

AD-A119 160

SOUTHEASTERN CENTER FOR ELECTRICAL ENGINEERING EDUCAT--ETC F/G 9/2
ADVANCED TECHNOLOGY MULTIPLE CRITERIA DECISION MODEL.(U)

NOV 81 P J SWEENEY, K B BERNER, J R FRAKER F33615-77-C-2059

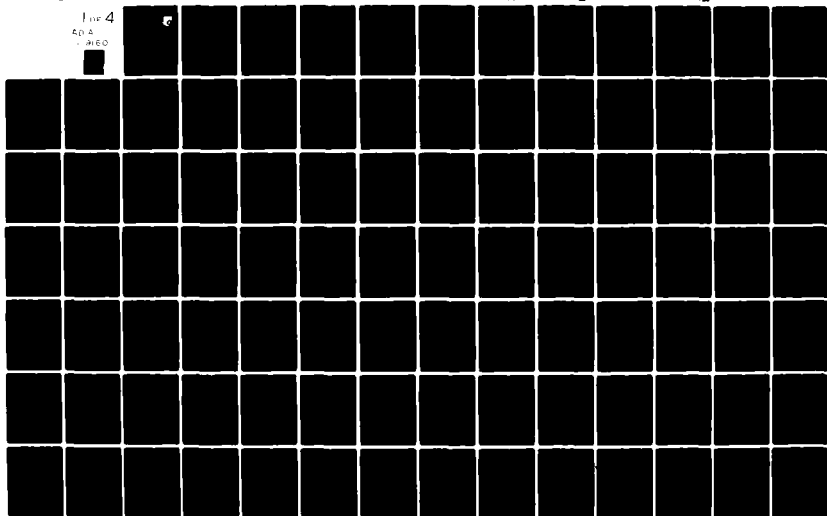
UNCLASSIFIED

AFWAL-TR-81-2112

NL

Line 4

AD A
11960



AD A119160

AFWAL-TR-81-2112

ADVANCED TECHNOLOGY MULTIPLE CRITERIA
DECISION MODEL



UNIVERSITY OF DAYTON
SCHOOL OF ENGINEERING
DAYTON, OHIO 45469

NOVEMBER, 1981

SEP 13 1982

A

FINAL REPORT FOR PERIOD 1 JUNE 1980 to 30 NOVEMBER 1981

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AERO PROPULSION LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

DTIC FILE COPY

033

NOTICE

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

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

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



PROJECT ENGINEER
Richard G. Honneywell



ROBERT R. BARTHELEMY
Chief, Energy Conversion Branch
Aerospace Power Division

FOR THE COMMANDER



JAMES D. REAMS
Chief, Aerospace Power Division
Aero Propulsion Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/POOC, W-PAPB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-81-2112	2. GOVT ACCESSION NO. AD-A119160	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ADVANCED TECHNOLOGY MULTIPLE CRITERIA DECISION MODEL		5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT 1 June 1980 to 30 November 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Patrick J. Sweeney Daniel J. Mick Kathleen B. Berner Judy A. Sweeney John R. Fraker		8. CONTRACT OR GRANT NUMBER(s) F33615-77-C-2059
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Dayton School of Engineering Dayton, OH 45469		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 62203F PROJECT 3145 TASK 314524 WORK UNIT 31452413
11. CONTROLLING OFFICE NAME AND ADDRESS AEROPROPULSION LABORTORY (AFWAL/POOC) AF WRIGHT AERONAUTICAL LABORATORIES (AFSC) WRIGHT-PATTERSON AFB OHIO 45433		12. REPORT DATE November 1981
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 330
		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)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Computer Software Technology Systems Decision Model Multiple Criteria Decision Model		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This computerized model provides a flexible, interactive capability to solve problems involving multiple criteria. The decision model is capable of analyzing up to 10 different technology systems over 5 time increments. The program accepts up to 20 quantitative and qualitative parametric descriptions for each technology system. The program provides a system value comparison based on the technology data base and an evaluation of the relative importance of the parameters.		

FOREWORD

This report prepared for the Aerospace Power Division, Aero Propulsion Laboratory, Wright-Patterson AFB, describes a computerized decision model capable of solving multiple criteria problems. The decision model is capable of analyzing up to 10 different technology systems over 5 time increments. Each technology system can be characterized by up to 20 quantitative and qualitative parametric descriptions for each technology system. A system value comparison is created based on the technology data base and an evaluation of the relative importance of the parameters.

The work reported herein was performed under project/task/work unit 3145/24/13 during the period 1 June 1980 to 30 November 1981, under the direction of Lt. Richard G. Honneywell, AFWAL/POOC, as project engineer. Dr. Patrick J. Sweeney, University of Dayton (School of Engineering), is technically responsible for the work.

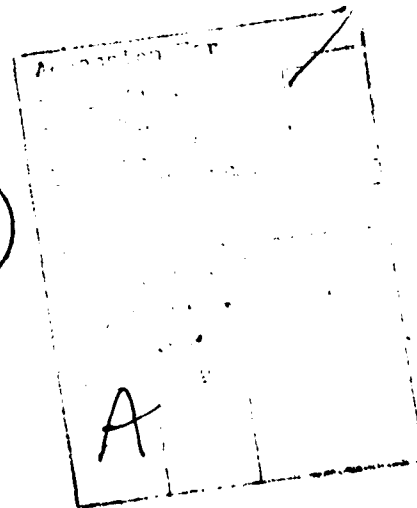


TABLE OF CONTENTS

SECTION		PAGE
I	INTRODUCTION	1
II	PROGRAMMING STANDARDS	9
III	ENVIRONMENT AND INTERFACES	13
IV	FEATURES OF THE PROGRAM	15
V	PROGRAMMING SPECIFICATIONS	85
VI	TEST/VERIFICATION SPECIFICATIONS	101
	APPENDIX A PROGRAM ANALYSIS	113
	APPENDIX B SYSTEM SELECTION MODEL IN BATCH MODE	117
	APPENDIX C EXAMPLES	121
	APPENDIX D UPPP FLOWCHART AND NARRATIVE	157
	APPENDIX E PPP FLOWCHART AND NARRATIVE	177
	APPENDIX F GRAPHX AND GRAPHT FLOWCHARTS AND NARRATIVES	193
	APPENDIX G SSMP FLOWCHART AND NARRATIVE	213
	APPENDIX H REFERENCES	241
	APPENDIX I GLOSSARY	243
	APPENDIX J LISTINGS	249

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	Flow Sheet of Three Programs	7
2	Problem Tree for College Selection Problem	16
3	Problem Tree for Generator Selection Problem	20
4	Generator Selection Problem Using PPP with CRTE	23
5	Generator Selection Problem Using UPPP with CRTE	29
6	Generator Selection Problem Using SSMP	33
7	Parameter Package Program Output	51
8	Output from the UPPP Program	79
9	Graphics Output for Generator Selection Problem	81
10	Overlay Tree for SSMP	87
11	Overlay Tree for GRAPHX	88
12	SSMP's Interaction with its Data Files	90
13	Data Structure of HEADER File	91
14	Data Structure of UPP File	93
15	Data Structure of PP File	94
16	Data Structure of SYSNUM File	95
17	HEADER File for the Generator Problem	96
18	UPP File for the Generator Problem	97
19	PP File for the Generator Problem	98
20	College Run	102

LIST OF TABLES

TABLE		PAGE
1	Ranking/Rating Example	5
2	Arrays That Can Be Modified	11

SECTION I

INTRODUCTION

1. BACKGROUND

Today's manager is often faced with the dilemma of selecting a course of action in the face of multiple and integrated goals or objectives. Rarely is a decision of any magnitude based solely on a single objective or goal. Instead, the decision maker often has several criteria upon which to base his decision. This program will assist the manager in his role as a decision maker by providing a flexible decision model that will analyze complex, multiple criteria decision problems. This decision model uses variable decision criteria and the analyst's evaluation of the relative importance of these criteria to select a preferred course of action. In addition, the model is structured to handle the change of the decision criteria values as a function of time. Results can be portrayed graphically as a function of time.

Suppose, for example, that one were tasked with the responsibility for selecting from a number of alternative weapon systems the weapon system that would best satisfy a particular mission. It is apparent that there would be many factors that would impact upon a decision of this magnitude. The potential destructive capability of the weapon system, the current research and development (R&D) program status, the expected improvements in the system as a function of time, the expected life cycle cost of the weapon system, etc., would significantly impact upon this decision. Initial cost would also certainly impact upon the economic feasibility of a particular weapon system.

It is also easily recognized that the factors noted above may vary in importance. It is in the model development and in the use of this particular technique that one identifies the criteria (parameters) that will be used in making that selection and then rates these parameters as a function of their importance. It can be seen that if the decision can be made at a future time, then the importance of particular parameters may change significantly with time. For example, the total ordnance delivery capability of a particular weapon system may vary significantly over a 25-year period. The R&D effort necessary and the expected results could significantly change the performance of a particular weapon system. It is with this in mind that the model was developed with the option to take into account the effects of change in parameter values as a function of time.

2. PURPOSE AND SCOPE

The purpose of this model is to provide a flexible, interactive computer program for use in multiple criteria decision problems. By following procedures provided interactively by the program on a computer terminal, the analyst is able to introduce the parameters, the data base, and the ratings of the parameters necessary to obtain a problem solution. The results are presented in tabular and graphic form.

3. GENERAL DESCRIPTION OF THE PROGRAM

The model consists basically of three major programs: the parameter package program (PPP); the user's preference package program (UPPP); and the system selection model program (SSMP).

The parameter package program allows the analyst to identify the parameters which are used as criteria in the decision model in choosing the optimal result and to enter

the parameter data for each system. PPP creates two data files: PP (or the parameter package), which contains the parameter data, and HEADER, which contains the problem size and various names of systems, subsystems, and parameters.

The user's preference package program is used to evaluate and rate the parameters. UPP creates a third data file, UPP (or the user's preference package) containing the user's normalized ratings for the parameters. The user's preference package is constructed after the parameter package has been developed. It is necessary that those parameters identified in the parameter package program be the same parameters that are identified in the user's preference package program. In the UPPP, the analyst is given a choice of either identifying the importance of the parameters by using the paired comparison technique or using his own evaluation technique for determining the relative importance (ranking) of each parameter. The analyst then ranks the parameters from the most to the least important and rates each parameter accordingly with a value from 1 to 0. The analyst can establish different sets of ratings for different situations.

Data from these two packages are then used in the system selection model program. This program integrates the data from the two packages and utilizes this information in selecting the optimal solution as a function of the parameters discussed above.

As an example, suppose one is asked to make a decision as to which auxiliary power system should be selected, procured, and introduced into the Air Force inventory in the years 1980 to 2000. First, the analyst would select those parameters that would be used to differentiate among the power systems. For example, some parameters may be quantitative such as acquisition cost, life cycle cost,

reliability, or other numerical descriptions of the systems. Once these parameters are identified, the corresponding values, either actual or forecasted, would be entered into the model. These values may be in units or dollars, gallons, or any form which enables the program to evaluate each specific parameter relative to each system. These parameters are next ranked and rated by the analyst according to the relative importance of each parameter in the set.

Certain parameter data may not be in a quantitative form. Therefore, the model has been structured to operate with nonquantitative data by using qualitative terms such as very poor, poor, fair, good, or excellent to define the values of particular nonquantitative parameters. In the model development of this selection problem, the analyst has defined the important parameters in this particular decision which would be applicable and has quantified them on a relative to the systems under consideration.

The analyst must next evaluate the importance of each of the different parameters. As mentioned above, acquisition cost, life cycle cost, reliability, and other parameters have been identified as important in this decision problem. In this particular portion of the program, referred to as the user's preference package, it is necessary for the analyst to rank all of the parameters on a scale from most important to least important. From this ranking, the analyst determines a rating for each of the parameters on a scale from 1 to 0.

For example, in the case mentioned above, the parameters may be ranked in order of importance by the decision maker as acquisition cost, reliability, life cycle cost, and other parameters. These rankings are next rated by the analyst from 1 to 0. Table 1 exemplifies this ranking methodology.

The analyst has now completed both the parameter package and the user's preference package programs. These data are automatically fed into the system selection model program

TABLE 1
RANKING/RATING EXAMPLE

Unranked Parameters	Parameter Ranking	Ranked Parameters	Parameter Rating
Acquisition Cost	1	Acquisition Cost	0.95
Life Cycle Cost	3	Reliability	0.80
Reliability	2	Life Cycle Cost	0.40
Other Parameters	4	Other Parameters	0.10

(SSMP) when in the interactive mode from a computer terminal. If desired, the SSMP outputs at the terminal the tabular and graphic data and, in this example, will show which systems should be procured in which years between 1980 and 2000.

Figure 1 shows how the three programs interact.

4. UNCERTAINTY OF THE DECISION MODEL

Rarely is an analyst certain that the data are absolutely correct. Neither is the analyst always confident as to which data are critical to the decision. Because of these uncertainties, it is important that current models address this topic of uncertainty.

In this multiple criteria decision model, uncertainty may exist in the values of the parameters in the parameter package and the values of the ratings in the user's preference package. The uncertainty associated with the parameters stems from the fact that all future parameter values are merely estimates and that current or past data may be biased by the system used in collecting the data. The user's preference ratings are at best subjective and may actually vary widely.

The example discussed earlier concerns the selection of power systems for the next twenty years. Obviously, the costs, performances, etc., will change over time, and estimating these factors up to twenty years hence can be risky indeed. No matter how exhaustive or expensive the data search, a guarantee of a certain value or future still does not exist. Therefore, the topic of uncertainty is important, especially when the decision model is used to predict a future condition.

When parameter data and user preferences are known to be one hundred percent accurate the results are risk free. However, this is rarely the case as the accuracy of future

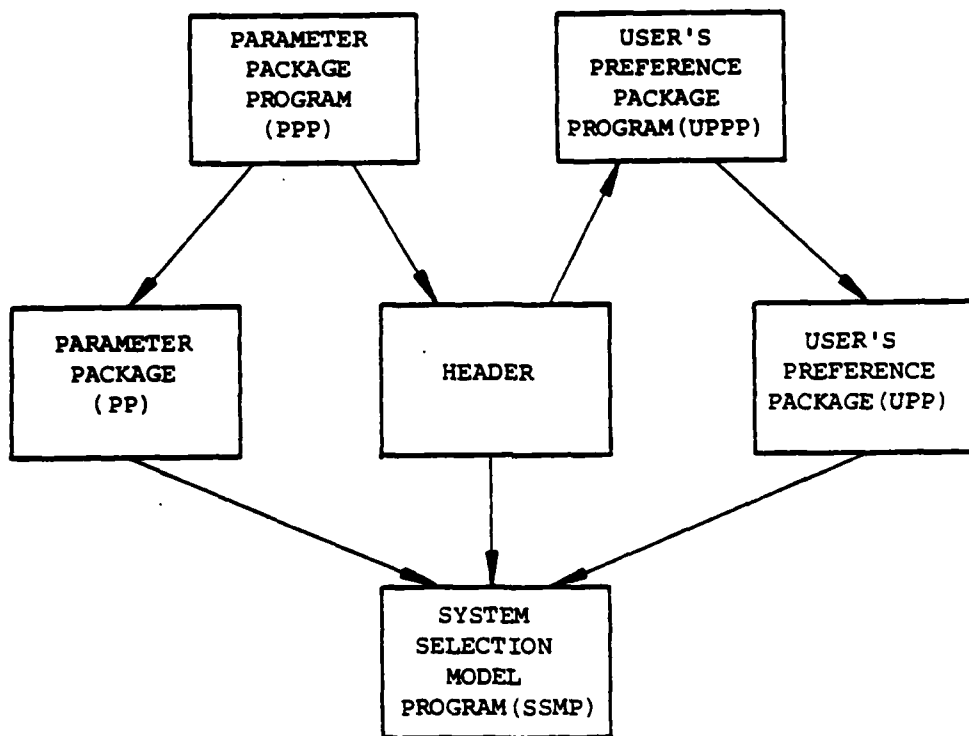


Figure 1. Flow Sheet of Three Programs

values is always uncertain or risky. This model is not structured such that a direct computation of system values given parameter uncertainties is possible. The authors suggest the following approach to addressing uncertainty with this model. First, determine the most important parameters. This can be accomplished by selecting those parameters with the highest user preference values and/or the highest relative valued parameters. Second, test the sensitivity of the model to small changes in the items identified in step one by executing the model with individual parameter value changes. Third, by experience, additional data, or other methodology, determine reasonable ranges for the sensitive parameters. This will establish upper and lower limits on the important and model sensitive parameters. Fourth, execute the model using combinations of the above parameters throughout the ranges identified in step three. A compilation of many runs will provide a mapping of the effects of varying the sensitive parameter values. This mapping will be a visual display of the system value variability as a function of the uncertainty of the parameter data and/or user preferences.

SECTION II

PROGRAMMING STANDARDS

1. PORTABILITY FEATURES

The programs described in this report are written in standard ANSI FORTRAN IV. To enhance the portability of these programs and in order to allow the programs to be used on other computer systems, certain conventions are followed. These conventions include:

- a. No more than five continuation cards are used.
- b. A maximum field width of four is used for the alpha numeric format, i.e., A4. This may differ in the graphics program.
- c. No EQUIVALENCE statements are used.
- d. Unit numbers in input/output statements appear as variables, and their values are located in a DATA statement. This feature allows modification of the unit numbers with one statement change.
- e. Hollerith formatting is used exclusively.

2. PROVISIONS FOR FUTURE MODIFICATION

The system selection model programs are designed to allow easy modification and expansion.

Features include:

- a. The same variable names and labeled COMMON statements are used throughout all programs but the graphics program. This is because the graphics programs are machine-dependent, and changes to the programs may be extensive if the program is to be used on another computer system.
- b. The input/output unit numbers can be modified easily by changing the appropriate DATA statements in the BLOCK DATA module in each program.

c. The do-loop indices, where applicable, are given by integer variables in place of constants.

d. The graphics programs are optional. As a separate segment they are easy to drop from the model.

e. The PROCEDURE files coordinate the attaching and returning of the data files. The PROCEDURE files are for convenience and have no bearing on the programs themselves.

f. If the programs are to be modified to run larger problems, the user should consult Table 2, to help facilitate array changes. Another program called system selection model program in background (SSMPB) is set up to run the larger problems. For further information consult Appendix B.

3. POTENTIAL DIFFICULTIES

Some features which may potentially cause problems include:

a. Labeled COMMONs are used.

b. Unformatted input/output is used.

c. The program SSMP and the two graphics programs are overlaid. Overlaying was used to allow these large programs to run interactively. The overlays can be changed to subroutines by: (1) deleting the OVERLAY cards; (2) changing PROGRAM cards to SUBROUTINE; (3) adding RETURN statements; and (4) changing CALL statements to call the subroutines.

d. The programs are executed using PROCEDURE files to attach and return data files. This feature allows the user to specify a different file name for each problem. PROCEDURE files are system-dependent; for further information, see Section IV.3.

TABLE 2

ARRAYS THAT CAN BE MODIFIED

Labeled COMMON Name	Array Name
/HDR1/	SUB1NM(NSUB1,4) SUB2NM(NSUB1,NSUB2,4) PARNAM(NPARM,4) NUM(NPARM) RATED(NPARM) YN(NPARM) INVERT(NPARM)
/HDR1/	SCNDSC(NSCEN,15)
/PARAM1/	CHPRM(NPARM) ABV(NPARM) BELW(NPARM) RLOWR(NPARM) HIR(NPARM) RTD(NPARM) IRANGE(NPARM) ISWNO(NPARM)
/PARAM2/	ELIM(INDEX,10,NPARM)
/PARAM3/	MATRIX(NPARM,INDEX,10)
/PARAM4/	SYSNRM(INDEX,10)
/RTG/	RTING(NPARM)
/RTNG/	RATING(NSCEN,NPARM)
/RTING/	RATING(NPARM)
None	PARAM(INDEX,10) VPARAM(INDEX,10) ICOUNT(NPARM) IORDER(NPARM) ICNT(NPARM) PSUM(NPARM) PAVE(NPARM)

Notes: NSUB1 = Number of Level 1 Subsystems
 NSUB2 = Number of Level 2 Subsystems
 NPARM = Number of Parameters
 INDEX = 1 for Case 1*
 NSUB1 for Case 2
 NSUB1 x NSUB2 for Case 3

* For an explanation of Cases 1, 2, and 3, see Section IV, 1.a page 15.

SECTION III

ENVIRONMENT AND INTERFACES

1. SYSTEM DESCRIPTION

The development of the system selection model programs has been conducted on the Wright-Patterson Air Force Base ASD computer system. Two large CDC computers support ASD, a CDC CYBER 175 and a CDC CYBER 74. The systems are for two different functions. The CYBER 175 is used for interactive support, while the CYBER 74 is used for batch processing and interactive graphics. It is possible to enter jobs on one computer system and have it processed on the other. This ability is due to the fact that both systems are architecturally compatible and share the same permanent disk facilities and tape drives. For further information, consult the Introduction to ASD Computer Center Guide (1).

2. HARDWARE CHARACTERISTICS AND CONSTRAINTS

Both systems have a large amount of central memory available. The CYBER 175 has 262,000 (60 bit) words, and the CYBER 74 has 131,000 words. Each system has 20 peripheral processing units available for input/output. Each system can handle 12 jobs in central memory at one time. Jobs not yet processed remain in queues. High speed readers of 1,200 characters per minute (cpm), card punches of 250 characters per minute, and line printers at 1,200 lines per minute (lpm) are available. There exist at present 7- and 9-track tape drives, with the 9-track tape drives capable of handling up to 6,250 bits per minute (bpm). The hardware facilities also include a 3.3 billion 6-bit character disk storage capability, with approximately 2.4 billion bytes available for user permanent file storage.

3. SOFTWARE ENVIRONMENT

The operating system in current use on the CDC systems is the Network Operating System/Batch Environment (NOS/BE). The decision model programs were developed on the CDC INTERCOM software. INTERCOM is the CDC software that supports the interactive access to the CDC computer system.

4. INTERFACE CHARACTERISTICS

The system selection model program (SSMP) will produce, as an option, TEKTRONIX graphs as output. TEKTRONIX graphics are accessed through a number of FORTRAN subroutines. The TEKTRONIX software used is PLOT-10/Advanced Graphics-II. For best results, the graphics program should be executed on a TEKTRONIX terminal that has at least a 5" X 7" screen. Terminals that are adequate include the TEKTRONIX 4010, 4051, and 4052 terminals.

SECTION IV

FEATURES OF THE PROGRAM

1. FUNCTIONAL OVERVIEW AND CAPABILITIES

The system selection model program (SSMP) determines the optimal, or best system from a number of choices. A problem must be able to meet certain specifications in order to execute SSMP.

a. Choices as Systems

The choices are termed systems. Sometimes the choices are complex and can be broken down into subsystems. As an example, a student must choose among different colleges. This student also cannot make up his/her mind as to which major he/she should take when attending college. Assume that the four choices of majors are chemical engineering, mechanical engineering, finance, and accounting. Assuming that the student can get accepted into any college with any choice of major, the problem can be set up as shown in Figure 2. In this figure, college is the system, the schools of business and engineering are level 1 subsystems, and the various departments are level 2 subsystems.

The breakdown of the system into subsystems is optional. The case where systems are not broken into subsystems is labeled Case 1. Case 2 is when the system is broken down into one level of subsystems. Case 3 is when the system is broken down into two levels of subsystems, as in this example.

To continue further with the example, the student specifies the colleges as: Harvard, Massachusetts Institute of Technology, University of Dayton, California Institute of Technology, and Yale. This particular problem needs to expand to include level 1 and level 2 subsystems, and it must do so in what is called a "balanced" fashion. That is, there

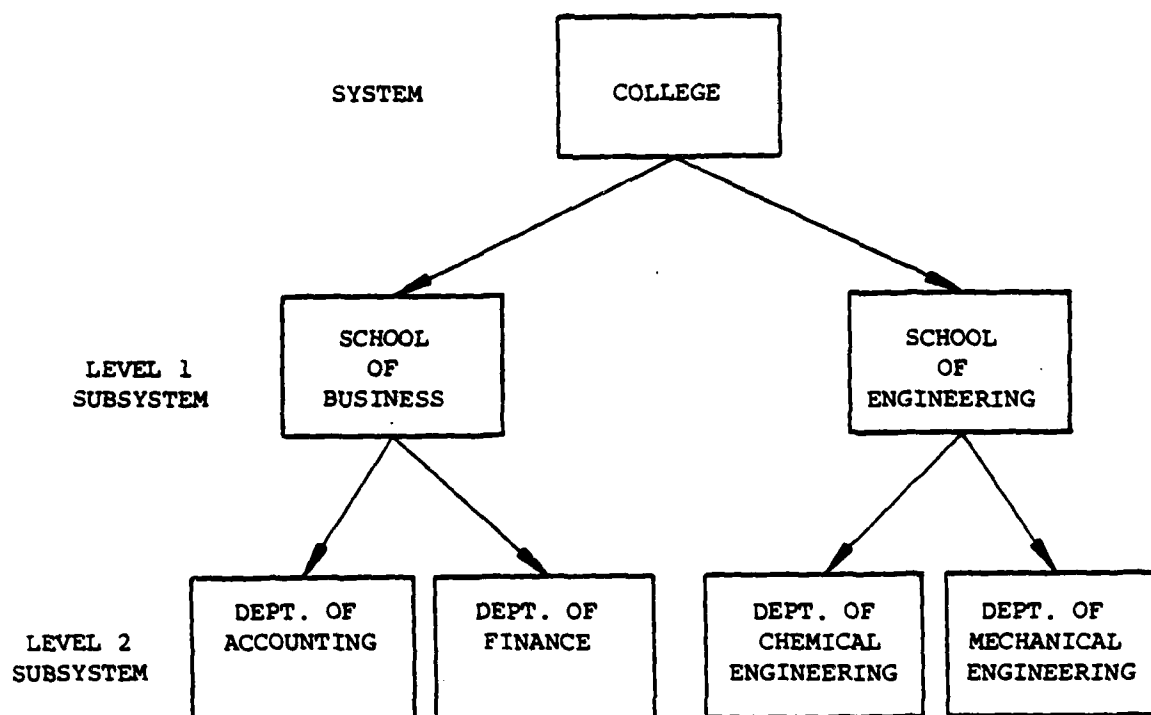


Figure 2. Problem Tree for College Selection Problem

must be at least two level 1 subsystems; if only one is desired, then another must be entered into the model to create a "balanced" tree. This may be accomplished by entering a level 1 subsystem named "BLANK" and entering zeros for its parameter data so that it does not influence the decision. Similarly, each level 1 subsystem must have at least two level 2 subsystems, and each must be broken down into the same number of level 2's. Of course, if the particular problem is Case 1 or 2, that is, deals only with systems or with systems and level 1's, then the stipulations on level 1 or level 2 subsystems may be ignored as they do not enter into the problem tree.

b. Determination of Parameters

The parameters to be used in judging the colleges must be determined. Some of the parameters for this example might include tuition, reputation of school, number of students in department, etc. The program recognizes three different types of parameters: (1) numerical (for example, tuition); (2) rated (for example, reputation of school can be specified as excellent, good, etc.); and (3) qualifying to indicate whether or not a system possesses a certain quality (such as off-campus housing for freshmen). The system will be eliminated if it does not possess the quality; data is entered in a yes/no form.

Several examples of this college selection problem appear in Appendix C.

Problems such as this example can be run interactively so long as the problem size is not too large. To execute these programs interactively on Wright-Patterson's CDC CYBER 175, the problem size is limited to:

- (1) No more than 10 systems.
- (2) No more than 40 parameters.

(3) No more than the product of seven for level 1 and level 2 subsystems. For example:

(a) Seven level 1 subsystems and no level 2 subsystems.

(b) Two level 1 subsystems and three level 2 subsystems.

(c) Three level 1 subsystems and two level 2 subsystems.

If the problem size is larger, the problem can be run in background or batch by changing the appropriate dimensions in the COMMON statements. The only limit that cannot be changed without some rewriting of the program is the number of systems. For instructions and examples on running SSMPB see Appendix B.

The user should consider the special case where time appears as a subsystem. If a problem's parameter data change with time, SSMP can be used to forecast the optimal system in the future. Within the graphics capability of the program a special program is available which provides a time graph for the specific problem of choosing the optimal system for five time intervals when time is a level 1 or 2 subsystem. This graph is different from the other graphs created by the normal histogram graphics feature, in that it plots time versus the system values. If the user believes that the output available is inadequate for his needs, the user can add other output routines rather easily. For more information on modification of these programs see Section II.

2. INPUT AND OUTPUT

This section describes line-by-line how to execute the parameter package program (PPP), the user's preference package program (UPPP), and the system selection model program (SSMP). Examples are provided to help illustrate each program. A generator selection problem is used as an

example of a Case 3 problem where four generators are the systems; the power sizes, 250 kW, 50 kW, and 10 kW are level 1 subsystems; and the level 2 subsystems are the years 1980 and 2000. The problem tree for the generator selection problem is shown in Figure 3.

a. Input

(1) Parameter package program (PPP) to create the HEADER file.

For PPP (using the command CRTE) to set up the data file HEADER, the user is required to specify:

(a) Data base name: specifies the problem name and should contain no more than 16 alphanumeric characters. If more than 16 characters are entered, the program truncates the name.

(b) Number of systems should not exceed ten.

(c) Number of level 1 subsystems: enter as a two-digit number; this specifies the number of level 1 subsystems per system.

(d) Number of level 2 subsystems: these specify the number of level 2 subsystems per level 1 subsystems and are entered as a two-digit number. If no subsystems are desired, be sure to enter the appropriate zeros. Also, level 2 subsystems cannot be specified if no level 1 subsystems exist.

(e) Number of parameters: enter as a two-digit number; the value should not exceed the value of 40 for the interactive version of SSMP.

(f) System setname: specifies the setname for the system. For example, the system setname for the diesel engine and Stirling engine could be generators. The setname should contain no more than 16 alphanumeric characters.

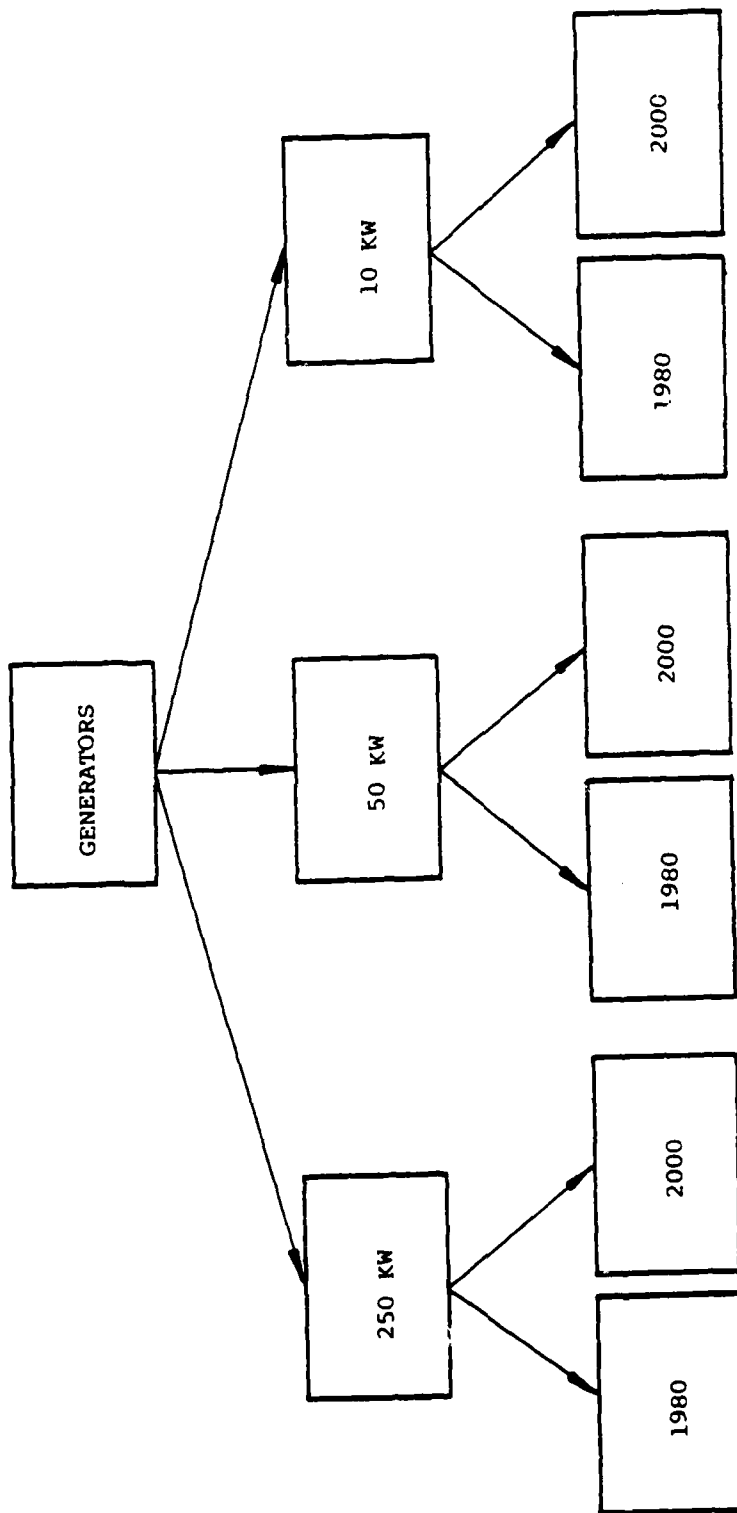


Figure 3. Problem Tree for Generator Selection Problem

(g) System names: these specify the specific names for each system, and each should have no more than 16 alphanumeric characters.

