
      KDEEP=LVL+1
      NWRN=NWRN+1
      NNODES=NNODES+1
      PRINT*,'THE LAST SUBOBJECTIVE ENTERED IS:'
      II=1
      DO 10 I=1,4
      WRITE(*,'(1X,4A10)')(ANSWER(J),J=II,II+3)
```

```
      SUBROUTINE SPAN

      II=II+4
   10 CONTINUE
      IF(NNODES.GT.2)THEN
      CALL OBJECT(LEVEL(LVL,1),OBJCTV)
      II=1
      PRINT*,' '
      PRINT*,'WHICH IS SUBOBJECTIVE NUMBER ',NNRN
      PRINT*,'FOR THE OBJECTIVE:'
      DO 30 J=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(J),J=II,II+3)
      II=II+4
   30 CONTINUE
      ENDIF
      CALL ISET(IFINDT,IFADD,NNODES)
      CALL ISET(NNODES,1,NNRN)
      CALL ISET(NNODES,2,-1)
      CALL ISET(NNODES,3,-1)
      CALL ISET(NNODES,4,IFINDT)
      CALL ASET(NNODES,1,0.0)
      CALL ASET(NNODES,2,0.0)
      CALL OSET(NNODES,ANSWER)
      IFADD=3
      IFINDT=NNODES
      GO TO 4
      ELSE
      IF(U.NE.'Y')THEN
      CALL CHECK
      ELSE
      CALL NEXT
      ENDIF
      IFADD=2
      GO TO 3
      ENDIF
      ENDIF
      WRITE(*,'(5(/),1X)')
      CALL RDATT
      RETURN
      END
```

--VARIABLE MAP--(LO:A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ANSWER	2B	/S$ABUSE/	SAV	CHAR*10	16
CH	43B	/S$ABUSE/	SAV	CHAR*1	
CONTRL	6B	/CHILD/		INTEGER	
I	50B	/S$ABUSE/	SAV	INTEGER	
ICONT	0B	/MEX/		INTEGER	
ICOUNT	46B	/S$ABUSE/	SAV	INTEGER	
IDATA	1B	/MEX/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
IFINDT	47B	/S$ABUSE/	SAV	INTEGER	
II	51B	/S$ABUSE/	SAV	INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20
ITOTL	2B	/MEX/		INTEGER	
J	52B	/S$ABUSE/	SAV	INTEGER	
LABEL	1B	/S$ABUSE/	UNUSED/SAV	CHAR*10	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NDEEP	1B	/C/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
NNODES	3B	/C/		INTEGER	
NNRN	45B	/S$ABUSE/	SAV	INTEGER	
OBJCTV	23B	/S$ABUSE/	SAV	CHAR*10	16
TLNSTR	22B	/S$ABUSE/	SAV	CHAR*10	
U	44B	/S$ABUSE/	SAV	CHAR*1	
USER	0B	/P/		CHAR*10	

```
        SUBROUTINE SPAN

--PROCEDURES--(LO*A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  ASET                    3      SUBROUTINE       OCET                    2      SUBROUTINE
  CHECK                   0      SUBROUTINE       PREMEX                  0      SUBROUTINE
  ISET                    3      SUBROUTINE       PRETOT                  0      SUBROUTINE
  NEXT                    0      SUBROUTINE       RDATT                   0      SUBROUTINE
  OBJECT                  2      SUBROUTINE

--STATEMENT LABELS--(LO*A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

      1     13B                    36          10  INACTIVE   DO-TERM       116
      2     50B                    49          20  INACTIVE   DO-TERM        8S
      3    130B                    74          30  INACTIVE   DO-TERM       12G
      4    224B                    88

--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  SPAN        5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH            1105B *  581
  CM LABELLED COMMON LENGTH       207B *  135
  CM STORAGE USED               62600B *  25984
  COMPILE TIME                 0.286 SECONDS
```

```
      SUBROUTINE STAT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE DISPLAYS THE CHARACTERISTICS OF THE TREE       *
C*   FILE CURRENTLY IN USE.  IT IS CALLED DIRECTLY VIA           *
C*   OPTION *** STA ***.  IT IS INVOKED AUTOMATICALLY WHEN       *
C*   CONDUCTING THE BETWEEN NODE CHECK.                          *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DUMMY,CHECK                                     *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  NDEEP,NNODES,NSYS                             *
C*                                                               *
C*          MODIFIED:  (NONE)                                    *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/SYS1/NSYS
      SAVE
      WRITE(*,'(5(/),1X)')
      PRINT*,'CURRENT NUMBER OF NODES: ',NNODES,'(MAX 500)'
      PRINT*,'CURRENT NUMBER OF LEVELS: ',NDEEP,'(MAX 20)'
      PRINT*,'CURRENT NUMBER OF SYSTEMS: ',NSYS,'(MAX 59)'
      RETURN
      END
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  NDEEP       1B  /C/                          INTEGER
  NNODES      0B  /C/                          INTEGER
  NSYS        0B  /SYS1/                       INTEGER

 - ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  STAT        5B    0
        SUBROUTINE STAT        74/74    OPT-0

--STATISTICS--

 PROGRAM-UNIT LENGTH                  65B =  53
 CM LABELLED COMMON LENGTH            4B =  4
 CM STORAGE USED                    60600B =  24960
 COMPILE TIME                       0.032 SECONDS
```

```
      SUBROUTINE TAPER
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ALLOWS THE USER TO SELECT THE DESIRED          *
C*   DATA FILE   THE VARIABLE NTREE RESTRICTS THE CHOICE         *
C*   TO THREE POSSIBLE FILES.                                    *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DUMMY                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  USER                                          *
C*                                                               *
C*          MODIFIED:  NNODES,NTAPE,NTREE                        *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/F/USER
      COMMON/CNTRL/ISAVD,NTAPE
      SAVE
      CHARACTER*10 USER
      NTREE=3
      NNODES=0
      PRINT*,' '
      PRINT*,'THE FOLLOWING INFORMATION WILL ALLOW YOU TO CHOOSE'
      PRINT*,'AN EXISTING (STORED) DATA FILE, OR TO CONSTRUCT A'
      PRINT*,'NEW ONE. ',USER,'.'
      PRINT*,' '
      PRINT*,'THE CURRENT TREE IS NUMBER ',NTAPE
      CALL TSAVE
1     PRINT*,' '
      PRINT*,'WITH WHICH TREE WOULD YOU LIKE TO WORK. ',USER,'?'
      READ(*,'(I1)')NTAPE
      IF((NTAPE.LT.1).OR.(NTAPE.GT.NTREE))THEN
      PRINT*,'ONLY VALUES BETWEEN 1 AND ',NTREE,' ARE ALLOWED.'
      GO TO 1
      ENDIF
      CALL TLOAD
      IF(NNODES.LE.0)THEN
      PRINT*,'FILE ',NTAPE,' HAS NO CURRENT TREE STRUCTURE.  YOU ARE'
      PRINT*,'BEING TRANSFERRED TO OPTION *** NEW ***.'
      CALL INITT
      ENDIF
      RETURN
      END
```

```
          SUBROUTINE TAPER

--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ISAVD        0B  /CNTRL/                        INTEGER
  NNODES       2B  /$$ABUSE/ SAV                  INTEGER
  NTAPE        1B  /CNTRL/                        INTEGER
  NTREE        1B  /$$HBUSE/ SAV                  INTEGER
  USER         0B  /P/                            CHAR*10

--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----

  INITT                    0      SUBROUTINE
  TLOAD                    0      SUBROUTINE
  TSAVE                    0      SUBROUTINE

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      1     31B                     37

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  TAPER        5B     0

--STATISTICS--

  PROGRAM-UNIT LENGTH              212B =  138
  CM LABELLED COMMON LENGTH          6B =  6
  CM STORAGE USED                 60680B =  24960
  COMPILE TIME                    0.066 SECONDS
```

```
      SUBROUTINE VALUE
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES THE DATA POINTS FROM WHICH THE       *
C*   INDIVIDUAL VALUE FUNCTIONS ARE ESTIMATED, AND CALLS         *
C*   ROUTINE REGRS TO DO THE ESTIMATION.                         *
C*                                                               *
C*                                                               *
C*   CALLED BY:  PRI,RDATT                                       *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  NATT,XMID                                     *
C*                                                               *
C*          MODIFIED:  CH,I,VAL,X                                *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/VAL/VAL(5)
      SAVE
      CHARACTER CH,ATT1*10
      DO 10 I=1,NATT
1     CALL ASK(I,ATT(I,1),ATT(I,2),XMID)
      VAL(1)=ATT(I,1)
      VAL(5)=ATT(I,2)
      VAL(3)=XMID
      X=XMID
      CALL ASK(I,ATT(I,1),X,XMID)
      VAL(2)=XMID
      CALL ASK(I,ATT(I,2),VAL(3),XMID)
      VAL(4)=XMID
      CALL REGRS(I)
      PRINT*,' '
      PRINT*,'THE ABOVE YIELDS A VALUE FUNCTION FOR ',ATT1(I)
      PRINT*,'WITH PARAMETERS:'
      PRINT*,'B0=',PARAM(I,1),'    B1=',PARAM(I,2)
      PRINT*,'SUM OF SQUARED ERROR=',PARAM(I,3)
      IF(PARAM(I,4).EQ.0.0)THEN
      PRINT*,'(LINEAR FORM):'
      PRINT*,'VALUE=B0+B1*(ATTRIBUTE LEVEL)'
      ELSEIF(PARAM(I,4).EQ.-1.0)THEN
      PRINT*,'(SQUARE-ROOT FORM)'
      PRINT*,'VALUE=B0+B1*(ATTRIBUTE LEVEL)**.5'
      ELSEIF(PARAM(I,4).EQ.1.0)THEN
      PRINT*,'(SQUARED FORM)'
      PRINT*,'VALUE=B0+B1*(ATTRIBUTE LEVEL)**2'
      ELSEIF(PARAM(I,4).EQ.-2.0)THEN
      PRINT*,'(LOGARITHMIC FORM)'
      PRINT*,'VALUE=B0+B1*LN(ATTRIBUTE LEVEL)'
      ELSEIF(PARAM(I,4).EQ.2.0)THEN
      PRINT*,'(EXPONENTIAL FORM)'
```

```
      SUBROUTINE VALUE

 58                 PRINT*,'VALUE=B0+B1*EXP(ATTRIBUTE LEVEL)'
 59                 ENDIF
 60                 CALL PICT.R(1)
 61           2     PRINT*,
 62                 PRINT*,'DOES THE ABOVE REPRESENTATION APPEAR REASONABLE? (Y/N)'
 63                 PRINT*,'      ?'
 64                 READ(*, (A1)')CH
 65                 IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
 66                 PRINT*,'YOU MUST ENTER "Y" OR "N"'
 67                 GO TO 2
 68                 ENDIF
 69                 IF(CH.EQ. N')GO TO 1
 70           10    CONTINUE
 71                 RETURN
 72                 END
```

```
VARIABLE MAP--(LO-A)
-NAME---ADDRESS--BLOCK-----PROPERTIES---- --TYPE --------SIZE
 CH          1B   /$$ABUSE/ SAV          CHAR*1
 I           2B   /$$ABUSE/ SAV          INTEGER
 NA/T        4B   ATTR*                  INTEGER

-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
 VAL         0B   /VAL/                          REAL          5
 X           4B   /$$ABUSE/ SAV                  REAL
 XMID        3B   /$$ABUSE/ SAV                  REAL

--PROCEDURES--(LO-A)
-NAME------TYPE---------ARGS------CLASS-----
 ASL                     4      SUBROUTINE
 ATT         REAL        2      FUNCTION
 ATT1        CHAR*10     1      FUNCTION

-NAME------TYPE--------ARGS------CLASS-----
 PARAM     REAL          2      FUNCTION
 PICTUR                  1      SUBROUTINE
 REGRS                   1      SUBROUTINE

--STATEMENT LABELS--(LO-A)
-LABEL-ADDRESS-----PROPERTIES----DEF
    1     14B                     29
    2    153B                     61
   10  INACTIVE   DO-TERM         70

--ENTRY POINTS--(LO-A)
-NAME---ADDRESS--ARGS---
 VALUE      5B      0

--STATISTICS--
 PROGRAM-UNIT LENGTH          467B =  311
 CM LABELLED COMMON LENGTH     13B =   11
 CM STORAGE USED            606000B = 24960
 COMPILE TIME               0.131 SECONDS
```

```
      OVERLAY(XFILE,2,0)
      PROGRAM UULOAD
C
C
C*****************************************************************
C*                                                               *
C*   THIS PROGRAM SERVES AS THE ENTRY POINT FOR OVERLAY(2,0).    *
C*   IT DIRECTS THE *** UUC *** OPTION.  THE OPTION ALLOWS       *
C*   THE ENTRY OF RELATIVE WEIGHTS OF NODES, ATTRIBUTE LEVELS    *
C*   OF THE ALTERNATIVE SYSTEMS AT EACH DATA NODE, AND           *
C*   CALCULATING THE CUMULATIVE WEIGHTS AND INTERIOR VALUES.     *
C*                                                               *
C*                                                               *
C*   CALLED BY:  (NONE)                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  CH                                        *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/C/NNODES,NUEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/ONE/CMD
      COMMON/P/USCR
      COMMON/RRAY1/ARRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      COMMON/SYS1/NSYS
      CHARACTER CH
1     WRITE(*,'(5(/),1X)')
      PRINT*,'YOU MAY NOW ENTER WEIGHTS, VALUES, OR (RE)CALCULATE'
      PRINT*,'THE TREE.  CHOOSE YOUR OPTION:'
      PRINT*,'  W(EIGHT      V(ALUES      C(ALCULATE      E(XIT'
      PRINT*,' '
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'W').AND.(CH.NE.'V').AND.(CH.NE.'C').AND.
     * (CH.NE.'E'))THEN
      PRINT*,'YOU MUST ENTER "W","V","C", OR "E" '
      GO TO 1
      ENDIF
      IF(CH.EQ.'C')THEN
      CALL CALC
      ELSEIF(CH.NE.'E')THEN
      CALL UULODI(CH)
      GO TO 1
      ENDIF
      RETURN
*     RETURN IN MAIN PROGRAM -- ACTS AS END
```

```
PROGRAM UVLOAD

 57                    END
 .

--VARIABLE MAP--(LO-A)
 NAME---ADDRESS -BLOCK-----PROPERTIES-------TYPE---------SIZE

 ARRAY      0B  /ARRAY1/                      REAL          80
 CH         0B                                CHARX1
 CMD        0B  /ONE/                         REAL
 ICONT      0B  /NEX/                         INTEGER
 IDATA      0B  /NEX/                         INTEGER
 IFADD      0B  /LEVEL/                       INTEGER
 IFIND      0B  /LEVEL/                       INTEGER
 INNPN      0B  /LEVEL/                       INTEGER       20
 ISAVD      0B  /CNTRL/                       INTEGER
 ISIN       0B  /RRAY2/                       INTEGER
 ITOTL      0B  /NEX/                         INTEGER

 PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----

 CALL                     0       SUBROUTINE
 UVLOD1                   1       SUBROUTINE

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1      0B                      38

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

 UVLOAD      0B      0

--STATISTICS--

 PROGRAM-UNIT LENGTH                0B =  0
 CM LABELLED COMMON LENGTH        262B =  178
 CM STORAGE USED                 60600B =  24960
 COMPILE TIME                   0.062 SECONDS