(h) Level 1 subsystem setname: if level 1 subsystems have been opted, the user enters the setnames describing the level 1 subsystems. A maximum of 16 alphanumeric characters is allowed.

(i) Level 1 subsystems names: if the level 1 subsystem option has been chosen, the user enters the specific names for the level 1 subsystems. A maximum of 16 alphanumeric characters is allowed.

(j) Level 2 subsystem setname: if the level 2 subsystems option has been chosen, the user enters the setnames describing the level 2 subsystems. A maximum of 16 alphanumeric characters is allowed.

(k) Level 2 subsystem names: if the level 2 subsystems option has been chosen, the user enters the specific names for the level 2 subsystems. A maximum of 16 alphanumeric characters is allowed.

(l) Parameter name: the parameter names are entered, and a maximum of 16 alphanumeric characters is allowed.

(m) Type of parameter: the user is required to identify the type of parameter, either numerical, rated or qualitative. When the parameter is specified numerically, the user is asked whether or not he wishes to invert the parameter. The SSMP determines the best system from the numerical data by considering the lowest values as the best value. See Appendix A for the inverting algorithm. In cases where this method of inverting may prove to be unsatisfactory, the user must treat the parameter data with a more appropriate method before entering it in the parameter

package program (PPP). Of course, the user should not indicate that the data are to be inverted if the data have already been inverted by some other means.

If the parameter is specified as a rated parameter, the user can describe the parameter in terms of very poor, poor, fair, good, or excellent. The program assigns numerical values to the ratings, and SSMP evaluates the parameters on the basis of these relative values.

If the parameter is specified as a qualifying parameter, this parameter is used to check whether or not the systems and its subsystems have this qualifier. If not, that system or subsystem is eliminated from the decision process. See Figure 4 for an example using the generator selection problem.

(2) Parameter package program (PPP) to create the parameter package (PP) file.

For PPP (using the command MNIN) to set up the data file PP, the user is required to specify:

(a) For a numerical parameter: the numerical data, in any format, of that system or subsystem.

(b) For a rated parameter: the rating of that system or subsystem.

(c) For a qualifying parameter: the user specifies whether or not that system or subsystem has that qualifier.

(3) User's preference package program to create the user's preference package (UPP) file.

For UPPP (using the command CRTE) to set up the data file UPP, the user is required to specify:

(a) Number of scenarios: scenarios describe the different situations where the rating values for the parameters would differ. For example, in the college selection problem, emphasis might shift for some parameters depending on how much a student could afford. Therefore,

```

*****
*               PARAMETER PACKAGE PROGRAM               *
*****

```

COMMANDS AVAILABLE:

```

CRTE - CREATES FILE OF NAMES
MNIN - ALLOWS MANUAL ENTERING OF PARAMETER VALUES
LIST - LISTS OUT NAMES AND PARAMETER VALUES
END - ENDS PROGRAM

```

COMMAND-->CRTE

```

ENTER LEVEL 1 SUBSYSTEM SETNAME (MAX. 16 CHARS)
KILOWATTS
ENTER NAME FOR KILOWATTS      , NO. 1  (MAX. 16 CHARS.)
250 KW
ENTER NAME FOR KILOWATTS      , NO. 2  (MAX. 16 CHARS.)
50 KW
ENTER NAME FOR KILOWATTS      , NO. 3  (MAX. 16 CHARS.)
10 KW
ENTER LEVEL 2 SUBSYSTEM SETNAME (MAX. 16CHARS.)
YEAR
ENTER NAME FOR YEAR           , 250 KW      , NO. 1  (MAX. 16 CHARS.)
1980
ENTER NAME FOR YEAR           , 250 KW      , NO. 2  (MAX. 16 CHARS.)
2000
ENTER NAME FOR YEAP           , 50 KW       , NO. 1  (MAX. 16 CHARS.)
1980
ENTER NAME FOR YEAR           , 50 KW       , NO. 2  (MAX. 16 CHARS.)
2000
ENTER NAME FOR YEAR           , 10 KW       , NO. 1  (MAX. 16 CHARS.)
1980
ENTER NAME FOR YEAR           , 10 KW       , NO. 2  (MAX. 16 CHARS.)
2000

```

Figure 4 - Generator Selection Problem Using
PPP With Command CRTE

ENTER LEVEL 1 SUBSYSTEM SETNAME (MAX. 16 CHARS)
 KILOWATTS
 ENTER NAME FOR KILOWATTS , NO. 1 (MAX. 16 CHARS.)
 250 KW
 ENTER NAME FOR KILOWATTS , NO. 2 (MAX. 16 CHARS.)
 50 KW
 ENTER NAME FOR KILOWATTS , NO. 3 (MAX. 16 CHARS.)
 10 KW
 ENTER LEVEL 2 SUBSYSTEM SETNAME (MAX. 16CHARS.)
 YEAR
 ENTER NAME FOR YEAR , 250 KW , NO. 1 (MAX. 16 CHARS.)
 1980
 ENTER NAME FOR YEAR , 250 KW , NO. 2 (MAX. 16 CHARS.)
 2000
 ENTER NAME FOR YEAR , 50 KW , NO. 1 (MAX. 16 CHARS.)
 1980
 ENTER NAME FOR YEAR , 50 KW , NO. 2 (MAX. 16 CHARS.)
 2000
 ENTER NAME FOR YEAR , 10 KW , NO. 1 (MAX. 16 CHARS.)
 1980
 ENTER NAME FOR YEAR , 10 KW , NO. 2 (MAX. 16 CHARS.)
 2000

ENTER NAME FOR PARAMETER NO. 1
 ACQUISITION COST
 ENTER NAME FOR PARAMETER NO. 2
 LIFE CYCLE COST
 ENTER NAME FOR PARAMETER NO. 3
 SYSTEM EFF.
 ENTER NAME FOR PARAMETER NO. 4
 START UP TIME
 ENTER NAME FOR PARAMETER NO. 5
 SHUTDOWN TIME
 ENTER NAME FOR PARAMETER NO. 6
 RELIABILITY
 ENTER NAME FOR PARAMETER NO. 7
 MAINT. AND OPER.
 ENTER NAME FOR PARAMETER NO. 8
 LIFETIME
 ENTER NAME FOR PARAMETER NO. 9
 THERMAL ENERGY
 ENTER NAME FOR PARAMETER NO.10
 VOLUME/SIZE
 ENTER NAME FOR PARAMETER NO.11
 WEIGHT
 ENTER NAME FOR PARAMETER NO.12
 FUEL USED
 ENTER NAME FOR PARAMETER NO.13
 GROWTH POTENTIAL
 ENTER NAME FOR PARAMETER NO.14
 ENVIRON. CONSTR.

Figure 4 - Continued

IS ACQUISITION COST A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR ACQUISITION COST ? (YES/NO)
 NO
 IS LIFE CYCLE COST A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR LIFE CYCLE COST ? (YES/NO)
 NO
 IS SYSTEM EFF. A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR SYSTEM EFF. ? (YES/NO)
 YES
 IS START UP TIME A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR START UP TIME ? (YES/NO)
 NO
 IS SHUTDOWN TIME A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR SHUTDOWN TIME ? (YES/NO)
 NO
 IS RELIABILITY A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR RELIABILITY ? (YES/NO)

Figure 4 - Continued

YES
 IS MAINT. AND OPER. A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR MAINT. AND OPER. ? (YES/NO)
 NO
 IS LIFETIME A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR LIFETIME ? (YES/NO)
 YES
 IS THERMAL ENERGY A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR THERMAL ENERGY ? (YES/NO)
 NO
 IS VOLUME/SIZE A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR VOLUME/SIZE ? (YES/NO)
 NO
 IS WEIGHT A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR WEIGHT ? (YES/NO)
 NO
 IS FUEL USED A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR FUEL USED ? (YES/NO)
 NO
 IS GROWTH POTENTIAL A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS GROWTH POTENTIAL A RATED PARAMETER ? (YES/NO)
 YES
 IS ENVIRON. CONSTR. A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS ENVIRON. CONSTR. A RATED PARAMETER ? (YES/NO)
 YES
 COMMAND-->END

Figure 4 - Concluded

several different scenarios could be set up. As examples, one scenario could give the student no limit on his tuition expenses while another scenario might show the student as having a \$2,000 scholarship. Therefore, the ratings on some parameters might vary from scenario to scenario. There is a maximum limit of 25 scenarios.

(b) Scenario description: the user can enter up to 60 alphanumeric characters describing each scenario.

To choose how to rate the parameters the user should specify:

(i) Paired comparison: the user can rank the parameters in importance by comparing pairs of parameters and choosing which parameter is more important than the other.

(ii) Directly rate the parameters: if the user performs the paired comparison, the program ranks the parameters and the user specifies a rating from 0 to 1. If the paired comparison is not chosen, the parameters are not ranked, and the user just rates the parameter from 0 to 1. For UPPP (using the ADD command) the user can add additional scenarios to the data file UPP. The only difference between this command and the command CRTE is that the user specifies the number of scenarios to be added. See Figure 5 for an example of the generator selection problem using UPPP with the commands CRTE and LIST.

(4) System selection model program (SSMP)

To perform the decision process, the user is required to specify the user's choice of either entering parameter ratings or using the parameter ratings that are set up in the scenarios in the UPP data file. SSMP asks the user, "Do you wish to enter your own ratings?" If yes, the

 USER'S PREFERENCE PACKAGE PROGRAM

NOTE: THE COMMANDS CRTE & ADD CAN BE ISSUED ONLY ONCE DURING
 A SESSION & YOU CANNOT ISSUE A LIST BEFORE YOU CREATE THAT FILE

COMMAND-->CRTE
 ENTER NO. OF SCENARIOS
 3
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 1
 AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 2
 AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 3
 AUX BACK-UP PWR FOR CRIT. FACILITIES REMOTE
 DO YOU WISH TO SKIP THE PAIRED COMPARSION ? (YES/NO)
 YES

AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS
 ACQUISITION COST
 .89
 LIFE CYCLE COST
 .7
 SYSTEM EFF.
 .65
 START UP TIME
 .5
 SHUTDOWN TIME
 .56
 RELIABILITY
 .75
 MAINT. AND OPER.
 .56
 LIFETIME
 .45
 THERMAL ENERGY
 .2
 VOLUME/SIZE
 .3
 WEIGHT
 .3
 FUEL USED
 .8
 GROWTH POTENTIAL
 .2
 ENVIPON. CONSTR.
 .3

Figure 5 - Generator Selection Problem Using
 UPPP With Command CRTE

AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS

ACQUISITION COST	.9
LIFE CYCLE COST	.87
SYSTEM EFF.	.76
START UP TIME	.6
SHUTDOWN TIME	.6
RELIABILITY	.8
MAINT. AND OPER.	.8
LIFETIME	.5
THERMAL ENERGY	.2
VOLUME/SIZE	.6
WEIGHT	.6
FUEL USED	.7
GROWTH POTENTIAL	.2
ENVIRON. CONSTR.	.1

Figure 5 - Continued

AUX BACK-UP PWR FOR CRIT. FACILITIES REMOTE

ACQUISITION COST	.9
LIFE CYCLE COST	.9
SYSTEM EFF.	.5
START UP TIME	.7
SHUTDOWN TIME	.5
RELIABILITY	.7
MAINT. AND OPER.	.7
LIFETIME	.6
THERMAL ENERGY	.2
VOLUME/SIZE	.7
WEIGHT	.8
FUEL USED	.8
GROWTH POTENTIAL	.3
ENVIRON. CONSTR.	.1
COMMAND-->LIST	

Figure 5 - Concluded

user enters a scenario description with a limit of 60 alphanumeric characters and parameter ratings ranging from 0 to 1. If no, the program prints out the available scenarios and the user chooses a scenario.

The user then chooses the number of parameters to be included in the decision process.

(a) Parameter choices: the user enters the integer number associated with the chosen parameters. The integer numbers must be entered from the lowest to the highest values.

(b) Parameter limits: the user can specify a limit on the parameters so that any systems or subsystems that do not satisfy this limit are eliminated from the decision process.

(i) If the parameter is numerical, the user can specify a numerical limit of greater than, lesser than, or both.

(ii) If the parameter is rated, the user can specify a rating that the parameter data has to satisfy or exceed.

(iii) If the parameter is qualifying, the user cannot specify a limit because the parameter either has that quality or does not.

(c) Choice of output: the user can choose between tabular, graphic, or both.

(i) If no graphic output is desired, the user can execute the program again.

(ii) If graphic output is chosen, the user can choose the standard graphic output or the time versus system rating graph, depending on whether or not time is a subsystem.

See Figure 6 for an example of the generator selection problem using SSMP.

.....
SYSTEM SELECTION MODEL PROGRAM
.....

LIST OF AVAILABLE PARAMETERS

- 1 -- ACQUISITION COST
- 2 -- LIFE CYCLE COST
- 3 -- SYSTEM EFF.
- 4 -- START UP TIME
- 5 -- SHUTDOWN TIME
- 6 -- RELIABILITY
- 7 -- MAINT. AND OPER.
- 8 -- LIFETIME
- 9 -- THERMAL ENERGY
- 10 -- VOLUME/SIZE
- 11 -- WEIGHT
- 12 -- FUEL USED
- 13 -- GROWTH PCTENTIAL
- 14 -- ENVIRON. CONSTR.

DO YOU WISH TO ENTER YOUR OWN RATINGS ? (YES/NO)
NO

SCENARIOS AVAILABLE

- 1 -- AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS
- 2 -- AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
- 3 -- AUX BACK-UP PWR FOR CRIT. FACILITIES REMOTE

ENTER NO OF CHOICE OF SCENARIO
2

Figure 6 - Generator Selection Problem Using SSMP

AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS

PARAMETERS		RATINGS
ACQUISITION COST		.900
LIFE CYCLE COST		.870
SYSTEM EFF.		.760
START UP TIME		.600
SHUTDOWN TIME		.600
RELIABILITY		.900
MAINT. AND OPER.		.900
LIFETIME		.500
THERMAL ENERGY		.200
VOLUME/SIZE		.600
WEIGHT		.600
FUEL USED		.700
GROWTH POTENTIAL		.200
ENVIRON. CONSTR.		.100

ENTER THE NUMBER OF PARAMETERS TO BE INCLUDED IN THE DECISION PROCESS
14

ENTER EACH INTEGER NO. ASSOCIATED WITH EACH CHOSEN
PARAMETER (FROM LOWEST TO HIGHEST)

PARAMETER NO. --1
PARAMETER NO. --2
PARAMETER NO. --3
PARAMETER NO. --4
PARAMETER NO. --5
PARAMETER NO. --6
PARAMETER NO. --7
PARAMETER NO. --8
PARAMETER NO. --9
PARAMETER NO. --10
PARAMETER NO. --11
PARAMETER NO. --12
PARAMETER NO. --13
PARAMETER NO. --14

Figure 6 - Continued

PARAMETER -- ACQUISITION COST
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 YES
 DO YOU WANT THE PARAMETER TO BE ABOVE, BELOW
 OR IN BETWEEN THE SPECIFIED LIMIT?
 (CHOOSE 1, 2, OR 3 RESPECTIVELY)
 2
 ACQUISITION COST<15000

PARAMETER -- LIFE CYCLE COST
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 YES
 DO YOU WANT THE PARAMETER TO BE ABOVE, BELOW
 OR IN BETWEEN THE SPECIFIED LIMIT?
 (CHOOSE 1, 2, OR 3 RESPECTIVELY)
 2
 LIFE CYCLE COST <20000

PARAMETER -- SYSTEM EFF.
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 YES
 DO YOU WANT THE PARAMETER TO BE ABOVE, BELOW
 OR IN BETWEEN THE SPECIFIED LIMIT?
 (CHOOSE 1, 2, OR 3 RESPECTIVELY)
 1
 SYSTEM EFF. >60

PARAMETER -- START UP TIME
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 YES
 DO YOU WANT THE PARAMETER TO BE ABOVE, BELOW
 OR IN BETWEEN THE SPECIFIED LIMIT?
 (CHOOSE 1, 2, OR 3 RESPECTIVELY)
 2
 START UP TIME <30
 START UP TIME >5

PARAMETER -- SHUTDOWN TIME
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 NO

PARAMETER -- RELIABILITY
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 NO

PARAMETER -- MAINT. AND OPER.
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 YES
 DO YOU WANT THE PARAMETER TO BE ABOVE, BELOW
 OR IN BETWEEN THE SPECIFIED LIMIT?
 (CHOOSE 1, 2, OR 3 RESPECTIVELY)
 2
 MAINT. AND OPER.<5000

Figure 6 - Continued

PARAMETER -- LIFETIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- THERMAL ENERGY
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- VOLUME/SIZE
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- WEIGHT
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- FUEL USED
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- GROWTH POTENTIAL
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
YES
THIS PARAMETER CAN BE RATED AS EXCELLENT, GOOD, FAIR, POOR, OR VERY POOR,
CHOOSE 1,2,3,4, OR 5, RESP.,
2

PARAMETER -- ENVIRON. CONSTR.
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
YES
THIS PARAMETER CAN BE RATED AS EXCELLENT, GOOD, FAIR, POOR, OR VERY POOR,
CHOOSE 1,2,3,4, OR 5, RESP.,
2

Figure 6 - Continued

 GENERATOR SELECT CASE 3 ELIMINATION TABLE
 AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 1

ELIMINATION TABLE

AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS

GENERATORS	KILOWATTS	YEAR	REASONS FOR ELIMINATION
GAS TURBINE	250 KW	1980	ACQUISITION COST > 15000.00
GAS TURBINE	250 KW	1980	LIFE CYCLE COST > 20000.00
GAS TURBINE	250 KW	1980	START UP TIME < 5.000
			CP
GAS TURBINE	250 KW	1980	START UP TIME > 30.000
GAS TURBINE	250 KW	1980	MAINT. AND OPER. > 5000.00
GAS TURBINE	250 KW	1980	GROWTH POTENTIAL < 6000
GAS TURBINE	250 KW	1980	ENVIRON. CONST. < FAIR

 GENERATOR SELECT CASE 3 ELIMINATION TABLE
 AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 2

GAS TURBINE	250 KW	2000	ACQUISITION COST > 15000.00
GAS TURBINE	250 KW	2000	LIFE CYCLE COST > 20000.00
GAS TURBINE	250 KW	2000	START UP TIME < 5.000
			CP
GAS TURBINE	250 KW	2000	START UP TIME > 30.000
GAS TURBINE	250 KW	2000	MAINT. AND OPER. > 5000.00
GAS TURBINE	250 KW	2000	GROWTH POTENTIAL < 6000
GAS TURBINE	50 KW	1980	ENVIRON. CONST. < FAIR
GAS TURBINE	50 KW	1980	ACQUISITION COST > 15000.00
GAS TURBINE	50 KW	1980	LIFE CYCLE COST > 20000.00
GAS TURBINE	50 KW	1980	START UP TIME < 5.000
			CP
GAS TURBINE	50 KW	1980	START UP TIME > 30.000
GAS TURBINE	50 KW	1980	GROWTH POTENTIAL < 6000

Figure 6 - Continued

```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 3
*****
GAS TURBINE 50 KW 1980 ENVIRON. CONSTR. <FAIR
GAS TURBINE 50 KW 2000 ACQUISITION COST > 15000.000
GAS TURBINE 50 KW 2000 LIFE CYCLE COST > 20000.000
GAS TURBINE 50 KW 2000 START UP TIME < 5.000
OP
START UP TIME > 30.000
GAS TURBINE 50 KW 2000 GROWTH POTENTIAL <GOOD
GAS TURBINE 50 KW 2000 ENVIRON. CONSTR. <FAIR
GAS TURBINE 10 KW 1980 START UP TIME < 5.000
CP
START UP TIME > 30.000
GAS TURBINE 10 KW 1980 GROWTH POTENTIAL <GOOD
GAS TURBINE 10 KW 1980 ENVIRON. CONSTR. <FAIR
GAS TURBINE 10 KW 2000 START UP TIME < 5.000
CP
START UP TIME > 30.000

```

```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 4
*****
GAS TURBINE 10 KW 2000 GROWTH POTENTIAL <GOOD
GAS TURBINE 10 KW 2000 ENVIRON. CONSTR. <FAIR
DIESEL 250 KW 1980 LIFE CYCLE COST > 20000.000
DIESEL 250 KW 1980 SYSTEM EFF. < 60.000
DIESEL 250 KW 1980 START UP TIME < 5.000
CP
START UP TIME > 30.000
DIESEL 250 KW 1980 GROWTH POTENTIAL <GOOD
DIESEL 250 KW 2000 ACQUISITION COST > 15000.000
DIESEL 250 KW 2000 LIFE CYCLE COST > 20000.000
DIESEL 250 KW 2000 START UP TIME < 5.000
CP
START UP TIME > 30.000
DIESEL 250 KW 2000 MAINT. AND OPER. > 5000.000

```

Figure 6 - Continued


```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE PAGE 5
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
*****
DIESEL 250 KW 2000 GROWTH POTENTIAL <6000
DIESEL 50 KW 1980 SYSTEM EFF. < 60.000
DIESEL 50 KW 1980 START UP TIME < 5.000
OR
START UP TIME > 30.000
DIESEL 50 KW 1980 GROWTH POTENTIAL <6000
DIESEL 50 KW 2000 ACQUISITION COST > 15000.000
DIESEL 50 KW 2000 START UP TIME < 5.000
OR
START UP TIME > 30.000
DIESEL 50 KW 2000 MAINT. AND OPER. > 5000.000
DIESEL 50 KW 2000 GROWTH POTENTIAL <6000
DIESEL 10 KW 1980 SYSTEM EFF. < 60.000
DIESEL 10 KW 1980 START UP TIME < 5.000
OR
START UP TIME > 30.000

```

```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE PAGE 6
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
*****
DIESEL 10 KW 1980 GROWTH POTENTIAL <6000
DIESEL 10 KW 2000 START UP TIME < 5.000
OR
START UP TIME > 30.000
DIESEL 10 KW 2000 GROWTH POTENTIAL <6000
DIESEL 250 KW 1980 SYSTEM EFF. < 60.000
FUEL CELL 250 KW 1980 START UP TIME < 5.000
OR
START UP TIME > 30.000
DIESEL 250 KW 2000 ACQUISITION COST > 15000.000
DIESEL 250 KW 2000 LIFE CYCLE COST > 20000.000
FUEL CELL 250 KW 2000 START UP TIME < 5.000
OR
START UP TIME > 30.000

```

Figure 6 - Continued

```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 7
*****
DIESEL 250 KW 2000 MAINT. AND OPER. > 5000.000
DIESEL 50 KW 1980 SYSTEM EFF. < 60.000
FUEL CELL 50 KW 1980 START UP TIME < 5.000
CP
START UP TIME > 30.000
ACQUISITION COST > 15000.000
DIESEL 50 KW 2000 ACQUISITION COST > 15000.000
DIESEL 10 KW 1980 SYSTEM EFF. < 60.000
STIRLING ENGINE 250 KW 1980 START UP TIME < 5.000
STIRLING ENGINE 250 KW 1980 CP
START UP TIME > 30.000
STIRLING ENGINE 250 KW 1980 GROWTH POTENTIAL < 6000
STIRLING ENGINE 250 KW 2000 ACQUISITION COST > 15000.000
STIRLING ENGINE 250 KW 2000 LIFE CYCLE COST > 20000.000

```

```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 8
*****
STIRLING ENGINE 250 KW 2000 START UP TIME < 5.000
CP
START UP TIME > 30.000
STIRLING ENGINE 250 KW 2000 GROWTH POTENTIAL < 6000
STIRLING ENGINE 50 KW 1980 SYSTEM EFF. < 60.000
STIRLING ENGINE 50 KW 1980 START UP TIME < 5.000
CP
START UP TIME > 30.000
STIRLING ENGINE 50 KW 1980 GROWTH POTENTIAL < 6000
STIRLING ENGINE 50 KW 2000 ACQUISITION COST > 15000.000
STIRLING ENGINE 50 KW 2000 SYSTEM EFF. < 60.000
STIRLING ENGINE 50 KW 2000 START UP TIME < 5.000
CP
START UP TIME > 30.000

```

Figure 6 - Continued

```

*****
GENERATOR SELECT CASE 3 ELIMINATION TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 9
*****
STIRLING ENGINE 50 KW 2000 GROWTH PCTENTIAL <6000
STIRLING ENGINE 10 KW 1980 SYSTEM EFF. < 60.000
STIRLING ENGINE 10 KW 1980 START UP TIME < 5.000
                                OP
                                START UP TIME > 30.000
STIRLING ENGINE 10 KW 1980 GROWTH PCTENTIAL <6000
STIRLING ENGINE 10 KW 2000 SYSTEM EFF. < 60.000
STIRLING ENGINE 10 KW 2000 START UP TIME < 5.000
                                OP
                                START UP TIME > 30.000
STIRLING ENGINE 10 KW 2000 GROWTH PCTENTIAL <6000

DO YOU WISH TO SEE TABULAR OUTPUT ? (YES/NO)
YES

```

Figure 6 - Continued

```

.....
GENERATOR SELECT CASE 3 UTILITY VALUES TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 1
.....

```

UTILITY VALUES FOR GENERATOR SELECT

GENERATORS

- A -- GAS TURBINE
- B -- DIESEL
- C -- FUEL CELL
- D -- STIRLING ENGINE

```

*** 250 Kw ***
-----

```

1980

```

ALL SYSTEMS HAVE BEEN ELIMINATED
2000
-----

```

ALL SYSTEMS HAVE BEEN ELIMINATED

```

*** 50 Kw ***
-----

```

Figure 6 - Continued

```

*****
GENERATOR SELECT CASE 3 UTILITY VALUES TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS PAGE 2
*****

```

1980

ALL SYSTEMS HAVE BEEN ELIMINATED
2000

ALL SYSTEMS HAVE BEEN ELIMINATED

*** 10 KW ***

1980

ALL SYSTEMS HAVE BEEN ELIMINATED
2000

A	B	C	D	OPTIMUM GENERATORS
0.000	0.000	1.000	0.000	C

DO YOU WISH TO EXECUTE THE PROGRAM AGAIN ? (YES/NO)
YES

Figure 6 - Continued

.....
SYSTEM SELECTION MODEL PROGRAM
.....

LIST OF AVAILABLE PARAMETERS

- 1 -- ACQUISITION COST
- 2 -- LIFE CYCLE COST
- 3 -- SYSTEM EFF.
- 4 -- START UP TIME
- 5 -- SHUTDOWN TIME
- 6 -- RELIABILITY
- 7 -- MAINT. AND OPER.
- 8 -- LIFETIME
- 9 -- THERMAL ENERGY
- 10 -- VOLUME/SIZE
- 11 -- WEIGHT
- 12 -- FUEL USED
- 13 -- GROWTH POTENTIAL
- 14 -- ENVIRON. CONSTR.

DO YOU WISH TO ENTER YOUR OWN RATINGS ? (YES/NO)
NO

SCENARIOS AVAILABLE

- 1 -- AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS
- 2 -- AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
- 3 -- AUX BACK-UP PWR FOR CRIT. FACILITIES REMOTE

ENTER NO OF CHOICE OF SCENARIO
1

Figure 6 - Continued

AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS

PARAMETERS	RATINGS
ACQUISITION COST	.290
LIFE CYCLE COST	.700
SYSTEM EFF.	.650
START UP TIME	.500
SHUTDOWN TIME	.560
RELIABILITY	.750
MAINT. AND OPER.	.560
LIFETIME	.450
THERMAL ENERGY	.200
VOLUME/SIZE	.300
WEIGHT	.300
FUEL USED	.800
GROWTH POTENTIAL	.200
ENVIRON. CONSTR.	.300

ENTER THE NUMBER OF PARAMETERS TO BE INCLUDED IN THE DECISION PROCESS

ENTER EACH INTEGER NO. ASSOCIATED WITH EACH CHOSEN
PARAMETER (FROM LOWEST TO HIGHEST)

PARAMETER NO. --1
PARAMETER NO. --2
PARAMETER NO. --3
PARAMETER NO. --4
PARAMETER NO. --5
PARAMETER NO. --6
PARAMETER NO. --7
PARAMETER NO. --8
PARAMETER NO. --9
PARAMETER NO. --10
PARAMETER NO. --11
PARAMETER NO. --12
PARAMETER NO. --13
PARAMETER NO. --14

Figure 6 - Continued

PARAMETER -- ACQUISITION COST
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- LIFE CYCLE COST
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- SYSTEM EFF.
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- START UP TIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- SHUTDOWN TIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- RELIABILITY
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- MAINT. AND OPER.
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- LIFETIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- THERMAL ENERGY
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- VOLUME/SIZE
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- WEIGHT
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

Figure 6 - Continued

PARAMETER -- FUEL USED
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- GROWTH POTENTIAL
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- ENVIRON. CONSTR.
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

DO YOU WISH TO SEE TABULAR OUTPUT ? (YES/NO)
YES

Figure 6 - Continued

 GENERATOR SELECT CASE 3 UTILITY VALUES TABLE
 AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS PAGE 1

UTILITY VALUES FOR GENERATOR SELECT

GENERATORS

A -- GAS TURBINE
 B -- DIESEL
 C -- FUEL CELL
 D -- STIRLING ENGINE

*** 250 KW ***

 1980

A	B	C	D	OPTIMUM GENERATORS
.832	.119	.070	.031	A

 GENERATOR SELECT CASE 3 UTILITY VALUES TABLE
 AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS PAGE 2

2000

A	B	C	D	OPTIMUM GENERATORS
.669	.526	.548	.810	D

*** 50 KW ***

 1980

A	B	C	D	OPTIMUM GENERATORS
.832	.119	.070	.031	A

Figure 6 - Continued

```

*****
GENERATOR SELECT CASE 3 UTILITY VALUES TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS PAGE 3
*****
2000
-----

```

A	B	C	D	OPTIMUM GENERATORS
.643	.447	.647	.744	C

```

*** 10 KW ***
-----
1980
-----

```

A	B	C	D	OPTIMUM GENERATORS
.754	.066	.544	.031	A

```

*****
GENERATOR SELECT CASE 3 UTILITY VALUES TABLE
AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS PAGE 4
*****
2000
-----

```

A	B	C	D	OPTIMUM GENERATORS
.542	.415	.566	.715	D

```

DO YOU WISH TO EXECUTE THE PROGRAM AGAIN ? (YES/NO)
NO

```

Figure 6 - Concluded

b. Output

(1) Parameter package program (PPF)

Using the command LIST, the program will print out a listing of parameter data for each system and subsystem. See Figure 7 for an example of output of the generator selection problem.

(2) User's preference package program (UPPP)

Using the command LIST, the program will print out a listing of parameter rating values for each scenario plus the average parameter rating for each parameter. See Figure 8 for an example of output of the generator selection problem.

(3) System selection model program (SSMP)

This program will print out:

(a) Elimination table: if the parameter limits eliminate data for some system or subsystem, an elimination table is printed. This table prints out the eliminated system and/or subsystem and the reason why it was eliminated.

(b) Utility value table: the program prints out a table showing the relative values of each system. The highest value indicates which system is the best or optimal system. Care has to be taken when specifying limits on the parameter, or all of the systems will be eliminated. If this condition occurs, the program will print out a message so indicating.