     1  ANSI     ERROR IN  UVLOAD
```

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

LEVEL       0B  /LEVEL/                       INTEGER       60
LVL         0B  /LEVEL/                       INTEGER
NATT        0B  /ATTR/                        INTEGER
NDEEP       0B  /C/                           INTEGER
NDIFF       0B  /LEVEL/                       INTEGER
NLOUD       0B  /RRAY2/                       INTEGER
NLVLS       0B  /LEVEL/                       INTEGER
NNODES      0B  /C/                           INTEGER
NSYS        0B  /SYS1/                        INTEGER
NTAPE       0B  /CNTRL/                       INTEGER
USER        0B  /P/                           REAL
```

```
      SUBROUTINE CALC
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE CALCULATES THE CUMULATIVE WEIGHT OF EACH       *
C*   NODE, AND THE ALTERNATIVE SYSTEM VALUES AT EACH             *
C*   INTERIOR (NON-DATA) NODE.                                   *
C*                                                               *
C*                                                               *
C*   CALLED BY:  WVLOAD                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ICONT,IDATA,IFIND,LEVEL,LVL,NNODES,NSYS       *
C*                                                               *
C*          MODIFIED:  I,ISTOP,J,JJ,NDEEP,VHOLD                  *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/C/NNODES,NDEEP
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/SYS1/NSYS
      SAVE
      IF(NNODES.LE.0)RETURN
      PRINT*,'INTERIOR TREE VALUES ARE BEING CALCULATED...'
      NDEEP=0
      CALL PRETOT
      CALL ASET(1,1,1.0)
      CALL ASET(1,2,1.0)
      CALL ASET(2,1,1.0)
      CALL ASET(2,2,1.0)
1     IF(ICONT.EQ.0)THEN
      CALL PRETOT
      DO 5 II=1,NNODES
      IF(ICONT.EQ.0)RETURN
      IF(IDATA.NE.1)THEN
      DO 4 JJ=1,NSYS
      VHOLD=VRAY(IFIND,JJ)/ARAY(IFIND,2)
      CALL VSET(IFIND,JJ,VHOLD)
4     CONTINUE
      ENDIF
      CALL NEXT
5     CONTINUE
      ENDIF
      IF(LVL.GT.NDEEP)NDEEP=LVL
      VHOLD=ARAY(IFIND,2)
      IF((LVL-1).GE.1)
     *VHOLD=ARAY(IFIND,1)*ARAY(LEVEL(LVL-1,1),2)
      CALL ASET(IFIND,2,VHOLD)
      IF(IDATA.NE.1)THEN
      DO 10 J=1,NSYS
10    CALL VSET(IFIND,J,0.0)
```

```
SUBROUTINE CALC

58                ELSE
59                ISTOP=LVL-1
60                DO 20 I=1,ISTOP
61                DO 20 J=1,NSYS
62                VHOLD=VRAY(LEVEL(I,1),J)+ARAY(IFIND,2)*VRAY(IFIND,J)
63                CALL VSET(LEVEL(I,1),J,VHOLD)
64          20    CONTINUE
65                ENDIF
66                CALL NEXT
67                GO TO 1
68                END
```

SUBROUTINE CALC

--STATISTICS--

PROGRAM-UNIT LENGTH	341B =	225
CM LABELLED COMMON LENGTH	1422 =	98
CM STORAGE USED	606003 =	24960
COMPILE TIME	0.115 SECONDS	

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
I	6B	/S$ABUSE/	SAV	INTEGER	
ICONT	0B	/NEX/		INTEGER	
IDATA	1B	/NEX/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
II	1B	/S$ABUSE/	SAV	INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20
ISTOP	5B	/S$ABUSE/	SAV	INTEGER	
ITOTL	2B	/NEX/		INTEGER	
J	4B	/S$ABUSE/	SAV	INTEGER	
JJ	2B	/S$ABUSE/	SAV	INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NDEEP	1B	/C/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
NNODES	0B	/C/		INTEGER	
NSYS	0B	/SYS1/		INTEGER	
VHOLD	3B	/S$ABUSE/	SAV	REAL	

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ARAY	REAL	2	FUNCTION
ASET		3	SUBROUTINE
NEXT		0	SUBROUTINE
PRETOT		0	SUBROUTINE
VRAY	REAL	2	FUNCTION
VSET		3	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
1	30B		37
4	INACTIVE	DO-TERM	45
5	INACTIVE	DO-TERM	48
10	INACTIVE	DO-TERM	57
20	INACTIVE	DO-TERM	64

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
CALC	5B	0

```
 1                SUBROUTINE DUIDE(N,A,D)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS ROUTINE DIVIDES EACH OF THE ELEMENTS (N OF THEM) OF    *
 8          C*   A VECTOR A BY D.                                            *
 9          C*                                                               *
10          C*                                                               *
11          C*   CALLED BY:  RDUT                                            *
12          C*                                                               *
13          C*                                                               *
14          C*   VARIABLES                                                   *
15          C*          USED:  D,N                                           *
16          C*                                                               *
17          C*          MODIFIED:  A                                         *
18          C*                                                               *
19          C*                                                               *
20          C*****************************************************************
21          C
22          C
23                REAL A(N)
24                DO 10 I=1,N
25          10    A(I)=A(I)/D
26                RETURN
27                END
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  A          2   DUMMY-ARG                    REAL       ADJ-ARY
  D          3   DUMMY-ARG                    REAL
  I         35B                               INTEGER
  N          1   DUMMY-ARG                    INTEGER


--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        25


--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  DUIDE      5B     3
        SUBROUTINE DUIDE


--STATISTICS--
 PROGRAM-UNIT LENGTH              41B =  33
 CM STORAGE USED                60000B =  24960
 COMPILE TIME                   0.025 SECONDS
```

```
      SUBROUTINE GMVEC(NCROSS,IFIND,AIN)
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE ALLOWS THE USER TO ENTER THE RELATIVE WEIGHTS  *
C*   ACROSS THE SPAN IMPLICITLY BY USING A PAIRWISE-COMPARISON   *
C*   MATRIX.  THE RELATIVE WEIGHTS ARE THE ELEMENTS OF THE       *
C*   GEOMETRIC MEAN VECTOR (ACROSS THE ROWS OF THE MATRIX)       *
C*   NORMALIZED TO SUM TO 1.                                     *
C*                                                               *
C*                                                               *
C*   CALLED BY:  RDUT                                            *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  IFIND,NCROSS,OBJCTV                           *
C*                                                               *
C*          MODIFIED:  AIN,ANSWER,I,II,IFINDT,J,K,KK,MATRIX,     *
C*                     OBJCT,PROD,X1,X2                          *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      REAL AIN(20),MATRIX(20,20)
      CHARACTER*10 OBJCT(10,16),OBJCTV(16)
      INTEGER ANSWER
      IFINDT=IRAY(IFIND,2)
      CALL OBJECT(IFINDT,OBJCTV)
      DO 20 I=1,NCROSS
      DO 10 J=1,16
10    OBJCT(I,J)=OBJCTV(J)
      IF(I.LT.NCROSS)THEN
      IFINDT=IRAY(IFINDT,3)
      CALL OBJECT(IFINDT,OBJCTV)
      ENDIF
20    CONTINUE
      DO 40 I=1,NCROSS
      IF(I.LT.NCROSS)THEN
      DO 30 J=I+1,NCROSS
      PRINT*,'USING THE SCALE:'
      PRINT*,'1-9 AS DEFINED IN THE USER''S MANUAL,'
      PRINT*,'HOW IMPORTANT IS FACTOR ',I,':'
      II=1
      DO 50 K=1,4
      WRITE(*,'(1X,4A10)')(OBJCT(I,KK),KK=II,II+3)
      II=II+4
50    CONTINUE
      PRINT*,' '
      PRINT*,'COMPARED TO FACTOR ',J,':'
      II=1
      DO 60 K=1,4
      WRITE(*,'(1X,4A10)')(OBJCT(J,KK),KK=II,II+3)
      II=II+4
60    CONTINUE
```

```
      SUBROUTINE GMVEC

 58                PRINT*,'ENTER THE NUMERATOR OF THE RATIO...'
 59                PRINT*,'      ?'
 60                READ(*,'(I1)')ANSWER
 61                X1=FLOAT(ANSWER)
 62                PRINT*,'ENTER THE DENOMINATOR OF THE RATIO...'
 63                PRINT*,'      ?'
 64                READ(*,'(I1)')ANSWER
 65                X2=FLOAT(ANSWER)
 66                MATRIX(I,J)=X1/X2
 67                MATRIX(J,I)=X2/X1
 68          30    CONTINUE
 69                ENDIF
 70                MATRIX(I,I)=1.0
 71                PROD=1.0
 72                DO 70 J=1,NCROSS
 73          70    PROD=PROD*MATRIX(I,J)
 74                AIN(I)=PROD**(1.0/FLOAT(NCROSS))
 75          40    CONTINUE
 76                RETURN
 77                END
```

```
                    SUBROUTINE GMVEC

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  GMVEC       58      3

--STATISTICS--

  PROGRAM-UNIT LENGTH          16058 =  901
  CM STORAGE USED              60600B =  24960
  COMPILE TIME                 0.161 SECONDS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AIN          3   DUMMY-ARG                REAL            20
  ANSWER    1560B                           INTEGER
  I         1562B                           INTEGER
  IFIND        2   DUMMY-ARG                INTEGER
  IFINDT    1561B                           INTEGER
  II        1570B                           INTEGER
  J         1564B                           INTEGER
  K         1571B                           INTEGER
```

```
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  KK        1573B                           INTEGER
  MATRIX     460B                           REAL          460
  NCROSS       1   DUMMY-ARG                INTEGER
  OBJCT     1300B                           CHAR*10       160
  OBJCTV    1540B                           CHAR*10        16
  PROD      1601B                           REAL
  X1        1577B                           REAL
  X2        1600B                           REAL
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE---------ARGS------CLASS-----

  FLOAT      REAL          1      INTRINSIC
  IPAY       INTEGER       2      FUNCTION
  OBJECT                   2      SUBROUTINE
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM      34
     20  INACTIVE   DO-TERM      39
     30  INACTIVE   DO-TERM      68
     40  INACTIVE   DO-TERM      76
```

```
 -LABEL-ADDRESS-----PROPERTIES----DEF

     50  INACTIVE   DO-TERM      50
     60  INACTIVE   DO-TERM      57
     70  INACTIVE   DO-TERM      73
```

```
      SUBROUTINE PDV
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ELICITS THE ATTRIBUTE LEVELS OF THE            *
C*   ALTERNATIVE SYSTEMS ACROSS ALL THE DATA NODES.              *
C*                                                               *
C*                                                               *
C*   CALLED BY:  UVLOD1                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  IFIND,NATT,NSYS,OBJCTV                        *
C*                                                               *
C*          MODIFIED:  ANSWER,I,II,J,XMAX,XMIN                   *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 LABEL,OBJCTV(16),SYSLBL,ATT1
      PRINT*,'WE ARE AT THE DATA NODE:'
      CALL OBJECT(IFIND,OBJCTV)
      II=1
      DO 5 I=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
5     CONTINUE
      PRINT*,'WHICH HAS THE ASSOCIATED ATTRIBUTE ',LABEL(IFIND)
      DO 10 I=1,NSYS
      PRINT*,' '
      PRINT*,'THE CURRENT LEVEL OF THE ATTRIBUTE ',LABEL(IFIND)
      PRINT*,'IS ',XLEVEL(VRAY(IFIND,I),LABEL(IFIND)),' FOR SYSTEM'
      PRINT*,SYSLBL(I)
      DO 20 J=1,NATT
      IF(LABEL(IFIND).EQ.ATT1(J))THEN
      PRINT*,' '
      PRINT*,'THE RANGE OF THE ATTRIBUTE IS ',ATT(J,1),' TO ',ATT(J,2)
      XMIN=AMIN1(ATT(J,1),ATT(J,2))
      XMAX=AMAX1(ATT(J,1),ATT(J,2))
      GO TO 1
      ENDIF
20    CONTINUE
1     PRINT*,' '
      PRINT*,'WHAT IS THE NEW LEVEL (REAL NUMBER)?'
      READ(*,'(F10.0)')ANSWER
      IF((ANSWER.LT.XMIN).OR.(ANSWER.GT.XMAX))THEN
      PRINT*,'THE LEVEL OF ',ANSWER,' IS OUT OF RANGE.'
      GO TO 1
      ENDIF
```

```
      SUBROUTINE RDV

                    ANSWER=VALU(ANSWER,LABEL(IFIND))
                    IF((ANSWER.LT.0.0).OR.(ANSWER.GT.1.0))THEN
                    ANSWER=AMIN1(ANSWER,1.0)
                    ANSWER=AMAX1(ANSWER,0.0)
                    PRINT*,'THE INPUT ATTRIBUTE LEVEL CAUSES'
                    PRINT*,'THE VALUE GENERATED (BASED ON THE '
                    PRINT*,'ESTIMATED INDIVIDUAL VALUE FUNCTION)'
                    PRINT*,'TO BE OUTSIDE THE RANGE (0.0-1.0).'
                    PRINT*,'IN ORDER TO REMAIN IN THE PROPER RANGE,'
                    PRINT*,'YOUR INPUT VALUE IS BEING CHANGED TO'
                    PRINT*,XLEVEL(ANSWER,LABEL(IFIND))
                    PRINT*,'IF THIS IS UNACCEPTABLE, USE ***ATT***'
                    PRINT*,'TO ADJUST THE VALUE FUNCTION, AFTER'
                    PRINT*,'EXITTING THIS OPTION.'
                    ENDIF
                    CALL VSET(IFIND,I,ANSWER)
              10    CONTINUE
                    PRINT*,' '
                    PRINT*,'ENTER COMMENTS ON THESE ENTRIES.'
                    CALL CSET(IFIND)
                    RETURN
                    END
```

```
--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      1     162B                      51
      5  INACTIVE    DO-TERM         35
     10  INACTIVE    DO-TERM         74
     20  INACTIVE    DO-TERM         50

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  RDV        5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              575B =  381
  CM LABELLED COMMON LENGTH        162B =  114
  CM STORAGE USED                62600B = 25984
  COMPILE TIME                   0.163 SECONDS
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ANSWER     27B  /S$ASVSE/ SAV          REAL
 I          22B  /S$ASV2E/ SAV          INTEGER
 ICONT       0B  /NEX/                  INTEGER
 IDATA       1B  /NEX/                  INTEGER
 IFADD      27B  /LEVEL/                INTEGER
 IFIND      25B  /LEVEL/                INTEGER
 II         21B  /S$ASVSE/ SAV          INTEGER
 INNRM       1B  /LEVEL/                INTEGER        20
 ITOTL       2B  /NEX/                  INTEGER
 J          24B  /S$ASVSE/ SAV          INTEGER
```

```
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 K          23B  /S$ASVSE/ SAV          INTEGER
 LEVEL      31B  /LEVEL/                INTEGER        60
 LVL        30B  /LEVEL/                INTEGER
 NATT        0B  /ATTR/                 INTEGER
 NDIFF      26B  /LEVEL/                INTEGER
 NLVLS       0B  /LEVEL/                INTEGER
 NSYS        0B  /SYS1/                 INTEGER
 OBJCTV      1B  /S$ASVSE/ SAV          CHAR*10        16
 XMAX       26B  /S$ASVSE/ SAV          REAL
 XMIN       25B  /S$ASVSE/ SAV          REAL
```

```
--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----

 AMAX1      REAL          7      INTRINSIC
 AMIN1      REAL          7      INTRINSIC
 ATT        REAL          2      FUNCTION
 ATT1       CHAR*12       1      FUNCTION
 CSET                     1      SUBROUTINE
 LABEL      CHAR*10       1      FUNCTION
```

```
 -NAME------TYPE--------ARGS------CLASS-----

 OBJECT                   2      SUBROUTINE
 SYSLBL     CHAR*10       1      FUNCTION
 VALU       REAL          2      FUNCTION
 VRAY       REAL          2      FUNCTION
 VSET                     3      SUBROUTINE
 XLEVEL     REAL          2      FUNCTION
```

```
      SUBROUTINE RDWT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ELICITS THE RELATIVE WEIGHTS OF ALL THE        *
C*   NODES.  THEY MAY BE INPUT DIRECTLY, OR INDIRECTLY           *
C*   (THROUGH THE USE OF A PAIRWISE-COMPARISON APPROACH).        *
C*   INPUTS ARE NORMALIZED TO SUM TO UNITY ACROSS EACH SPAN.     *
C*                                                               *
C*                                                               *
C*   CALLED BY:  WULOD1                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  LEVEL,OBJCTV                                  *
C*                                                               *
C*          MODIFIED:  AIN,CH,I,IFIND,IFINDT,II,ITOTL,J,K,       *
C*                     NCROSS,ZNORM                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INMRH(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      CHARACTER*10 OBJCTV(16),CH*1
      REAL AIN(20)
      IFIND=LEVEL(LVL,1)
      IFINDT=IRAY(IFIND,2)
      NCROSS=1
      PRINT*,'WE ARE WEIGHTING THE NODE SET:'
      CALL OBJECT(IFINDT,OBJCTV)
      II=1
      DO 5 J=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
5     CONTINUE
      PRINT*,'THE ABOVE OBJECTIVE IS FACTOR 1'
1     IFINDT=IRAY(IFINDT,3)
      IF(IFINDT.GT.0)THEN
      CALL OBJECT(IFINDT,OBJCTV)
      II=1
      DO 10 J=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
10    CONTINUE
      NCROSS=NCROSS+1
      PRINT*,'THE ABOVE OBJECTIVE IS FACTOR ',NCROSS
      GO TO 1
      ENDIF
2     PRINT*,' '
      PRINT*,'DO YOU WISH TO ENTER THE RELATIVE WEIGHTS DIRECTLY,'
      PRINT*,'(Y/N)  ?'
```