(c) Graphics: the graphs show the same information as the utility value table. The graphics plot the system value for each system. See Figure 9 for examples of output using the generator selection problem.

```

*****
*                                     *
*               PARAMETER PACKAGE PROGRAM               *
*                                     *
*****

```

COMMANDS AVAILABLE:

```

CRTE - CREATES FILE OF NAMES
MNIN - ALLOWS MANUAL ENTERING OF PARAMETER VALUES
LIST - LISTS OUT NAMES AND PARAMETER VALUES
END - ENDS PROGRAM

```

COMMAND-->LIST

ACQUISITION COST			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
90800.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
90800.000	162500.000	125000.000	
STIRLING ENGINE			
100000.000			

Figure 7 - Parameter Package Program Output

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
20600.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
20600.000	35000.000	25000.000	
STIRLING ENGINE			
22500.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
6800.000	0.000	20000.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
6800.000	8000.000	5000.000	
STIRLING ENGINE			
3500.000			

Figure 7 - Continued

LIFE CYCLE COST			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
120200.000	64700.000	0.000	
	STIRLING ENGINE		
	0.000		
2000			
GAS TURBINE	DIESEL	FUEL CELL	
114350.000	64700.000	38262.500	
	STIRLING ENGINE		
	56000.000		

Figure 7 - Continued

50 KW			

1980			

GAS TURBINE	DIESEL	FUEL CELL	
32300.000	16100.000	0.000	
STIRLING ENGINE			
0.000			

2000			

GAS TURBINE	DIESEL	FUEL CELL	
26825.000	16100.000	7837.500	
STIRLING ENGINE			
12900.000			

10 KW			

1980			

GAS TURBINE	DIESEL	FUEL CELL	
7900.000	5500.000	2880.000	
STIRLING ENGINE			
0.000			

2000			

GAS TURBINE	DIESEL	FUEL CELL	
6475.000	5500.000	1548.750	
STIRLING ENGINE			
3810.000			

Figure 7 - Continued

SYSTEM EFF.			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
82.500	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
72.250	63.000	60.000	
STIRLING ENGINE			
60.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
86.500	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
75.750	65.000	60.000	
STIRLING ENGINE			
55.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
87.500	0.000	65.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
77.750	67.000	60.000	
STIRLING ENGINE			
55.000			

Figure 7 - Continued

START UP TIME			
250 KW			
1980			
GAS TURBINE	.580	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	.580	DIESEL	.170
		FUEL CELL	90.000
		STIRLING ENGINE	1.000

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	1.000	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	1.000	DIESEL	.170
		FUEL CELL	30.000
		STIRLING ENGINE	1.000

10 KW			
1980			
GAS TURBINE	1.000	DIESEL	0.000
		FUEL CELL	15.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	1.000	DIESEL	.170
		FUEL CELL	15.000
		STIRLING ENGINE	1.000

Figure 7 - Continued

SHUTDOWN TIME			
250 KW			
1980			
GAS TURBINE	.500	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	.500	DIESEL	.170
		FUEL CELL	45.000
		STIRLING ENGINE	1.000

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	1.000	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	1.000	DIESEL	.170
		FUEL CELL	15.000
		STIRLING ENGINE	1.000

10 KW			
1980			
GAS TURBINE	1.000	DIESEL	0.000
		FUEL CELL	7.800
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	1.000	DIESEL	.170
		FUEL CELL	7.800
		STIRLING ENGINE	1.000

Figure 7 - Continued

RELIABILITY			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	10.000	1.000	
STIRLING ENGINE			
8.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	10.000	1.000	
STIRLING ENGINE			
8.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	10.000	1.000	
STIRLING ENGINE			
8.000			

Figure 7 - Continued

MAINT. AND OPER.			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
6205.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
6205.000	8300.000	6250.000	
STIRLING ENGINE			
3094.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
4715.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
4715.000	5200.000	1000.000	
STIRLING ENGINE			
1125.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
3700.000	0.000	1000.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
3700.000	3400.000	250.000	
STIRLING ENGINE			
160.000			

Figure 7 - Continued

LIFETIME			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
11.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
11.000	22.000	0.000	
STIRLING ENGINE			
25.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
11.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
11.000	13.000	0.000	
STIRLING ENGINE			
25.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
11.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
11.000	9.000	0.000	
STIRLING ENGINE			
25.000			

Figure 7 - Continued

THERMAL ENERGY			
250 KW			
1980			
GAS TURBINE	7.590	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	8.690	DIESEL	3.100
		FUEL CELL	9.200
		STIRLING ENGINE	3.700

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	.340	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	.700	DIESEL	8.500
		FUEL CELL	8.400
		STIRLING ENGINE	8.900

10 KW			
1980			
GAS TURBINE	.860	DIESEL	0.000
		FUEL CELL	0.000
		STIRLING ENGINE	0.000
2000			
GAS TURBINE	.940	DIESEL	6.700
		FUEL CELL	6.100
		STIRLING ENGINE	7.800

Figure 7 - Continued

VOLUME/SIZE			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
150.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
150.000	180.000	170.000	
STIRLING ENGINE			
150.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
47.500	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
47.500	70.000	30.000	
STIRLING ENGINE			
30.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
12.500	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
12.500	21.000	6.000	
STIRLING ENGINE			
12.000			

Figure 7 - Continued

WEIGHT			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
4500.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
4500.000	5538.000	12500.000	
STIRLING ENGINE			
4800.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
1000.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
1000.000	1895.000	4000.000	
STIRLING ENGINE			
600.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
200.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
200.000	483.000	400.000	
STIRLING ENGINE			
220.000			

Figure 7 - Continued

FUEL USED			
250 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
210.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
135.000	70000.000	40000.000	
STIRLING ENGINE			
67000.000			

Figure 7 - Continued

50 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
55.000	0.000	0.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
30.250	14900.000	8140.000	
STIRLING ENGINE			
12600.000			

10 KW			
1980			
GAS TURBINE	DIESEL	FUEL CELL	
12.000	0.000	4260.000	
STIRLING ENGINE			
0.000			
2000			
GAS TURBINE	DIESEL	FUEL CELL	
6.420	3100.000	3760.000	
STIRLING ENGINE			
2540.000			

Figure 7 - Continued

GROWTH POTENTIAL		
250 KW		
1980		
GAS TURBINE POOR	DIESEL POOR	FUEL CELL GOOD
STIRLING ENGINE POOR		
2000		
GAS TURBINE POOR	DIESEL POOR	FUEL CELL GOOD
STIRLING ENGINE POOR		

Figure 7 - Continued

50 KW		
1980		
GAS TURBINE POOR	DIESEL POOR	FUEL CELL GOOD
STIRLING ENGINE POOR		
2000		
GAS TURBINE POOR	DIESEL POOR	FUEL CELL GOOD
STIRLING ENGINE POOR		

10 KW		
1980		
GAS TURBINE POOR	DIESEL POOR	FUEL CELL GOOD
STIRLING ENGINE POOR		
2000		
GAS TURBINE POOR	DIESEL POOR	FUEL CELL GOOD
STIRLING ENGINE POOR		

Figure 7 - Continued

ENVIRON. CONSTR.		
250 KW		
1980		
GAS TURBINE VERY POOR	DIESEL FAIR	FUEL CELL EXCELLENT
STIRLING ENGINE GOOD		
2000		
GAS TURBINE VERY POOR	DIESEL FAIR	FUEL CELL EXCELLENT
STIRLING ENGINE GOOD		

Figure 7 - Continued

50 KW		
1980		
GAS TURBINE VERY POOR	DIESEL FAIR	FUEL CELL EXCELLENT
STIRLING ENGINE GOOD		
2000		
GAS TURBINE VERY POOR	DIESEL FAIR	FUEL CELL EXCELLENT
STIRLING ENGINE GOOD		

10 KW		
1980		
GAS TURBINE VERY POOR	DIESEL FAIR	FUEL CELL EXCELLENT
STIRLING ENGINE GOOD		
2000		
GAS TURBINE VERY POOR	DIESEL FAIR	FUEL CELL EXCELLENT
STIRLING ENGINE GOOD		

COMMAND-->END

Figure 7 - Concluded


```

*****
GENERATOR SELECT      CASE 3  USER'S PREFERENCE PACKAGE
AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS
*****

```

PAGE 1

ACQUISITION COST	.890
LIFE CYCLE COST	.700
SYSTEM EFF.	.650
START UP TIME	.500
SHUTDOWN TIME	.560
RELIABILITY	.750
MAINT. AND OPEP.	.560
LIFETIME	.450
THERMAL ENERGY	.200
VOLUME/SIZE	.300
WEIGHT	.300
FUEL USED	.800
GROWTH POTENTIAL	.200
ENVIRON. CONSTP.	.300

```

*****
GENERATOR SELECT      CASE 3  USER'S PREFERENCE PACKAGE
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
*****

```

PAGE 2

ACQUISITION COST	.900
LIFE CYCLE COST	.870
SYSTEM EFF.	.760
START UP TIME	.600
SHUTDOWN TIME	.600
RELIABILITY	.800
MAINT. AND OPER.	.800
LIFETIME	.500
THERMAL ENERGY	.200
VOLUME/SIZE	.600
WEIGHT	.600
FUEL USED	.700
GROWTH POTENTIAL	.200
ENVIRON. CONSTR.	.100

Figure 8 - Output From The UPPP Program

```

*****
GENERATOR SELECT CASE 3 USER'S PREFERENCE PACKAGE
AUX BACK-UP PWR FOR CRIT. FACILITIES REMOTE PAGE 3
*****

```

ACQUISITION COST	.900
LIFE CYCLE COST	.900
SYSTEM EFF.	.500
START UP TIME	.700
SHUTDOWN TIME	.500
RELIABILITY	.700
MAINT. AND OPEP.	.700
LIFETIME	.600
THERMAL ENERGY	.200
VOLUME/SIZE	.700
WEIGHT	.800
FUEL USED	.600
GROWTH POTENTIAL	.300
ENVIRON. CONSTR.	.100

```

*****
GENERATOR SELECT CASE 3 USER'S PREFERENCE PACKAGE
AVERAGE VALUES FOR EACH PARAMETER PAGE 4
*****

```

ACQUISITION COST	.897
LIFE CYCLE COST	.823
SYSTEM EFF.	.637
START UP TIME	.600
SHUTDOWN TIME	.553
RELIABILITY	.750
MAINT. AND OPER.	.687
LIFETIME	.517
THERMAL ENERGY	.200
VOLUME/SIZE	.533
WEIGHT	.567
FUEL USED	.767
GROWTH POTENTIAL	.233
ENVIRON. CONSTR.	.167

COMMAND-->END

Figure 8 - Concluded

POWER SIZE : 10 MW 1 HOUR
 YEAR : 1990

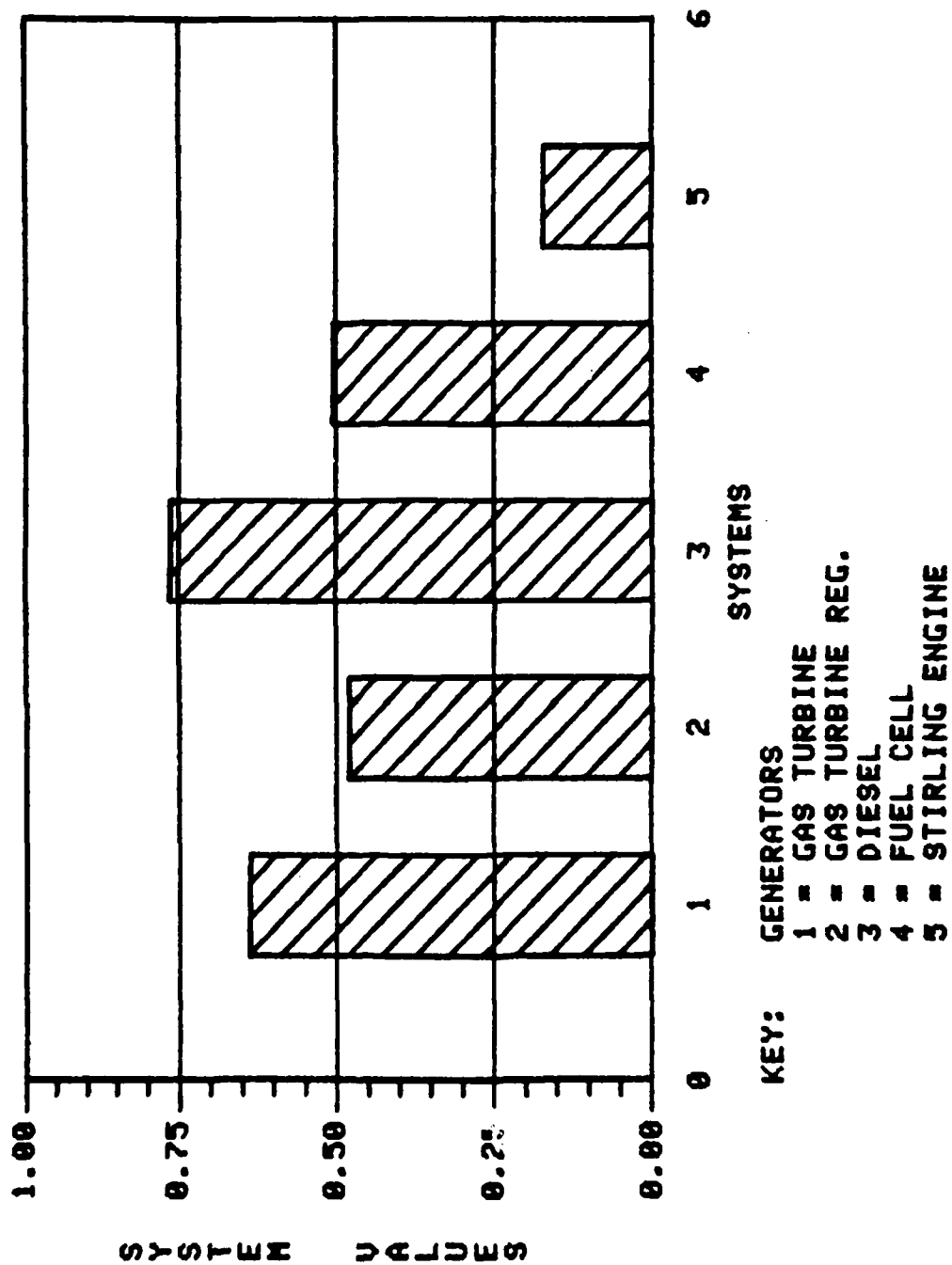


Figure 9. Graphics Output For Generator Selection Problem

3. PROCEDURES FOR USE

To execute the programs interactively, a PROCEDURE file has been set up so that the user has only one command to issue in order to run the programs.

A PROCEDURE file is a file containing system commands. The first line is the header statement, and it defines the file as a PROCEDURE file and identifies any keywords. When a PROCEDURE file is called up, the values for the keywords are substituted for any keywords specified in the file. The procedure is called by a BEGIN command. The BEGIN command states the procedure name as on the header statement, the name of the PROCEDURE file, and, if any, values for the keywords.

In the PROCEDURE files, file names are given as keywords. Therefore, different file names can be specified when running different problems. Different PROCEDURE files are available when the programs are used for different functions because the handling of the files differs from function to function. When the programs are used to create the data files, the PROCEDURE file issues the necessary commands to set up the new files as permanent files. The user then issues the CATALOG command to make the files permanent. If the data files are already present, another PROCEDURE file is used. Listings of these PROCEDURE files are available in Appendix J. Further information of PROCEDURE files with the CDC CYBER 175 is available in NOS/BE Version 1: Reference Manual.

On Wright-Patterson's CDC computer the following command should be issued to gain access to INTERCOM (interactive mode): LOGIN, problem number, password. Then the following commands should be issued for the various programs:

a. Parameter package program (PPP)

(1) To create the parameter package (PP) and the HEADER file:

ATTACH,PPROCl

BEGIN,PPPP,PPROCl,filename1,filename2

where: filename1 is the desired permanent filename for the HEADER file to be created, and filename2 is the similar name for the PP file.

NOTE: No filenames discussed in this section may be exactly the names "HEADER", "PP", or "UPP".

(2) PPP program commands:

CRTE--creates a new data base, or in other words, a new HEADER file

MNIN--allows entering of parameter values, if user has no data file already made up

LIST--lists all parameter data contained in PP

END --ends the program

b. User's preference package program (UPPP)

(1) To create the user's preference package (UPP) file:

ATTACH,UPROCl

filename3 BEGIN,UPPPP,UPROCl,filename1,cyl,filename2,

where: filename1 and cyl are the filename and cycle number of the HEADER file, which must already exist as a permanent file and must not be attached, and filename2 and filename3 are the desired filenames of the UPP file and the new, annotated HEADER file, both of which are to be created.

(2) UPPP program commands:

CRTE--creates new user's preference package (UPP)

ADD --adds scenarios to an already-existing UPP

LIST--lists the scenarios and their parameter ratings

END --ends the program

(3) The new HEADER file and the UPP file created must be CATALOGed as above.

(4) To add scenarios to a previously created UPP:

ATTACH,UPROC2

BEGIN,UPPPP,UPROC2,filename1,cyl,filename2,cy2,
filename3,filename4

where: filename1,cyl and filename2,cy2 represent the filenames and cycle numbers of the existing HEADER and UPP files, and filename3 and filename4 are the new filenames for the appended versions of these files.

c. System selection model program (SSMP)

(1) To run program:

ATTACH,IPROC

BEGIN,SSMPP,IPROC,filename1,cyl,filename2,cy2,
filename3,cy3

where: filename1,cyl pertain to the PP file,
filename2,cy2 to the UPP file,
and filename3,cy3 to the HEADER file

To run larger problems, all these programs may be executed in batch mode. This is particularly desirable when entering data interactively becomes a long, cumbersome task, and for problems larger than the size limits specified above, SSMP must be executed in batch mode. Instructions and further explanation are contained in Appendix B.

SECTION V

PROGRAMMING SPECIFICATIONS

1. PROGRAM OVERVIEW

The main program is the system selection model program (SSMP) which is used to select the best system by integrating the system parameters and the user's parameter ratings. The SSMP has three data bases: (1) parameter package (PP), which contains the system parameter data; (2) user's preference package (UPP), which contains the user's ratings of the system parameters; and (3), HEADER which contains information on the structure of the problem and titles.

Two supporting programs develop these data bases. The parameter package program (PPP) develops the PP and HEADER files. The user's preference package program (UPPP) uses the information from HEADER as input and outputs UPP as a file. See Figure 8.

To set up a problem the PPP and UPP are executed in order to provide the data bases for the SSMP. Once these data bases are set up, the system selection model program can be executed as many times as needed. It is not necessary to execute the two supporting programs each time the SSMP is run.

Each of these programs can be executed interactively on Wright-Patterson's CDC CYBER 175. The SSMP consists of two programs, the first of which contains everything but the graphics. The second set of programs, the graphics programs GRAPHX and GRAPHT, contain the FORTRAN codings to utilize the Tektronix PLOT-10 software subroutines which actually perform the graphics work.

The user of the SSMP cannot tell that there are two separate programs because the programs are executed together under a PROCEDURE file. This PROCEDURE file not only executes the two programs but also attaches and returns data files. The reason for this separation is two-fold: first, each SSMP program requires so much memory allocation that it is impossible to have the programs run interactively together as one program; and second, the graphics programs are presented as an option.

To further reduce the amount of storage required, the three programs for SSMP are divided into overlays. The program modules are structured into overlay trees as shown in Figures 10 and 11. The main overlay which is designated as level (0,0) remains in memory during job execution. Beneath the main overlay are the primary overlays, which are called from the main overlay. The primary levels are, for example, (1,0), (2,0), (3,0)... Under a primary overlay, a hierarchy of secondary overlays can exist. The secondary overlays are loaded in response to a CALL statement issued in the primary overlay. Examples of secondary overlays associated with the primary overlay (1,0) are: (1,1), (1,2), (1,3), and (1,4).

The array dimensions are kept consistent throughout SSMP and the supporting programs. The dimensions for each array are set at the maximum value that will allow the program to run interactively. The dimensions of these arrays therefore limit the sizes of the problems that can be run interactively. To run larger problems will require the user to increase array dimensions and run the program in batch mode. The program that can handle larger problems is called SSMPB and is described in Appendix B. Of course, the dimensions may change from computer system to computer system.

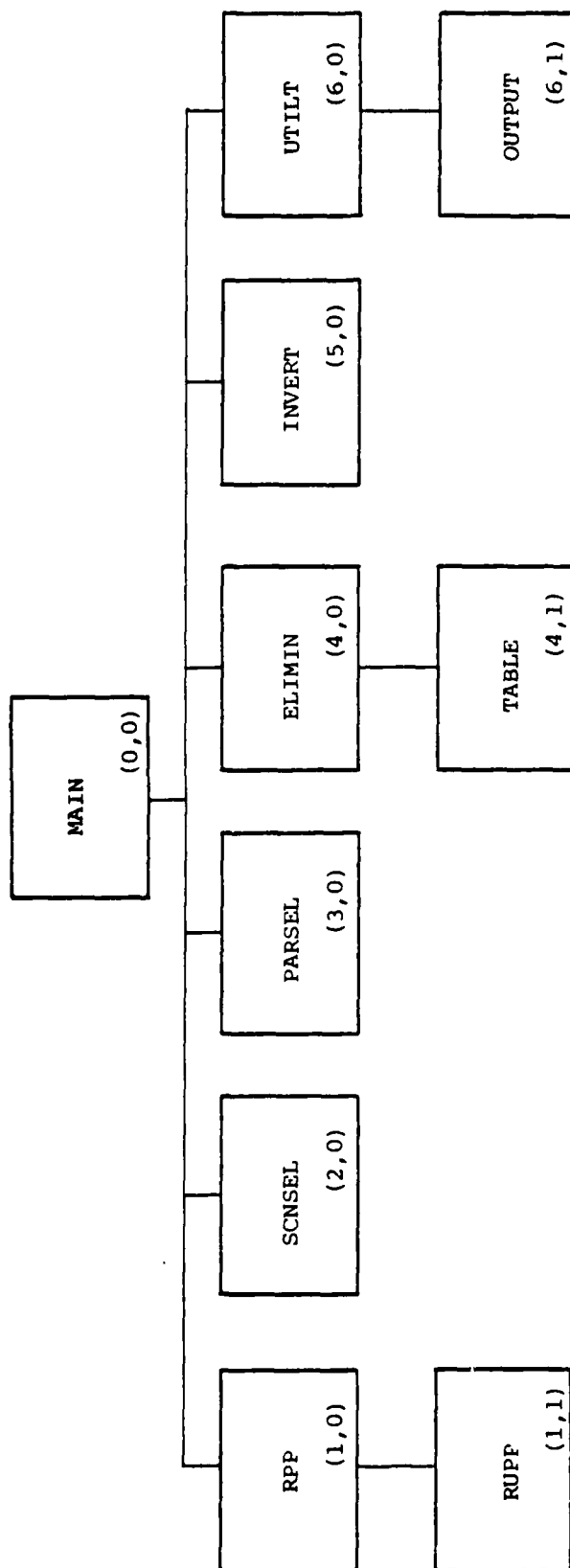


Figure 10. Overlay Tree for SSMP

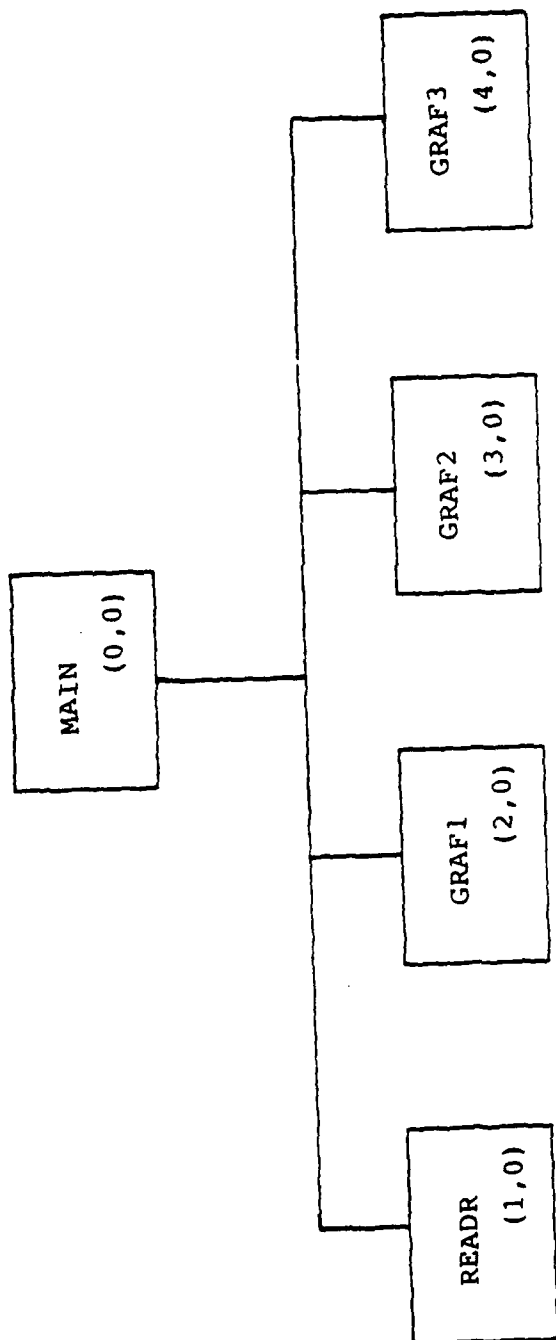


Figure 11. Overlay Tree for GRAPHX

A user on a different computer system may be able to set up these programs as a library. To facilitate this potential setup, the COMMON statements and variable names are kept the same between programs.

Section II suggests methods to change the overlay program modules to subroutines.

Besides the three data files discussed above, other data files exist which are used to reduce the number of arrays. These files are created by SSMP during its execution. These files are VPARM and SYSNUM. The interaction of SSMP and its files is shown in Figure 12. The structure and description of the data in these files are given in Section V.2.

2. DATA STRUCTURE TABLES

This section describes the data files used in the system selection model program. Examples of the first three data files for the generator selection problem are given in Figures 17, 18, and 19.

a. HEADER

HEADER is an output file from the supporting program PPP and an input file to UPPP and SSMP.

Figure 13 shows the structure of this file.

b. User's preference package (UPP)

UPP is an output file of the user's preference package program (UPPP) and an input file to SSMP.

The file is structured in Figure 14.

c. Parameter package (PP)

PP is an output file from the parameter package program and an input file to SSMP.

The file is structured in Figure 15.

d. VPARM

VPARM is a file created by SSMP and used to store data during the execution of SSMP.

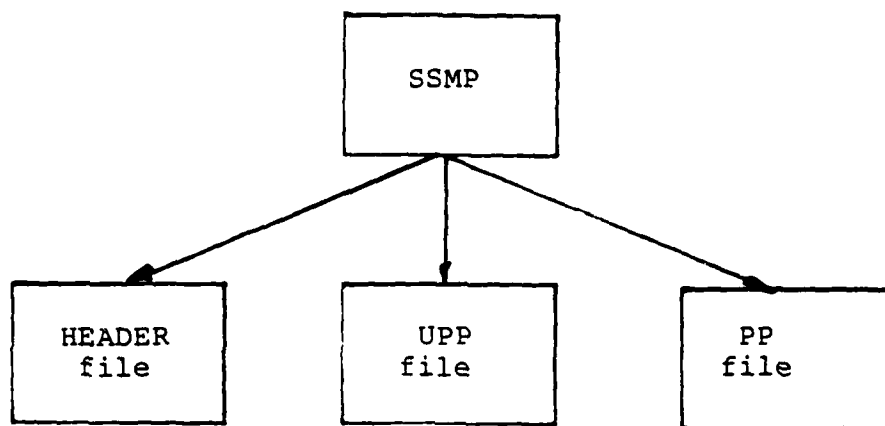


Figure 12. SSMP's Interaction with its Data File

```

SYSDSN
NSYS
NSUB1
NSUB2
NPARM
SYSSET(4)
SB1SET(4)
SB2SET(4)
SYSNAM(NSYS,4)
SUB1NM(NSUB1,4)
SUB2NM(NSUB2,4)
PARNAM(NPARM,4), NUM(NPARM), RATED(NPARM), YN(NPARM)
INVERT(NPARM)
NSCEN
SCNDSC(NSCEN,15)

```

Description:

SYSDSN-INTEGER VALUE: indicates the description of the problem

- 1--Indicates systems to be included.
- 2--Indicates systems and level 1 subsystems to be included.
- 3--Indicates systems, level 1 and level 2 subsystems to be included.

FORMAT (I1)

NSYS-INTEGER VALUE: number of systems.

FORMAT (I2)

NSUB1-INTEGER VALUE: number of level 1 subsystems.

FORMAT (I2)

NSUB2-INTEGER VALUE: number of level 2 subsystems per level 1 subsystem.

FORMAT (I2)

NPARM-INTEGER VALUE: number of parameters.

FORMAT (I2)

SYSSET-ALPHANUMERIC: general setname for the systems.

FORMAT (4A4)

Figure 13 - Data Structure of HEADER File

AD-A119 160

SOUTHEASTERN CENTER FOR ELECTRICAL ENGINEERING EDUCAT--ETC F/G 9/2
ADVANCED TECHNOLOGY MULTIPLE CRITERIA DECISION MODEL.(U)

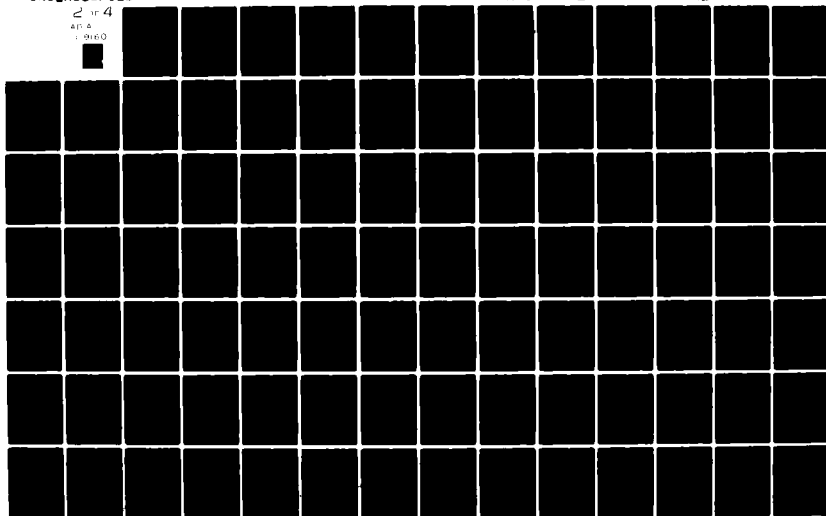
NOV 81 P J SWEENEY, K B BERNER, J R FRAKER F33615-77-C-2059

UNCLASSIFIED

AFWAL-TR-81-2112

NL

2 of 4
415 6
1 0160



SB1SET-ALPHANUMERIC: general setname for the level 1
subsystems.

FORMAT (4A4)

SB2SET-ALPHANUMERIC: general setname for the level 2
subsystems.

FORMAT (4A4)

Figure 13 - Concluded

RATING (1)

RATING (2)

.

.

.

.

.

RATING (NPARM)

A similar block appears in the file for each scenario in the problem.

Description:

RATING-REAL VALUE: A rating for each parameter, entered by the user in the execution of the UPPP program, which indicates the relative importance of each of the parameters.

FORMAT (F5.3)

Figure 14. Data Structure of UPP File


```

PARAM(1,1), PARAM(1,2).....PARAM(1,NSYS)
PARAM(2,1), PARAM(2,2).....PARAM(2,NSYS)
      .
      .
      .
      .
      .
PARAM(INDEX,1), PARAM(INDEX,2).....PARAM(INDEX,NSYS)

```

This block appears in the data file once for each parameter existing in the problem.

Description:

PARAM-REAL VALUE: contains the actual parameter data.

UNFORMATTED

NSYS-INTEGER VALUE: number of systems.

FORMAT (I2)

INDEX-INTEGER VALUE: Case 1--INDEX = 1
Case 2--INDEX = NSUB1
Case 3--INDEX = NSUB1*NSUB2

Figure 15. Data Structure of PP File

```

SYSNRM(1,1), SYSNRM(1,2).....SYSNRM(1,NSYS)
SYSNRM(2,1), SYSNRM(2,2).....SYSNRM(2,NSYS)
      .
      .
      .
      .
SYSNRM(INDEX,1), SYSNRM(INDEX,2).....SYSNRM(INDEX,NSYS)

```

Description:

SYSNRM-REAL VALUE: normalized system utility values
indicating ranking of systems.

NSYS-INTEGER VALUE: number of systems.

FORMAT (I2)

INDEX-INTEGER VALUE: Case 1--INDEX = 1
Case 2--INDEX = NSUB1
Case 3--INDEX = NSUB1*NSUB2

Figure 16. Data Structure of SYSNUM File

```

3
4
3
2
14
GENERATORS
KILOWATTS
YEAR
GAS TURBINE
DIESEL
FUEL CELL
STIRLING ENGINE
250 KW
50 KW
10 KW
1980
2000
1980
2000
1980
2000
ACQUISITION COST 1000
LIFE CYCLE COST 1000
SYSTEM EFF. 1001
START UP TIME 1000
SHUTDOWN TIME 1000
RELIABILITY 1001
MAINT. AND OPER. 1000
LIFETIME 1001
THERMAL ENERGY 1000
VOLUME/SIZE 1000
WEIGHT 1000
FUEL USED 1000
GROWTH POTENTIAL 0100
ENVIRON. CONSTR. 0100
3
AUX BACK-UP PWR FOR CRIT. FACILITIES CONUS
AUX BACK-UP PWR FOR CRIT. FACILITIES OVERSEAS
AUX BACK-UP PWR FOR CRIT. FACILITIES REMOTE

```

Figure 17. HEADER File for the Generator Problem

.890
.700
.650
.500
.560
.750
.560
.450
.200
.300
.300
.800
.200
.300
.900
.870
.760
.600
.600
.600
.800
.800
.500
.200
.600
.600
.700
.200
.100
.900
.900
.500
.700
.500
.700
.700
.600
.200
.700
.800
.800
.300
.100

Figure 18. UPP File for Generator Problem

90800. 0. 0. 0.
 90800. 162500. 125000. 100000.
 23600. 0. 0. 0.
 23600. 35000. 25000. 22500.
 6800. 0. 20000. 0.
 6800. 8000. 5000. 3500.
 120200. 64700. 0. 0.
 114350. 64700. 38262.5 56000.
 32300. 16100. 0. 0.
 26825. 16100. 7837.5 12900.
 7900. 5500. 2830. 0.
 6475. 5500. 1548.75 3810.
 82.5 0. 0. 0.
 72.25 63. 60. 60.
 86.5 0. 0. 0.
 75.75 65. 60. 55.
 97.5 0. 65. 0.
 77.75 67. 60. 55.
 .58 0. 0. 0.
 .58 .17 90. 1.
 1. 0. 0. 0.
 1. .17 30. 1.
 1. 0. 15. 0.
 1. .17 15. 1.
 .58 0. 0. 0.
 .58 .17 45. 1.
 1. 0. 0. 0.
 1. .17 15. 1.
 1. 0. 7.8 0.
 1. .17 7.8 1.
 12. 0. 0. 0.
 12. 10. 1. 8.
 12. 0. 0. 0.
 12. 10. 1. 8.
 12. 0. 0. 0.
 12. 10. 1. 8.
 6205. 0. 0. 0.
 6205. 8300. 6250. 3094.
 4715. 0. 0. 0.
 4715. 5200. 1000. 1125.
 3700. 0. 1000. 0.
 3700. 3400. 250. 150.
 11. 0. 0. 0.
 11. 22. 0. 25.
 11. 0. 0. 0.

Figure 19 - PP File for Generator Problem

11. 13. 0. 25.
 11. 0. 0. 0.
 11. 9. 0. 25.
 7.59 0. 0. 0.
 8.69 3.1 9.2 3.7
 .34 0. 0. 0.
 .7 8.5 8.4 8.9
 .86 0. 0. 0.
 .94 6.7 5.1 7.8
 150. 0. 0. 0.
 150. 180. 170. 150.
 47.5 0. 0. 0.
 47.5 70. 32. 30.
 12.5 0. 0. 0.
 12.5 21. 6. 12.
 4500. 0. 0. 0.
 4500. 5538. 12500. 4800.
 1000. 0. 0. 0.
 1000. 1895. 4000. 600.
 200. 0. 0. 0.
 200. 483. 400. 220.
 210. 0. 0. 0.
 135. 70000. 40000. 67000.
 55. 0. 0. 0.
 30.25 14900. 8140. 12600.
 12. 0. 4260. 0.
 6.42 3100. 3760. 2540.
 4. 4. 2. 4.
 4. 4. 2. 4.
 4. 4. 2. 4.
 4. 4. 2. 4.
 4. 4. 2. 4.
 4. 4. 2. 4.
 5. 3. 1. 2.
 5. 3. 1. 2.
 5. 3. 1. 2.
 5. 3. 1. 2.
 5. 3. 1. 2.
 5. 3. 1. 2.

Figure 19 - Concluded

SECTION VI

TEST/VERIFICATION SPECIFICATIONS

1. PRODUCTION TESTING

During the coding of these programs, input data were used to run through each branch of the programs. Input used to violate program conditions included:

- a. Empty input files.
- b. Extraneous characters in data.
- c. User input and/or file input with values which were too large or too small.
- d. An improperly constructed problem tree.
- e. Problem size too large for interactive executive.

2. ACCEPTANCE TEST SPECIFICATIONS

The programs will be considered suitable when:

- a. The programs respond correctly to the program violations as described in Section VI.1.
- b. Error-free performance is observed in several test cases.
- c. The correct answers are found for test data. These test data are shown in Section VI.3.

3. TEST CASES

To show that SSMP performs its calculations correctly, several test runs were made, and the results were confirmed to be accurate. Figure 20 is a test case that depicts an actual run and shows not only the final results but also the interactive commands with typical responses.

```

*****
*                                     *
*               PARAMETER PACKAGE PROGRAM               *
*                                     *
*****
COMMANDS AVAILABLE:
  CRTE - CREATES FILE OF NAMES
  MAIN - ALLOWS MANUAL ENTERING OF PARAMETER VALUES
  LIST - LISTS OUT NAMES AND PARAMETER VALUES
  END - ENDS PROGRAM

```

COMMAND-->CRTE

```

ENTER DATA BASE NAME (MAX 16 CHARS)
REPORT EXAMPLE
ENTER NO. OF SYSTEMS (2 DIGIT NO.)
05
ENTER NO. OF LEVEL 1 SUBSYSTEMS (2 DIGIT NO.)
02
ENTER NO. OF LEVEL 2 SUBSYSTEMS (2 DIGIT NO.)
02
ENTER NO. OF PARAMETERS (2 DIGIT NO.)
05
ENTER SYSTEM SETNAME (MAX. 16 CHARS.)
COLLEGES
ENTER NAME FOR COLLEGES          NO. 1 (MAX. 16 CHARS)
YALE
ENTER NAME FOR COLLEGES          NO. 2 (MAX. 16 CHARS)
U.C.
ENTER NAME FOR COLLEGES          NO. 3 (MAX. 16 CHARS)
HARVARD
ENTER NAME FOR COLLEGES          NO. 4 (MAX. 16 CHARS)
U.S.C.
ENTER NAME FOR COLLEGES          NO. 5 (MAX. 16 CHARS)
SEMPRELY

```

```

ENTER NAME FOR PARAMETER NO. 1
COST
ENTER NAME FOR PARAMETER NO. 2
DISTANCE
ENTER NAME FOR PARAMETER NO. 3
HOUSING
ENTER NAME FOR PARAMETER NO. 4
TEACHERS
ENTER NAME FOR PARAMETER NO. 5
CO-ED LIVING?

```

Figure 20 - College Run

Copy available to DTIC does not
 permit fully legible reproduction
 102

IS COST A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR COST ? (YES/NO)
 NO
 IS DISTANCE A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR DISTANCE ? (YES/NO)
 NO
 IS HOUSING A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS HOUSING A RATED PARAMETER ? (YES/NO)
 YES
 IS TEACHERS A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS TEACHERS A RATED PARAMETER ? (YES/NO)
 YES
 IS CO-ED LIVING? A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS CO-ED LIVING? A RATED PARAMETER ? (YES/NO)
 NO
 IS CO-ED LIVING? A QUALITATIVE PARAMETER ? (YES/NO)
 YES
 COMMAND-->MNIN

Figure 20 - Continued

COST

THIS IS A NUMERIC PARAMETER - ENTER A NUMBER

YALE	--	7500
U.C.	--	6000
HARVARD	--	6000
U.S.C.	--	5500
BERKELEY	--	7000
DISTANCE		

THIS IS A NUMERIC PARAMETER - ENTER A NUMBER

YALE	--	400
U.C.	--	20
HARVARD	--	300
U.S.C.	--	560
BERKELEY	--	600
HOUSING		

THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR

YALE	--	2
U.C.	--	2
HARVARD	--	1
U.S.C.	--	3
BERKELEY	--	2

TEACHERS

THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR

YALE	--	1
U.C.	--	2
HARVARD	--	2
U.S.C.	--	3
BERKELEY	--	1
CO-ED LIVING?		

THIS IS A QUALITATIVE PARAMETER - ENTER 1 FOR YES, 0 FOR NO

YALE	--	0
U.C.	--	0
HARVARD	--	0
U.S.C.	--	1
BERKELEY	--	1
COMMAND-->LIST		

Figure 20 - Continued

COST				
YALE	7500.000	U.C.	6000.000	HARVARD
				8000.000
	U.S.C.	5500.000	BERKELY	7000.000

DISTANCE				
YALE	400.000	U.C.	20.000	HARVARD
				300.000
	U.S.C.	500.000	BERKELY	600.000

HOUSING				
YALE	GOOD	U.C.	GOOD	HARVARD
				EXCELLENT
	U.S.C.	FAIR	BERKELY	GOOD

Figure 20 - Continued

TEACHERS		
YALE	U.C.	HARVARD
EXCELLENT	GOOD	GOOD
U.S.C.	BERKELEY	
FAIR	EXCELLENT	

CO-ED LIVING?		
YALE	U.C.	HARVARD
NO	NO	NO
U.S.C.	BERKELEY	
YES	YES	

COMMAND-->END

Figure 20 - Continued

.....
USER'S PREFERENCE PACKAGE PROGRAM
.....

NOTE: THE COMMANDS CRTE & ADD CAN BE ISSUED ONLY ONCE DURING
A SESSION & YOU CANNOT ISSUE A LIST BEFORE YOU CREATE THAT FILE

COMMAND-->CRTE
ENTER NO. OF SCENARIOS
02
ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 1
SUPER STUDENT WITH FINANCIAL NEED
ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 2
STUDENT WHO LOOKS AT COLLEGE FOR FUN
DO YOU WISH TO SKIP THE PAIRED COMPARSION ? (YES/NO)
YES

SUPER STUDENT WITH FINANCIAL NEED
COST
.880
DISTANCE
.450
HOUSING
.440
TEACHERS
.890
CO-ED LIVING?
.100

STUDENT WHO LOOKS AT COLLEGE FOR FUN
COST
.340
DISTANCE
.450
HOUSING
.880
TEACHERS
.100
CO-ED LIVING?
.990
COMMAND-->LIST

Figure 20 - Continued

```

*****
COLLEGE EXAMPLE      CASE 1  USER'S PREFERENCE PACKAGE
SUPER STUDENT WITH FINANCIAL NEED
*****

```

PAGE 1

```

COST                .880
DISTANCE            .450
HOUSING             .440
TEACHERS            .890
CO-ED LIVING?      .100

```

```

*****
COLLEGE EXAMPLE      CASE 1  USER'S PREFERENCE PACKAGE
STUDENT WHO LOOKS AT COLLEGE FOR FUN
*****

```

PAGE 2

```

COST                .340
DISTANCE            .450
HOUSING             .880
TEACHERS            .100
CO-ED LIVING?      .990

```

```

*****
COLLEGE EXAMPLE      CASE 1  USER'S PREFERENCE PACKAGE
AVERAGE VALUES FOR EACH PARAMETER
*****

```

PAGE 3

```

COST                .610
DISTANCE            .450
HOUSING             .660
TEACHERS            .495
CO-ED LIVING?      .545

```

COMMAND-->END

Figure 20 - Continued

.....
 SYSTEM SELECTION MODEL PROGRAM

LIST OF AVAILABLE PARAMETERS

- 1 -- COST
- 2 -- DISTANCE
- 3 -- HOUSING
- 4 -- TEACHERS
- 5 -- CO-ED LIVING?

DO YOU WISH TO ENTER YOUR OWN RATINGS ? (YES/NO)
 NO

SCENARIOS AVAILABLE

- 1 -- SUPER STUDENT WITH FINANCIAL NEED
- 2 -- STUDENT WHO LOOKS AT COLLEGE FOR FUN

ENTER NO OF CHOICE OF SCENARIO
 1

SUPER STUDENT WITH FINANCIAL NEED

.....	
PARAMETERS	RATINGS
.....	
COST	.880
DISTANCE	.450
HOUSING	.450
TEACHERS	.890
CO-ED LIVING?	.100
.....	

Figure 20 - Continued

ENTER THE NUMBER OF PARAMETERS TO BE INCLUDED IN THE DECISION PROCESS

ENTER EACH INTEGER NO. ASSOCIATED WITH EACH CHOSEN
PARAMETER (FROM LOWEST TO HIGHEST)

PARAMETER NO. -- 1
PARAMETER NO. -- 2
PARAMETER NO. -- 3
PARAMETER NO. -- 4
PARAMETER NO. -- 5

PARAMETER -- COST
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- DISTANCE
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- HOUSING
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- TEACHERS
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- CO-ED LIVING?
A QUALITATIVE PARAMETER--CANNOT SPECIFY A RANGE

Figure 20 - Continued


```

*****
COLLEGE EXAMPLE      CASE 1  ELIMINATION TABLE
SUPER STUDENT WITH FINANCIAL NEED
*****

```

PAGE 1

ELIMINATION TABLE

SUPER STUDENT WITH FINANCIAL NEED

COLLEGES	REASONS FOR ELIMINATION
YALE	CO-ED LIVING?
U.D.	CO-ED LIVING?
HARVARD	CO-ED LIVING?

DO YOU WISH TO SEE TABULAR OUTPUT ? (YES/NO)
YES

```

*****
COLLEGE EXAMPLE      CASE 1  UTILITY VALUES TABLE
SUPER STUDENT WITH FINANCIAL NEED
*****

```

PAGE 1

UTILITY VALUES FOR COLLEGE EXAMPLE

COLLEGES					
A	B	C	D	E	OPTIMUM COLLEGES
0.000	0.000	0.000	.516	.520	E

DO YOU WISH TO EXECUTE THE PROGRAM AGAIN ? (YES/NO)
NO

Figure 20 - Concluded

APPENDIX A - PROGRAM ANALYSIS

The SSMP user selects the parameters and, if desired, parameter limits and parameter ratings. This information is used to determine the best system by mathematically evaluating the parameter data which have been weighed by the parameter ratings. The user can specify parameter data limits. If a system and/or subsystem do not satisfy these parameter limits, that system and/or subsystem are eliminated from the decision process.

The program module ELIMIN evaluates the user's parameter choices and limits and eliminates unwanted parameter data and systems and/or subsystems that do not satisfy parameter limits.

The data base PP (parameter package) is set up so that SSMP evaluates the data one parameter at a time and all of the data for the systems or subsystems for that parameter at the same time.

The program module sets up two arrays, ELIM and MATRIX. The array ELIM stores the parameter that eliminated the system or subsystem. This array is used to print out the reasons for elimination of a system or subsystem and is detailed in the elimination table. The array MATRIX stores a zero when a parameter does not satisfy a limit or if a parameter is not chosen to participate in the decision process. If a system and/or subsystem is eliminated, ELIMIN puts zeros in the array MATRIX for that system and/or subsystem parameter data. The SSMP considers zeros in the parameter package (PP) as blanks and does not do any calculations or evaluations for those data.

The program module ELIMIN then calls up TABLE, which prints out the elimination table. Next, the program module INVRT is called to invert any parameter data that do not show the best value as the lowest value. If a parameter value has

been indicated as inverted, INVRT reads in that parameter's data. For each line of systems or subsystems, INVRT finds the largest value. The inverted parameter data are determined by the following equation:

$$\text{inverted parameter} = \frac{\text{largest value}}{\text{parameter data}}$$

If the parameter data do not need inverting, they are written out onto the data file VPARM untouched.

SSMP calls the program module UTILT, which determines the best system. The UTILT module first reads in the parameter data from the data file VPARM. Next, it eliminates any parameter data for any eliminated system and/or subsystems by multiplying the PARAM array by the MATRIX array. The zeros in the MATRIX indicate the eliminated systems and/or subsystems. The UTILT module finds the smallest and largest value for each line of parameter data and calculates the utility of each system parameter by the following equation:

$$U_a = \frac{\text{largest value} - \text{current value}}{\text{largest value} - \text{smallest value}}$$

For example, if the parameter values under consideration are 9, 7, 3, 2, 1, and 5, 9 would be the largest value and 1 the smallest. If the system being evaluated has the parameter rating of 3, this would be the current value, and the utility of the parameter for the system under consideration would equal 6/8, or 0.75. This utility value of a specified system is based upon an individual parameter value relative to the parameter ratings of all systems under consideration. If the largest value equals the smallest value, the utility is assigned unity.

To calculate the overall utility for each system, the utility values for each parameter are multiplied by their

respective preference ratings, obtained from the UPP file,
and the resultant products are summed:

$$U_{\text{sys}} = K_1 U_1 + K_2 U_2 + \dots + K_n U_n$$

K_a = User's preference rating for each parameter

U_a = Utility value for each parameter

SSMP prints out the results in the utility table.

APPENDIX B - SYSTEM SELECTION MODEL IN BATCH MODE

When a particular problem is too large to be run interactively with the programs described earlier, the problem can be solved by running the programs in batch mode. The batch process submits the job to a system batch queue so that it may be run independently of the user's control. The batch process allows the user to specify whether the job is to be printed, punched, or sent to a remote terminal.

When a program is run interactively the user must input data from the terminal as the program is executing. Batch mode input differs from interactive in that the input data are sent in the batch file, along with the job control cards that execute the program. This makes it possible for the user to perform other tasks while the program is executing.

Setting up a program to run in batch mode is a simple process. The user should take the following steps in preparing and entering the batch file:

- a. Determine the dimensions needed for the arrays to allow the program to process the particular size problem. The reader will recall that array dimensions are set to maximum interactive memory limits. A list of the arrays and their dimensions appears in Section II, Table 2. Make the appropriate changes in the source files and recompile the source. For the decision model use SYSSELB and SSMPB as the source file and object file, respectively.

- b. The user should next begin to set up the batch file itself. The first card of the file should be the job card. The job card parameters consist of the user's box number, the estimated CPU and I/O times in seconds, and the estimated requirements for core memory in words. The last three parameters must be coded in octal, or base eight, numerals. The specification of these particular parameters is recommended to increase system efficiency and improve job

turnaround time. Typical values might be T10, IO50, CM70000, meaning 8 seconds cpu time, 40 seconds I/O time, and 28672 words of memory (all values given in base 10). The next thing to appear on the job card is a period, followed by comments to the operator so that your job may be verified. These consist of the problem number, address and phone number in case a problem should arise. The next cards will ATTACH the data files needed to run the program, and then the object file for the program itself must be ATTACHED. Following the ATTACH cards will be a card containing the object file name followed by a period, used to execute the file. Next in the file appears a *EOR, which separates the job control statements from the data to follow the *EOR. The next set of cards hold the data, which consist of all the responses normally typed by the user during interactive execution. The user should run the program a few times interactively to make certain no responses are left out of the batch file data section. After the data cards are finished, the file is ended with *EOR and *EOF cards.

The following is a review of the basic form of the file:

Jobname, Tnn, IOmm, CMxxxxx.problem#, address, phone.

ATTACH, data filename.

ATTACH, data filename.

ATTACH, object filename.

Object filename.

*EOR

All input data, one datum per line.

⋮

*EOR

*EOF

This file should be saved and made permanent.

(c) The next step is to enter the file into the batch queue. This is accomplished with the BATCH command. Two options exist as to where output from the batch job will be sent, and the user should decide at this point whether or not he wishes hardcopy output that must be obtained at the computer facility. If the user wishes to obtain a hardcopy of his output, the following command should be issued;

BATCH, batch filename,INPUT.

If the user wishes to obtain his results at a remote terminal, the following command is issued:

BATCH, batch filename,INPUT,HERE.

When the user issues either of these commands, he has entered his program in batch. The program will remain in the queue until it is executed. If the user has chosen the second form, results may be examined by using the FIND command. Format for the FIND command is as follows:

FIND,jobid.

where jobid consists of enough characters of the jobname (listed on the jobcard of the batch file) to identify it uniquely. See the NOS/BE 1 Reference Manual for more details.

The file will first appear in the input queue, identified by the five-character jobname plus two system-generated characters, then will enter the execute queue, and eventually will be listed as residing in the output queue, indicating the task is completed. The user may examine his output by first making the file local with another form of the BATCH command:

BATCH,output filename,LOCAL.

and by using either the PAGE function or the EDITOR program.

APPENDIX C - EXAMPLES

These examples are provided to demonstrate to the user the capabilities of SSMP and the supporting programs' options. Included are several examples which demonstrate exactly how the problem is entered into the model and which provide some insight as to how the structure of a problem is chosen. One example, the generator selection, is an example of a problem which lends itself to a Case 3 type breakdown. The horse example, however, is a good example of a decision which could not logically be broken down further than a Case 1 problem.

Note that, in all the examples, the SSMP may be executed as many times as is desired in order to see how different limitations on the decision process affect the final outcome.


```

*****
*               PARAMETER PACKAGE PROGRAM               *
*****
COMMANDS AVAILABLE:
  CRTE - CREATES FILE OF NAMES
  MNIN - ALLOWS MANUAL ENTERING OF PARAMETER VALUES
  LIST - LISTS OUT NAMES AND PARAMETER VALUES
  END - ENDS PROGRAM
COMMAND-->CRTE

```

```

ENTER DATA BASE NAME (MAX 16 CHARS)
REPORT EXAMPLE
ENTER NO. OF SYSTEMS (2 DIGIT NO.)
04
ENTER NO. OF LEVEL 1 SUBSYSTEMS (2 DIGIT NO.)
00
ENTER NO. OF LEVEL 2 SUBSYSTEMS (2 DIGIT NO.)
00
ENTER NO. OF PARAMETERS (2 DIGIT NO.)
14
ENTER SYSTEM SETNAME (MAX. 16 CHARS.)
GENERATORS
ENTER NAME FOR GENERATORS      NO. 1 (MAX. 16 CHARS)
GAS TURBINE
ENTER NAME FOR GENERATORS      NO. 2 (MAX. 16 CHARS)
DIESEL
ENTER NAME FOR GENERATORS      NO. 3 (MAX. 16 CHARS)
FUEL CELL
ENTER NAME FOR GENERATORS      NO. 4 (MAX. 16 CHARS)
STIRLING ENGINE

```

ENTER NAME FOR PARAMETER NO. 1
ACQUISITION COST
ENTER NAME FOR PARAMETER NO. 2
LIFE CYCLE COST
ENTER NAME FOR PARAMETER NO. 3
SYSTEM EFF.
ENTER NAME FOR PARAMETER NO. 4
START UP TIME
ENTER NAME FOR PARAMETER NO. 5
SHUTDOWN TIME
ENTER NAME FOR PARAMETER NO. 6
RELIABILITY
ENTER NAME FOR PARAMETER NO. 7
MAINT. AND OPER.
ENTER NAME FOR PARAMETER NO. 8
LIFETIME
ENTER NAME FOR PARAMETER NO. 9
THERMAL ENERGY
ENTER NAME FOR PARAMETER NO.10
VOLUME/SIZE
ENTER NAME FOR PARAMETER NO.11
WEIGHT
ENTER NAME FOR PARAMETER NO.12
FUEL USED
ENTER NAME FOR PARAMETER NO.13
GROWTH POTENTIAL
ENTER NAME FOR PARAMETER NO.14
ENVIRON.CONSTR.

IS ACQUISITION COST A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR ACQUISITION COST ? (YES/NO)
 NO
 IS LIFE CYCLE COST A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR LIFE CYCLE COST ? (YES/NO)
 NO
 IS SYSTEM EFF. A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS SYSTEM EFF. A RATED PARAMETER ? (YES/NO)
 YES
 IS START UP TIME A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR START UP TIME ? (YES/NO)
 NO
 IS SHUTDOWN TIME A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR SHUTDOWN TIME ? (YES/NO)
 NO
 IS RELIABILITY A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS RELIABILITY A RATED PARAMETER ? (YES/NO)
 YES
 IS MAINT. AND OPER. A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS MAINT. AND OPER. A RATED PARAMETER ? (YES/NO)
 YES

IS LIFETIME A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR LIFETIME ? (YES/NO)
 NO
 IS THERMAL ENERGY A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS THERMAL ENERGY A RATED PARAMETER ? (YES/NO)
 YES
 IS VOLUME/SIZE A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR VOLUME/SIZE ? (YES/NO)
 NO
 IS WEIGHT A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR WEIGHT ? (YES/NO)
 NO
 IS FUEL USED A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR FUEL USED ? (YES/NO)
 NO
 IS GROWTH POTENTIAL A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS GROWTH POTENTIAL A RATED PARAMETER ? (YES/NO)
 YES
 IS ENVIRON.CONSTR. A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS ENVIRON.CONSTR. A RATED PARAMETER ? (YES/NO)
 NO
 IS ENVIRON.CONSTR. A QUALITATIVE PARAMETER ? (YES/NO)
 YES
 COMMAND-->MNIN

ACQUISITION COST
 THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
 GAS TURBINE -- 2100
 DIESEL -- 2000
 FUEL CELL -- 3000
 STIRLING ENGINE -- 1200
 LIFE CYCLE COST
 THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
 GAS TURBINE -- 30000
 DIESEL -- 40000
 FUEL CELL -- 23000
 STIRLING ENGINE -- 14000
 SYSTEM EFF.
 THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
 1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR
 GAS TURBINE -- 2
 DIESEL -- 1
 FUEL CELL -- 3
 STIRLING ENGINE -- 1
 START UP TIME
 THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
 GAS TURBINE -- 24
 DIESEL -- 8
 FUEL CELL -- 48
 STIRLING ENGINE -- 72
 SHUTDOWN TIME
 THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
 GAS TURBINE -- 12
 DIESEL -- 6
 FUEL CELL -- 2
 STIRLING ENGINE -- 24

RELIABILITY
THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR
GAS TURBINE -- 1
DIESEL -- 2
FUEL CELL -- 3
STIRLING ENGINE -- 4
MAINT. AND OPER.
THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR
GAS TURBINE -- 3
DIESEL -- 2
FUEL CELL -- 2
STIRLING ENGINE -- 1
LIFETIME
THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
GAS TURBINE -- 100
DIESEL -- 50
FUEL CELL -- 23
STIRLING ENGINE -- 22
THERMAL ENERGY
THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR
GAS TURBINE -- 3
DIESEL -- 2
FUEL CELL -- 2
STIRLING ENGINE -- 1
VOLUME/SIZE
THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
GAS TURBINE -- 400
DIESEL -- 300
FUEL CELL -- 200
STIRLING ENGINE -- 450

WEIGHT
 THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
 GAS TURBINE -- 300
 DIESEL -- 340
 FUEL CELL -- 200
 STIRLING ENGINE -- 1200
 FUEL USED
 THIS IS A NUMERIC PARAMETER - ENTER A NUMBER
 GAS TURBINE -- 34
 DIESEL -- 23
 FUEL CELL -- 67
 STIRLING ENGINE -- 12
 GROWTH POTENTIAL
 THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
 1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR
 GAS TURBINE -- 1
 DIESEL -- 2
 FUEL CELL -- 3
 STIRLING ENGINE -- 4
 ENVIRON.CONSTR.
 THIS IS A QUALITATIVE PARAMETER - ENTER 1 FOR YES, 0 FOR NO
 GAS TURBINE -- 1
 DIESEL -- 1
 FUEL CELL -- 1
 STIRLING ENGINE -- 1
 COMMAND-->LIST

ACQUISITION COST		
GAS TURBINE	DIESEL	FUEL CELL
2100.000	2000.000	3000.000
STIRLING ENGINE		
1200.000		

LIFE CYCLE COST		
GAS TURBINE	DIESEL	FUEL CELL
30000.000	40000.000	23000.000
STIRLING ENGINE		
14000.000		

SYSTEM EFF.		
GAS TURBINE	DIESEL	FUEL CELL
GOOD	EXCELLENT	FAIR
STIRLING ENGINE		
EXCELLENT		


```

*****
*                                     *
*                               START UP TIME                               *
*                                     *
*   GAS TURBINE      DIESEL      FUEL CELL                               *
*       24.000              8.000      48.000                             *
*                                     *
*   STIRLING ENGINE                                         72.000         *
*                                     *
*****

```

```

*****
*                                     *
*                               SHUTDOWN TIME                               *
*                                     *
*   GAS TURBINE      DIESEL      FUEL CELL                               *
*       12.000              6.000      2.000                             *
*                                     *
*   STIRLING ENGINE                                         24.000         *
*                                     *
*****

```

```

*****
*                                     *
*                               RELIABILITY                               *
*                                     *
*   GAS TURBINE      DIESEL      FUEL CELL                               *
*   EXCELLENT              GOOD      FAIR                             *
*                                     *
*   STIRLING ENGINE                                         POOR          *
*                                     *
*****

```

```

*****
*
*                               MAINT. AND OPER.
*
*****
*   GAS TURBINE      DIESEL      FUEL CELL
*   FAIR              GOOD        GOOD
*
*   STIRLING ENGINE
*   EXCELLENT
*
*****

```

```

*****
*
*                               LIFETIME
*
*****
*   GAS TURBINE      DIESEL      FUEL CELL
*   100.000          50.000      23.000
*
*   STIRLING ENGINE
*   22.000
*
*****

```

```

*****
*
*                               THERMAL ENERGY
*
*****
*   GAS TURBINE      DIESEL      FUEL CELL
*   FAIR              GOOD        GOOD
*
*   STIRLING ENGINE
*   EXCELLENT
*
*****

```

VOLUME/SIZE			
GAS TURBINE	DIESEL	FUEL CELL	
400.000	300.000	200.000	
	STIRLING ENGINE		
	450.000		

WEIGHT			
GAS TURBINE	DIESEL	FUEL CELL	
300.000	340.000	200.000	
	STIRLING ENGINE		
	1200.000		

FUEL USED			
GAS TURBINE	DIESEL	FUEL CELL	
34.000	23.000	67.000	
	STIRLING ENGINE		
	12.000		

.....
 USER'S PREFERENCE PACKAGE PROGRAM

NOTE: THE COMMANDS CRTE & ADD CAN BE ISSUED ONLY ONCE DURING
 A SESSION & YOU CANNOT ISSUE A LIST BEFORE YOU CREATE THAT FILE

COMMAND-->CPT
 ENTER NO. OF SCENARIOS
 03
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 1
 CONUS
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 2
 OVERSEAS
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 3
 REMOTE
 DO YOU WISH TO SKIP THE PAIRED COMPARISON ? (YES/NO)
 YES

CONUS
 ACQUISITION COST
 .450
 LIFE CYCLE COST
 .340
 SYSTEM EFF.
 .560
 START UP TIME
 .780
 SHUTDOWN TIME
 .900
 RELIABILITY
 .450
 MAINT. AND OPER.
 .560
 LIFETIME
 .120
 THERMAL ENERGY
 .890
 VOLUME/SIZE
 .600
 WEIGHT
 .340
 FUEL USED
 .780
 GROWTH POTENTIAL
 .560
 ENVIRON.CONSTP.
 .340

OVERSEAS
ACQUISITION COST
.340
LIFE CYCLE COST
.450
SYSTEM EFF.
.890
START UP TIME
.990
SHUTDOWN TIME
.450
RELIABILITY
.120
MAINT. AND OPER.
.780
LIFETIME
.560
THERMAL ENERGY
.230
VOLUME/SIZE
.790
WEIGHT
.450
FUEL USED
.780
GROWTH POTENTIAL
.770
ENVIRON.CONSTP.
.500

REMOTE
ACQUISITION COST
.450
LIFE CYCLE COST
.770
SYSTEM EFF.
.220
START UP TIME
.100
SHUTDOWN TIME
.600
RELIABILITY
.450
MAINT. AND OPER.
.300
LIFETIME
.900
THERMAL ENERGY
.300
VOLUME/SIZE
.600
WEIGHT
.800
FUEL USED
.900
GROWTH POTENTIAL
.100
ENVIRON.CONSTR.
.570
COMMAND-->LIST

.....
 REPORT EXAMPLE CASE 1 USER'S PREFERENCE PACKAGE PAGE :
 CONUS

ACQUISITION COST	.450
LIFE CYCLE COST	.340
SYSTEM EFF.	.560
START UP TIME	.760
SHUTDOWN TIME	.600
RELIABILITY	.450
MAINT. AND OPER.	.560
LIFETIME	.120
THERMAL ENERGY	.990
VOLUME/SIZE	.600
WEIGHT	.340
FUEL USED	.760
GROWTH POTENTIAL	.560
ENVIRON.CONSTR.	.340

.....
 REPORT EXAMPLE CASE 1 USER'S PREFERENCE PACKAGE PAGE 2
 OVERSEAS

ACQUISITION COST	.340
LIFE CYCLE COST	.450
SYSTEM EFF.	.890
START UP TIME	.990
SHUTDOWN TIME	.450
RELIABILITY	.120
MAINT. AND OPER.	.780
LIFETIME	.560
THERMAL ENERGY	.230
VOLUME/SIZE	.790
WEIGHT	.450
FUEL USED	.780
GROWTH POTENTIAL	.770
ENVIRON.CONSTP.	.500

 REPORT EXAMPLE CASE 1 USER'S PREFERENCE PACKAGE PAGE 4
 AVERAGE VALUES FOR EACH PARAMETER

ACQUISITION COST	.413
LIFE CYCLE COST	.520
SYSTEM EFF.	.557
START UP TIME	.623
SHUTDOWN TIME	.650
RELIABILITY	.340
MAINT. AND OPER.	.547
LIFETIME	.527
THERMAL ENERGY	.473
VOLUME/SIZE	.663
WEIGHT	.530
FUEL USED	.820
GROWTH POTENTIAL	.477
ENVIRON.CONSTP.	.470

COMMAND-->END

.....
SYSTEM SELECTION MODEL PROGRAM
.....

LIST OF AVAILABLE PARAMETERS

- 1 -- ACQUISITION COST
- 2 -- LIFE CYCLE COST
- 3 -- SYSTEM EFF.
- 4 -- START UP TIME
- 5 -- SHUTDOWN TIME
- 6 -- RELIABILITY
- 7 -- MAINT. AND OPER.
- 8 -- LIFETIME
- 9 -- THERMAL ENERGY
- 10 -- VOLUME/SIZE
- 11 -- WEIGHT
- 12 -- FUEL USED
- 13 -- GROWTH POTENTIAL
- 14 -- ENVIRON.CONSTR.