SUBROUTINE RDUT

```
58                  READ(*,'(A1)')CH
59                  IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
60                  PRINT*,'ENTER 'Y' OR 'N' '
61                  GO TO 2
62                  ENDIF
63                  IF(CH.EQ.'N')THEN
64                  CALL GNVEC(NCROSS,IFIND,AIN)
65                  ELSE
66                  PRINT*,'ENTER THE (UNNORMALIZED) WEIGHTS.'
67                  DO 20 I=1,NCROSS
68                  PRINT*,'WHAT IS THE WEIGHT FOR FACTOR ',I
69                  PRINT*,'      ?'
70                  READ*,AIN(I)
71                  IF(AIN(1).EQ.0.0)GO TO 4
72            20    CONTINUE
73            4     IF(AIN(1).EQ.0.0)THEN
74                  ITOTL=0
75                  RETURN
76                  ENDIF
77                  ENDIF
78                  IF(NCROSS.EQ.1)CALL ASET(IRAY(IFIND,2),1,0.0)
79                  ZNORM=SUM(NCROSS,AIN)
80                  IF(ZNORM.LT.0.0)RETURN
81                  IF(ZNORM.GT.0.0)CALL DVIDE(NCROSS,AIN,ZNORM)
82                  PRINT*,'NORMALIZED:',((IFIX(100.0*AIN(I)+.5)),I=1,NCROSS)
83            3     PRINT*,' '
84                  PRINT*,'ARE YOU HAPPY WITH THESE RELATIVE WEIGHTS? (Y/N)'
85                  PRINT*,'      ?'
86                  READ(*,'(A1)')CH
87                  IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
88                  PRINT*,'PLEASE ENTER 'Y' OR 'N' '
89                  GO TO 3
90                  ENDIF
91                  IF(CH.EQ.'N')GO TO 2
92                  IFINDT=IRAY(IFIND,2)
93                  DO 11 I=1,NCROSS
94                  CALL ASET(IFINDT,1,AIN(I))
95                  IFINDT=IRAY(IFINDT,3)
96            11    CONTINUE
97                  PRINT*,'ENTER COMMENTS ON THESE WEIGHTS'
98                  CALL CSET(IFIND)
99                  RETURN
100                 END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
AIN	22B	/SAUSE/	SAV	REAL	20
CH	21B	/SAUSE/	SAV	CHAR*1	
I	53B	/SAUSE/	SAV	INTEGER	
ICONT	0B	/NEX/		INTEGER	
IDATA	1B	/NEX/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
IFINDT	46B	/SAUSE/	SAV	INTEGER	
II	50B	/SAUSE/	SAV	INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
INNRM	1B	/LEVEL/		INTEGER	20
ISAUD	0B	/CNTRL/		INTEGER	
ITOTL	2B	/NEX/		INTEGER	
J	51B	/SAUSE/	SAV	INTEGER	
K	52B	/SAUSE/	SAV	INTEGER	
LEVEL	3B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NCROSS	47B	/SAUSE/	SAV	INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	

```
        SUBROUTINE RDWT        74/74   OPT=0
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 HLULS        0B   /LEVEL/                    INTEGER
 HTAPE        1B   /CNTRL/                    INTEGER

--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 ASET                    3       SUBROUTINE
 CSET                    1       SUBROUTINE
 DVIDE                   3       SUBROUTINE
 UNVEC                   3       SUBROUTINE

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

     1     66B                     43
     2    144B                     55
     3    300B                     83
     4    227B                     73

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 RDWT        5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              631B =  409
  CM LABELLED COMMON LENGTH        207B =  135
  CM STORAGE USED                 606000 =  24960
  COMPILE TIME                   0.192 SECONDS
```

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 OBJCTV       1B   /$$ABUSE/ SAV              CHAR*10        16
 ZNORM       54B   /$$ABUSE/ SAV              REAL

-NAME------TYPE--------ARGS------CLASS-----

 IFIX       INTEGER        1       INTRINSIC
 IRAY       INTEGER        2       FUNCTION
 OBJECT                    2       SUBROUTINE
 SUM        REAL           2       FUNCTION

-LABEL-ADDRESS-----PROPERTIES----DEF

     5   INACTIVE   DO-TERM        41
    10   INACTIVE   DO-TERM        50
    11   INACTIVE   DO-TERM        96
    20   INACTIVE   DO-TERM        72
```

```
 1                FUNCTION SUM(N,A)
 2          C
 3          C
 4          C*****************************************************************
 5          C*                                                               *
 6          C*                                                               *
 7          C*   THIS FUNCTION COMPUTES THE SUM OF THE ELEMENTS OF A         *
 8          C*   VECTOR A (N ELEMENTS).                                      *
 9          C*                                                               *
10          C*                                                               *
11          C*   CALLED BY:  RDWT                                            *
12          C*                                                               *
13          C*                                                               *
14          C*   VARIABLES                                                   *
15          C*          USED:  A,N                                           *
16          C*                                                               *
17          C*          MODIFIED:  I                                         *
18          C*                                                               *
19          C*                                                               *
20          C*****************************************************************
21          C
22          C
23                REAL A(N)
24                SUM=0.0
25                DO 10 I=1,N
26          10    SUM=SUM+A(I)
27                RETURN
28                END

--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  A          2   DUMMY-ARG                    REAL       ADJ-ARY
  I         36B                               INTEGER
  N          1   DUMMY-ARG                    INTEGER
  SUM       35B                               REAL

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        26

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  SUM        6B     2
          FUNCTION SUM:          74/74   OPT=0

--STATISTICS--

  PROGRAM-UNIT LENGTH              42B =  34
  CM STORAGE USED                 606000B =  24960
  COMPILE TIME                    0.823 SECONDS
```

```
      SUBROUTINE WULCD:(OPT)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE DIRECTS THE INPUT OF RELATIVE                  *
C*   WEIGHTS OR ATTRIBUTE LEVELS.                                *
C*                                                               *
C*                                                               *
C*   CALLED BY:  WULOAD                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ICONT,OPT                                     *
C*                                                               *
C*          MODIFIED:  CH,CH1,LABEL                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/NEX/ICONT,IDATA,ITOTL
      SAVE
      CHARACTER CH,CH1,OPT,LABEL*10
      IF(OPT.EQ.'W')THEN
      LABEL='WEIGHTS'
      ELSE
      LABEL='VALUES'
      ENDIF
1     PRINT*,' '
      PRINT*,LABEL,':  ALL      SELECT'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'A').AND.(CH.NE.'S'))THEN
      PRINT*,'I AM SORRY, BUT I CAN ONLY ACCEPT "A" OR "S".'
      GO TO 1
      ENDIF
      PRINT*,' '
      CALL PRETOT
2     IF(CH.EQ.'S')CALL PRENEX
      IF(ICONT.EQ.0)THEN
      PRINT*,'NRN ENTERED IS INVALID.'
      PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH1
      ENDIF
      IF((ICONT.EQ.0).AND.(CH.EQ.'S'))GO TO 2
3     IF((IDATA.EQ.0).AND.(OPT.EQ.'W'))CALL RDWT
      IF((IDATA.EQ.1).AND.(OPT.EQ.'V'))CALL RDV
      IF(ITOTL.EQ.1)CALL NEXT
      IF((ICONT.NE.0).AND.(ITOTL.EQ.1))GO TO 3
      IF(ITOTL.EQ.0)GO TO 2
      RETURN
      END
```

```
        SUBROUTINE UVLOD1

--VARIABLE MAP--(LO-A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 CH          1B   /SSASUBE/ SAV          CHAR*1
 CH1         2B   /SSASUBE/ SAV          CHAR*1
 ICONT       0B   /NEX/                  INTEGER
 LDATA       1B   /NEX/                  INTEGER

-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ITOTL       2B   /NEX/                  INTEGER
 LABEL       3B   /SSASUBE/ SAV          CHAR*10
 OPT         1    DUMMY-ARG              CHAR*1

--PROCEDURES--(LO-A)
-NAME------TYPE--------ARGS------CLASS-----

 NEXT                    0       SUBROUTINE
 PRENEX                  0       SUBROUTINE
 PRETOT                  0       SUBROUTINE
 RDU                     0       SUBROUTINE
 RDUT                    0       SUBROUTINE

--STATEMENT LABELS--(LO-A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    1     24B                     31
    2     53B                     41
    3    103B                     49

--ENTRY POINTS--(LO-A)
-NAME---ADDRESS--ARGS---

 UVLOD1      5B      1

--STATISTICS--

 PROGRAM-UNIT LENGTH              271B =  185
 CM LABELLED COMMON LENGTH         7B =  7
 CM STORAGE USED                 60600B =  24960
 COMPILE TIME                    0.090 SECONDS
```

```
      OVERLAY(XFILE,3,0)
      PROGRAM HUM
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE IS THE ENTRY POINT FOR OVERLAY(3,0).  IT       *
C*   ACTS AS A DRIVER FOR THE DISPLAY/REVIEW OPTIONS.            *
C*                                                               *
C*                                                               *
C*   CALLED BY:  (NONE)                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  CMD,IDATA                                     *
C*                                                               *
C*          MODIFIED:  CH                                        *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INNRM(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/ONE/CMD
      COMMON/P/USER
      COMMON/RRAY1/ARRAY(80)
      COMMON/RRAY2/ISIN,NLOUD
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*3 CMD,CH*1
      CALL PRETOT
      IF(CMD.EQ.'DIS')THEN
1     WRITE(*,'(5(/),1X)')
      PRINT*,'ENTER NODE TO BE DISPLAYED...'
      CALL NODIN
      IF(IDATA.EQ.1)THEN
      PRINT*,'NODE IS A DATA NODE, AND CANNOT BE DISPLAYED'
      PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      GO TO 1
      ENDIF
      ELSE
      CALL PRETOT
2     WRITE(*,'(5(/),1X)')
      PRINT*,'HOW MUCH DO YOU WANT TO REVIEW...'
      PRINT*,'      A(LL      S(ELECT'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'A').AND.(CH.NE.'S'))THEN
      PRINT*,'ENTER "A" OR "S"'
      GO TO 2
```

```
      58                ENDIF
      59                IF(CH.EQ.'S')CALL PRENEX
      60                ENDIF
      61                PRINT*,'IF ANY MODIFICATIONS HAVE BEEN MADE TO THE TREE'
      62                PRINT*,'SINCE IT WAS BEEN CALCULATED ,NUMERICAL VALUES'
      63                PRINT*,'WILL BE INCORRECT.'
      64                PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
      65                PRINT*,'      ?'
      66                READ(*,'(A1)')CH
      67                IF(CMD.EQ.'DIS')THEN
      68                CALL DISPLA
      69                ELSEIF((CMD.EQ.'NUM').OR.(CMD.EQ.'REV'))THEN
      70                CALL NUMREV
      71                ENDIF
      72                RETURN
ANSI     *        RETURN IN MAIN PROGRAM -- ACTS AS END
      73                END
```



```
--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  NUM          0B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH                  05 =  0
  CM LABELLED COMMON LENGTH          264B =  180
  CM STORAGE USED                  606003 =  24960
  COMPILE TIME                     0.025 SECONDS

     1  ANSI      ERROR IN  NUM
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ARRAY       0B  /RRAY1/                    REAL           80
  CH          0B  /S0A$USE/ SAV              CHAR*1
  CMD         0B  /ONE/                      CHAR*3
  ICONT       0B  /NEX/                      INTEGER
  IDATA       0B  /NEX/                      INTEGER
  IFADD       0B  /LEVEL/                    INTEGER
  IFIND       0B  /LEVEL/                    INTEGER
  INNRN       0B  /LEVEL/                    INTEGER        20
  ISAVD       0B  /CNTRL/                    INTEGER
  ISIN        0B  /RRAY2/                    INTEGER
  ITOTL       0B  /NEX/                      INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  LEVEL       0B  /LEVEL/                    INTEGER        60
  LVL         0B  /LEVEL/                    INTEGER
  NATT        0B  /ATTR/                     INTEGER
  NDEEP       0B  /C/                        INTEGER
  NDIFF       0B  /LEVEL/                    INTEGER
  NLOUD       0B  /RRAY2/                    INTEGER
  NLVLS       0B  /LEVEL/                    INTEGER
  NNODES      0B  /C/                        INTEGER
  NSYS        0B  /SYS1/                     INTEGER
  NTAPE       0B  /CNTRL/                    INTEGER
  USER        0B  /P/                        REAL
```

```
--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS-------CLASS-----

  DISPLA                 0       SUBROUTINE
  NODIN                  0       SUBROUTINE
  NUMREV                 0       SUBROUTINE
  PRENEX                 0       SUBROUTINE
  PRETOT                 0       SUBROUTINE
```

```
--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1      0B                     30
     2      6B                     50
```

```
      SUBROUTINE DISPLA
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES A TABULAR DISPLAY OF THE             *
C*   INFORMATION ASSOCIATED WITH A SELECTED NODE.                *
C*                                                               *
C*                                                               *
C*   CALLED BY:  MAN                                             *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*           USED:  IDATA,IFIND,NSYS,OBJCTV                      *
C*                                                               *
C*           MODIFIED:  DUM,I,IFINDT,II,IJ,INDEX,K,NCROSS,       *
C*                      TSYS,VAL,UT                              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INMRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/SYS1/NSYS
      CHARACTER*10 OBJCTV(16),DUM,SYSLBL,TSYS(4)
      REAL VAL(4)
      IF(IDATA.EQ.1)THEN
      PRINT*,'A DATA NODE MAY NOT BE DISPLAYED.'
      PRINT*,' '
      RETURN
      ENDIF
      PRINT*,'THE PARENT OBJECTIVE IS:'
      CALL OBJECT(IFIND,OBJCTV)
      II=1
      DO 10 I=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
10    CONTINUE
      NCROSS=1
      IFINDT=IRAY(IFIND,2)
      DO 20 I=1,10
      IF(IRAY(IFINDT,3).GT.0)THEN
      NCROSS=NCROSS+1
      IFINDT=IRAY(IFINDT,3)
      ENDIF
20    CONTINUE
      PRINT*,' '
      PRINT*,'IT HAS ',NCROSS,' SUBOBJECTIVES (FACTORS).'
      IFINDT=IRAY(IFIND,2)
      DO 40 I=1,NCROSS
      CALL OBJECT(IFINDT,OBJCTV)
      PRINT*,'FACTOR ',I,':'
      II=1
      DO 30 J=1,4
```

```
      SUBROUTINE DISPLA

115                 VAL(IJ)=100.0*VRAY(IFIND,K)
116                 IJ=IJ+1
117           71    CONTINUE
118                 WRITE(*,'(1X,4(A10,5X))')(TSYS(K),K=1,IJ-1)
119                 WRITE(*,'(1X,4(F10.2,5X))')(VAL(K),K=1,IJ-1)
120                 ENDIF
121                 PRINT*,' '
122                 CALL DSPLOT
123                 RETURN
124                 END
```

SUBROUTINE DISPLA

--STATISTICS--

PROGRAM-UNIT LENGTH	1137D	= 607
CM LABELLED COMMON LENGTH	133B	= 91
CM STORAGE USED	62600D	= 25984
COMPILE TIME	0.292 SECONDS	

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
DUM	1072B			CHAR*10	
I	1104B			INTEGER	
ICONT	0B		/NEX/	INTEGER	
IDATA	1B		/NEX/	INTEGER	
IFADD	27B		/LEVEL/	INTEGER	
IFIND	25B		/LEVEL/	INTEGER	
IFINDT	1111B			INTEGER	
II	1103B			INTEGER	
IJ	1123B			INTEGER	
INDEX	1121B			INTEGER	
INHRN	1B		/LEVEL/	INTEGER	20
ISAVD	0B		/CNTRL/	INTEGER	
ITOTL	2B		/NEX/	INTEGER	
J	1114B			INTEGER	
K	1106B			INTEGER	
LEVEL	31B		/LEVEL/	INTEGER	60
LU	30B		/LEVEL/	INTEGER	
MCROSS	1110B			INTEGER	
NDIFF	26B		/LEVEL/	INTEGER	
NLUIS	0B		/LEVEL/	INTEGER	
NSYS	9B		/SYS1/	INTEGER	
NTAPE	1B		/CNTRL/	INTEGER	
OBJCTV	1052B			CHAR*10	16
TSYS	1073B			CHAR*10	4
VAL	1077B			REAL	4
VT	1120B			REAL	

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ARAY	REAL	2	FUNCTION
DSPLOT		0	SUBROUTINE
IRAY	INTEGER	2	FUNCTION
OBJECT		2	SUBROUTINE
SYSLBL	CHAR*10	1	FUNCTION
VRAY	REAL	2	FUNCTION

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	41
20	INACTIVE	DO-TERM	49
30	INACTIVE	DO-TERM	60
40	INACTIVE	DO-TERM	63
49	INACTIVE	DO-TERM	81
50	INACTIVE	DO-TERM	85
51	INACTIVE	DO-TERM	92
60	INACTIVE	DO-TERM	97
69	INACTIVE	DO-TERM	106
70	INACTIVE	DO-TERM	110
71	INACTIVE	DO-TERM	117

--ENTRY POINTS--(LO=A)

NAME	ADDRESS	ARGS
DISPLA	5B	0

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      SUBROUTINE DSPLOT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES A PLOTTED DISPLAY OF THE             *
C*   INFORMATION ASSOCIATED WITH A SELECTED NODE (SIMILAR TO     *
C*   THE INFORMATION PROVIDED BY THE DISPLA TABULAR OUTPUT).     *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DISPLA                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  IFIND,NSYS                                    *
C*                                                               *
C*          MODIFIED:  ALFA,DSPL,DUM,I,IFINDT,IFROM,II,          *
C*                     IINDEX,INDEX,IPUT,J,NCOUNT,WT             *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER ALFA(50),DSPL(51),DUM,SYSLIL*10
      DATA ALFA/'A','B','C','D','E','F','G','H','I','J','K','L',
     * 'M','N','O','P','Q','R','S','T','U','V','W','X','Y',
     * 'Z','1','2','3','4','5','6','7','8','9','0','-','+',
     * '$','$','%','&','(',')','@','*','<','>','?','^'/
      PRINT*,' '
      PRINT*,'BRANCH    ',
     * '0--------20--------40--------60--------80--------100'
      IFROM=2
      IFINDT=IFIND
      NCOUNT=0
1     IFINDT=IRAY(IFINDT,IFROM)
      IFROM=3
      IF(IFINDT.GT.0)THEN
      NCOUNT=NCOUNT+1
      DO 10 I=2,50
10    DSPL(I)=' '
      DO 20 I=1,51,10
20    DSPL(I)='+'
      DUM='*'
      WT=1.0
      IF(IRAY(IFINDT,2).GT.0)THEN
      DUM=' '
      ENDIF
      INDEX=MIN0(NSYS,50)
      DO 30 J=1,INDEX
      IPUT=(100.0*VRAY(IFINDT,J)/WT)/2.0+1.9
      DSPL(IPUT)=ALFA(J)
```

```
      SUBROUTINE DSPLOT
 58          30   CONTINUE
 59               PRINT*,'FACTOR ',NCOUNT,DUM,DSPL
 60               GO TO 1
 61               ENDIF
 62               PRINT*,'BRANCH    ',
 63              * '0--------20--------40--------60--------80--------100'
 64               PRINT*,' '
 65               PRINT*,'LEGEND:'
 66               II=1
 67               IINDEX=IKDEX/2
 68               DO 40 J=1,IINDEX
 69               PRINT*,'SYMBOL:',ALFA(II),' IS SYSTEM: ',SYSLBL(II),
 70              * 'SYMBOL:',ALFA(II+1),' SYSTEM:',SYSLBL(II+1)
 71               II=II+2
 72          40   CONTINUE
 73               IF(2*IINDEX.LT.INDEX)THEN
 74               PRINT*,'SYMBOL:',ALFA(INDEX),' SYSTEM:',SYSLBL(INDEX)
 75               ENDIF
 76               PRINT*,' '
 77               RETURN
 78               END
```

```
                                  SUBROUTINE DSPLOT

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     1     22B                      41
    10  INACTIVE   DO-TERM          46
    20  INACTIVE   DO-TERM          48
    30  INACTIVE   DO-TERM          58
    40  INACTIVE   DO-TERM          72

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DSPLOT      5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH            403B =  259
  CM LABELLED COMMON LENGTH      162B =  114
  CM STORAGE USED              60600B = 24960
  COMPILE TIME                 0.168 SECONDS
```

```
-VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ALFA        1B  /S$A$V$E/ SAV          CHAR*1        50
 DSPL        6B  /S$A$V$E/ SAV          CHAR*1        51
 DUM        14B  /S$A$V$E/ SAV          CHAR*1
 I          20B  /S$A$V$E/ SAV          INTEGER
 ICONT       0B  /NEX/                  INTEGER
 IDATA       1B  /NEX/                  INTEGER
 IFADD      27B  /LEVEL/                INTEGER
 IFIND      25B  /LEVEL/                INTEGER
 IFINDT     16B  /S$A$V$E/ SAV          INTEGER
 IFROM      15B  /S$A$V$E/ SAV          INTEGER
 II         25B  /S$A$V$E/ SAV          INTEGER
 IINDEX     26B  /S$A$V$E/ SAV          INTEGER
 INDEX      22B  /S$A$V$E/ SAV          INTEGER
```

```
 -NAME---ADDRESS--$LOCK-----PROPERTIES-------TYPE---------SIZE

 IINRN       1B  /LEVEL/                INTEGER        20
 IPUT       24B  /S$A$V$E/ SAV          INTEGER
 ISAVD       0B  /CNTRL/                INTEGER
 ITOTL       2B  /NEX/                  INTEGER
 J          23B  /S$A$V$E/ SAV          INTEGER
 LEVEL      31B  /LEVEL/                INTEGER        60
 LVL        30B  /LEVEL/                INTEGER
 NCOUNT     17B  /S$A$V$E/ SAV          INTEGER
 NDIFF      26B  /LEVEL/                INTEGER
 NLVLS       0B  /LEVEL/                INTEGER
 NSYS        0B  /SYS1/                 INTEGER
 NTAPE       1B  /CNTRL/                INTEGER
 UT         21B  /S$A$V$E/ SAV          REAL
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS-------CLASS-----

  IRAY      INTEGER        2      FUNCTION
  MIN0      INTEGER        7      INTRINSIC
  SYSLBL    CHARX1@        1      FUNCTION
  URAY      REAL           2      FUNCTION
```

```
      SUBROUTINE NEWPG
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE PROVIDES A HEADER FOR EACH NODE IN THE         *
C*   *** NUK *** OPTION.                                         *
C*                                                               *
C*                                                               *
C*   CALLED BY:  NUMREV                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*           USED:  (NONE)                                       *
C*                                                               *
C*           MODIFIED:  (NONE)                                   *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      WRITE(*,'(S(/),1X)')
      PRINT*,'                    REVIEW'
      CALL CRAY(1)
      PRINT*,' '
      PRINT*,'NODE REFERENCE NUMBER(AND OBJECTIVE):'
      RETURN
      END
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  CRAY                      1       SUBROUTINE

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  NEWPG       5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH              692 =  4B
  CM STORAGE USED                68600B =  24960
  COMPILE TIME                  0.024 SECONDS
```

```
      SUBROUTINE NUMREV
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE CONDUCTS A NODE-BY-NODE REVIEW OF THE          *
C*   TREE.    THE *** NUM *** OPTION GIVES NUMERICAL             *
C*   INFORMATION IN ADDITION TO THE INFORMATION PROVIDED         *
C*   BY THE *** REV *** OPTION.                                  *
C*                                                               *
C*                                                               *
C*   CALLED BY:  NUM                                             *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  CMD,ICONT,IFIND,INNRN                         *
C*                                                               *
C*          MODIFIED:  CH,II,IJ,INDEX,J,K,TSYS,VAL               *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
      COMMON/ONE/CMD
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 CH*1,CMD*3,OBJCTU(16),SYSLBL,TSYS(4)
      REAL VAL(4)
1     IF(CMD.EQ.'REV')THEN
      WRITE(*,'(5(/),1X)')
      CALL CRAY(1)
      ELSE
      CALL NEUPG
      ENDIF
      WRITE(*,'(20I3)')(INNRN(I),I=1,LUL)
      CALL OBJECT(IFIND,OBJCTU)
      II=1
      DO 10 J=1,4
      WRITE(*,'(1X,4A10)')(OBJCTU(K),K=II,II+3)
      II=II+4
10    CONTINUE
      IF(CMD.EQ.'NUM')THEN
      PRINT*,'RELATIVE WEIGHT: ',ARRAY(IFIND,1)
      PRINT*,'CUMULATIVE WEIGHT: ',ARRAY(IFIND,2)
      PRINT*,' '
      PRINT*,'SYSTEM VALUES:'
      II=1
      INDEX=NSYS/4
      DO 20 J=1,INDEX
      IJ=1
      DO 19 K=II,II+3
      TSYS(IJ)=SYSLBL(K)
      VAL(IJ)=100.0*VARAY(IFIND,K)
      IJ=IJ+1
19    CONTINUE
```

```
SUBROUTINE NUMREV

58                WRITE(*,'(1X,4(A10,5X))')(TSYS(K),K=1,4)
59                WRITE(*,'(1X,4(F10.2,5X))')(VAL(K),K=1,4)
60                II=II+4
61          20    CONTINUE
62                IF(4*INDEX.LT.NSYS)THEN
63                IJ=1
64                DO 21 K=INDEX*4+1,NSYS
65                TSYS(IJ)=SYSLBL(K)
66                VAL(IJ)=100.0*VRAY(IFIND,K)
67                IJ=IJ+1
68          21    CONTINUE
69                WRITE(*, (1X,4(A10,5X))')(TSYS(K),K=1,IJ-1)
70                WRITE(*,'(1X,4(F10.2,5X))')(VAL(K),K=1,IJ-1)
71                ENDIF
72                PRINT*,' '
73                CALL CRAY(IFIND)
74                ELSE
75                PRINT*,' '
76                ENDIF
77                CALL NEXT
78                PRINT*,'(PRESS ANY LETTER TO CONTINUE (EXCEPT 'E'))'
79                PRINT*,'(PRESS 'E' TO EXIT)'
80                PRINT*,'      ?'
81                READ(5,'(A1)')CH
82                IF((CH.NE.'E').AND.(ICONT.NE.0))GO TO 1
83                RETURN
84                END
```

SUBROUTINE NUMREV

--STATEMENT LABELS--(LO-A)

LABEL	ADDRESS	PROPERTIES		DEF
1	7B			31
10	INACTIVE	DO-TERM		43
19	INACTIVE	DO-TERM		57
20	INACTIVE	DO-TERM		61
21	INACTIVE	DO-TERM		63

--ENTRY POINTS--(LO-A)

NAME	ADDRESS	ARGS
NUMREV	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	555B = 365
CM LABELLED COMMON LENGTH	170B = 120
CM STORAGE USED	60600B = 24960
COMPILE TIME	0.178 SECONDS

--VARIABLE MAP--(LO-A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CH	1B	/SAUSE/	SAV	CHAR*1	
CMD	0B	/ONE/		CHAR*3	
I	32B	/SAUSE/	SAV	INTEGER	
ICONT	40B	/SAUSE/	UND/SAV	INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
II	33B	/SAUSE/	SAV	INTEGER	
IJ	37B	/SAUSE/	SAV	INTEGER	
INDEX	36B	/SAUSE/	SAV	INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20
J	34B	/SAUSE/	SAV	INTEGER	
K	35B	/SAUSE/	SAV	INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
NSYS	0B	/SYS1/		INTEGER	
OBJCTV	2B	/SAUSE/	SAV	CHAR*10	16
TSYS	22B	/SAUSE/	SAV	CHAR*10	4
VAL	26B	/SAUSE/	SAV	REAL	4

--PROCEDURES--(LO-A)

NAME	TYPE	ARGS	CLASS
ARAY	REAL	2	FUNCTION
CRAY		1	SUBROUTINE
NEWPG		0	SUBROUTINE
NEXT		0	SUBROUTINE

NAME	TYPE	ARGS	CLASS
OBJECT		2	SUBROUTINE
SYSLBL	CHAR*10	1	FUNCTION
VRAY	REAL	2	FUNCTION

```
      OVERLAY(XFILE,4,0)
      PROGRAM SENSTU
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS PROGRAM IS THE ENTRY POINT FOR OVERLAY(4,0).           *
C*   IT DIRECTS THE VARIOUS SENSITIVITY ANALYSIS OPTIONS.        *
C*                                                               *
C*                                                               *
C*   CALLED BY:  (NONE)                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  CH,CH1,SENS                               *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/C/NNODES,NDEEP
      COMMON/CNTRL/ISAVD,NTAPE
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/ONE/CKD
      COMMON/P/USER
      COMMON/RRAY1/ARRAY(90)
      COMMON/RRAY2/ISIN,NLOUD
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*8 SENS*5,SYSNAM,CH*1,CH1*1,ATTNAM
1     WRITE(*,'(5(/),1X)')
      PRINT*,'                    SENSITIVITY ANALYSIS'
      PRINT*,' '
      PRINT*,'CHOOSE YOUR OPTION:'
      PRINT*,'   C)UMUT     R)ELWT     L)EVEL     S)YSTEM     E)XIT'
      PRINT*,'     ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'C').AND.(CH.NE.'R').AND.(CH.NE.'L').AND.
     * (CH.NE.'E').AND.(CH.NE.'S'))THEN
      PRINT*,'PLEASE ENTER "C","R","L", "S", OR "E" '
      GO TO 1
      ENDIF
      CALL PRETOT
      IF(CH.NE.'E')THEN
      IF(CH.EQ.'C')THEN
      SENS='CUMUT'
      ELSEIF(CH.EQ.'R')THEN
      SENS='RELWT'
      ELSEIF(CH.EQ.'S')THEN
      SENS='SYSTM'
      ELSE
```