DO YOU WISH TO ENTER YOUR OWN RATINGS ? (YES/NO)
NO

SCENARIOS AVAILABLE

- 1 -- CONUS
- 2 -- OVERSEAS
- 3 -- REMOTE

ENTER NO OF CHOICE OF SCENARIO
2

OVERSEAS

PARAMETERS	RATINGS
ACQUISITION COST	.400
LIFE CYCLE COST	.780
SYSTEM EFF.	.110
START UP TIME	.770
SHUTDOWN TIME	.450
RELIABILITY	.800
MAINT. AND OPER.	.780
LIFETIME	.450
THERMAL ENERGY	.560
VOLUME/SIZE	.500
WEIGHT	.780
FUEL USED	.760
GROWTH POTENTIAL	.430
ENVIRON.CONSTR.	.670

14
ENTER THE NUMBER OF PARAMETERS TO BE INCLUDED IN THE DECISION PROCESS

ENTER EACH INTEGER NO. ASSOCIATED WITH EACH CHOSEN
PARAMETER (FROM LOWEST TO HIGHEST)

PARAMETER NO. -- 1
PARAMETER NO. -- 2
PARAMETER NO. -- 3
PARAMETER NO. -- 4
PARAMETER NO. -- 5
PARAMETER NO. -- 6
PARAMETER NO. -- 7
PARAMETER NO. -- 8
PARAMETER NO. -- 9
PARAMETER NO. -- 10
PARAMETER NO. -- 11
PARAMETER NO. -- 12
PARAMETER NO. -- 13
PARAMETER NO. -- 14

PARAMETER -- ACQUISITION COST
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- LIFE CYCLE COST
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- SYSTEM EFF.
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- START UP TIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- SHUTDOWN TIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- RELIABILITY
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- MAINT. AND OPER.
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- LIFETIME
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- THERMAL ENERGY
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- VOLUME/SIZE
WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)
NO

PARAMETER -- WEIGHT
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 NO

PARAMETER -- FUEL USED
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 NO

PARAMETER -- GROWTH POTENTIAL
 WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
 HAS TO SATISFY ? (YES/NO)
 NO

PARAMETER -- ENVIRON.CONSTR.
 A QUALITATIVE PARAMETER--CANNOT SPECIFY A RANGE

DO YOU WISH TO SEE TABULAR OUTPUT ? (YES/NO)
 YES

 REPORT EXAMPLE CASE 1 UTILITY VALUES TABLE PAGE 1
 OVERSEAS

UTILITY VALUES FOR REPORT EXAMPLE

GENERATORS

A -- GAS TURBINE
 B -- DIESEL
 C -- FUEL CELL
 D -- STIRLING ENGINE
 OPTIMUM
 GENERATORS

A	B	C	D	
.605	.687	.574	.547	B

DO YOU WISH TO EXECUTE THE PROGRAM AGAIN ? (YES/NO)
 NO

```
*****
*                                     *
*               PARAMETER PACKAGE PROGRAM               *
*                                     *
*****
```

```
COMMANDS AVAILABLE:
  CRTE - CREATES FILE OF NAMES
  MNIN - ALLOWS MANUAL ENTERING OF PARAMETER VALUES
  LIST - LISTS OUT NAMES AND PARAMETER VALUES
  END - ENDS PROGRAM
```

```
COMMAND-->CRTE
```

```
ENTER DATA BASE NAME (MAX 16 CHARS)
REPORT EXAMPLE
ENTER NO. OF SYSTEMS (2 DIGIT NO.)
05
ENTER NO. OF LEVEL 1 SUBSYSTEMS (2 DIGIT NO.)
00
ENTER NO. OF LEVEL 2 SUBSYSTEMS (2 DIGIT NO.)
00
ENTER NO. OF PARAMETERS (2 DIGIT NO.)
06
ENTER SYSTEM SETNAME (MAX. 16 CHARS.)
HORSES
ENTER NAME FOR HORSES          NO. 1 (MAX. 16 CHARS)
LUCKY STRIKER
ENTER NAME FOR HORSES          NO. 2 (MAX. 16 CHARS)
MAN OF WAR
ENTER NAME FOR HORSES          NO. 3 (MAX. 16 CHARS)
ENTERPRISE
ENTER NAME FOR HORSES          NO. 4 (MAX. 16 CHARS)
THE SEEKER
ENTER NAME FOR HORSES          NO. 5 (MAX. 16 CHARS)
ALEX THE GREAT
```


ENTER NAME FOR PARAMETER NO. 1
 SPEED
 ENTER NAME FOR PARAMETER NO. 2
 GATE START
 ENTER NAME FOR PARAMETER NO. 3
 BACKGROUND
 ENTER NAME FOR PARAMETER NO. 4
 PREV. REC.
 ENTER NAME FOR PARAMETER NO. 5
 FIRST RACE?
 ENTER NAME FOR PARAMETER NO. 6
 LUCK

IS SPEED A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR SPEED ? (YES/NO)
 NO
 IS GATE START A NUMERICAL PARAMETER ? (YES/NO)
 YES
 FOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS
 CONSIDERED THE "BEST" VALUE.
 DO YOU WISH TO INVERT THIS FOR GATE START ? (YES/NO)
 NO
 IS BACKGROUND A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS BACKGROUND A RATED PARAMETER ? (YES/NO)
 YES
 IS PREV.PEC A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS PPREV.PEC A RATED PARAMETER ? (YES/NO)
 YES
 IS FIRST RACE? A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS FIRST RACE? A RATED PARAMETER ? (YES/NO)
 NO
 IS FIRST RACE? A QUALITATIVE PARAMETER ? (YES/NO)
 YES
 IS LUCK A NUMERICAL PARAMETER ? (YES/NO)
 NO
 IS LUCK A RATED PARAMETER ? (YES/NO)
 YES
 COMMAND-->MNIN

SPEED

THIS IS A NUMERIC PARAMETER - ENTER A NUMBER

LUCKY STRIKER -- 200
MAN OF WAR -- 244
ENTERPRISE -- 300
THE SEEKER -- 201
ALEX THE GREAT -- 190
GATE START

THIS IS A NUMERIC PARAMETER - ENTER A NUMBER

LUCKY STRIKER -- 2
MAN OF WAR -- 3
ENTERPRISE -- 4
THE SEEKER -- 1
ALEX THE GREAT -- 5
BACKGROUND

THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR

LUCKY STRIKER -- 2
MAN OF WAR -- 3
ENTERPRISE -- 2
THE SEEKER -- 1
ALEX THE GREAT -- 2
PREV.REC

THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR

LUCKY STRIKER -- 1
MAN OF WAR -- 2
ENTERPRISE -- 2
THE SEEKER -- 3
ALEX THE GREAT -- 1

FIRST PACE?

THIS IS A QUALITATIVE PARAMETER - ENTER 1 FOR YES, 0 FOR NO

LUCKY STRIKER -- 1
MAN OF WAR -- 1
ENTERPRISE -- 1
THE SEEKER -- 1
ALEX THE GREAT -- 1
LUCK

THIS IS A RATED PARAMETER - RATE AS FOLLOWS:
1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR

LUCKY STRIKER -- 3
MAN OF WAR -- 2
ENTERPRISE -- 2
THE SEEKER -- 1
ALEX THE GREAT -- 4
COMMAND-->LIST

SPEED		
LUCKY STRIKER	MAN OF WAR	ENTERPRISE
200.000	244.000	300.000
THE SEEKER	ALEX THE GREAT	
201.000	190.000	

GATE START		
LUCKY STRIKER	MAN OF WAR	ENTERPRISE
4.000	3.000	4.000
THE SEEKER	ALEX THE GREAT	
1.000	5.000	

BACKGROUND		
LUCKY STRIKER	MAN OF WAR	ENTERPRISE
GOOD	FAIR	GOOD
THE SEEKER	ALEX THE GREAT	
EXCELLENT	GOOD	

```

*****
*                                     *
*                               PREV.REC                               *
*-----*-----*-----*-----*-----*-----*-----*-----*-----*
* LUCKY STRIKER    MAN OF WAR    ENTERPRISE                          *
* EXCELLENT        GOOD          GOOD                                *
*
* THE SEEKER      ALEX THE GREAT                                     *
* FAIR            EXCELLENT                                          *
*-----*-----*-----*-----*-----*-----*-----*-----*-----*

```

```

*****
*                                     *
*                               FIRST RACE?                             *
*-----*-----*-----*-----*-----*-----*-----*-----*-----*
* LUCKY STRIKER    MAN OF WAR    ENTERPRISE                          *
* YES              YES          YES                                *
*
* THE SEEKER      ALEX THE GREAT                                     *
* YES              YES                                          *
*-----*-----*-----*-----*-----*-----*-----*-----*-----*

```

```

*****
*                                     *
*                               LUCK                                    *
*-----*-----*-----*-----*-----*-----*-----*-----*-----*
* LUCKY STRIKER    MAN OF WAR    ENTERPRISE                          *
* FAIR             GOOD          GOOD                                *
*
* THE SEEKER      ALEX THE GREAT                                     *
* EXCELLENT       POOR                                                  *
*-----*-----*-----*-----*-----*-----*-----*-----*-----*

```

COMMAND-->END

.....
 USER'S PREFERENCE PACKAGE PROGRAM

NOTE: THE COMMANDS CRTE & ADD CAN BE ISSUED ONLY ONCE DURING
 A SESSION & YOU CANNOT ISSUE A LIST BEFORE YOU CREATE THAT FILE

COMMAND-->CRTE
 ENTER NO. OF SCENARIOS
 02
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 1
 FAST HORSE
 ENTER SCENARIO DESCRIPTION FOR SCENARIO NO. 2
 LONG DISTANCE HORSE
 DO YOU WISH TO SKIP THE PAIRED COMPARSION ? (YES/NO)
 YES

FAST HORSE

SPEED
 .990
 GATE START
 .560
 BACKGROUND
 .230
 PREV.PEC.
 .670
 FIRST RACE?
 .550
 LUCK
 .990

LONG DISTANCE HORSE

SPEED
 .665
 GATE START
 .505
 BACKGROUND
 .560
 PREV.PEC.
 .830
 FIRST RACE?
 .665
 LUCK
 .885
 COMMAND-->LIST

 REPORT EXAMPLE CASE 1 USER'S PREFERENCE PACKAGE PAGE 1
 FAST HORSE

SPEED	.990
GATE START	.560
BACKGROUND	.230
PREV.REC.	.670
FIRST RACE?	.550
LUCK	.990

 REPORT EXAMPLE CASE 1 USER'S PREFERENCE PACKAGE PAGE 2
 LONG DISTANCE HORSE

SPEED	.340
GATE START	.450
BACKGROUND	.890
PREV.REC.	.990
FIRST RACE?	.780
LUCK	.780

```

*****
REPORT EXAMPLE      CASE 1  USER'S PREFERENCE PACKAGE
      AVEPAGE VALUES FOR EACH PARAMETER
*****

```

PAGE 3

```

      SPEED                .665
      GATE START           .505
      BACKGROUND           .560
      PREV.REC.            .830
      FIRST RACE?          .665
      LUCK                  .885

```

COMMAND-->END

 SYSTEM SELECTION MODEL PROGRAM

LIST OF AVAILABLE PARAMETERS

- 1 -- SPEED
- 2 -- GATE START
- 3 -- BACKGROUND
- 4 -- PREV.REC.
- 5 -- FIRST RACE?
- 6 -- LUCK

DO YOU WISH TO ENTER YOUR OWN RATINGS ? (YES/NO)
 NO

SCENARIOS AVAILABLE

- 1 -- FAST HORSE
- 2 -- LONG DISTANCE HORSE

ENTER NO OF CHOICE OF SCENARIO
 2

LONG DISTANCE HORSE

PARAMETERS	RATINGS

SPEED	.340
GATE START	.450
BACKGROUND	.890
PREV.REC.	.990
FIRST RACE?	.780
LUCK	.780

ENTER THE NUMBER OF PARAMETERS TO BE INCLUDED IN THE DECISION PROCESS

6

ENTER EACH INTEGER NO. ASSOCIATED WITH EACH CHOSEN
PARAMETER (FROM LOWEST TO HIGHEST)

PARAMETER NO. -- 1
PARAMETER NO. -- 2
PARAMETER NO. -- 3
PARAMETER NO. -- 4
PARAMETER NO. -- 5
PARAMETER NO. -- 6

PARAMETER -- SPEED

WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)

NO

PARAMETER -- GATE START

WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)

NO

PARAMETER -- BACKGROUND

WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)

NO

PARAMETER -- PREV.REC.

WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)

NO

PARAMETER -- FIRST RACE?

A QUALITATIVE PARAMETER--CANNOT SPECIFY A RANGE

PARAMETER -- LUCK

WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER
HAS TO SATISFY ? (YES/NO)

NO

DO YOU WISH TO SEE TABULAR OUTPUT ? (YES/NO)

YES

```

*****
REPORT EXAMPLE      CASE 1  UTILITY VALUES TABLE      PAGE 1
LONG DISTANCE HORSE
*****

```

UTILITY VALUES FOR REPORT EXAMPLE

HORSES

```

A -- LUCKY STRIKER
B -- MAN OF WAR
C -- ENTERPRISE
D -- THE SEEKER
E -- ALEX THE GREAT
    OPTIMUM
    HORSES

```

A	B	C	D	E	
.738	.518	.556	.758	.604	D

DO YOU WISH TO EXECUTE THE PROGRAM AGAIN ? (YES/NO)
NO

APPENDIX D - UPPP FLOWCHART AND NARRATIVE

UPPP, the User's Preference Package Program, creates the User's Preference Package (UPP) file, the file containing the set of scenarios and their associated sets of parameter preference ratings. The program uses the HEADER file, created by PPP, to present the parameters to the user and allow the user to rate them, and then incorporates the data into a revised HEADER file and a file named UPP. Appendix D consists of a general flowchart and narrative for the UPPP.

- 1.1 USERS creates the user's preference package,
 which includes scenarios and the parameter ratings for
 these scenarios.
- 1.2-1.3 Read in HEADER file.
- 1.4 Take command from user--CRTE, LIST, ADD, or
 END.
- 1.5-1.6 If command is END, stop execution.
- 1.7-1.8 If command is CRTE, call subroutine CREATE.
- 1.9-1.10 If command is not CRTE and ITERN = 0
 (indicating this is the first run on an already-existing
 user's preference package), call READR2 to read in the
 second part of HEADER and UPP.
- 1.11-1.12 Command ADD calls subroutine ADD to add
 scenarios.
- 1.13-1.14 Command LIST causes calling of subroutine
 LIST, to list scenarios and their associated ratings.
- 1.15 Read next command.
- 1.16 Set ITERN = 1 to indicate command sequence
 has been executed at least once, and return to process
 new command.

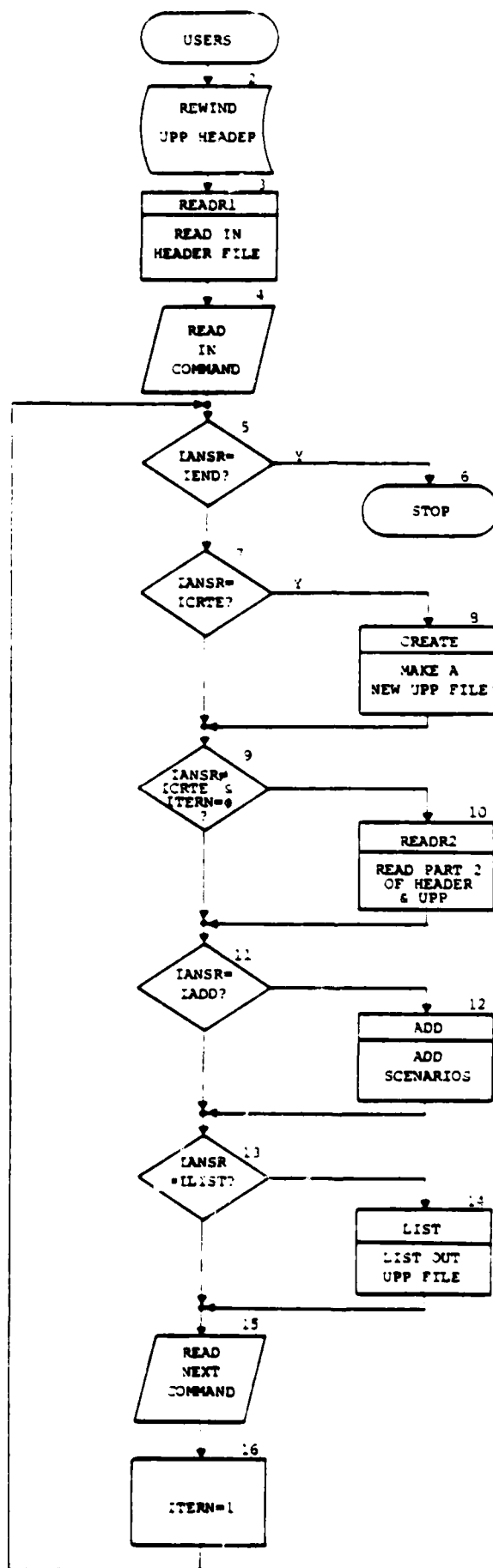


Chart Number 1.3
Module Name USERS

- 1.3.1 READR1 reads in first part of HEADER file.
- 1.3.2-1.3.4 Rewind, read problem size and names of systems, subsystems, etc.
- 1.3.5 Return to main.

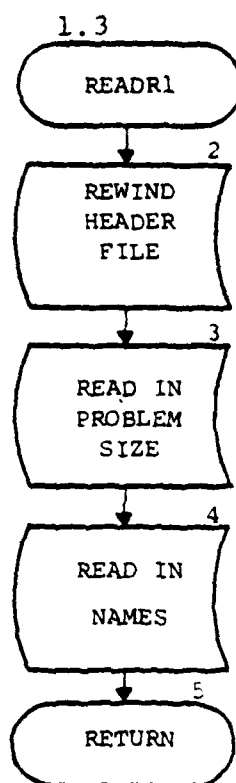


Chart Number 1.8
Module Name USERS

- 1.8.1 CReaTE creates a User's Preference Package,
adding scenarios and ratings to the problem.
- 1.8.2-1.8.3 Read number of scenarios and their names, or
descriptions.
- 1.8.4 Write out to new HEADER file with WRTR.
- 1.8.5 Call COMPAR to allow user to enter ratings.
- 1.8.6 Return to main.

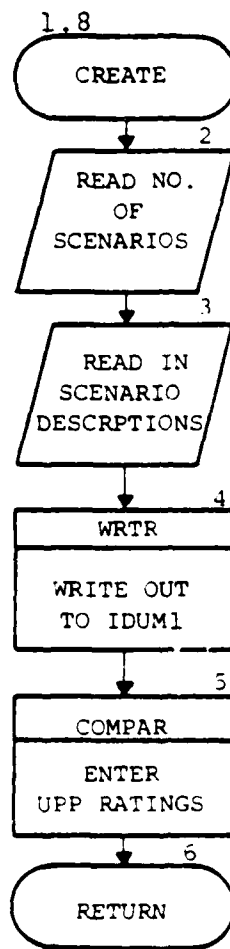


Chart Number 1.4
Module Name USERS

- 1.8.4.1 WRTR writes HEADER's data and any scenario data to file IDUM1, which becomes the new header file.
- 1.8.4.2-1.8.4.3 Rewind IDUM1 and fill with HEADER information.
- 1.8.4.4 Return to ADD.

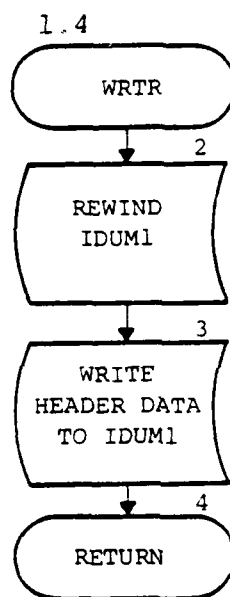


Chart Number 1.8.5
Module Name USERS

- 1.8.5.1 COMPAR allows entering of ratings, and writes them to UPP file.
- 1.8.5.2 User is asked if he wishes to skip the paired comparison.
- 1.8.5.3 Parameters are presented in pairs; user picks most important.
- 1.8.5.4 LARGE puts parameters in order after they are compared, from most-chosen to least.
- 1.8.5.5 The parameters are presented in descending importance order.
- 1.8.5.6-1.8.4.7 User enters ratings and they are written to the UPP file.
- 1.8.5.8 Return to subroutine CREATE.

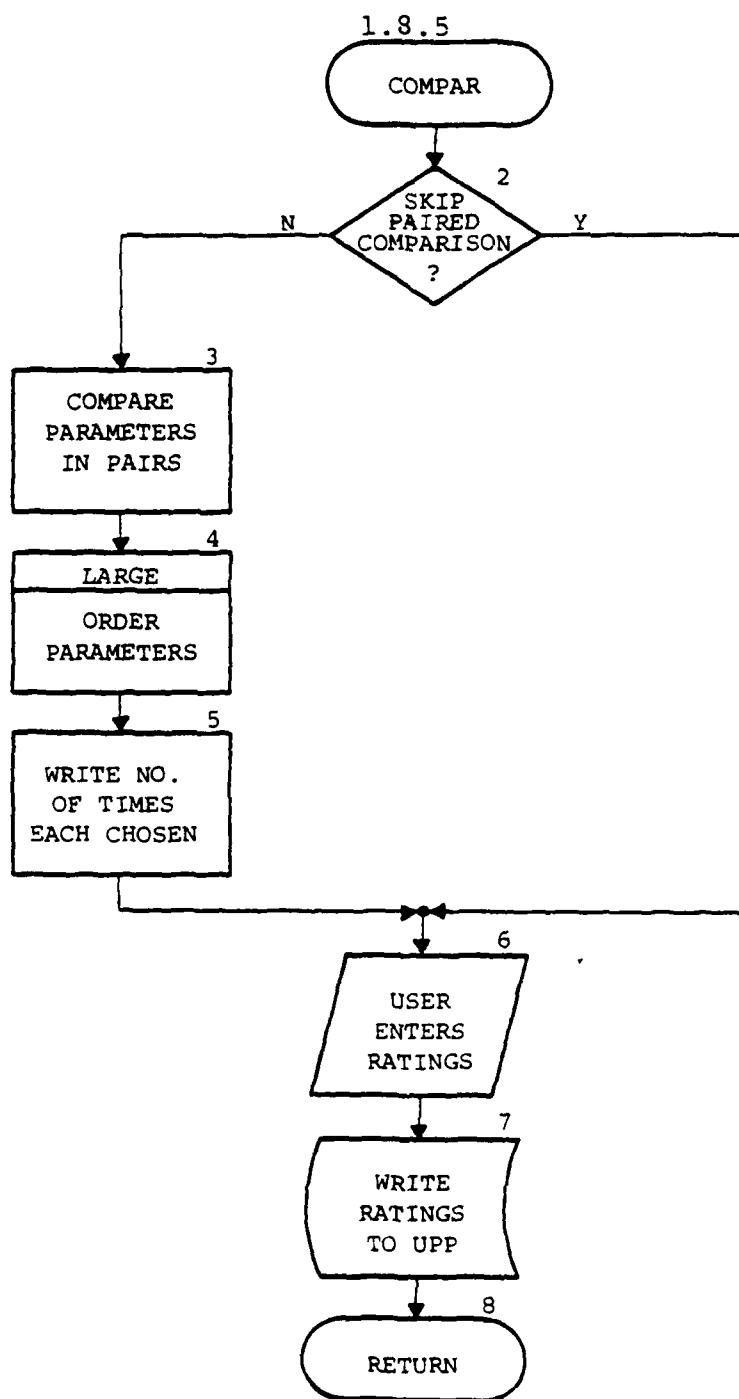


Chart Number 1.8.5.4
Module Name USERS

- 1.8.5.4.1 LARGE sorts parameters by numbers of times
 chosen in the paired comparison and saves the sort order
 by saving the subscripts of the array elements in
 another array.
- 1.8.5.4.2 Determine order by a comparison sort, com-
 paring current element to relative maximum.
- 1.8.5.4.3 Save sort order, not by changing elements
 themselves but by saving the subscripts in array IORDER.
- 1.8.5.4.4 Return to COMPAR.

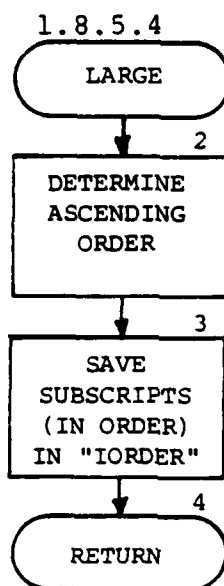


Chart Number 1.10
Module Name USERS

- 1.10.1 READR2 reads the second part of HEADER and the UPP file; used as a result of executing USERS to ADD to or LIST an already-existing user's preference package.
- 1.10.2 Read NSCEN, the number of scenarios, from HEADER file.
- 1.10.3 Read scenario descriptions from HEADER.
- 1.10.4 Read in ratings from UPP file.
- 1.10.5 Return to main.

1.10

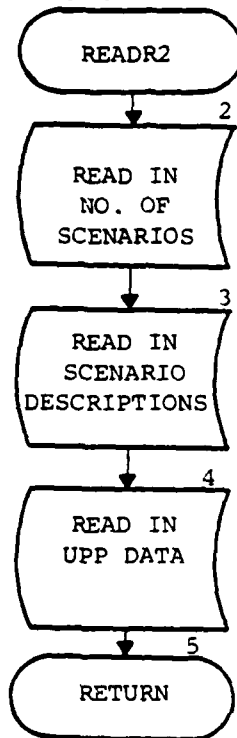


Chart Number 1.12
Module Name USERS

- 1.12.1 ADD is called when the users wish to add more scenarios to the problem.
- 1.12.2-1.12.3 Read number of scenarios to add and their descriptions, or names.
- 1.12.4 Call WRTR to write this new information in the HEADER file. See Chart 1.8.4 for further documentation.
- 1.12.5 Call COMPAR to allow user to enter ratings. See Chart 1.8.4 for further documentation.
- 1.12.6 Return to main program.

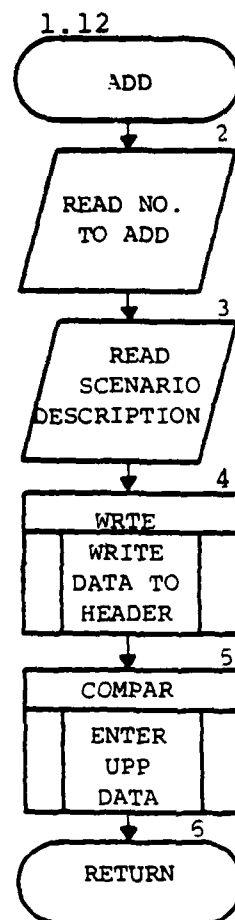
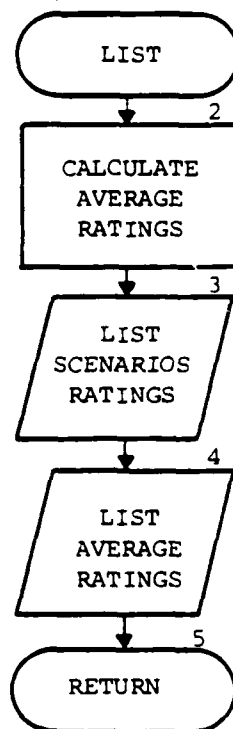


Chart Number 1.14
Module Name USERS

- 1.14.1 LIST lists out the user's preference package, scenarios, ratings, and average ratings for each parameter.
- 1.14.2 Calculate average parameter ratings over all scenarios.
- 1.14.3-1.14.4 List out ratings, for each scenario as well as averaged ratings for all scenarios.
- 1.14.5 Return to main.

1.14



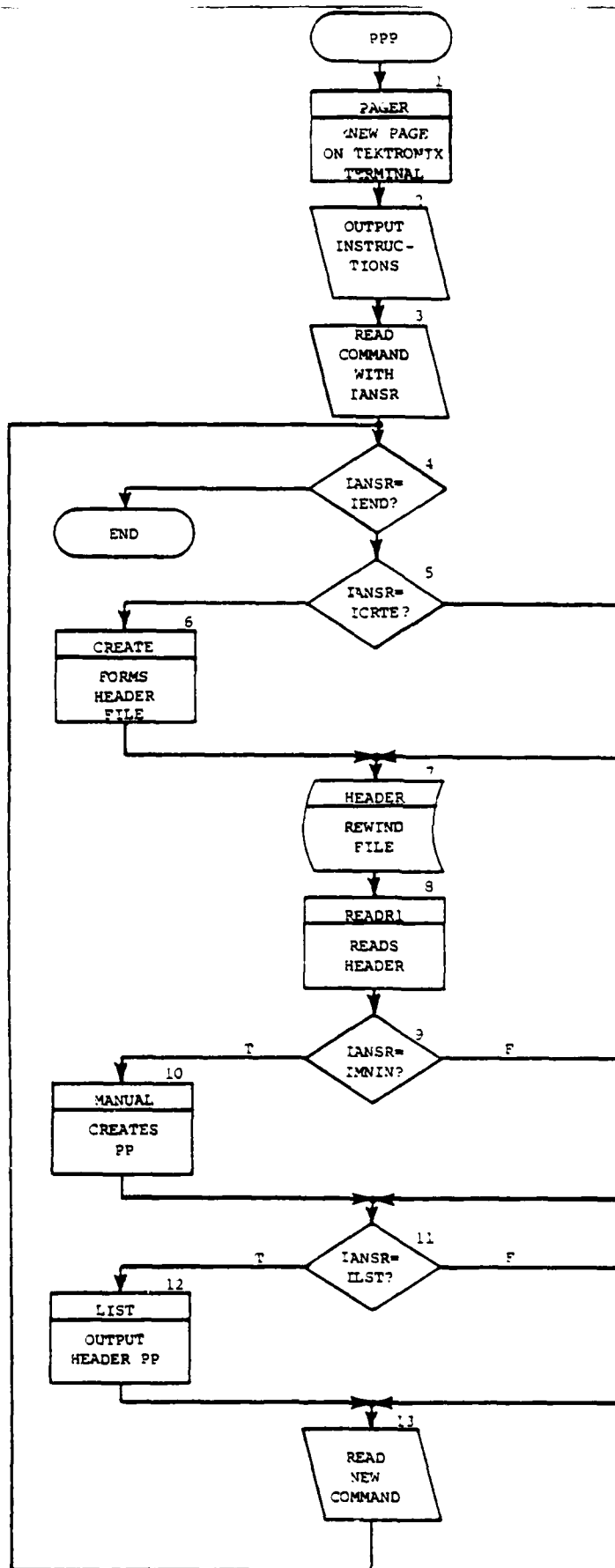
APPENDIX E - PPP FLOWCHART AND NARRATIVE

PPP, the Parameter Package Program, creates the Parameter Package, which consists of the PP and HEADER data files. The HEADER file contains the size of the problem and the names given to the various entities in the problem, such as the systems, subsystems, parameters, etc. The PP file contains numerical values for the parameters, either coded qualitative values (1 meaning yes, 2 meaning no) or actual numerical data (6875 gallons of fuel usage per year). PPP is the first program in the series comprising the model, and Appendix E consists of a general flowchart and narrative for the program.

(1) Parameter Package Program

This module reads the user's command and calls the appropriate subroutines.

- .1 PAGER clears screen on TEKTRONIX terminal, returns cursor to home.
- .2 Print instructions, list and describe commands.
- .3 Read in IANSR command string.
- .4 End program if "END" is typed.
- .5-.6 Create new HEADER file with subroutine CREATE if "CRTE".
- .7-.8 Rewind and read in HEADER.
- .9-.10 Create PP file if "MNIN" is typed in using subroutine MANUAL.
- .11-.12 If "LIST" is typed, call LIST to output HEADER and PP.
- .13 Read in new command and return.



(1.1) TEKTRONIX PAGER

This module contains TEKTRONIX subroutine calls that clears the screen and returns the cursor to home.

- .1-.2 Subroutine INITT, BINITT initialized PLOT-10.
- .3 Subroutine NEWPAG is PLOT-10 clearing function.
- .4 Subroutine FINITT closes out PLOT-10.

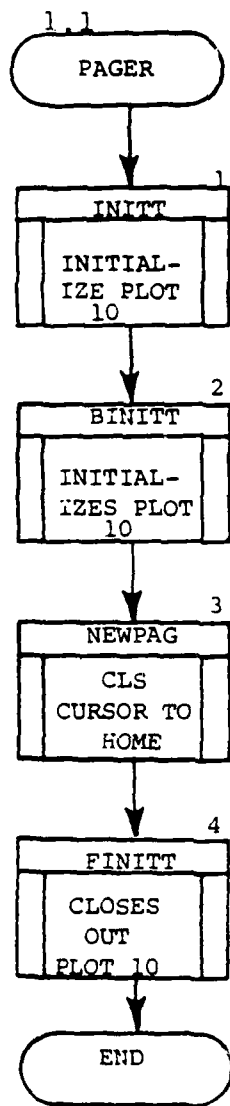


Chart No. 1.6
Module Name CREATE

(1.6) CREATES the HEADER file

This module creates the HEADER file.

- .1 Read in all system and parameter names, parameter types,
number of levels.
- .2 Rewind file.
- .3 Write names, etc. to file.

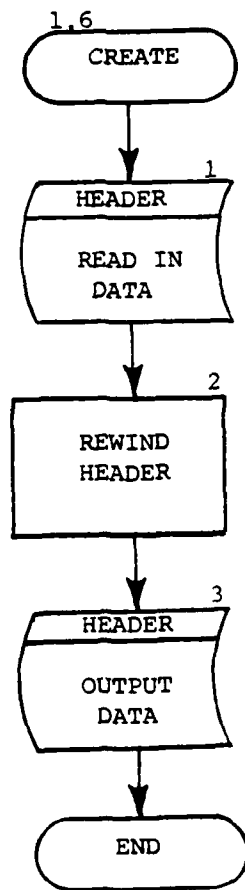
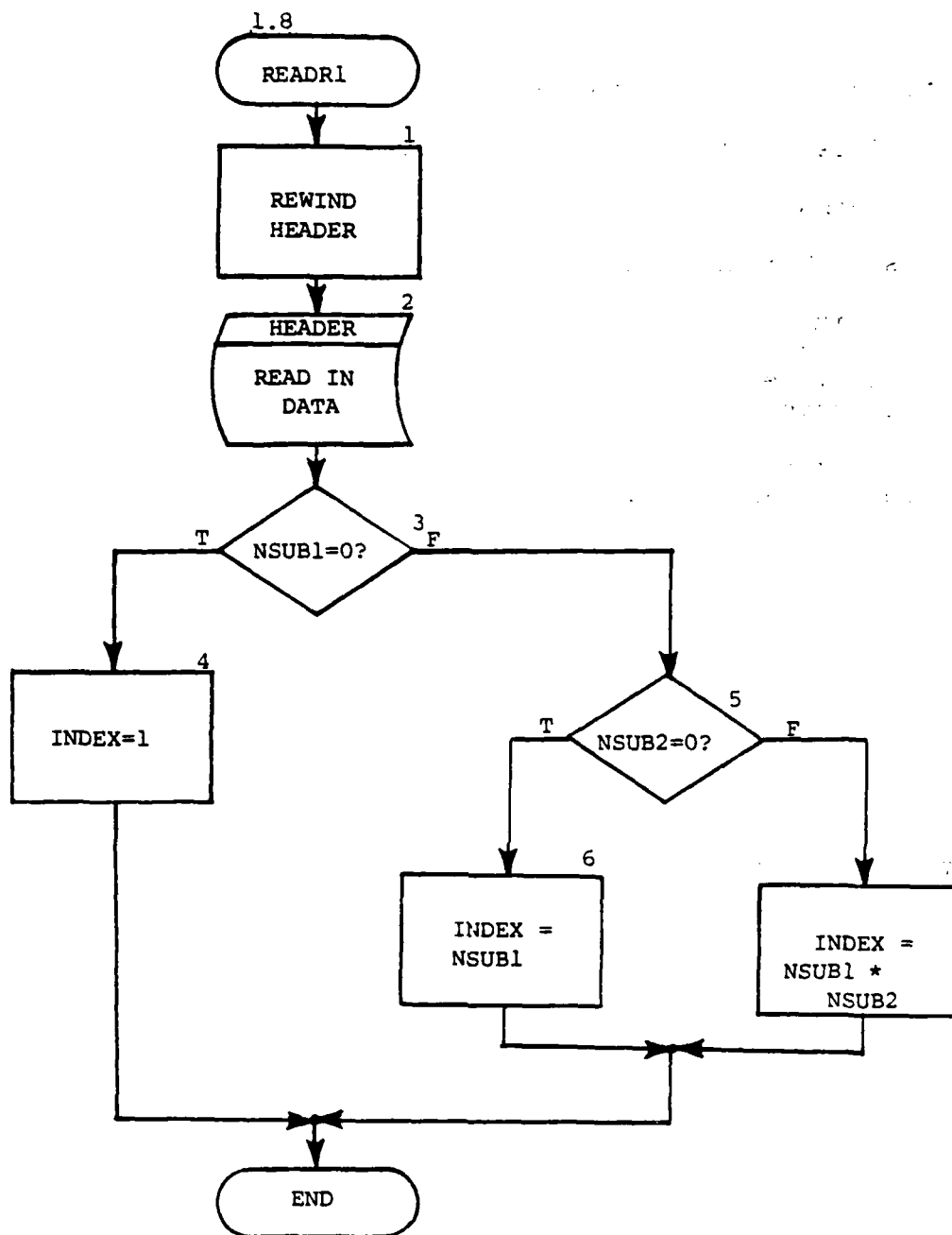


Chart No. 1.1.8
Module Name READR1

(1.8) READ in HEADER

This module reads in the data file HEADER.