```
PROGRAM $SENSTU

      58                SENS='LEVEL'
      59                ENDIF
      60                IF(SENS.NE.'SYSTM')THEN
      61                CALL DETNOD
      62                ELSE
      63                CALL SYSSEN
      64                ENDIF
      65                CALL COMPUT
      66          2     WRITE(*,'(5(/),1X)')
      67                PRINT*,SENS,' ANALYSIS:   T)ABULAR   G)RAPHICAL   E)XIT'
      68                PRINT*,'      ?'
      69                READ(*,'(A1)')CH1
      70                IF((CH1.NE.'T').AND.(CH1.NE.'G').AND.(CH1.NE.'E'))THEN
      71                PRINT*,'ENTER 'T','G', OR 'E' '
      72                GO TO 2
      73                ENDIF
      74                IF(CH1.EQ.'T')THEN
      75                CALL TABDIS
      76                ELSEIF(CH1.EQ.'G')THEN
      77                CALL GRAPH
      78                ENDIF
      79                IF(CH1.NE.'E')GO TO 2
      80                ENDIF
      81                RETURN
ANSI      *         RETURN IN MAIN PROGRAM -- ACTS AS END
      82                END
```

```
                                                    PROGRAM SENSTU

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

      1      0B                    37
      2      0B                    66

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  SENSTU      0B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH                0B =  0
  CM LABELLED COMMON LENGTH        270B =  184
  CM STORAGE USED                60600B =  24960
  COMPILE TIME                   0.114 SECONDS

     1  ANSI      ERROR IN  SENSTU
```

```
--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ARRAY       0B  /RRAY1/                    REAL           80
  ATTNAM      0B  /SEN1/                     CHAR*10
  CH          0B  /S$ASUSE/ SAV              CHAR*1
  CH1         0B  /S$ASUSE/ SAV              CHAR*1
  CMD         0B  /ONE/                      REAL
  ICONT       0B  /NEX/                      INTEGER
  IDATA       0B  /NEX/                      INTEGER
  IFADD       0B  /LEVEL/                    INTEGER
  IFIND       0B  /LEVEL/                    INTEGER
  IHNRN       0B  /LEVEL/                    INTEGER        20
  ISAVD       0B  /CNTRL/                    INTEGER
  ISIN        0B  /RRAY2/                    INTEGER
  ITOTL       0B  /NEX/                      INTEGER

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  LEVEL       0B  /LEVEL/                    INTEGER        60
  LVL         0B  /LEVEL/                    INTEGER
  NATT        0B  /ATTR/                     INTEGER
  NDEEP       0B  /C/                        INTEGER
  NDIFF       0B  /LEVEL/                    INTEGER
  NLOUD       0B  /RRAY2/                    INTEGER
  NLVLS       0B  /LEVEL/                    INTEGER
  NNODES      0B  /C/                        INTEGER
  NSYS        0B  /SYS1/                     INTEGER
  NTAPE       0B  /CNTRL/                    INTEGER
  SENS        0B  /SEN1/                     CHAR*5
  SYSNAM      0B  /SEN1/                     CHAR*10
  USER        0B  /P/                        REAL
```

```
--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----

  COMPUT                  0       SUBROUTINE
  DETNOD                  0       SUBROUTINE
  GRAPH                   0       SUBROUTINE

 -NAME------TYPE--------ARGS------CLASS-----

  PRETOT                  0       SUBROUTINE
  SYSSEN                  0       SUBROUTINE
  TABDIS                  0       SUBROUTINE
```

```
      SUBROUTINE COMPUT
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE COMPUTES THE MATRIX OF OVERALL VALUES          *
C*   ASSOCIATED WITH EACH SENSITIVITY ANALYSIS.                  *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SENSTV                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  FINDST,INDXST,ISYS,KIDSET,MTYPE,NSET,NSYS,    *
C*                 SENS                                          *
C*                                                               *
C*          MODIFIED:  ATTNAM,CONTRL,I,IFIND,IFINDT,IM,INDX,J,   *
C*                     K,KKK,NCOUNT,NCROSS,SUM,TEMP,TEST,WDLTA,  *
C*                     WHOLD,X                                   *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLVLS,INMRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN2/WMIN,WMAX,NCROSS,IMDX,ISYS
      COMMON/SEN3/WHOLD(0:20,0:80)
      COMMON/SEN4/MTYPE
      COMMON/SEN5/FINDST(50),NSET,INDXST(50),KIDSET(50)
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*5 SENS,MTYPE,TEST,LABEL*10,SYSNAM*10,ATTNAM*10
      INTEGER CONTRL,FINDST,KIDSET,INDXST
      WDLTA=(WMAX-WMIN)/20.0
      DO 5 I=0,20
      DO 5 J=0,NSYS
5     WHOLD(I,J)=0.0
      IF(SENS.NE.'SYSTR')THEN
      TEST=SENS
      ELSE
      TEST=MTYPE
      ENDIF
      IF(TEST.EQ.'CURVT')THEN
      CONTRL=1
      ELSEIF(TEST.EQ.'LEVEL')THEN
      CONTRL=2
      ELSEIF(TEST.EQ.'VALUE')THEN
      CONTRL=2
      ELSE
      CONTRL=3
      ENDIF
      NCOUNT=1
      IF(SENS.EQ.'SYSTR')THEN
      IFIND=FINDST(NCOUNT)
      NCROSS=KIDSET(NCOUNT)
```

SUBROUTINE COMPUT

```
      INDX=INDXST(NCOUNT)
      ENDIF
      IF(CONTRL.EQ.3)THEN
      SUM=0.0
      IFINDT=IRAY(IFIND,4)
      DO 7 I=1,10
      IF(IRAY(IFINDT,2).EQ.IFINDT+1)THEN
      GO TO 3
      ENDIF
      IFINDT=IRAY(IFINDT,4)
7     CONTINUE
3     IFINDT=IRAY(IFINDT,2)
      DO 10 I=1,NCROSS
      IF(INDX.NE.IRAY(IFINDT,1))SUM=SUM+ARAY(IFINDT,1)
      IFINDT=IRAY(IFINDT,3)
10    CONTINUE
      ENDIF
      DO 40 I=0,20
      X=FLOAT(I)
      X=X*WDLTA+WMIN
      WHOLD(I,0)=X
      IF(SENS.EQ.'SYSTM')THEN
11    K=NCOUNT
      IFIND=FINDST(NCOUNT)
      NCROSS=KIDSET(NCOUNT)
      INDX=INDXST(NCOUNT)
      IF(CONTRL.EQ.3)THEN
      SUM=0.0
      IFINDT=IRAY(IFIND,4)
      DO 66 II=1,10
      IF(IRAY(IFINDT,2).EQ.IFINDT+1)THEN
      GO TO 2
      ENDIF
      IFINDT=IRAY(IFINDT,4)
66     CONTINUE
2     IFINDT=IRAY(IFINDT,2)
      DO 12 II=1,NCROSS
      IF(INDX.NE.IRAY(IFINDT,1))SUM=SUM+ARAY(IFINDT,1)
      IFINDT=IRAY(IFINDT,3)
12    CONTINUE
      ENDIF
      IF(CONTRL.EQ.1)THEN
      WHOLD(I,K)=(100.0*VRAY(IFIND,ISYS)*X)+((1.0-X)*((100.0
     * *VRAY(2,ISYS)
     * -(100.0*VRAY(IFIND,ISYS)*ARAY(IFIND,2)))/(1.0-ARAY(IFIND,2))))
      ELSEIF(CONTRL.EQ.2)THEN
      ATTNAM=LABEL(IFIND)
      TEMP=X/100.0
      WHOLD(I,K)=100.0*VRAY(2,ISYS)+ARAY(IFIND,2)
     * *(100.0*VALU(TEMP,ATTNAM)-100.0*VRAY(IFIND,ISYS))
      ELSE
      IFINDT=IRAY(IFIND,4)
      DO 9 IM=1,10
      IF(IRAY(IFINDT,2).EQ.IFINDT+1)THEN
      GO TO 6
      ENDIF
      IFINDT=IRAY(IFINDT,4)
```

SUBROUTINE COMPUT

```
9       CONTINUE
6       UHOLD(I,K)=100.0*URAY(2,ISYS)-100.0*URAY(IFINDT,ISYS)
      * *ARAY(IFINDT,2)
        DO 60 L=1,NCROSS
        IF(INDX.NE.L)THEN
        UHOLD(I,K)=UHOLD(I,K)+(1.0-X)*100.0*URAY(IFINDT+L,ISYS)*
      *  ARAY(IFINDT+L,1)/SUM*ARAY(IFINDT,2)
        ELSE
        UHOLD(I,K)=UHOLD(I,K)+X*100.0*URAY(IFINDT+L,ISYS)*ARAY(IFINDT,2)
        ENDIF
60      CONTINUE
        ENDIF
        NCOUNT=NCOUNT+1
        IF(NCOUNT.LE.NSET)GO TO 11
        ELSE
        DO 30 K=1,NSYS
        IF(CONTRL.EQ.1)THEN
        UHOLD(I,K)=(100.0*URAY(IFIND,K)*X)+((1.0-X)*((100.0*URAY(2,K)
      * -(100.0*URAY(IFIND,K)*ARAY(IFIND,2)))/(1.0-ARAY(IFIND,2))))
        ELSEIF(CONTRL.EQ.2)THEN
        IF(K.EQ.ISYS)THEN
        UHOLD(I,K)=100.0*URAY(2,K)+ARAY(IFIND,2)
      * *(100.0*VALU(X,ATTNAM)-100.0*URAY(IFIND,K))
        ELSE
        UHOLD(I,K)=100.0*URAY(2,K)
        ENDIF
        ELSE
        IFINDT=IRAY(IFIND,4)
        DO 8 JM=1,10
        IF(IRAY(IFINDT,2).EQ.IFINDT+1)THEN
        GO TO 4
        ENDIF
        IFINDT=IRAY(IFINDT,4)
8       CONTINUE
4       UHOLD(I,K)=100.0*URAY(2,K)-100.0*URAY(IFINDT,K)*ARAY(IFINDT,2)
        DO 20 L=1,NCROSS
        IF(INDX.NE.L)THEN
        UHOLD(I,K)=UHOLD(I,K)+(1.0-X)*100.0*URAY(IFINDT+L,K)*
      *  ARAY(IFINDT+L,1)/SUM*ARAY(IFINDT,2)
        ELSE
        UHOLD(I,K)=UHOLD(I,K)+X*100.0*URAY(IFINDT+L,K)*ARAY(IFINDT,2)
        ENDIF
20      CONTINUE
        ENDIF
30      CONTINUE
        ENDIF
        NCOUNT=1
40      CONTINUE
        PRINT*,' '
        RETURN
        END
```

```
        SUBROUTINE COMPUT

--VARIABLE MAP--(LO-A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ATTNAM      0B   SEN1/                      CHAR*10                 LVL        30B  /LEVEL/                  INTEGER
  CONTRL      2B  /S$ARUSE/ SAV               INTEGER                 MTYPE       0B  /SEN4/                   CHAR*5
  FINDST      9B  /SEN5/                      INTEGER        50       NCOUNT      5B  /S$ABUSE/ SAV            INTEGER
  I           4B  /S$ABUSE/ SAV               INTEGER                 NCROSS      2B  /SEN2/                   INTEGER
  IFADD      27B  /LEVEL/                     INTEGER                 NDIFF      26B  /LEVEL/                  INTEGER
  IFIND      25B  /LEVEL/                     INTEGER                 NLVLS      32B  /LEVEL/                  INTEGER
  IFINDT     70I  /S$ABUSE/ SAV               INTEGER                 NSET       62B  /SEN5/                   INTEGER
  II         13B  /S$ABUSE/ SAV               INTEGER                 NSYS        0B  /SYS1/                   INTEGER
  IM         15B  /S$ABUSE/ SAV               INTEGER                 SENS        1B  /SEN1/                   CHAR*5
  INDX        3B  /SEN2/                      INTEGER                 SUM         7B  /S$ABUSE/ SAV            REAL
  INDXST     63B  /SEN5/                      INTEGER        50       SYSNAM      1B  /SEN1/                   CHAR*10
  INNRN       1B  /LEVEL/                     INTEGER        20       TEMP       14B  /S$ABUSE/ SAV            REAL
  ISYS        4B  /SEN2/                      INTEGER                 TEST        1B  /S$ABUSE/ SAV            CHAR*5
  J           5B  /S$ABUSE/ SAV               INTEGER                 UDELTA      3B  /S$ABUSE/ SAV            REAL
  *          12B  /S$ABUSE/ SAV               INTEGER                 UHOLD       0B  /SEN3/                   REAL          1701
  KIDSLT     145B  /SEN5/                     INTEGER        50       UMAX        1B  /SEN2/                   REAL
  L          16B  /S$ABUSE/ SAV               INTEGER                 UMIN        0B  /SEN2/                   REAL
  LEVEL      31B  /LEVEL/                     INTEGER        60       X         11B  /S$ABUSE/ SAV             REAL

--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  ARRY      REAL          2      FUNCTION         LABEL     CHAR*10       1      FUNCTION
  FLOAT     REAL          1      INTRINSIC        VALU      REAL          2      FUNCTION
  IRAY      INTEGER       2      FUNCTION         VRAY      REAL          2      FUNCTION

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

     2    254B                      93          8  INACTIVE   DO-TERM      148          29  INACTIVE   DO-TERM      157
     3    144B                      69          9  INACTIVE   DO-TERM      115          30  INACTIVE   DO-TERM      159
     4    671B                     149         10  INACTIVE   DO-TERM       73          40  INACTIVE   DO-TERM      162
     5  INACTIVE   DO-TERM          39         11    212B                  80          60  INACTIVE   DO-TERM      125
     6    423B                     116         12  INACTIVE   DO-TERM       97          66  INACTIVE   DO-TERM       92
     7  INACTIVE   DO-TERM          68

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  COMPUT      5B      0

--STATISTICS--

  PROGRAM-UNIT LENGTH            12173 =  655
  CM LABELLED COMMON LENGTH      36523 =  1962
  CM STORAGE USED               626608 =  25984
  COMPILE TIME                 0.382 SECONDS
```

```
      SUBROUTINE DETNOD
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ELICITS THAT NODE AT WHICH A DIRECT            *
C*   LSA, CSA, OR RSA IS TO BE PERFORMED.  IT ALSO               *
C*   INITIALIZES THOSE VARIABLES REQUIRED BY COMPUT.             *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SENSTU                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  IDATA,ICONT,IFIND,NATT,OBJCTU,SENS            *
C*                                                               *
C*          MODIFIED:  ATTNAM,CH,I,IFINDT,II,INDEX,K,            *
C*                     NCROSS,UMAX,UMAX1,UMIN,UMIN1              *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/ATTR/NATT
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN2/UMIN,UMAX,NCROSS,INDX,ISYS
      SAVE
      CHARACTER*10 SENS*5,CH*1,OBJCTU(16),ATT1,LABEL,SYSNAM,ATTNAM
      INDX=0
1     WRITE(*,'(5(/),1X)')
      PRINT*,'SENSITIVITY ANALYSIS FOLLOWS...'
      PRINT*,'ENTER NRN FOR WHICH ',SENS,' IS'
      PRINT*,'TO BE PERTURBED...'
      CALL PREMFX
      IF(ICONT.EQ.0)THEN
      PRINT*,'NODE DOES NOT EXIST.'
      ELSE
      IF((SENS.EQ.'LEVEL').AND.(IDATA.NE.1))THEN
      PRINT*,'A LEVEL ANALYSIS MAY ONLY BE PERFORMED
      PRINT*,'ON A DATA NODE.'
      ICONT=0
      ENDIF
      IF(ICONT.EQ.0)THEN
      PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
      PRINT*,'     ?'
      READ(*,'(A1)')CH
      ENDIF
      ENDIF
      IF(ICONT.NE.1)GO TO 1
      PRINT*,' '
      CALL OBJECT(IFIND,OBJCTU)
      II=1
      DO 10 I=1,4
      WRITE(*,'(1X,4A10)')(OBJCTU(K),K=II,II+3)
```

SUBROUTINE DETNOD

```
      II=II+4
10    CONTINUE
      IF(SENS.EQ.'CUMUT')THEN
      PRINT*,' '
      PRINT*,'CURRENT CUMLT IS ',ARAY(IFIND,2)
      ELSEIF(SENS.EQ.'LEVEL')THEN
      ATTNAM=LABEL(IFIND)
      CALL SENVAL
      ELSE
      PRINT*,' '
      PRINT*,'CURRENT RELUT IS ',ARAY(IFIND,1)
      ENDIF
      INDX=IRAY(IFIND,1)
      NCROSS=0
      IFINDT=IRAY(IFIND,4)
      DO 5 I=1,10
      IF(IRAY(IFINDT,2).EQ.IFINDT+1)THEN
      GO TO 6
      ENDIF
      IFINDT=IRAY(IFINDT,4)
5     CONTINUE
6     IFIND=IRAY(IFINDT,2)
      DO 20 I=1,10
      IF(IFINDT.GT.0)THEN
      NCROSS=NCROSS+1
      IFINDT=IRAY(IFINDT,3)
      ELSE
      GO TO 4
      ENDIF
20    CONTINUE
4     IF(SENS.EQ.'LEVEL')THEN
      DO 30 I=1,NATT
      IF(ATTI(I).EQ.LABEL(IFIND))THEN
      UMAX=AMAX1(ATT(I,1),ATT(I,2))
      UMIN=AMIN1(ATT(I,1),ATT(I,2))
      PRINT*,'WE ARE WORKING WITH ATTRIBUTE: ',ATTNAM
      ENDIF
30    CONTINUE
      ELSE
      UMAX=1.0
      UMIN=0.0
      ENDIF
2     PRINT*,' '
      PRINT*,'MINIMUM ',SENS,' (',UMIN,'-',UMAX,') IS?'
      READ(*,'(F5.2)')UMIN1
      IF((UMIN1.LT.UMIN).OR.(((UMIN1.GT.1.0).AND.(SENS.NE.'LEVEL'))
     * .OR.((UMIN1.GT.UMAX).AND.(SENS.EQ.'LEVEL'))))GO TO 2
      UMIN=UMIN1
3     PRINT*,' '
      PRINT*,'MAXIMUM ',SENS,' (',UMIN,'-',UMAX,') IS?'
      READ(*,'(F5.2)')UMAX1
      IF((UMAX1.LT.UMIN).OR.(((UMAX1.GT.1.0).AND.(SENS.NE.'LEVEL'))
     * .OR.((UMAX1.GT.UMAX).AND.(SENS.EQ.'LEVEL'))))GO TO 3
      UMAX=UMAX1
      RETURN
      END
```

```
        SUBROUTINE DETMOD

--VARIABLE MAP--(LO-HI)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ATTNAM      0B  /SEN1/                    CHAR*10                 LEVEL     51B  /LEVEL/                    INTEGER        60
  CH          1F  /S$4$USE/ SAV             CHAR*1                  LVL      10B  /LEVEL/                    INTEGER
  I         256   /S$ABUSE/ SAV             INTEGER                 NATT      0B  /ATTR/                     INTEGER
  IDATA       1B  /NEX/                     INTEGER                 NCROSS    3B  /SEN-/                     INTEGER
  IFADD     278   /LEVEL/                   INTEGER                 NDIFF     3B  /LEVEL/                    INTEGER
  IFINT     258   /LEVEL/                   INTEGER                 NLULS     6B  /LEVEL/                    INTEGER
  IFINIT    263   /S$ABUSE/ SAV             INTEGER                 OBJCTV    2B  /S$ABUSE/ SAV              CHAR*10        16
  II        223   /S$ABUSE/ SAV             INTEGER                 SENS      1B  /SEN1/                     CHAR*5
  INDX        3B  /SEN2/                    INTEGER                 SYSNAM    1B  /SEN1/                     CHAR*10
  INMRN       1B  /LEVEL/                   INTEGER        20       UMAX      1B  /SEN2/                     REAL
  ISYS        4B  /SEN2/                    INTEGER                 UMAX1    27B  /S$ABUSE/ SAV              REAL
  ITOTL       2B  /NEX/                     INTEGER                 UMIN      0B  /SEN2/                     REAL
  K         24B   /S$ABUSE/ SA              INTEGER                 UMIN1    2CB  /S$ABUSE/ SAV              REAL

--PROCEDURES--(LO-A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  AMAX1     REAL          7     INTRINSIC        IRAY     INTEGER        2     FUNCTION
  AMIN1     REAL          7     INTRINSIC        LABEL    CHAR*10        1     FUNCTION
  ARAY      REAL          2     FUNCTION         OBJECT                  2     SUBROUTINE
  ATT       REAL          2     FUNCTION         PRENEX                  0     SUBROUTINE
  ATTI      CHAR*10       1     FUNCTION         SENVAL                  0     SUBROUTINE

--STATEMENT LABELS--(LO-A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

     1     11B                     33          4    245B                    67          10  INACTIVE   DO-TERM       59
     2    321B                    100          5  INACTIVE   DO-TERM         72          20  INACTIVE   DO-TERM       87
     3    351B                    106          6    216B                    91          30  INACTIVE   DO-TERM       95

--ENTRY POINTS--(LO-A)
 -NAME---ADDRESS--ARGS---

  DETMOD      5B     0

--STATISTICS--

  PROGRAM-UNIT LENGTH            7105 =  456
  CM LABELLED COMMON LENGTH      1719 =  121
  CM STORAGE USED              626003 =  25984
  COMPILE TIME                 0.231 SECONDS
```

```
      SUBROUTINE GHDR(I,CH)
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES THE X-AXIS LABELS FOR THE            *
C*   GRAPHICAL DISPLAY OF SENSITIVITY RESULTS.                   *
C*                                                               *
C*                                                               *
C*   CALLED BY:  GRAPH                                           *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  CH,I,NSET,NSYS,SENS,WHOLD                     *
C*                                                               *
C*          MODIFIED:  IDLTA,IDUM,II,K,MAX,MIN,MM,WMAX,WMIN      *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN2/WMIN,WMAX,NCROSS,IHDX,ISYS
      COMMON/SEN3/WHOLD(20,80)
      COMMON/SEN5/FINDST(50),NSET,INDXST(50),KIDSET(50)
      COMMON/SYS1/NSYS
      INTEGER IDUM(0:5)
      CHARACTER CH,ATTNAM*10,SENS*5,SYSNAM*10
      WMIN=100.0
      WMAX=0.0
      IF(CH.EQ.'E')THEN
      IF(SENS.NE.'SYSTM')THEN
      MM=NSYS
      ELSE
      MM=NSET
      ENDIF
      DO 10 II=1,20
      DO 10 K=1,MM
      IF(WMIN.GT.WHOLD(II,K))WMIN=WHOLD(II,K)
      IF(WMAX.LT.WHOLD(II,K))WMAX=WHOLD(II,K)
10    CONTINUE
      ELSE
      WMIN=0.0
      WMAX=100.0
      ENDIF
      MIN=INT(WMIN/20.0)*20
      MAX=INT((WMAX+19.0)/20.0)*20
      IDLTA=(MAX-MIN)/5
      IF(I.EQ.1)THEN
      PRINT*,' '
      PRINT*,'                                                    OVERALL VALUE'
      DO 20 II=0,5
20    IDUM(II)=MIN+IDLTA*II
      WRITE(*,'(2X,6(5X,I5))')(IDUM(II),II=0,5)
      ELSE
      DO 30 II=0,5
```

```
      SUBROUTINE CHDR

 58          30     IDUM(II)=MIN+IDLTA*II
 59                 WRITE(*,'(2X,G(5X,I5))')(IDUM(II),II=0,5)
 60                 PRINT*,'                                    OVERALL VALUE'
 61                 ENDIF
 62                 RETURN
 63                 END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ATTNAM      0B  /SEN1/                    CHAR*10
  CH           2  DUMMY-ARG                 CHAR*1
  FINDST      0B  /SEN5/                    REAL          50
  I            1  DUMMY-ARG                 INTEGER
  IDLTA     274B                            INTEGER
  IDUM      257B                            INTEGER        6
  II        266B                            INTEGER
  INDX        3B  /SEN2/                    INTEGER
  INDXST     63B  /SEN5/                    INTEGER       50
  ISYS        4B  /SEN2/                    INTEGER
  K         270B                            INTEGER
  KIDSET    140B  /SEN5/                    INTEGER       50

 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  MAX       273B                            INTEGER
  MIN       272B                            INTEGER
  NN        265B                            INTEGER
  NCROSS      2B  /SEN2/                    INTEGER
  NSET       62B  /SEN5/                    INTEGER
  NSYS        0B  /SYS1/                    INTEGER
  SEN5        1B  /SEN1/                    CHAR*5
  SYSNAM      1B  /SEN1/                    CHAR*10
  WHOLD       0B  /SEN3/                    REAL          1701
  WMAX        1B  /SEN2/                    REAL
  WMIN        0B  /SEN2/                    REAL
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  INT       GENERIC        )       INTRINSIC
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----REF

     10  INACTIVE   DO-TERM        42
     20  INACTIVE   DO-TERM        54
     30  INACTIVE   DO-TERM        58
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  CHDR        5B     2
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH              301B =  193
  CM LABELLED COMMON LENGTH       3505B =  1861
  CM STORAGE USED                 536048 =  24960
  COMPILE TIME                    0.109 SECONDS
```

```
      SUBROUTINE GRAPH
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES A GRAPHICAL DISPLAY OF THE           *
C*   SENSITIVITY RESULTS.                                        *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SENSTV                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ATTNAM,FINDST,ISYS,NSET,OBJCTV,SENS,          *
C*                 SYSNAM                                        *
C*                                                               *
C*          MODIFIED:  CH,I,II,IN,INDEX,IPUT,J,JJ,JN,K,          *
C*                     LIMIT,LINE,M,SYMBOL,XVAL                  *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN2/WMIN,WMAX,NCROSS,INDX,ISYS
      COMMON/SEN3/WHOLD(0:20,0:50)
      COMMON/SEN5/FINDST(50),NSET,INDXST(50),KIDSET(50)
      COMMON/SYS1/NSYS
      SAVE
      INTEGER FINDST,KIDSET,SYSNUM
      CHARACTER CH,SYMBOL(50),SYSLBL*10,LINE(51),ATTNAM*10,SENS*15,
     * SYSNAM*10,OBJCTV(16)*10
      DATA SYMBOL/'A','B','C','D','E','F','G','H','I','J','K','L',
     * 'M','N','O','P','Q','R','S','T','U','V','W','X','Y',
     * 'Z','1','2','3','4','5','6','7','8','9','0','+','*',
     * '$','%','&','(',')','@','#','<','>','?','^'/
1     WRITE(*,'(5(/),1X)')
      PRINT*,'GRAPHICAL ANALYSIS:     N(ORMAL     E(XPANDED'
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'N').AND.(CH.NE.'E'))THEN
      PRINT*,' ENTER "N" OR "E" '
      GO TO 1
      ENDIF
      IF(SENS.NE.'SYSTM')THEN
      LIMIT=MIN0(50,NSYS)
      ELSE
      LIMIT=NSET
      ENDIF
      II=1
      INDEX=LIMIT/2
      PRINT*,'LEGEND:'
      DO 10 I=1,INDEX
      IF(SENS.NE.'SYSTM')THEN
      PRINT*,'SYMBOL=',SYMBOL(II),' IS SYSTEM:',SYSLBL(II),
     * '      SYMBOL=',SYMBOL(II+1),' IS SYSTEM:',SYSLBL(II+1)
```

SUBROUTINE GRAPH

```
                ELSE
                DO 5 JJ=0,1
                PRINT*,'SYMBOL=',SYMBOL(II+JJ),' IS NODE:'
                IN=1
                CALL OBJECT(FINDST(II+JJ),OBJCTV)
                DO 6 JN=1,4
                WRITE(*,'(1X,4A10)')(OBJCTV(M),M=IN,IN+3)
                IN=IN+4
        6       CONTINUE
                PRINT*,' '
        5       CONTINUE
                ENDIF
                II=II+2
       10       CONTINUE
                IF(SENS.NE.'SYSTM')THEN
                IF(INDEX*2.LT.LIMIT)PRINT*,'SYMBOL=',SYMBOL(LIMIT),
               * ' IS SYSTEM:',SYSLBL(LIMIT)
                ELSE
                IF(INDEX*2.LT.LIMIT)THEN
                IN=1
                CALL OBJECT(FINDST(LIMIT),OBJCTV)
                PRINT*,'SYMBOL=',SYMBOL(LIMIT),' IS NODE:'
                DO 7 JN=1,4
                WRITE(*,'(1X,4A10)')(OBJCTV(M),M=IN,IN+3)
                IN=IN+4
        7       CONTINUE
                ENDIF
                ENDIF
                IF(SENS.NE.'LEVEL')THEN
                PRINT*,'SENSITIVITY ANALYSIS ON ',SENS
                IF(SENS.EQ.'SYSTM')PRINT*,'FOR SYSTEM ',SYSLBL(ISYS)
                ELSE
                PRINT*,'LEVEL ANALYSIS ON SYSTEM ',SYSNAM
                PRINT*,'FOR ATTRIBUTE ',ATTNAM
                ENDIF
                CALL GHDR(1,CH)
                DO 20 I=1,20
                DO 30 J=2,50
                IF((I.EQ.1).OR.(I.EQ.5).OR.(I.EQ.10).OR.(I.EQ.15).OR.
               * (I.EQ.20))THEN
                LINE(J)='+'
                ELSE
                LINE(J)=' '
                ENDIF
       30       CONTINUE
                DO 40 J=1,51,10
       40       LINE(J)='+'
                DO 50 J=1,LIMIT
                XVAL=(UHOLD(I,J)-UMIN)/((UMAX-UMIN)/50.0)
                IPUT=INT(XVAL+.75*(UMAX-UMIN)/50.0)
                IF(IPUT.LE.0)IPUT=1
                IF(IPUT.GT.51)IPUT=51
                LINE(IPUT)=SYMBOL(J)
       50       CONTINUE
                IF((I.EQ.1).OR.(I.EQ.5).OR.(I.EQ.10).OR.(I.EQ.15).OR.
               * (I.EQ.20))THEN
                WRITE(*,'(1X,F10.2,51A1)')UHOLD(I,0),(LINE(K),K=1,51)
```

SUBROUTINE GRAPH