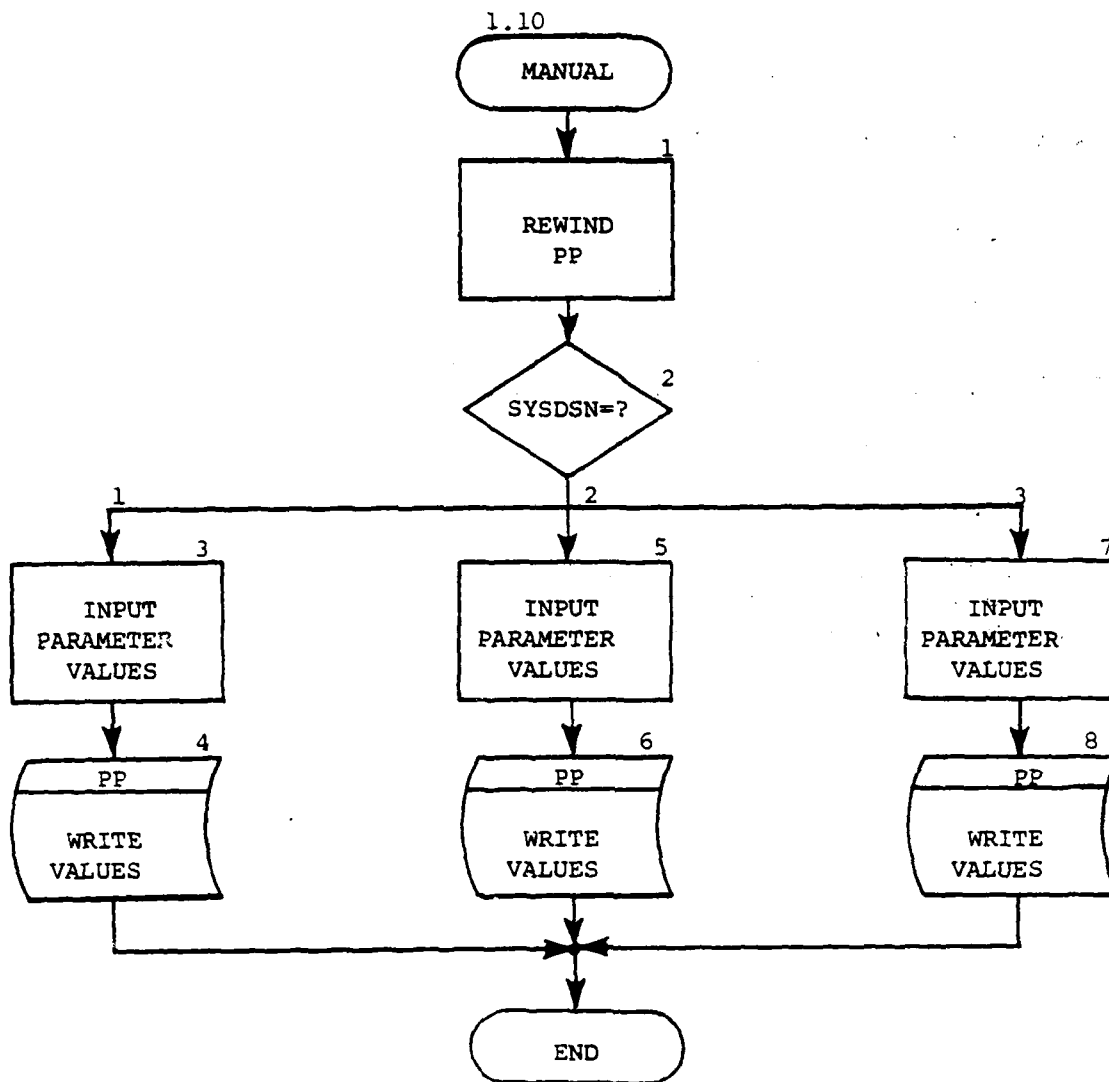
- .1 Rewind the file.
- .2 Read in all names, numbers, parameter types.
- .3-.4 If no level 1 subsystems, INDEX=1.
- .5-.6 If no level 2 subsystems, INDEX=number of level 1 sub-
 systems.
- .7 Otherwise, INDEX=product of numbers of level 1 and 2
 subsystems.



(1.10) MANUAlY enter the PP

The module allows the user to enter in the parameter values.

- .1 Rewind PP file.
- .2 SYSDSN indicates the correct branching.
- .3-.4 Parameters entered and written to file for systems only.
- .5-.6 Parameters are entered and written to file for systems
 and subsystems.
- .7-.8 Parameters are entered and written to file for systems
 and level 1 and 2 subsystems.



(1.12) LIST the parameter names and values

This module prints a table containing the parameter names and values.

- .1 Rewind PP.
- .2 SYSDSN indicates the correct branching.
- .3-.5 For systems, with PAGER and write out parameter names
 WRTOUT outputs system names and parameter values.
- .6-.8 For systems and level 1 subsystems page with PAGER and
 write out parameter names. WRTOUT outputs system and sub-
 system names and parameter values.
- .9-.11 For systems and level 1 and 2 subsystems, page with
 PAGER and write out parameter names. WRTOUT output systems
 and subsystem names and parameters values.

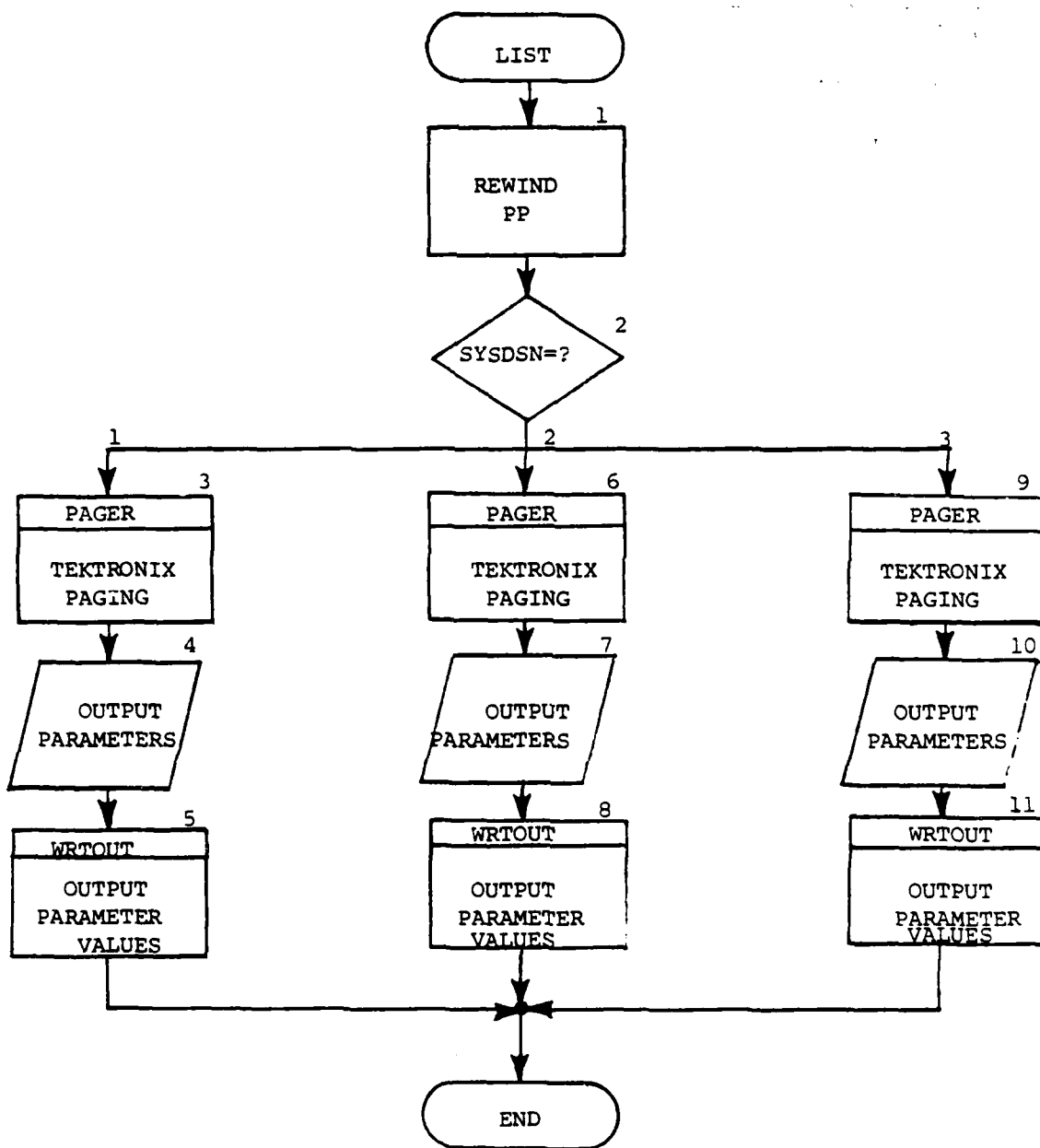
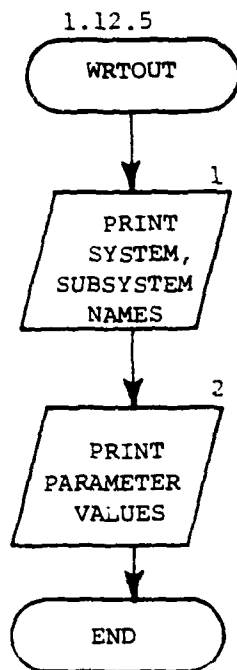


Chart No. 1.12.5
Module Name WRTOUT

(1.12.5) WriteOUT the parameter values

This module outputs the system and subsystems names and parameter values.

- .1 Print names
- .2 Print parameter values corresponding to names.



APPENDIX F - GRAPHX AND GRAPHT FLOWCHART AND NARRATIVE

GRAPHX and GRAPHT, the two graphics programs for the system selection model, provide both histogram and data-point graphical output. GRAPHX creates graphs that plot system values versus systems in a vertical histogram format, and GRAPHT creates graphs of system values versus time, to demonstrate effects of time-variant parameters on the decision results. The time intervals must be input to the system as the lowest-level subsystem, and the problem must of course be Case 1 or 2 to accommodate this requirement. The "time graphs," as they are referred to, contain one curve per system on each graph. The following set of flowcharts and narratives describe the general structure of the programs; however, it is recommended that those who wish to really understand the programs refer to the Tektronix manuals referenced in the narratives.

- 1.1 Program GRAPHT graphs system values against time at specified intervals, with a separate curve for each system. Time intervals must be specified in either level 1 or level 2 subsystems, and must be five in number.
- 1.2 Rewind HEADER, containing names and numbers, and SYSNUM, containing final system rating values.
- 1.3 Read in HEADER via program READR.
- 1.4 Read SYSNUM values into array SYSNRM.
- 1.5 Obtain frm user IBAUD and JTERM, baud rate and terminal type used by PLOT-10 initialization routines.
- 1.6 Problem must be Case 2 or Case 3, and must have exactly five lowest level subsystems. (Error message if conditions are not satisfied.)
- 1.7-1.9 Call appropriate overlay to make graph.

AD-A119 160

SOUTHEASTERN CENTER FOR ELECTRICAL ENGINEERING EDUCAT--ETC F/G 9/2
ADVANCED TECHNOLOGY MULTIPLE CRITERIA DECISION MODEL.(U)

NOV 81 P J SWEENEY, K B BERNER, J R FRAKER F33615-77-C-2059

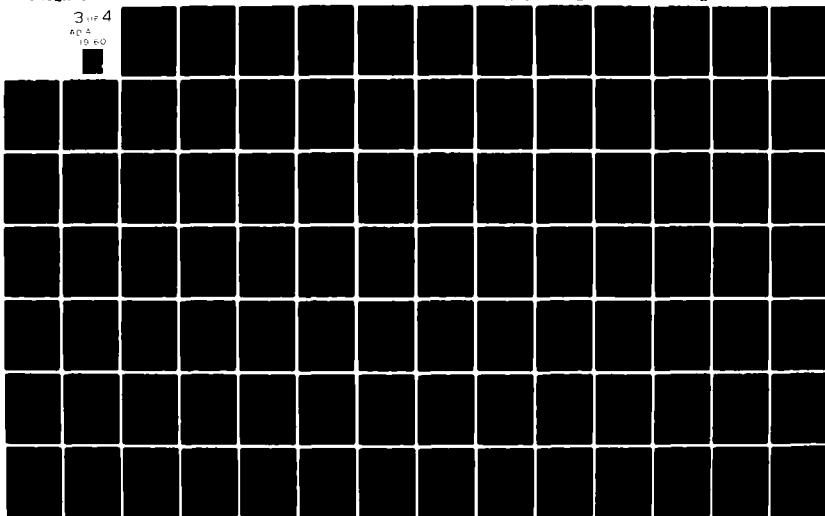
UNCLASSIFIED

AFWAL-TR-81-2112

NL

3 112 4

AD A
10-80



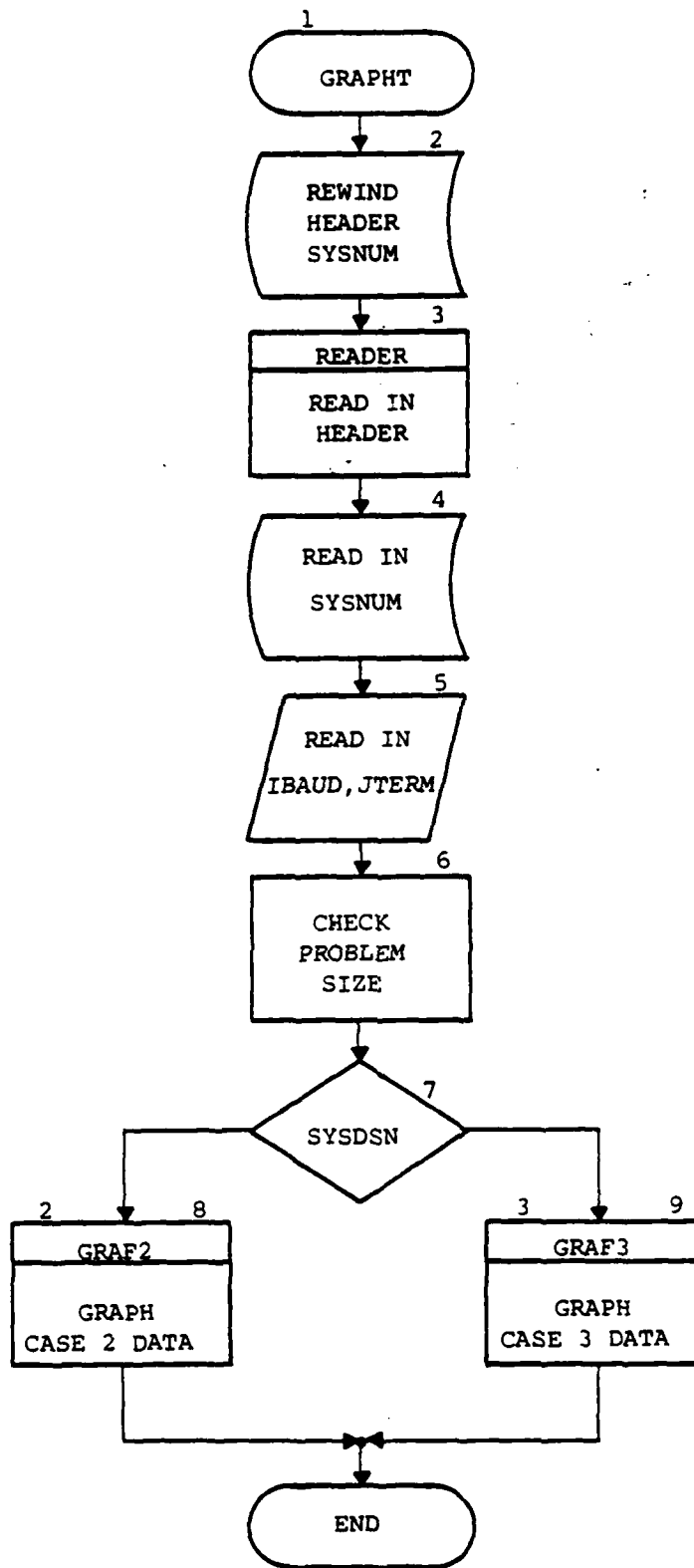


Chart Number 1.3
Module Name GRAPHT

1.3.1 Program READR reads in HEADER data, figures
 INDEX.
1.3.2 Read in data.
1.3.3-1.3.7 INDEX calculated, based on problem type:
 Case 1--INDEX = 1
 Case 2--INDEX = NSUB1
 Case 3--INDEX = NSUB1*NSUB2

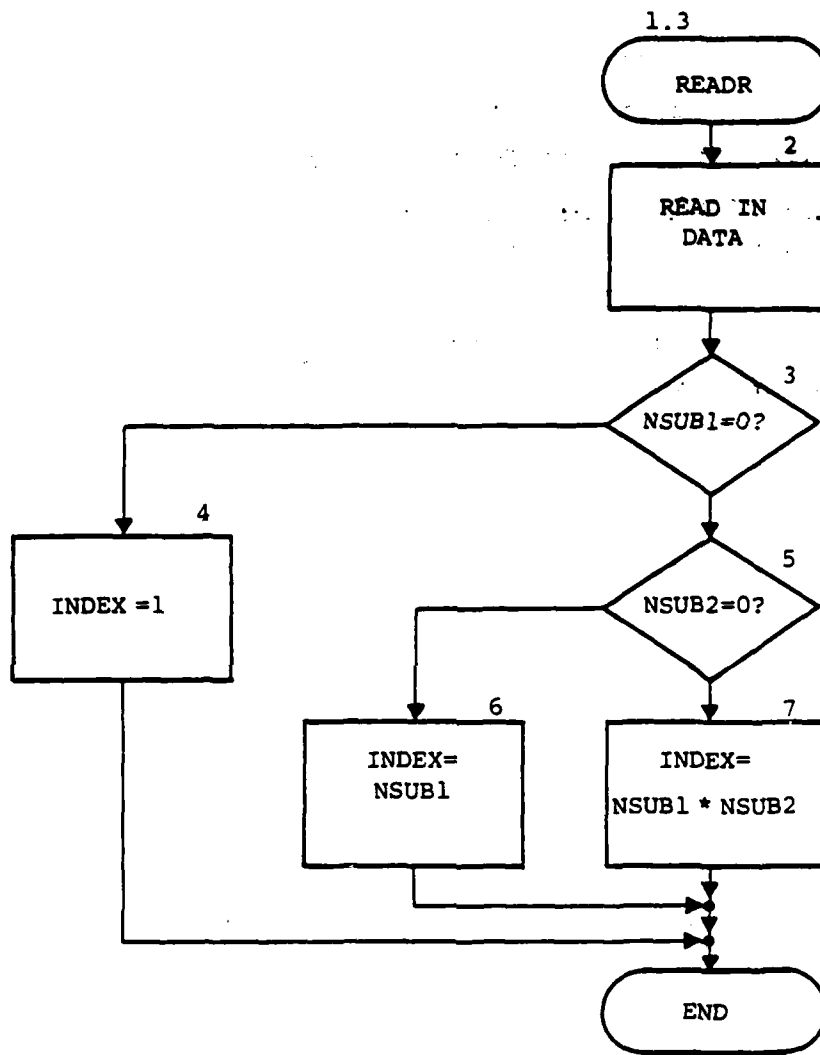


Chart Number 1.8
Module Name GRAPHT

- 1.8.1 Program GRAF2 creates one graph of Case 2 data.
- 1.8.2-1.8.4 TEKTRONIX PLOT-10 software subroutines; for more information on these and their operation, see TEKTRONIX manuals for Terminal Control System (TCS) and Advanced Graphics (AG-II), part numbers 062-1474-00 and 062-1530-00, respectively.
- 1.8.5 TINPUT waits for any key followed by an ETX, then continues; used as delay for examination/hard-copying of graph.

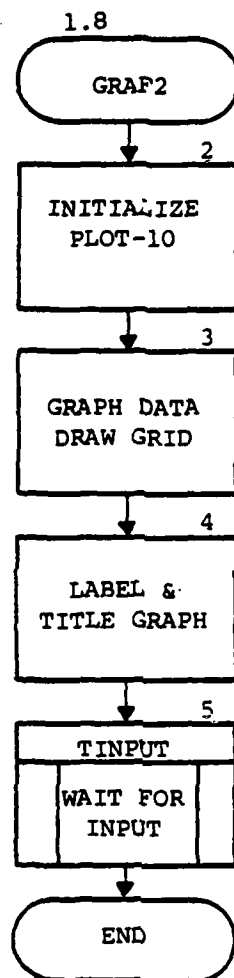


Chart Number 1.9
Module Name GRAPHT

- 1.9.1 Program GRAF3 creates (NSUB1) graphs of the
 Case 3 data.
- 1.9.2-1.9.2.5 Subroutines to draw, label, adjust graph.
- 1.9.6 TINPUT delays, waiting for character and
 ETX input.

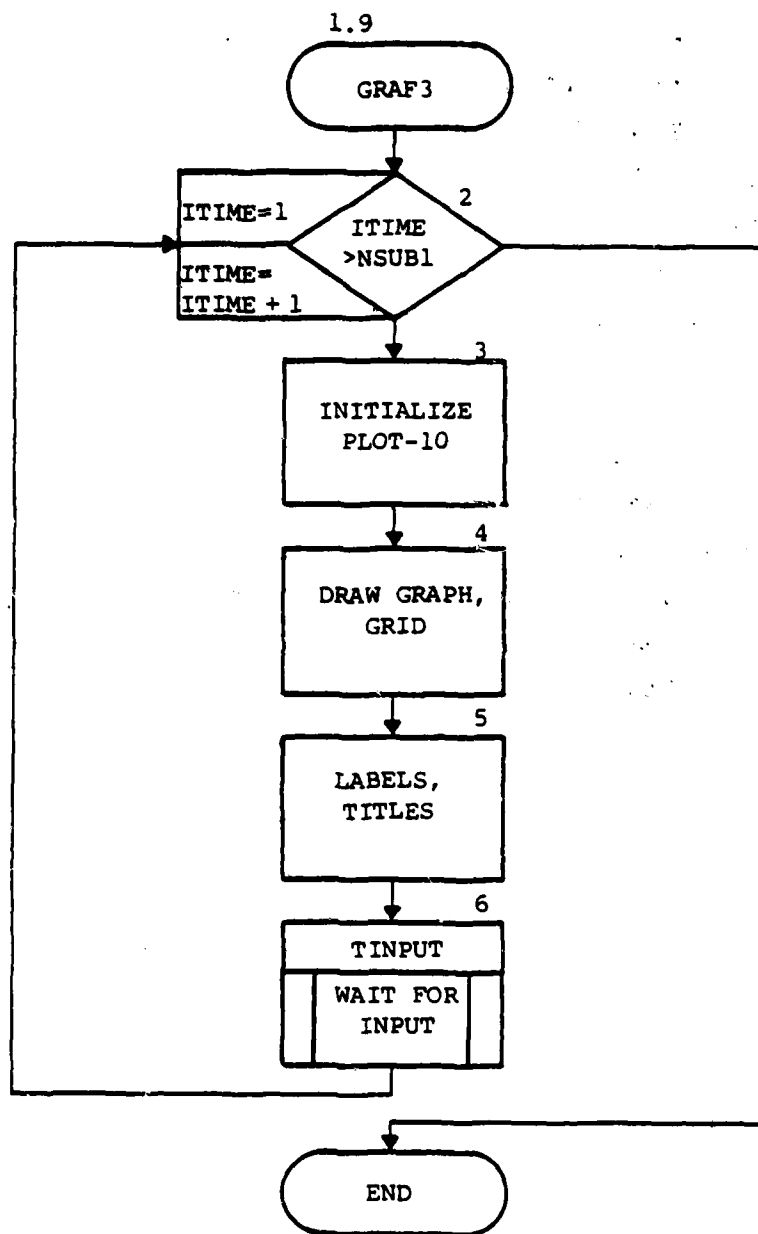


Chart Number 1
Module Name GRAPHX

- 1.1 Program GRAPHX does a bar graph output of the results from SSMP.
- 1.2 Rewind data files. HEADER contains size and names for problem, SYSNUM contains system rating values (from SSMP).
- 1.3 Read initialization parameters (baud rate and terminal type) from user. This is necessary for TEXTRONIX PLOT-10 software routines.
- 1.4 Read in HEADER via program READR.
- 1.5 Read in SYSNUM data into SYSNRM array.
- 1.6 Assign x-values in linear expansion form (see PLOT-10 manuals).
- 1.7-1.10 Go to appropriate program, according to value of SYSDSN.
- 1.11 Stop execution.

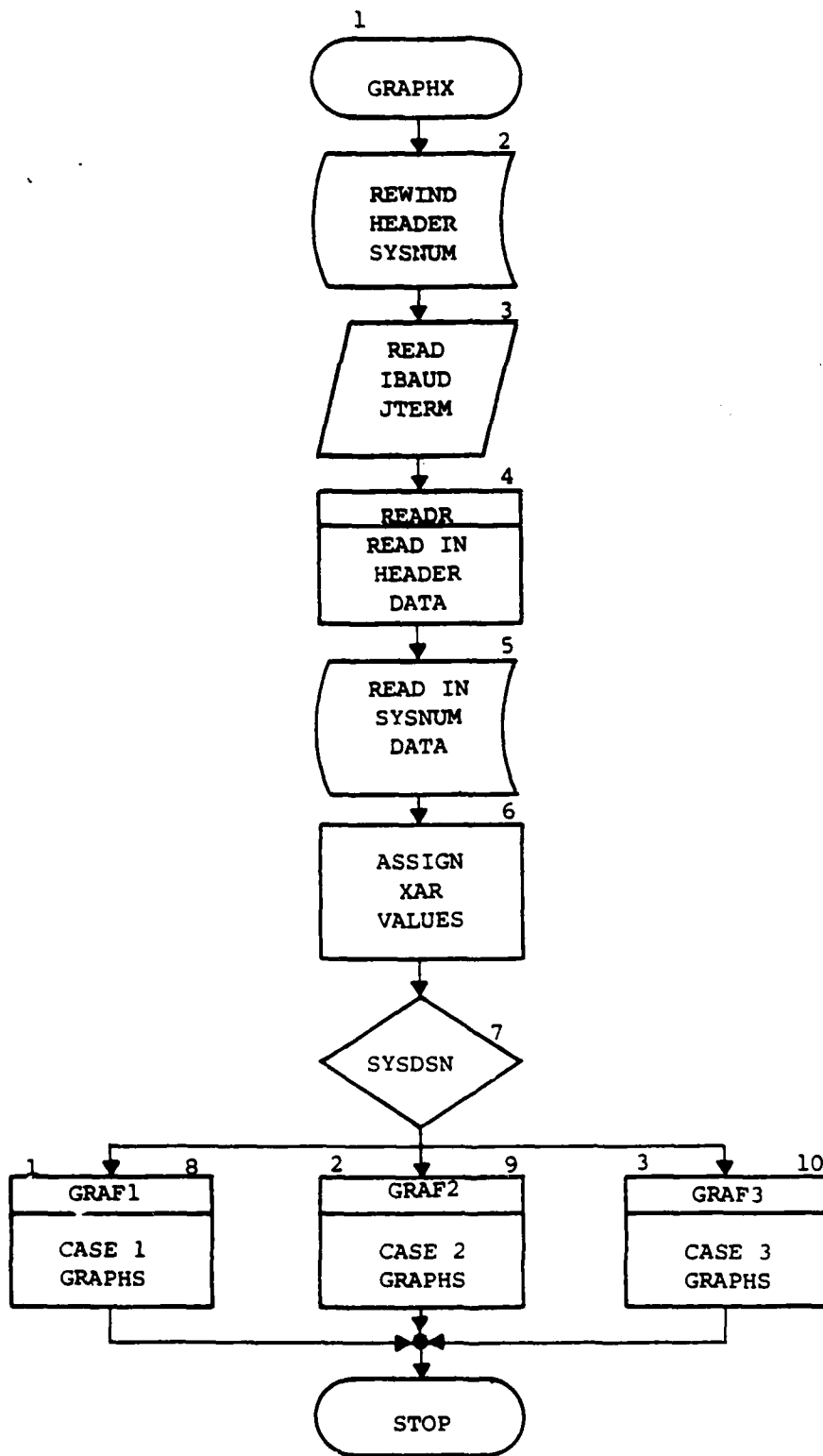
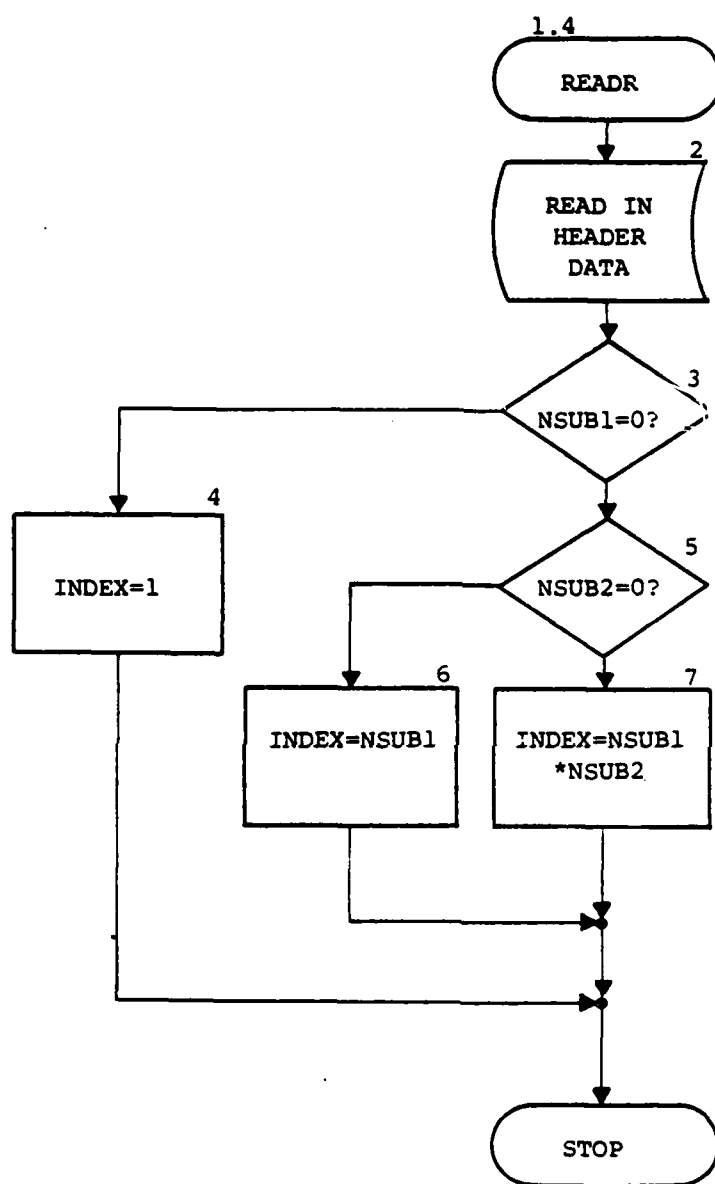


Chart Number 1.4
Module Name GRAPHX

1.4.1 Program READR reads in HEADER data.
1.4.2 Read in values from file HEADER.
1.4.3-1.4.7 Assign value to INDEX according to size of
problem as follows:
Case 1--INDEX = 1
Case 2--INDEX = NSUB1
Case 3--INDEX = NSUB1*NSUB2
1.4.8 Terminate.