```
115                 ELSE
116                 PRINT*,'                    ',LINE
117                 ENDIF
118          20     CONTINUE
119                 CALL CHDR(2,CH)
120                 PRINT*,' '
121                 PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
122                 PRINT*,'  ?'
123                 READ(*,'(A1)')CH
124                 RETURN
125                 END
```

--VARIABLE MAP--(LO*A)

-NAME---	-ADDRESS--	-BLOCK-----	-PROPERTIES-------	-TYPE---------	-SIZE
ATTNAM	0B	/SEN1/		CHAR*10	
CH	2B	/SAV$E/	SAV	CHAR*1	
FINDST	0B	/SEN5/		INTEGER	50
I	41B	/SAV$E/	SAV	INTEGER	
II	37B	/SAV$E/	SAV	INTEGER	
IN	43B	/SAV$E/	SAV	INTEGER	
INDEX	40B	/SAV$E/	SAV	INTEGER	
INDX	3B	/SEN2/		INTEGER	
INDXST	63B	/SEN5/		INTEGER	50
IPUT	50B	/SAV$E/	SAV	INTEGER	
ISYS	4B	/SEN2/		INTEGER	
J	46B	/SAV$E/	SAV	INTEGER	
JJ	42B	/SAV$E/	SAV	INTEGER	
JN	44B	/SAV$E/	SAV	INTEGER	
K	51B	/SAV$E/	SAV	INTEGER	
KIDSET	145B	/SEN5/		INTEGER	50

--PROCEDURES--(LO*A)

-NAME------	-TYPE--------	-ARGS------	-CLASS-----
CHDR		2	SUBROUTINE
INT	GENERIC	1	INTRINSIC
MIN0	INTEGER	7	INTRINSIC
OBJECT		2	SUBROUTINE
SYSLBL	CHAR*10	1	FUNCTION

--STATEMENT LABELS--(LO*A)

-LABEL-	-ADDRESS-----	-PROPERTIES----	-DEF
1	10B		3B
5	INACTIVE	DO-TERM	6B
6	INACTIVE	DO-TERM	56
7	INACTIVE	DO-TERM	83
10	INACTIVE	DO-TERM	71
20	INACTIVE	DO-TERM	11S
30	INACTIVE	DO-TERM	102
40	INACTIVE	DO-TERM	104
50	INACTIVE	DO-TERM	111

SUBROUTINE GRAPH

--ENTRY POINTS--(LO*A)

-NAME---	-ADDRESS--	-ARGS---
GRAPH	5B	0

--STATISTICS--

PROGRAM-UNIT LENGTH	10538	• 555
CM LABELLED COMMON LENGTH	35578	• 1903
CM STORAGE USED	62600B	• 25984
COMPILE TIME	0.306 SECONDS	

-NAME---	-ADDRESS--	-BLOCK-----	-PROPERTIES-------	-TYPE---------	-SIZE
LIMIT	36B	/SAV$E/	SAV	INTEGER	
LINE	10B	/SAV$E/	SAV	CHAR*1	51
M	45B	/SAV$E/	SAV	INTEGER	
NCROSS	2B	/SEN2/		INTEGER	
NSET	62B	/SEN5/		INTEGER	
NSYS	0B	/SYS1/		INTEGER	
OBJCTV	16B	/SAV$E/	SAV	CHAR*10	16
SENS	1B	/SEN1/		CHAR*5	
SYMBOL	3B	/SAV$E/	SAV	CHAR*1	50
SYSNAM	1B	/SEN1/		CHAR*10	
SYSNUM	1B	/SAV$E/	UNUSED/SAV	INTEGER	
UHOLD	2B	/SEN3/		REAL	1701
UMAX	1B	/SEN2/		REAL	
UMIN	0B	/SEN2/		REAL	
XVAL	47B	/SAV$E/	SAV	REAL	

```
      SUBROUTINE HEADER(I,OBJCTV,ANSWER)
C
C*****************************************************************
C                                                                 *
C*                                                                *
C*   THIS ROUTINE PROVIDES THE HEADING FOR THE TABULAR FORMAT     *
C*   OF THE TABULAR FORMAT OF THE OUTPUT OF SENSITIVITY RESULTS.  *
C*                                                                *
C*                                                                *
C*   CALLED BY:  TABDIS                                           *
C*                                                                *
C*                                                                *
C*   VARIABLES                                                    *
C*          USED:  ATTNAM,I,IFIND,MTYPE,OBJCTV,SENS,SYSNAM        *
C*                                                                *
C*          MODIFIED:  ANSWER,TEST                                *
C*                                                                *
C*****************************************************************
C
      COMMON/LEVEL/NLVLS,INNRM(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN2/VMIN,VMAX,NCROSS,INDX,ISYS
      COMMON/SEN4/MTYPE
      SAVE
      CHARACTER*10 ANSWER*70,SENS*5,SYSNAM,OBJCTV(16),ATTNAM,
     * MTYPE*5,TEST*5
      ANSWER=' '
      IF(I.EQ.1)THEN
      IF(SENS.EQ.'SYSTM')THEN
      TEST=MTYPE
      ELSE
      TEST=SENS
      ENDIF
      IF(TEST.EQ.'LEVEL')THEN
      ANSWER=ATTNAM//ANSWER//'('//SYSNAM//')'
      ANSWER=TEST//'1'//ANSWER//' SENSITIVITY ANALYSIS'
      ENDIF
      ELSEIF(I.EQ.2)THEN
      ANSWER='FOR NODE1'
      CALL OBJECT(IFIND,OBJCTV)
      ENDIF
      RETURN
      END
```

--VARIABLE MAP--(LO-A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ANSWER	3	DUMMY-ARG		CHAR*70	
ATTNAM	0B	/SEN1/		CHAR*10	
I	1	DUMMY-ARG		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
INDX	3B	/SEN2/		INTEGER	
INNRM	1B	/LEVEL/		INTEGER	20
ISYS	4B	/SEN2/		INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	33B	/LEVEL/		INTEGER	
MTYPE	0B	/SEN4/		CHAR*5	
NCROSS	2B	/SEN2/		INTEGER	

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
 MDIFF      26B  /LEVEL/                    INTEGER
 NLVLS       0B  /LEVEL/                    INTEGER
 OBJCTV      2   DUMMY-ARG                  CHAR*10        16
 SEMS        1B  /SEM1/                     CHAR*5

-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
 SYSNAM      13  /SEM1/                     CHAR*10
 TEST        1B  /SBABUSE/ SAV              CHAR*5
 VMAX        1B  /SEM2/                     REAL
 VMIN        6B  /SEM2/                     REAL

--PROCEDURES--(LO=A)
 NAME------TYPE--------ARGS------CLASS-----

 OBJECT                  2      SUBROUTINE

--ENTRY POINTS--(LO=A)
 NAME---ADDRESS--ARGS---

 HEADER      5B     3

--STATISTICS--

 PROGRAM-UNIT LENGTH              163B =  115
 CM LABELLED COMMON LENGTH        140B =  96
 CM STORAGE USED                60600B = 24960
 COMPILE TIME                   0.065 SECONDS
```

```
      SUBROUTINE NODSET
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ELICITS THE SET OF NODES OVER WHICH            *
C*   THE SSA IS TO BE PERFORMED (ANALOGOUS TO DETNOD).           *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SYSSEN                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  IDATA,IFIND,MTYPE,OBJCTV                      *
C*                                                               *
C*          MODIFIED:  CH,FINDST,ICONT,II,INDXST,K,              *
C*                     NCOUNT,NSET,UMAX,UMAXI,UMIN,UMINI         *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLVLS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/NEX/ICONT,IDATA,ITOTL
      COMMON/SEN1/ATTNAM,SEN5,SYSNAM
      COMMON/SEN2/UMIN,UMAX,NCROSS,INDX,ISYS
      COMMON/SEN4/MTYPE
      COMMON/SEN5/FINDST(50),NSET,INDXST(50),KIDSET(50)
      SAVE
      CHARACTER*10 SEN5*5,CH*1,OBJCTV(16),ATT1,LABEL,SYSNAM,ATTNAM,
     * MTYPE*5
      INTEGER FINDST,INDXST,KIDSET
1     WRITE(*,'(S(/),1X)')
      PRINT*,'HOW MANY NODES WOULD YOU LIKE TO EXAMINE? (1-50)'
      PRINT*,'  ?'
      READ(*,'(I2)' NSET
      IF((NSET.LT.1).OR.(NSET.GT.50))THEN
      PRINT*,'INPUT VALUE OUT OF RANGE'
      GO TO 1
      ENDIF
      NCOUNT=0
2     WRITE(*,'(S(/),1X)')
      INDXST(NCOUNT+1)=0
      PRINT*,'ENTER NRN FOR WHICH ',MTYPE,' IS'
      PRINT*,'TO BE PERTURBED...'
      CALL PRENEX
      IF(ICONT.EQ.0)THEN
      PRINT*,'NODE DOES NOT EXIST'
      ELSE
      IF((MTYPE.EQ.'VALUE').AND.(IDATA.NE.1))THEN
      PRINT*,'A VALUE ANALYSIS MAY ONLY BE PERFORMED'
      PRINT*,'ON A DATA NODE.'
      ICONT=0
      ENDIF
      IF(ICONT.EQ.0)THEN
      PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
```

SUBROUTINE NODSET

```
      PRINT*,'      ?'
      READ(*,'(A1)')CH
      ENDIF
      ENDIF
      IF(ICONT.NE.1)GO TO 2
      NCOUNT=NCOUNT+1
      FINDST(NCOUNT)=IFIND
      PRINT*,' '
      CALL OBJECT(IFIND,OBJCTV)
      PRINT*,'THE OBJECTIVE IS('
      II=1
      DO 10 I=1,4
      WRITE(1,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
10    CONTINUE
      INDXST(NCOUNT)=IRAY(IFIND,1)
      KIDSET(NCOUNT)=0
      IFINDT=IRAY(IFIND,4)
      DO 20 I=1,10
      IF(IRAY(IFINDT,2).EQ.IFINDT+1)GO TO 3
      IFINDT=IRAY(IFINDT,4)
20    CONTINUE
3     IFINDT=IRAY(IFINDT,2)
      DO 30 I=1,10
      IF(IFINDT.GT.0)THEN
      KIDSET(NCOUNT)=KIDSET(NCOUNT)+1
      IFINDT=IRAY(IFINDT,3)
      ELSE
      GO TO 4
      ENDIF
30    CONTINUE
4     IF(NCOUNT.LT.NSET)GO TO 2
      UMIN=0.0
      UMAX=1.0
      IF(MTYPE.EQ.'VALUE')UMAX=100.0
5     PRINT*,' '
      PRINT*,'MINIMUM ',MTYPE,'(',UMIN,'-',UMAX,') IS?'
      READ(*,'(F5.2)')UMIN1
      IF((UMIN.LT.0.0).OR.(UMIN1.GT.UMAX))THEN
      PRINT*,'VALUE OUT OF RANGE'
      GO TO 5
      ENDIF
      UMIN=UMIN1
6     PRINT*,' '
      PRINT*,'MAXIMUM ',MTYPE,'(',UMIN,'-',UMAX,') IS?'
      READ(*,'(F5.2)')UMAX1
      IF((UMAX1.LT.UMIN).OR.(UMAX1.GT.UMAX))THEN
      PRINT*,'INPUT IS OUT OF RANGE'
      GO TO 6
      ENDIF
      UMAX=UMAX1
      RETURN
      END
```

```
      SUBROUTINE NODSET

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
                                            CHAR*10
 ATTNAM      0B   /SEN1/                    CHAR*16
 ATT1       22B   /S$ASUSE/ UNUSED/SAV      CHAR*1
 CH          1B   /S$ASUSE/ SAV             INTEGER        50
 FINDST      0B   /SEN5/                    INTEGER
 I          26B   /S$ASUSE/ SAV             INTEGER
 ICONT       0B   /NEX/                     INTEGER
 IDATA       1B   /NEX/                     INTEGER
 IFADD      27B   /LEVEL/                   INTEGER
 IFIND      25B   /LEVEL/                   INTEGER
 IFINDT     30B   /S$ASUSE/ SAV             INTEGER
 II         25B   /S$ASUSE/ SAV             INTEGER
 INDX        3B   /SEN2/                    INTEGER        50
 INDXST     63B   /SEN5/                    INTEGER        20
 INNRM       1B   /LEVEL/                   INTEGER
 ISYS        4B   /SEN2/                    INTEGER
 ITOTL       2B   /NEX/                     INTEGER
 K          27B   /S$ASUSE/ SAV

-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE
                                            INTEGER        50
 KIDSET    145B   /SEN5/                    CHAR*10
 LABEL      23B   /S$ASUSE/ UNUSED/SAV      INTEGER        60
 LEVEL      31B   /LEVEL/                   INTEGER
 LVL        30B   /LEVEL/                   CHAR*5
 MTYPE       0B   /SEN4/                    INTEGER
 NCOUNT     24B   /S$ASUSE/ SAV             INTEGER
 NCROSS      2B   /SEN2/                    INTEGER
 NDIFF      26B   /LEVEL/                   INTEGER
 NLVLS       0B   /LEVEL/                   INTEGER
 NSET       62B   /SEN5/                    CHAR*10        16
 OBJCTV      2B   /S$ASUSE/ SAV             CHAR*5
 SENS        1B   /SEN1/                    CHAR*10
 SYSNAM      1B   /SEN1/                    REAL
 XMAX        1B   /SEN2/                    REAL
 XMAX1      32B   /S$ASUSE/ SAV             REAL
 XMIN        0B   /SEN2/                    REAL
 XMIN1      31B   /S$ASUSE/ SAV

--PROCEDURES--(LO=A)
-NAME------TYPE---------ARGS------CLASS-----
 IRAY      INTEGER        2       FUNCTION
 OBJECT                   2       SUBROUTINE
 PRENEX                   0       SUBROUTINE

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF
     1      7B                    34
     2     32B                    43
     3    211B                    80

-LABEL-ADDRESS-----PROPERTIES----DEF
     4    241B                    89
     5    255B                    93
     6    276B                   101

-LABEL-ADDRESS-----PROPERTIES----DEF
    10   INACTIVE   DO-TERM       72
    20   INACTIVE   DO-TERM       79
    30   INACTIVE   DO-TERM       88

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---
 NODSET     5B     0

--STATISTICS--
  PROGRAM-UNIT LENGTH              574B =  380
  CM LABELLED COMMON LENGTH        423B =  275
  CM STORAGE USED                 60E00B =  24960
  COMPILE TIME                    0.209 SECONDS
```

```
      SUBROUTINE SENTVP
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ESTABLISHES THE SUBTYPE OF SSA TO BE           *
C*   PERFORMED.                                                  *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SYSSEN                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  CH,MTYPE                                  *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/SEN4/MTYPE
      SAVE
      CHARACTER*5 MTYPE,CH*1
      PRINT*,' '
      PRINT*,'WHAT TYPE OF ANALYSIS WOULD YOU LIKE TO DO...'
1     PRINT*,'C)UMWT     R)ELWT     V)ALUE'
      PRINT*,'     ?'
      READ(*,'(A1)')CH
      IF((CH.NE.'C').AND.(CH.NE.'R').AND.(CH.NE.'V'))THEN
      PRINT*,'PLEASE INPUT "C" OR "R" OR "V" '
      GO TO 1
      ENDIF
      IF(CH.EQ.'C')THEN
      MTYPE='CUMWT'
      ELSEIF(CH.EQ. R')THEN
      MTYPE='RELWT'
      ELSE
      MTYPE='VALUE'
      ENDIF
      RETURN
      END
```

```
--VARIABLE MAP--(LO-A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

CH          1C  /SEGBUNE/ SAV              CHARE2
MTYPE       E8  /SEN4/                     CHARE5
```

```
--STATEMENT LABELS--(LO-A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    1     138                          88

--ENTRY POINTS--(LO-A)
-NAME---ADDRESS--ARGS---

 SENTVP     58     0

--STATISTICS--

 PROGRAM-UNIT LENGTH                 1638 =  115
 CM LABELLED COMMON LENGTH             38 =    3
 CM STORAGE USED                   606088 =  24960
 COMPILE TIME                      0.052 SECONDS
```

```
      SUBROUTINE SENVAL
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE ESTABLISHES THE SPECIFIC ALTERNATIVE SYSTEM    *
C*   TO BE EXAMINED FOR LSA AND SSA.                             *
C*                                                               *
C*                                                               *
C*   CALLED BY:  DETMOD,SYSSEN                                   *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ATTNAM,IFIND,MTYPE,NSYS,SENS                  *
C*                                                               *
C*          MODIFIED:  ANSWER,CH,I,II,IJ,INDEX,ISYS,K,SYSNAM,TSYS*
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN2/WMIN,WMAX,NCROSS,INDX,ISYS
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 SYSLBL,ANSWER,SYSNAM,SENS*5,CH*1,TSYS(4),ATTNAM
1     WRITE(*,'(5(/),1X)')
      PRINT*,'SYSTEMS AVAILABLE:'
      INDEX=NSYS/4
      II=1
      DO 10 I=1,INDEX
      IJ=1
      DO 9 K=II,II+3
      TSYS(IJ)=SYSLBL(K)
      IJ=IJ+1
9     CONTINUE
      WRITE(*,'(1X,4(A10,5X))')(TSYS(K),K=1,4)
      II=II+4
10    CONTINUE
      IF(INDEX*4.LT.NSYS)THEN
      IJ=1
      DO 11 K=INDEX*4+1,NSYS
      TSYS(IJ)=SYSLBL(K)
      IJ=IJ+1
11    CONTINUE
      WRITE(*,'(1X,4(A10,5X))')(TSYS(K),K=1,IJ-1)
      ENDIF
      PRINT*,' '
      IF(SENS.NE.'SYSTM')THEN
      PRINT*,'ENTER SYSTEM OF WHICH ',ATTNAM,' IS TO BE'
      PRINT*,'PERTURBED...'
      ELSE
      PRINT*,'WHAT SYSTEM IS TO BE ANALYZED?'
      ENDIF
      PRINT*,'     ?'
```

```
      SUBROUTINE SENVAL

 58                 READ(*,'(A10)')ANSWER
 59                 ISYS=0
 60                 DO 20 I=1,NSYS
 61                 IF(ANSWER.EQ.SYSLBL(I))THEN
 62                 ISYS=I
 63                 SYSNAM=ANSWER
 64                 ENDIF
 65           20    CONTINUE
 66                 IF(ISYS.EQ.0)THEN
 67                 PRINT*,'SYSTEM ENTERED IS NOT VALID.'
 68                 PRINT*,'(PRESS ANY LETTER TO CONTINUE)'
 69                 PRINT*,'    ?'
 70                 READ(*,'(A1)')CH
 71                 GO TO 1
 72                 ENDIF
 73                 PRINT*,' '
 74                 IF(SENS.NE.'SYSTM')THEN
 75                 PRINT*, CURRENT MODE LEVEL IS ',XLEVEL(VRAY(IFIND,ISYS),ATTNAM)
 76                 ENDIF
 77                 RETURN
 78                 END
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ANSWER	1B	/SAVSE/	SAV	CHAR*10	
ATTNAM	0B	/SEN1/		CHAR*10	
CH	2B	/SAVSE/	SAV	CHAR*1	
I	11B	/SAVSE/	SAV	INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
II	10B	/SAVSE/	SAV	INTEGER	
IJ	12B	/SAVSE/	SAV	INTEGER	
INDEX	7B	/SAVSE/	SAV	INTEGER	
INDX	3B	/SEN2/		INTEGER	
INNRN	1B	/LEVEL/		INTEGER	20
ISYS	4B	/SEN2/		INTEGER	
K	13B	/SAVSE/	SAV	INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	60
LVL	30B	/LEVEL/		INTEGER	
NCROSS	2B	/SEN2/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
NLVLS	0B	/LEVEL/		INTEGER	
NSYS	0B	/SYS1/		INTEGER	
SENS	1B	/SEN1/		CHAR*5	
SYSNAM	1B	/SEN1/		CHAR*10	
TSYS	3B	/SAVSE/	SAV	CHAR*10	4
VMAX	1B	/SEN2/		REAL	
VMIN	0B	/SEN2/		REAL	

--PROCEDURES--(LO=A)

-NAME------	TYPE--------	ARGS--------	CLASS-----
SYSLBL	CHAR*10	1	FUNCTION
VRAY	REAL	2	FUNCTION
XLEVEL	REAL	2	FUNCTION

SUBROUTINE IENVAL

--ENTRY POINTS--(LO=A)

-NAME---	ADDRESS--	ARGS---
SENVAL	5B	0

--STATEMENT LABELS--(LO=A)

-LABEL-	ADDRESS-----	PROPERTIES----	DEF
1	7B		2B
9	INACTIVE	DO-TERM	3B
10	INACTIVE	DO-TERM	41
11	INACTIVE	DO-TERM	47
20	INACTIVE	DO-TERM	6C

--STATISTICS--