- 1.8.1 Program GRAF1 graphs Case 1 data; creates one graph.
- 1.8.2-1.8.4 Assign y-values for graph from SYSNUM data (in array SYSNRM).
- 1.8.5 Initialize PLOT-10 software system.
- 1.8.6-1.8.7 All routines for drawing, labeling, and titling graphs, as well as certain parameters to set special features of graphs. For more information, see TEKTRONIX manuals for Terminal Control System (TCS) and Advanced Graphics (AG-II), part numbers 062-1474-00 and 062-1530-00, respectively.
- 1.8.8 TINPUT waits for user input; any keystroke and ENTER will cause the program to continue.
- 1.8.9-1.8.10 Exit PLOT-10 and terminate.

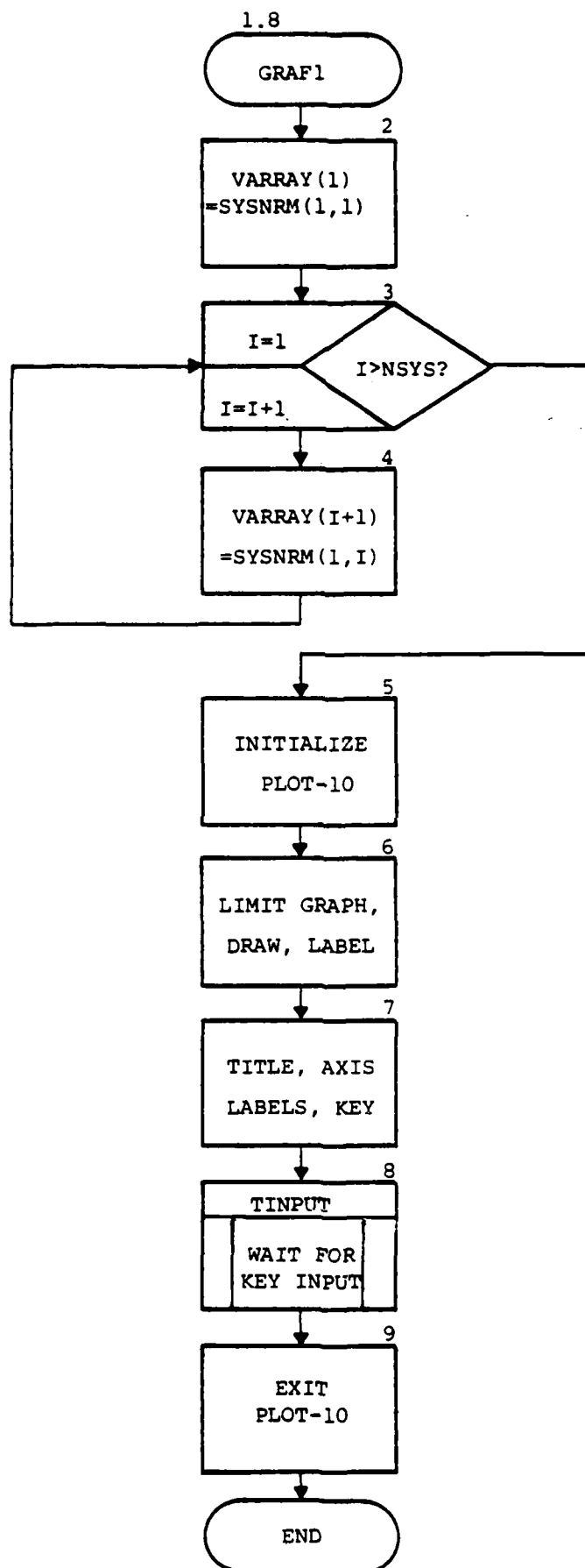


Chart Number 1.9
Module Name GRAPHX

- 1.9.1 Program GRAF2 graphs Case 2 data and creates
 NSUB1 graphs.
- 1.9.2 Do-loop to create graphs, number = number of
 level 1 subsystems.
- 1.9.3-1.9.7 Assign y-values, initialize system, graph,
 wait for input to do next graphs.
- 1.9.8-1.9.9 Exit PLOT-10 and terminate.

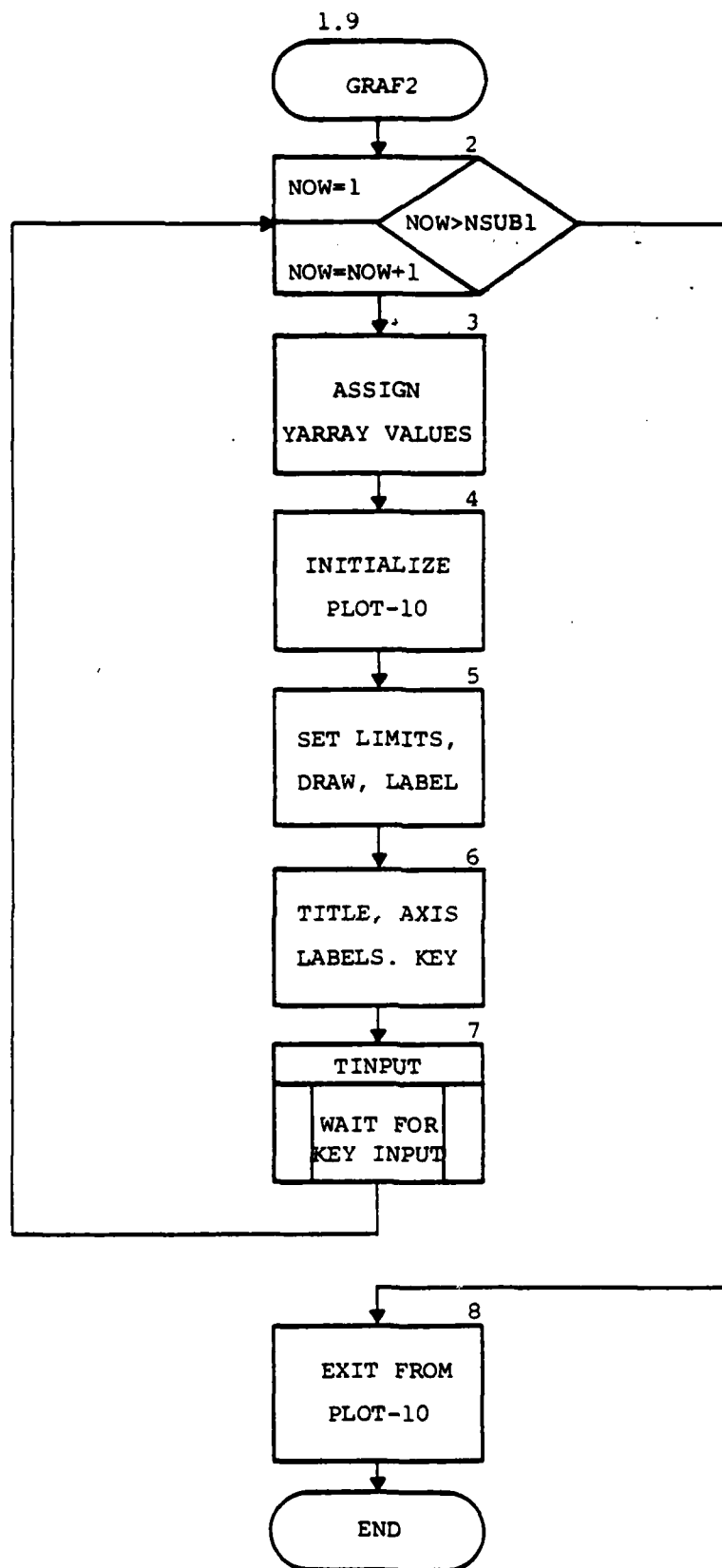
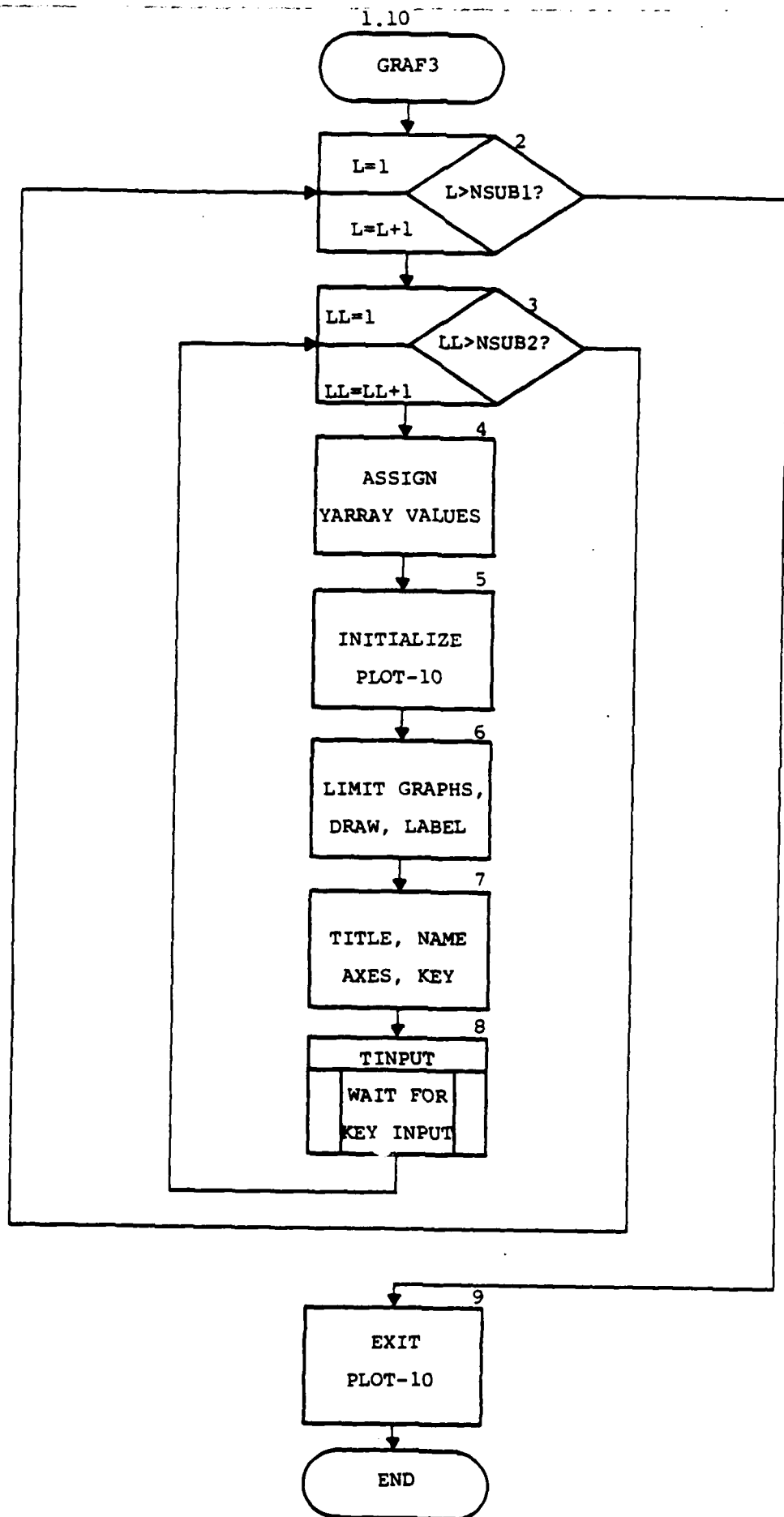


Chart Number 1.10
Module Name GRAPHX

- 1.10.1 Program GRAF3. This graphs Case 3 data,
 creates NSUB1*NSUB2 graphs.
- 1.10.2-1.10.8 Do graphs, wait for input to clear and do
 another.
- 1.10.9-1.10.10 Exit PLOT-10 and terminate.

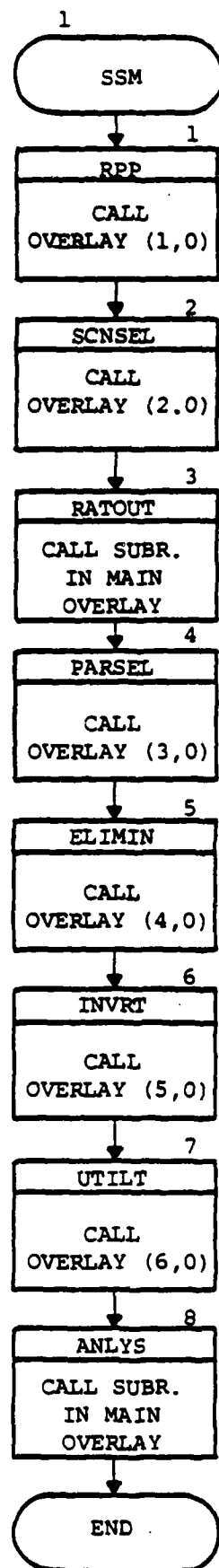


APPENDIX G - SSMP FLOWCHART AND NARRATIVE

SSMP, the System Selection Model Program, is the final step in executing the decision model. SSMP takes the three data files created by PPP and UPPP and integrates them into the problem, asks the user how the decision should be limited and output, and then performs the calculations and manipulations to eliminate and rate each system. This appendix contains a general flowchart and explanatory narrative of SSMP.

(1) System Selection Model

- .1 Reads in Parameter Package (PP) and calls RUPP that Reads in User's Preference Packag (UPP).
- .2 Allows the user to select a scenario to enter parameter ratings.
- .3 Prints out the parameter ratings.
- .4 Allows the user to select the parameters to be in the decision process.
- .5 Eliminates parameter data as specified by parameter limits and calls TABLE to print out eliminated system and/or sub-system.
- .6 If the numerical parameters chosen need inverting then this model performs the necessary calculations.
- .7 Calculates the utility values for each system and calls OUTPUT to print out the results.
- .8 Allows the user to execute the program again. This option can be used to an analysis of the uncertainty of the decision model.



(1.1) Read in Parameter Package

This procedure reads in the Parameter Package (PP) from the data file HEADER and stores the information in labeled COMMON statements.

- .1 Read DBNAME, the problem name for the data bases.
- .2 Read SYSDSN; 1=Case 1 (no subsystems); 2=Case 2 (level 1 subsystems); 3=Case 3 (level 1 and 2 subsystems).
- .3 Read in NSYS, NSUB2 and NPARM.
- .4 Read in the setnames for the systems and, if any, subsystems.
- .5 Read in the specific names for the systems and, if any, level 1 and 2 subsystems.
- .6 Read in parameter names and the values indicating the type of parameter and if it is to be inverted. The value of 1 means YES, and 0 means NO.
- .7 INDEX is calculated to help facilitate the indexing of arrays. Since there is a maximum of 3 dimensional arrays, a fourth dimension is combined with another dimension for Case 3.
- .8 The RUPP module reads in the UPP data file.

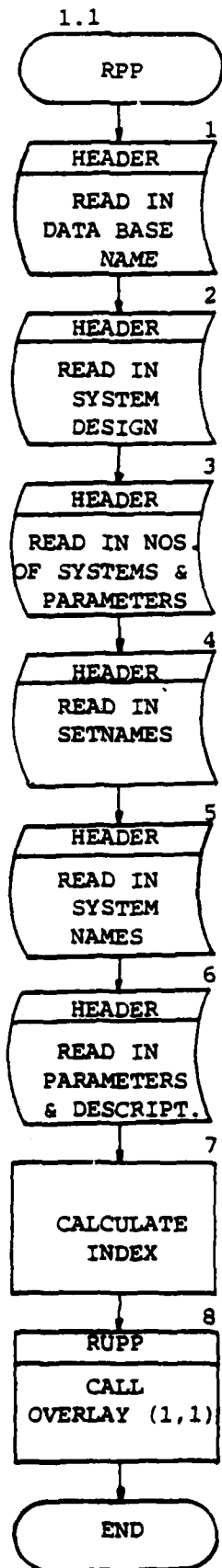
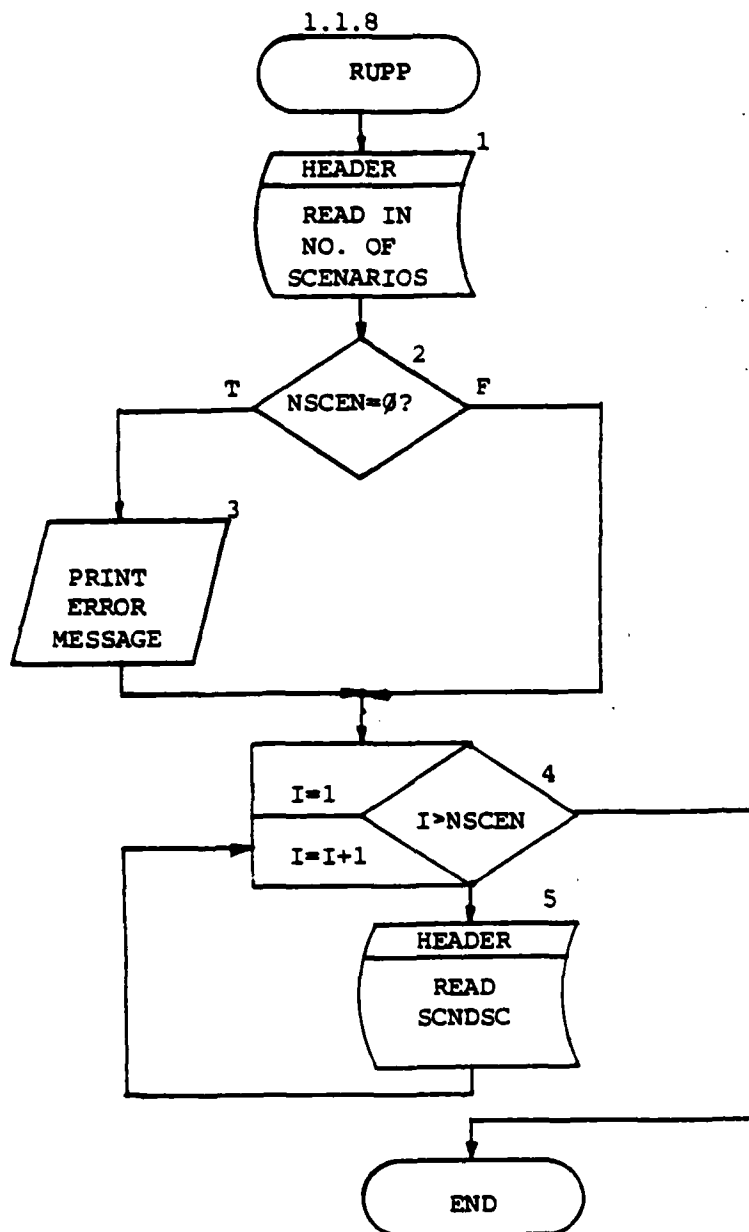


Chart No. 1.1.8
Module Name RUPP

(1.1.8) Read in User's Preference Package

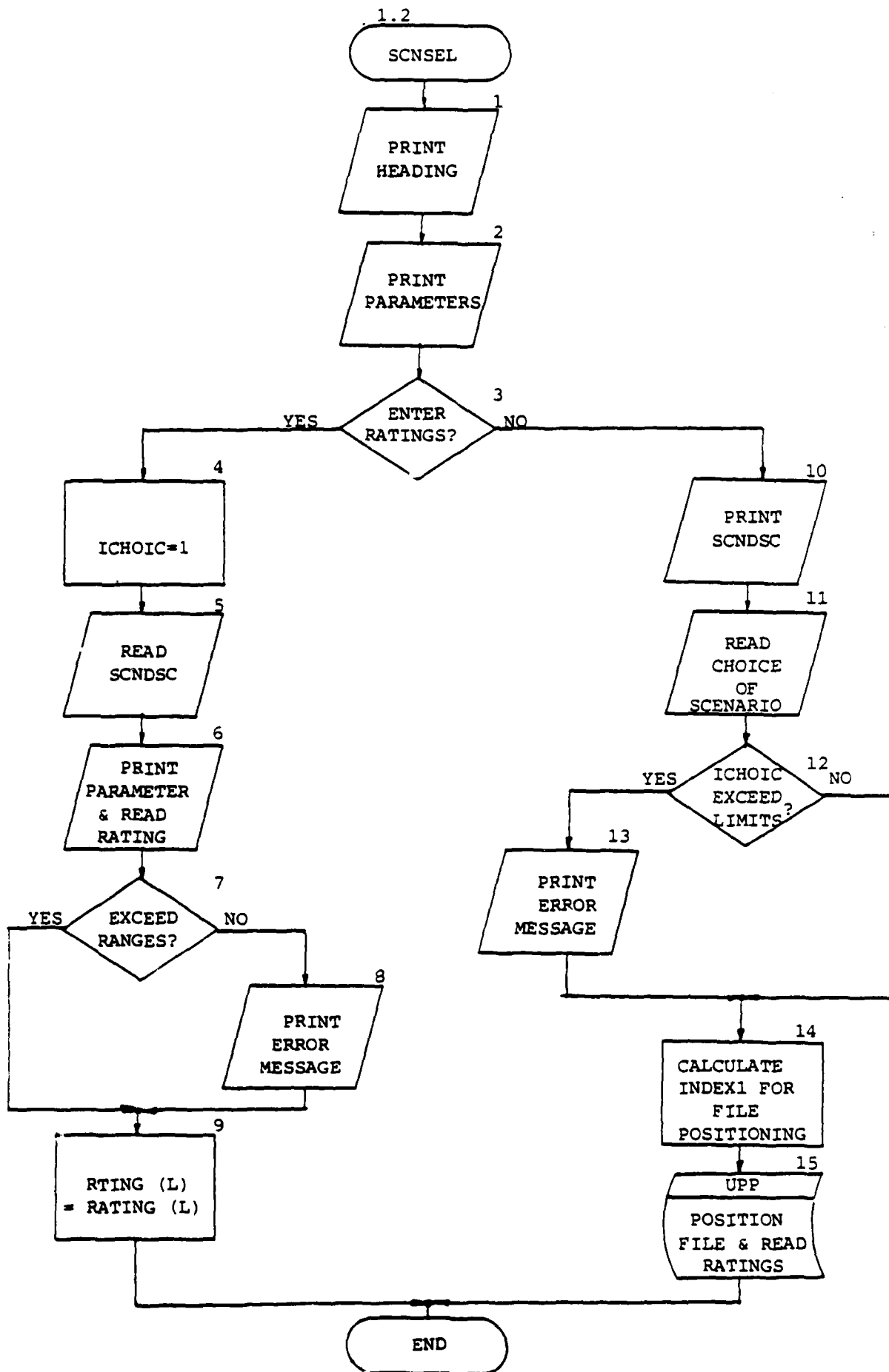
This module reads in the data file called user's preference package from the file called UPP and stores the information in labeled COMMON statements.

- .1 Read in the number of scenarios available, NSCEN.
- .2-.3 Check to see if NSCEN is greater than zero, if not, print an error message.
- .4-.5 Read in the scenario description, SCNDSC, with an alpha-numeric format of 15A4 giving a limit of 60 characters.



(1.2) SCeNario SELECTION

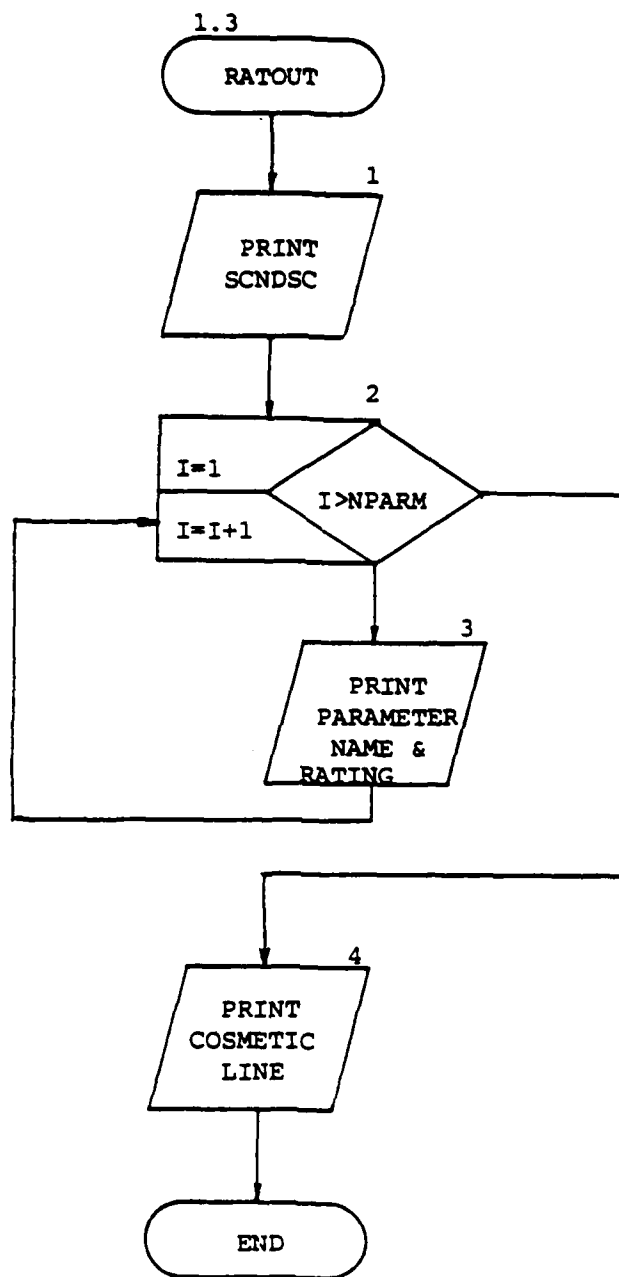
- .1-.2 Print out the available parameters.
- .3 The user chooses between entering his or her own ratings or the ratings available under the different scenarios.
- .4 Set the variable that indicates the choice of scenario to 1.
- .5 Read in the scenario description, SCNDSC, maximum 60 characters.
- .6 Print out each parameter and then read in a corresponding parameter rating.
- .7 Check each parameter rating to see if exceed its ranges.
- .8 If the rating is not a correct value an error message is printed.
- .9 The ratings are stored in another array for future use.
- .10 If the user decides to use an available scenario, the program prints out the descriptions of available scenarios.
- .11 The user indicates his or her choice of a scenario by an integer variable ICHOIC.
- .12 Check to see if ICHOIC does not exceed any ranges.
- .13 If ICHOIC is incorrect the program prints an error message.
- .14 Calculate INDEX1 to help facilitate file indexing.
- .15 After positioning the UPP data file to the correct scenario then read in that scenario's ratings.



(1.3) RATings OUTput

This module prints out the parameters ratings that either been entered by the user or from the scenario that has been chosen.

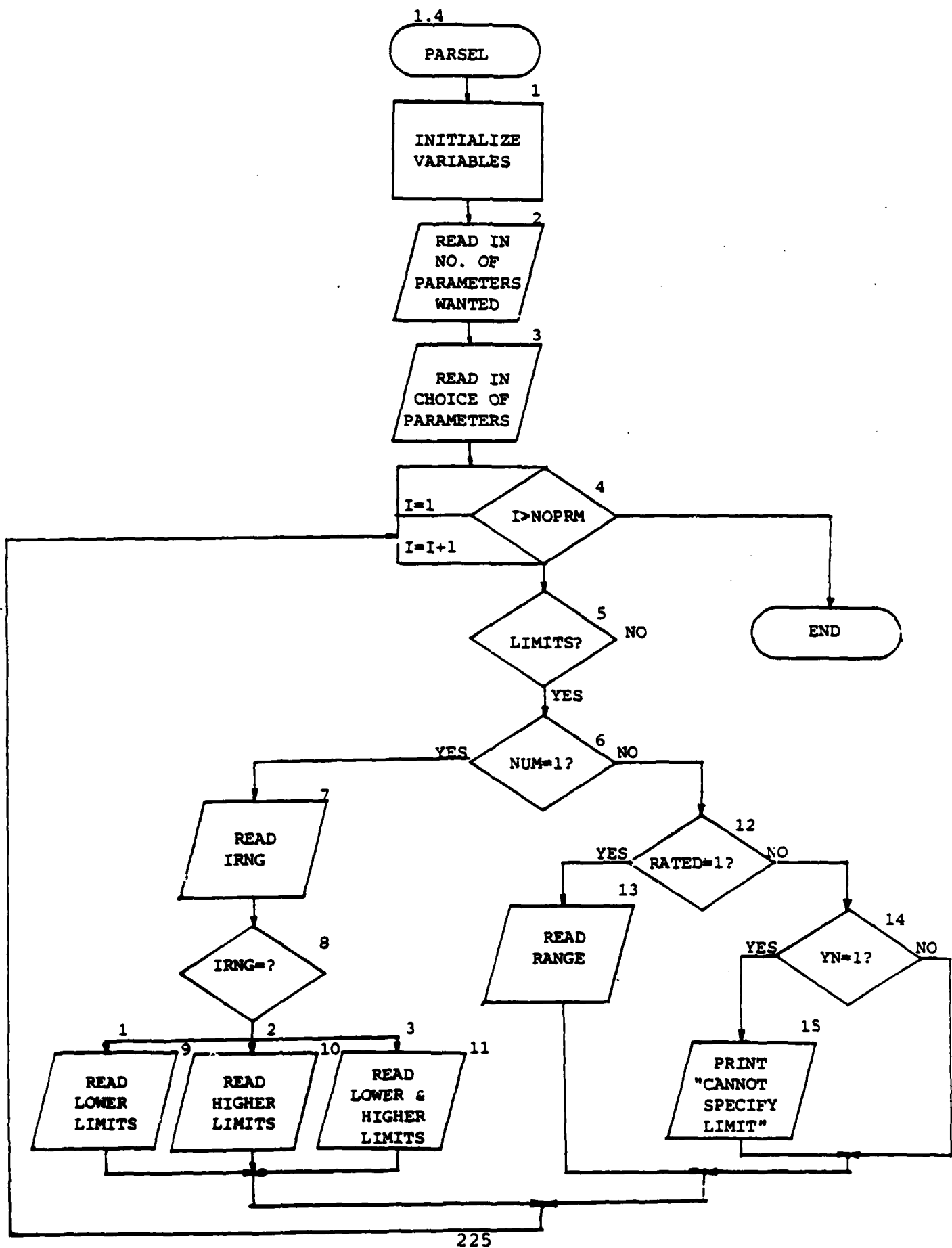
- .1 Print the 60 character scenario description, SCNDSC.
- .2-.3 Print the parameter name and rating.
- .4 Finish printing the box.
- .5 The module returns to the main overlay.



(1.4) PARAmeter SElection

This module allows the user to select the parameters and limits that will be used to participate in the decision process.

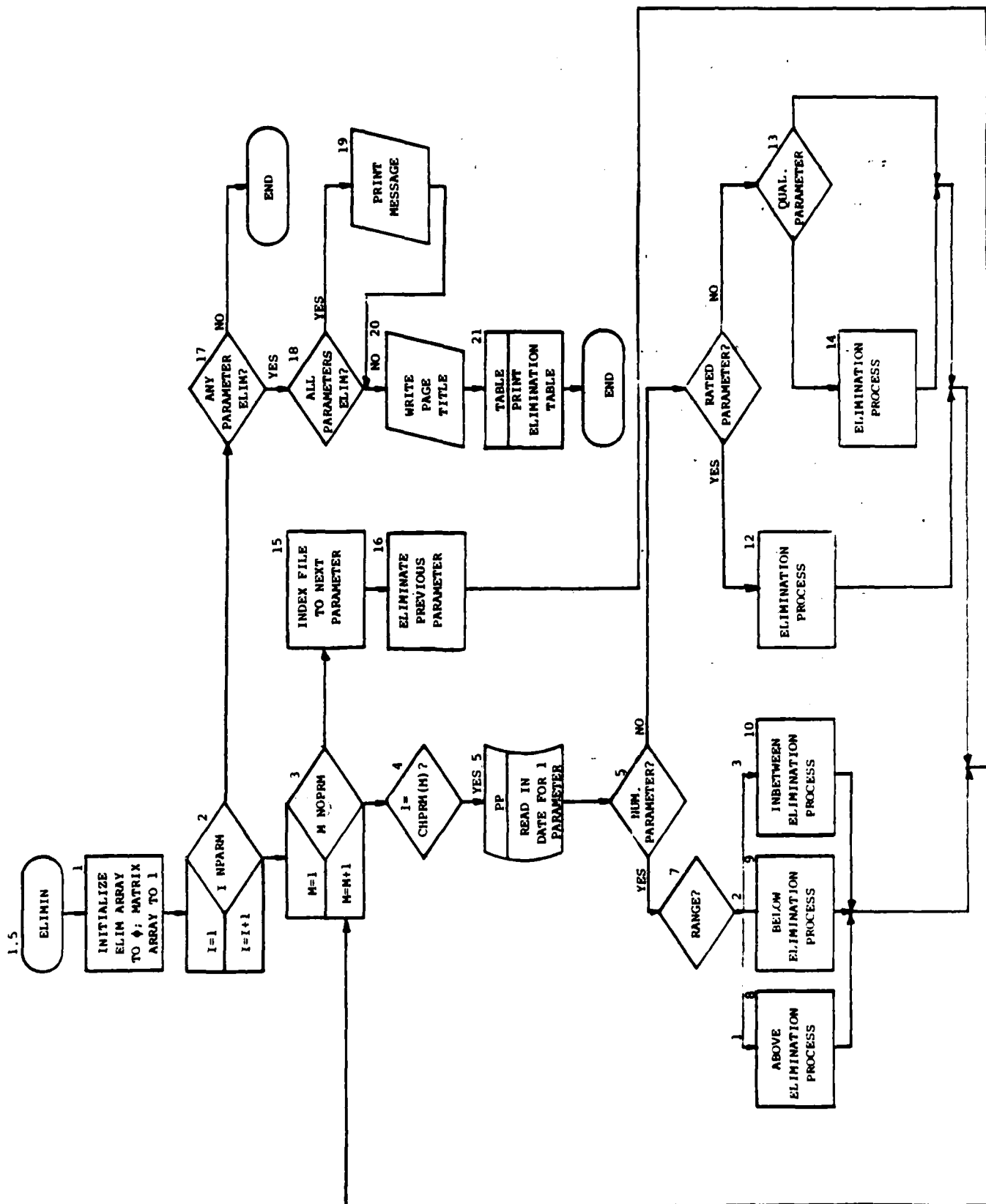
- .1 Initialize several variables.
- .2 Read the user's number of parameters wanted in the decision process.
- .3 Then read in the choice of parameters, CHPRM.
- .4-.5 Query the user to find out if any parameter limits are to be applied.
- .6 Check to see if the parameter is numerical.
- .7 Check to see if parameter is limited from above, below or both.
- .8 Read, IRNG, if:
 - .9 1, then the program reads the lower limits, RLOWR.
 - .10 2, then the program reads the higher limit, HIR.
 - .11 3, then the program reads both the higher and lower limits.
- .12 If the parameter is not numerical then check and see if it is rated.
- .13 Read the integer number (1-5) that will limit to the rated parameter to either, excellent, good, fair, poor, or very poor.
- .14-.15 If the parameter is not either numerical or rated it is then considered a qualitative and a message prints out.