```
PROGRAM-UNIT LENGTH                 4448 =  292
CM LABELLED COMMON LENGTH           1528 =  106
CM STORAGE USED                   60600B =  24960
COMPILE TIME                    0.144 SECONDS
```

```
      SUBROUTINE SYSSEN
C
C
C*****************************************************************
C*                                                               *
C*   THIS ROUTINE INITIALIZES THE PROGRAM FOR CONDUCTING         *
C*   SSA.                                                        *
C*                                                               *
C*   CALLED BY:  SEMSTV                                          *
C*                                                               *
C*   VARIABLES
C*          USED:  (NONE)                                        *
C*                                                               *
C*          MODIFIED:  (NONE)                                    *
C*                                                               *
C*****************************************************************
C
C
      CALL SENVAL
      CALL SENTYP
      CALL NODSET
      RETURN
      END
```

```
--PROCEDURES--(LO*A)
 -NAME------TYPE--------ARGS------CLASS-----

  NODSET                  0       SUBROUTINE
  SENTYP                  0       SUBROUTINE
  SENVAL                  0       SUBROUTINE

--ENTRY POINTS--(LO*A)
 -NAME---ADDRESS--ARGS---

  SYSSEN      5B      0

 --STATISTICS--

  PROGRAM-UNIT LENGTH              25B =  21
  CM STORAGE USED                 66400B =  24960
  COMPILE TIME                    6.817 SECONDS
```

```
      SUBROUTINE TABDIS
C
C
C*****************************************************************
C*                                                               *
C*                                                               *
C*   THIS ROUTINE GENERATES THE TABULAR OUTPUT FOR               *
C*   SENSITIVITY RESULTS.                                        *
C*                                                               *
C*                                                               *
C*   CALLED BY:  SENSTV                                          *
C*                                                               *
C*                                                               *
C*   VARIABLES                                                   *
C*          USED:  ANSWER,FINDST,RTYPE,NSYS,OBJCTV,SENS,         *
C*                 SYSNAM,.HOLD                                  *
C*                                                               *
C*          MODIFIED:  DUM,I,IFIND,II,IJ,INDEX,J,K,MM,           *
C*                     NCOUNT,NSET,NUM,STAR,TEST,TSYS,U          *
C*                                                               *
C*                                                               *
C*****************************************************************
C
C
      COMMON/SEN1/ATTNAM,SENS,SYSNAM
      COMMON/SEN3/UNHOLD(0:20,0:50)
      COMMON/SEN4/RTYPE
      COMMON/SEN5/FINDST(50),NSET,INDXST(50),KIDSET(50)
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*10 OBJCTV(16),ANSWER*70,TSYS(4),SYSLBL,SENS*5,SYSNAM,
     * ATTNAM,DUM(0:4)*5,TEST*5,RTYPE*5
      REAL NUM(0:4)
      INTEGER STAR
      WRITE(*,'(5(/),1X)')
      NCOUNT=1
      IF(SENS.NE.'SYSTM')NSET=1
      IF(SENS.EQ.'SYSTM')IFIND=FINDST(NCOUNT)
      CALL HEADER(1,OBJCTV,ANSWER)
      PRINT*,ANSWER
      IF(SENS.NE.'SYSTM')THEN
      CALL HEADER(2,OBJCTV,ANSWER)
      PRINT*,ANSWER
      II=1
      DO 10 I=1,4
      WRITE(*,'(1X,4A10)')(OBJCTV(K),K=II,II+3)
      II=II+4
   10 CONTINUE
      ELSE
      PRINT*,'FOR SYSTEM ',SYSNAM
      ENDIF
      IF(SENS.EQ.'SYSTM')THEN
      MM=NSET
      ELSE
      MM=NSYS
      ENDIF
      INDEX=MM/4
```

```
      DO 20 I=1,INDEX
      IJ=1
      DO 19 J=I*4-3,I*4
      IF(SENS.NE.'SYSTM')THEN
      TSYS(IJ)=SYSLBL(J)
      ELSE
      TSYS(IJ)='MODE'
      ENDIF
      IJ=IJ+1
19    CONTINUE
      IF(SENS.NE.'SYSTM')THEN
      TEST=SENS
      ELSE
      TEST=MTYPE
      ENDIF
      WRITE(*,'(1X,A5,T15,4(5X,A10)) )TEST,(TSYS(K),K=1,4)
      DO 30 J=0,20
      U=0
      STAR=0
      DO 40 K=I*4-3,I*4
      IF(UHOLD(J,K).GT.U)THEN
      STAR=K
      U=UHOLD(J,K)
      ENDIF
40    CONTINUE
      DUM(0)='  '
      NUM(0)=UHOLD(1,0)
      IJ=1
      DO 50 K=I*4-3,I*4
      IF(K.EQ.STAR)THEN
      NUM(IJ)=UHOLD(J,K)
      DUM(IJ)='*  '
      IJ=IJ+1
      ELSE
      NUM(IJ)=UHOLD(J,K)
      DUM(IJ)='  '
      IJ=IJ+1
      ENDIF
50    CONTINUE
      WRITE(*,'(1X,5(F10.4,A5))')NUM(0),DUM(0),NUM(1),DUM(1),NUM(2),
     * DUM(2),NUM(3),DUM(3),NUM(4),DUM(4)
30    CONTINUE
20    CONTINUE
      IF(INDEX*4.LT.NM)THEN
      IJ=1
      DO 21 I=INDEX*4+1,NM
      IF(SENS.NE.'SYSTM')THEN
      TSYS(IJ)=SYSLBL(I)
      ELSE
      TSYS(IJ)='MODE'
      ENDIF
      IJ=IJ+1
21    CONTINUE
      IF(SENS.NE.'SYSTM')THEN
      TEST=SENS
      ELSE
      TEST=MTYPE
```

```
SUBROUTINE TABDIS      74/74    OPT=0

                 ENDIF
                 WRITE(*,'(1X,A6,T15,4(6X,A10))')TEST,(TSYS(K),K=1,IJ-1)
                 DO 31 J=0,20
                 W=0
                 STAR=0
                 DO 41 K=INDEX*4+1,MM
                 IF(WHOLD(J,K).GT.W)THEN
                 STAR=K
                 W=WHOLD(J,K)
                 ENDIF
          41     CONTINUE
                 DUM(0)='
                 NUM(0)=WHOLD(J,0)
                 IJ=1
                 DO 51 K=INDEX*4+1,MM
                 IF(K.EQ.STAR)THEN
                 NUM(IJ)=WHOLD(J,K)
                 DUM(IJ)='*     '
                 IJ=IJ+1
                 ELSE
                 NUM(IJ)=WHOLD(J,K)
                 DUM(IJ)='      '
                 IJ=IJ+1
                 ENDIF
          51     CONTINUE
                 WRITE(*,'(1X,5(F10.4,A5))')(NUM(I),DUM(I),I=0,IJ-1)
          31     CONTINUE
                 ENDIF
                 RETURN
                 END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ANSUER	21B	/SAU$E/	SAV	CHAR*70	
ATTNAM	0B	/SEN1/		CHAR*10	
DUM	34B	/SAU$E/	SAV	CHAR*5	5
FIKDST	0B	/SEN5/		REAL	50
I	51B	/SAU$E/	SAV	INTEGER	
IFIKD	47B	/SAU$E/	SAV	INTEGER	
II	50B	/SAU$E/	SAV	INTEGER	
IJ	55B	/SAU$E/	SAV	INTEGER	
INDEX	54B	/SAU$E/	SAV	INTEGER	
INDXST	63B	/SEN5/		INTEGER	50
J	56B	/SAU$E/	SAV	INTEGER	
K	52B	/SAU$E/	SAV	INTEGER	
KIDSET	145B	/SEN6/		INTEGER	50
MM	53B	/SAU$E/	SAV	INTEGER	
MTYPE	0B	/SEN4/		CHAR*5	
NCOUNT	46B	/SAU$E/	SAV	INTEGER	
NSET	62B	/SEN5/		INTEGER	
NSYS	0B	/SYS1/		INTEGER	
NUM	40B	/SAU$E/	SAV	REAL	5
OBJCTV	1B	/SAU$E/	SAV	CHAR*10	16
SENS	1B	/SEN1/		CHAR*5	
STAR	45B	/SAU$E/	SAV	INTEGER	
SYSNAM	1B	/SEN1/		CHAR*10	
TEST	37B	/SAU$E/	SAV	CHAR*5	
TSYS	30B	/SAU$E/	SAV	CHAR*10	4
W	57B	/SAU$E/	SAV	REAL	
WHOLD	0B	/SEN3/		REAL	1701

```
      SUBROUTINE TABDIG

--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 HEADER                    3      SUBROUTINE
 BVSLBL     CHAR*10        1      FUNCTION

--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM      48
    19  INACTIVE   DO-TERM      67
    20  INACTIVE   DO-TERM     100
    21  INACTIVE   DO-TERM     110

-LABEL-ADDRESS-----PROPERTIES----DEF

    30  INACTIVE   DO-TERM      99
    31  INACTIVE   DO-TERM     141
    40  INACTIVE   DO-TERM      82

-LABEL-ADDRESS-----PROPERTIES----DEF

    41  INACTIVE   DO-TERM     125
    50  INACTIVE   DO-TERM      96
    51  INACTIVE   DO-TERM     139

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 TABDIG      58      0

--STATISTICS--

 PROGRAM-UNIT LENGTH          10048 =  516
 CM LABELLED COMMON LENGTH    35618 =  1905
 CM STORAGE USED              62600B =  25984
 COMPILE TIME                 0.285 SECONDS
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GOR/AA/81D-1	2. GOVT ACCESSION NO. A111 105	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MADAM: MULTIPLE-ATTRIBUTE DECISION ANALYSIS MODEL VOLUME II		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Wayne A. Stimpson, 2Lt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, OHIO 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1981 13. NUMBER OF PAGES 164
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

28 JAN 1982

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE AFR 190-17.

Fredric C. Lynch

FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Multiple-Attribute
Decision Analysis
Computer Aid
Preference
Value

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

The complex multifaceted decision situations present today suggest the need for a timely, automated tool. A decision-maker is forced into comparing alternative actions or systems over an entire set of different measures of merit. This effort is an on-line, real-time, computer-based decision aid designed to assist the decision-maker in clarifying preferences in a complex decision environment. It is applicable to problems which may be represented by a hierarchy of objectives to be satisfied. The program is MADAM: Multiple-Attribute Decision Analysis Model, and it is written in FORTRAN V and is implemented on the CYBER 175

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

system. MADAM is designed to aid the decision-maker as he or she progresses through problem formulation, parameterization, sensitivity analyses, and a decision, including storage of all data and rationales. Deterministic problems are analyzed through Multi-Attribute Utility Theory concepts and an additive value function is utilized for sensitivity analysis. Pairwise preferential independence is tested between attributes. The sensitivity analyses include a cumulative weight analysis, a relative weight analysis, and an attribute level analysis. The analyses may be conducted by fixing an objective to be considered and conducting the analysis across the alternative systems or actions, or conversely by fixing the alternative to be considered and conducting the analysis across a desired set of objectives.

The work is divided into two volumes. Volume I is a theoretical presentation and includes a user's manual. It requires no programming expertise and may be used independently of Volume II. Volume II is a programming manual including the source code. It may not be used independently of Volume I.