(1.5) ELIMINation of Systems/Subsystems

This module eliminates systems/subsystems based on parameter limits. For more information see Appendix 7.3.

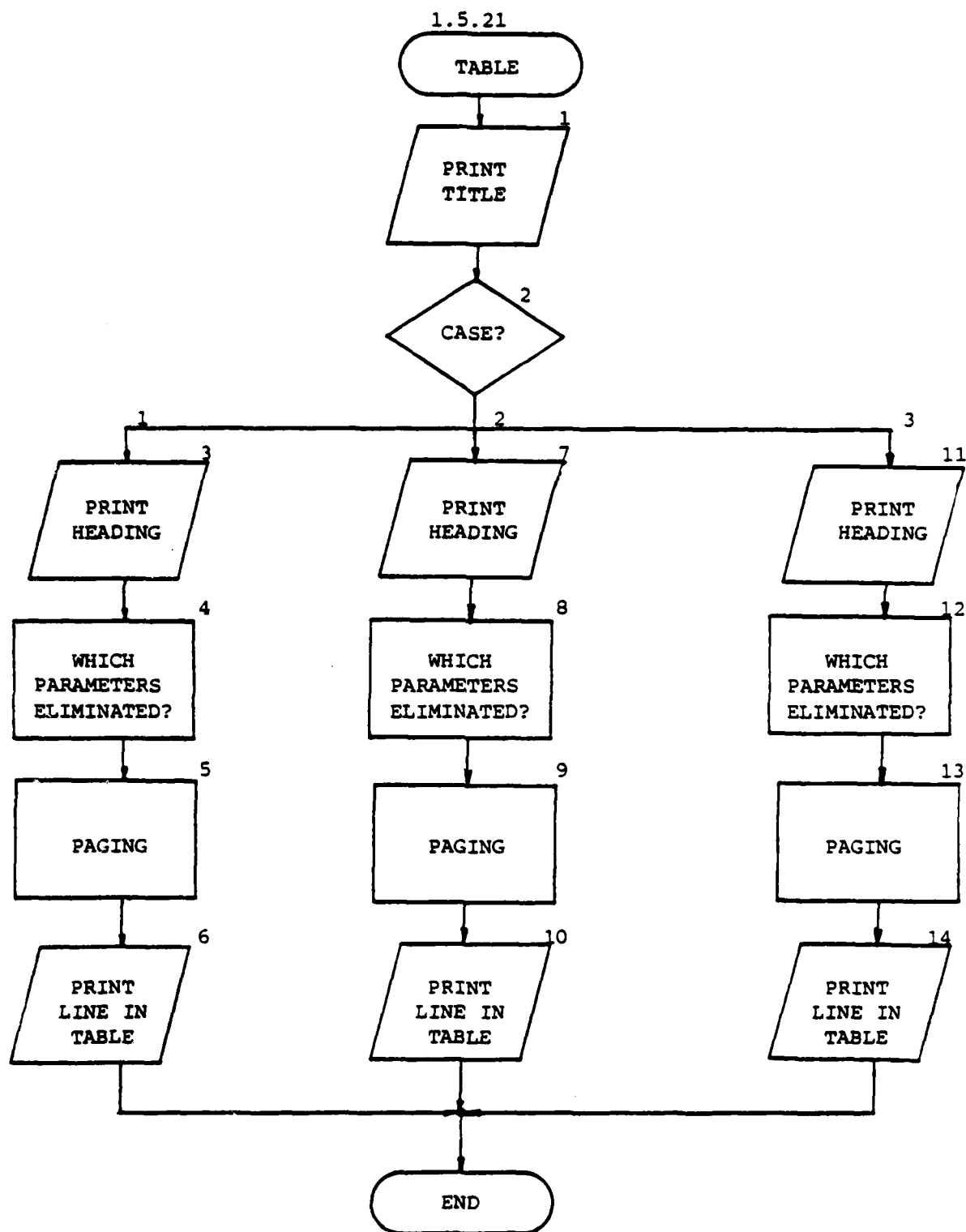
- .1 Initialize the ELIM array to zero and the MATRIX array to one.
- .2 Set up do-loop for all parameters.
- .3 Set up do-loop for chosen parameters.
- .4 Check to see if current parameter is a chosen parameter.
- .5 If a chosen parameter then read in parameter data from PP data file.
- .6-.7 Check to see if a numerical parameter and check range.
- .8 If 1, then perform elimination process for parameter limited from above.
- .9 If 2, then perform elimination process for parameter limited from below.
- .10 If 3, then perform elimination process for parameter limited from above and below.
- .11 If not a numerical parameter then check to see if a rated parameter.
- .12 If a rated parameter then perform elimination process.
- .13 Check to see if a qualitative parameter.
- .14 Then the elimination process is performed.
- .15 After the elimination process is finished then the data file is indexed to next parameter.
- .16 If systems/subsystems have been eliminated then all of the parameter data for these systems or subsystems must also be eliminated.
- .17 After all of the parameters have been checked then test to see if any system or subsystems have been eliminated. If nothing has been eliminated then the program module ends.
- .18 If a system/subsystem has been eliminated then check to see if all systems and subsystems have been eliminated.
- .19 If yes, then print message indicated that the parameter limits were such that no system/subsystem qualified.
- .20 Write page title for output.
- .21 Call program module TABLE that prints the elimination table.



(1.5.21) Elimination TABLE

This module prints out a table containing the reasons why a system or a subsystem was eliminated from the decision model.

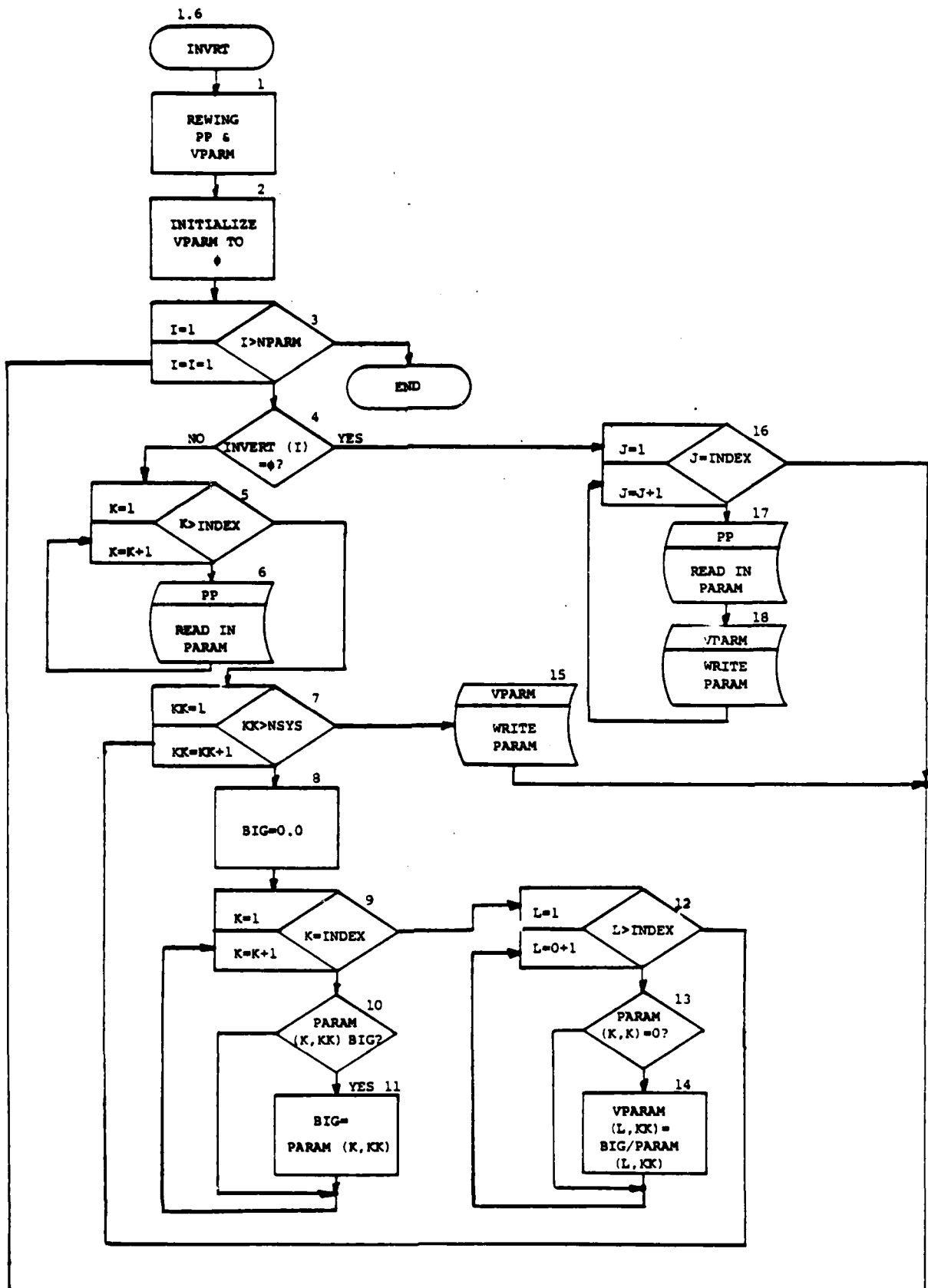
- .1 Print title.
- .2 Check to see if Case 1, 2, or 3.
- .3 Print heading for Case 1.
- .4 Find out which parameters are eliminated.
- .5 After a certain number of lines are printed out the output is started on a new page.
- .6 Print out reason for elimination.
- .7 Print heading for Case 2.
- .8 Find out which parameters are eliminated.
- .9 After a certain number of lines are printed out the output is started on a new page.
- .10 Print out reason for elimination.
- .11 Print heading for Case 3.
- .12 Find out which parameters are eliminated.
- .13 After a certain number of lines are printed out the output is started on a new page.
- .14 Print out reason for elimination.



(1.6) INVERT the Parameter Data

This module inverts indicated parameter data. The parameter data must be such that the lowest value of the parameter data is the "best" data.

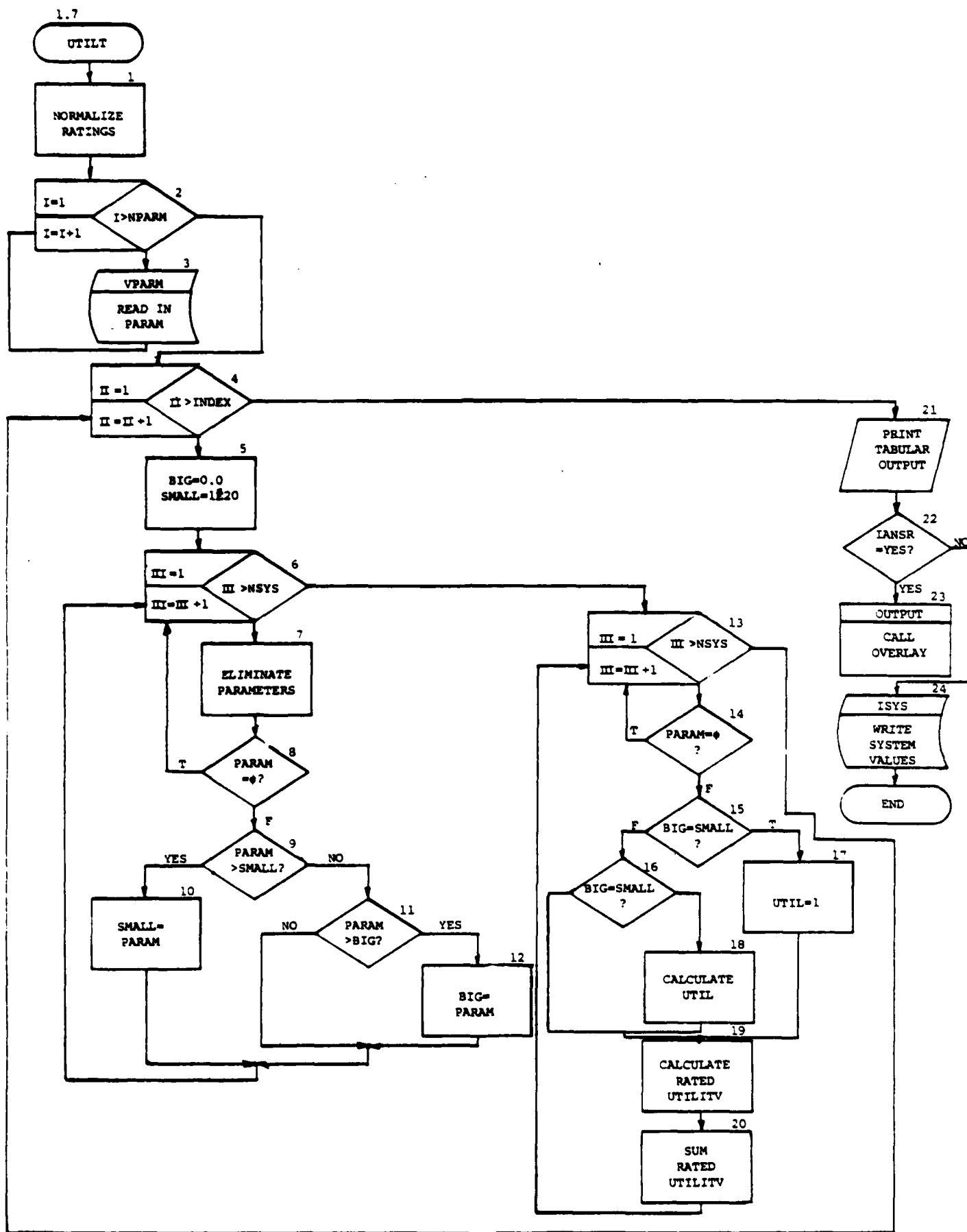
- .1 Rewind PP and VPARAM files.
- .2 Initialize VPARAM array to zero.
- .3 Set up do-loop.
- .4 Test to see if INVERT is zero.
- .5 If no, then set up do-loop.
- .6 Read in a line of parameter data from the data file.
- .7 Set up do-loop to check data for every system.
- .8 Initialize BIG to zero.
- .9-.11 Find the largest value among the system or subsystems for that parameter.
- .12 Set up do-loop.
- .13 If the parameter data equals zero it is considered blank and skips over calculation.
- .14 The parameter data is inverted by dividing the largest value by each system's or subsystem's parameter data.
- .15 After the data has been inverted then write the data out on to the VPARAM data file.
- .16-.18 If the parameter data for that parameter is not to be inverted then write out the parameter data to the VPARAM data file.



(1.7) Calculate the UTILiTy values

This module determines the best system or subsystem based on the highest utility value. See Appendix 7.3 for more information.

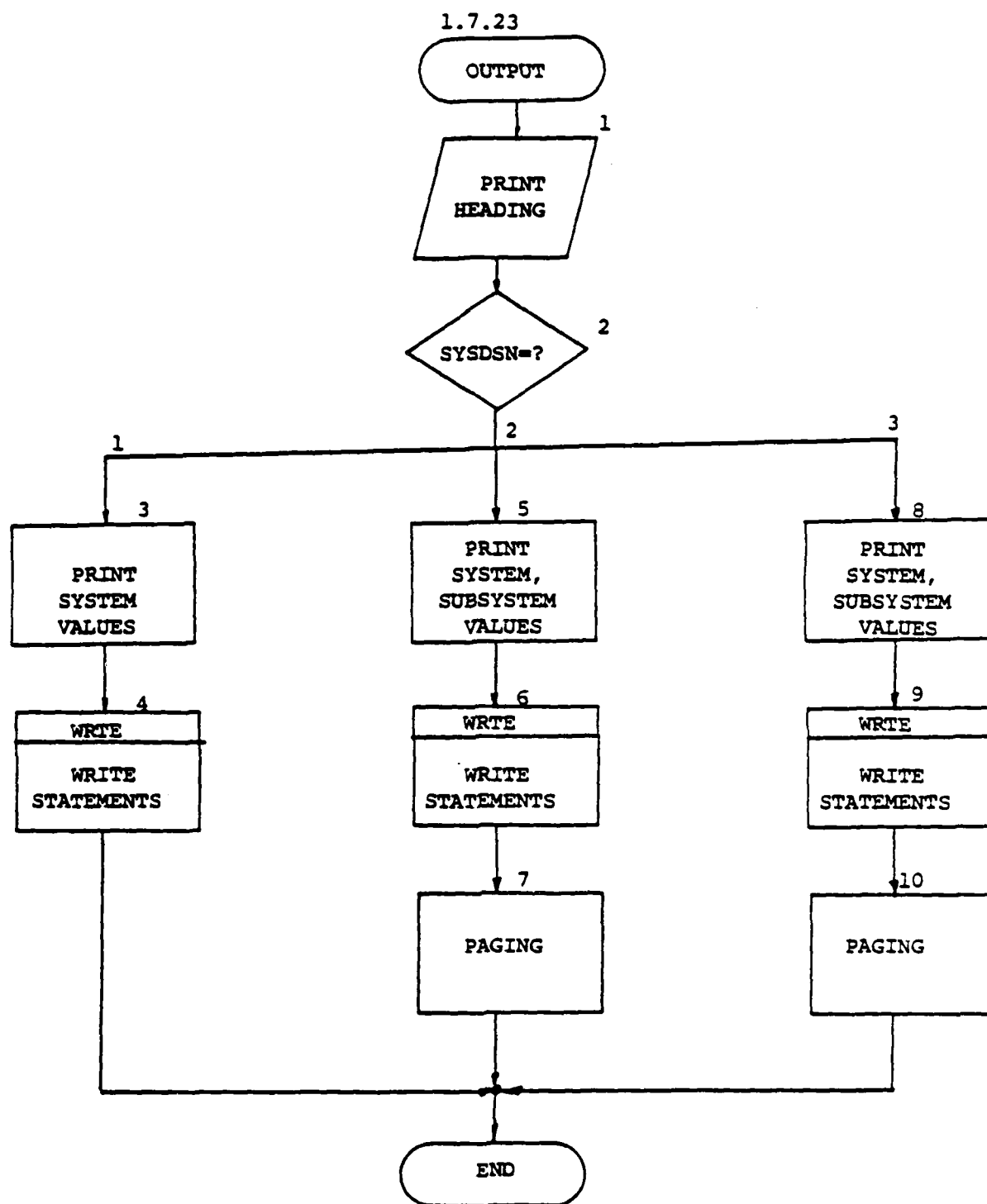
- .1 The RATING values are normalized.
- .2-.3 Read in parameter values from file VPARM.
- .4-.8 Eliminate parameter data systems that have been eliminated.
- .9-.10 Determine the smallest parameter value for each parameter between the systems or subsystems.
- .11-.12 Determine the biggest parameter value for each parameter between the systems or subsystems.
- .13-.14 If any parameter data is equal to zero skip over calculations.
- .15-.18 Calculate the utility value using the largest and smallest value.
- .19 Calculate the rated utility value using the RATING value.
- .20 Add the rated utility values for each system or subsystem.
- .21 Print headings.
- .22-.23 Call overlay containing module OUTPUT that prints table.
- .24 Write system values out on file SYSNUM.



(1.7.23) Tabular OUTPUT

This module prints the system values calculated from the utility values that determines the "best" system or subsystem.

- .1 Print headings.
- .2 SYSDSN indicates how the problem is structured (Case).
- .3-.4 Print table.
- .5-.6 Print table.
- .7 After a number of lines start a new page.
- .8-.9 Print table.
- .10 After a number of lines start a new page.

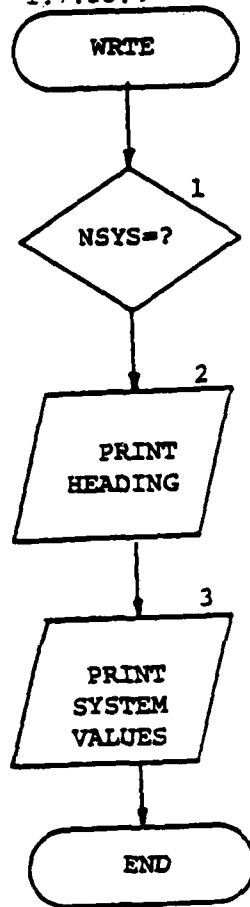


(1.7.23.4) WRiTE out system values

This module contains the write and format statements for the utility table.

- .1 Choose appropriate write and format statement dependent on the number of systems, NSYS.
- .2 Print heading.
- .3 Print the normalized system values, SYSNRM and the optimum system.

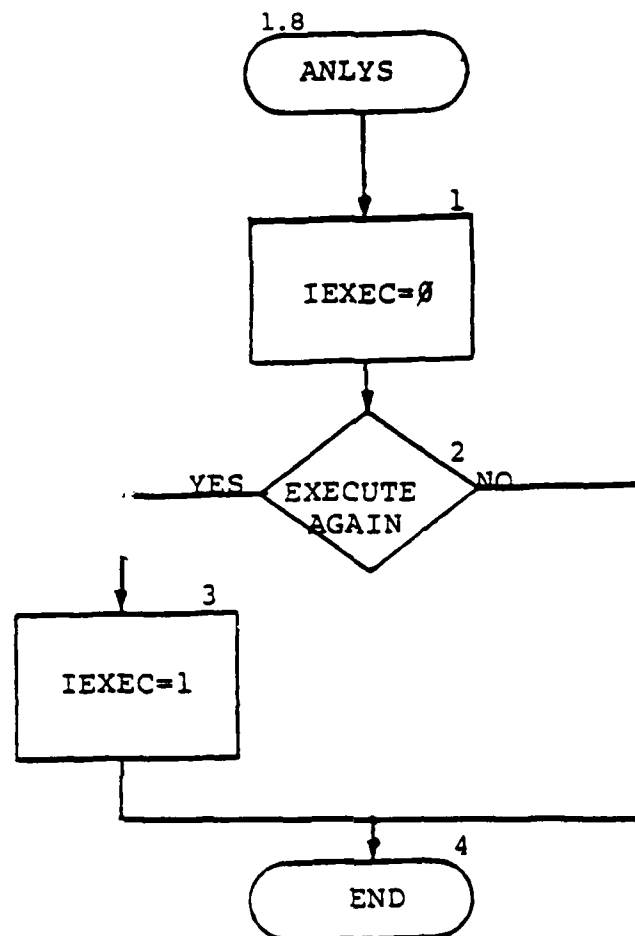
1.7.23.4



(1.8) ANaLYSis

This module allows the user to execute the program over and over again, this is useful if one is performing an uncertainty analysis.

- .1 Set IEXEC to zero.
- .2 Ask the user if he or she wishes to execute the program again.
- .3 If yes, then set IEXEC to zero.
- .4 Return to the main overlay.



APPENDIX H--REFERENCES

APPENDIX H REFERENCES

1. FORTTRAN Extended Version 4 Reference Manual, Control Data Corp., Sunnyvale, CA, 1978.
2. INTERCOM Version 5 Reference Manual, Control Data Corp., Sunnyvale, CA, 1980.
3. Introduction to ASD Computer Center, Wright-Patterson Air Force Base, OH, September, 1979.
4. NOS/BE Version 1 Reference Manual, Control Data Corp., Sunnyvale, CA, 1978.
5. Standardized Development of Computer Software. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.
6. Standardized Development of Computer Software, Part II, Standards. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1979.
7. Van Tassel, Dennie, Program Style, Design, Efficiency, Debugging, and Testing. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1978.
8. TEKTRONIX/PLOT-10 Terminal Control System. Tektronix, Inc., Beaverton, Oregon, 1973.
9. TEKTRONIX/PLOT-10 Advanced Graphing II, Tektronix, Inc., Beaverton, Oregon, 1973.

APPENDIX I GLOSSARY

APPENDIX I GLOSSARY

This is a listing which contains the names of program modules, variables, constants, and any special terms used in this document.

ABV The array that contains the upper limit for the parameter value.

AVE A variable that indicates the average value of the ratings according to scenario.

BELW The array that contains the lower limit for the parameter value.

BIG Indicates the biggest value.

CHPRM The integer array that contains the chosen parameters.

DBNAME Data base name; an integer array that contains the alphanumeric name of the PP data base.

ELIM The array that contains the parameter numbers that are to be eliminated.

HIR The array that contains the higher limit for the parameter value.

IADD An integer variable that is used as a command to indicate the add mode.

IANSR The integer variable that indicates a yes or no answer.

IBAUD The integer value indicating the baud rate to the TEKTRONIX PLOT-10 software.

IBLANK An integer value that contains the blank character.

ICHAR An integer value that contains alphanumeric characters.

ICHARA An integer value that contains the alphanumeric character "A".

ICHARB An integer value that contains the alphanumeric character "B".

ICHOIC The integer value that indicates the choice of scenario.

ICOUNT The integer variable that counts the number of times the parameter is chosen.

ICNT The integer variable that counts the number of times the parameter is chosen.

ICRTE Integer variable that is used as a command to indicate the create mode.

IDUM1 An integer unit number for I/O for the file DUM1.

IDUM2 An integer unit number for I/O for the file DUM2.

IEND An integer variable that is used as a command to indicate the end of the program.

IEXEC An integer variable that indicates whether or not the program should execute again.

IHDR An integer unit number for I/O for the file HEADER.

ILIST An integer variable that is used as a command to indicate the list mode.

IMNIN The integer variable that is used as a command to indicate the manual input of the data.

INDEX An integer value used in INDEX arrays. If SYSDSN=1, INDEX=1; if SYSDSN=2, INDEX=NSUB1; if SYSDSN=3, INDEX=NSUB1*NSUB2.

INDEX1 The integer value for indexing.

INDICE The integer value that holds the value for an indice.

INO An integer value that contains the character no.

INVERT The inverted parameter indicator which is a binary 1, if the parameter is to be inverted; 0, if not.

IOPT An integer value that indicates which optimal system has been chosen.

IPP An integer unit number for I/O for the file PP.

IPPH An integer unit number for I/O for the PP part of the file HEADER.

IQUALT An integer array that contains the alphanumeric names for the rated parameters.

IRANGE The integer array that contains the chosen ratings for the rated parameter.

IRNG An integer value that indicates if range of elimination is above, below, or in between the parameter.

ISCEN The integer variable that indicates the number of scenarios.

ISWNO An integer array that contains a 1 if the parameter has no limits and 0 if it does have limits.

ISYS An integer unit number for I/O for the SYSNAM file.

ITTYIN An integer unit number for the terminal input.

ITYOUT An integer unit number for the terminal output.

IUPP An integer unit number for I/O for the file UPP.

IUPPH An integer unit number for I/O for the UPP part of the file HEADER.

IVBIN An integer value to test if the parameter has already been eliminated.

IVPARAM An integer unit number for I/O for the VPARAM file.

IYES An integer value that contains the character yes.

JTERM An integer value signifying terminal type to the TEKTRONIX PLOT-10 software.

LINE The integer variable that counts the number of lines of output.

MATRIX The array that contains 0 and 1 values. A 0 value indicates that a parameter is to be eliminated. It is used to multiply the PARAM array.

NCALL The integer value that indicates the value for the indice when new scenarios are to be added.

NOLD An integer value that indicates the value for the indice when new scenarios are to be added.

NOPRM The number of parameters chosen to participate in the decision process.

NPAGE The integer value that contains the number of pages.

NPARM Number of parameters; an integer value that indicates the number of parameters.

NSCEN Number of scenarios; an integer value that indicates the number of scenarios.

NSUB1 Number of subsystems level 1; an integer value that indicates the number of level 1 subsystems.

NSUB2 Number of subsystems level 2; an integer value that indicates the largest number of level 2 subsystems for each level 1 subsystem.

NSYS Number of systems; an integer value that shows the number of systems.

NUM Numerical parameter which is given a binary value of 1, if numerical parameter; 0, if not.

PARNAM Parameter names; an integer array that contains the alphanumeric names of the parameters.

PAVE The array of the averages of the ratings.

PSUM A variable used to sum up the rating values for each parameter.

RATED Rated parameter; binary-defined as 1 if rated parameter; 0, if not.

RATING Ratings; an array that contains the user's ratings.

RCALL The real value of NCALL.

RLOWR The array that contains the lower limit for the parameter value.

RSCEN Real number of scenarios; the real value of NSCEN.

RTD The array that contains the chosen ratings for the rated parameter.

RTING Common block label.

SB1SET Subsystem level 1 setname; an integer array that contains the setname of the subsystem level 1 setname.

SB2SET Subsystem level 2 setname; an integer array that contains the setname of the subsystem level 2 setname.

SCNDSC Scenario descriptions; an integer array that contains the alphanumeric names of the scenarios.

SMALL Indicates the smallest value.

SUB1NM Subsystem level 1 names; an integer array that contains the alphanumeric names of the level 1 subsystems.

SUB2NM Subsystem level 2 names; an integer array that contains the alphanumeric names of the level 2 subsystems.

SUM An array used to sum up the ratings by scenarios.

SUMM Used to sum rating values.

SW1 A Boolean variable.

SW2 A Boolean variable.

SYSDSN System design indicator; an integer value that indicates the description of the problem.
 1--indicates systems to be included.
 2--indicates systems and level 1 subsystems.
 3--indicates systems and levels 1 and 2 subsystems.

SYSNAM System names; an integer array containing the alphanumeric names of the systems.

SYSNRM System values normalized; an array that contains the normalized system values that indicated the relative optimal system.

SYSSET System setname; an integer array containing the setname of the systems.

VPARAM The array that contains the inverted parameters.

UTIL The array that contains the utility values for the systems and subsystems.

YN Qualitative parameter indicator; a binary 1 indicates, yes, the system or subsystems have that quality; a binary 0 indicates, no, the quality is not present.

APPENDIX J - LISTINGS

PROCEDURE FILE PPROC1

.PROC,PPPP,FHDR,FPP.
REQUEST,FHCR,*PF.
REQUEST,FPP,*PF.
ATTACH,TEKLIB,TEKLIB,SN=ASD,ID=LIBRARY,CY=999.
LIBRARY,TEKLIB.
ATTACH,PPP,PPP,CY=1.
PPP.
REWIND,HEADER.
REWIND,PP.
COPY,HEADER,FHDR.
COPY,PP,FPP.
RETURN,HEADER,PP,PPP,TEKLIB.

PROCEDURE FILE PPROC2

.PROC,PPPP,FHDR,CY1,FPP,CY2.
ATTACH,HEADER,FHDR,CY=CY1.
ATTACH,PP,FPP,CY=CY2.
ATTACH,PPP,PPP,CY=1.
PPP.
RETURN,PPP,PP,HEADER.

PROCEDURE FILE UPROC1

.PROC,UPPPP,OLD1,CY1,OLD2,CY2,NEW1,NEW2.
ATTACH,HDP,OLD1,CY=CY1.
ATTACH,UPP,OLD2,CY=CY2.
ATTACH,UPPP,UPPP,CY=1.
ATTACH,TEKLIB,TEKLIB,ID=LIBRARY,SN=ASD,CY=999.
LIBRARY,TEKLIB.
REQUEST,NEW1,*PF.
REQUEST,NEW2,*PF.
UPPP.
REWIND,DUM1.
REWIND,DUM2.
COPY,DUM1,NEW1.
COPY,DUM2,NEW2.
RETURN,HDP,UPP,UPPP,DUM1,DUM2.

PROCEDURE FILE UPROC2

.PROC,UPPPP,OLD1,CY1,NEW1,NEW2.
ATTACH,HDR,OLD1,CY=CY1.
REQUEST,NEW1,*PF.
REQUEST,NEW2,*PF.
ATTACH,UPPP,UPPP,CY=1.
ATTACH,TEKLIB,TEKLIB,SN=ASD,ID=LIBRARY,CY=999.
LIBRARY,TEKLIB.
UPPP.
REWIND,DUM1.
REWIND,DUM2.
COPY,DUM1,NEW2.
COPY,DUM2,NEW1.
RETURN,HDR,UPP,UPPP,DUM1,DUM2,TEKLIB.

PROCEDURE FILE IPROC

```
.PROC,SSMPP,FL1,CY1,FL2,CY2,FL3,CY3.  
ATTACH,PP,FL1,CY=CY1.  
ATTACH,UPP,FL2,CY=CY2.  
ATTACH,HEADER,FL3,CY=CY3.  
ATTACH,SSMP,SSMP,CY=1.  
SSMP.  
RETURN,PP,UPP,SSMP,SSM,VPARM.  
REWIND,SYSNUM.  
ATTACH,TEKLIB,TEKLIB,SN=ASD,ID=LIBRARY,CY=999.  
LIBRARY,TEKLIB.  
ATTACH,GRAFX,GRAFX,CY=1.  
GRAFX.  
RETURN,GRF,GRAFX.  
ATTACH,TGRAFX,TGRAFX,CY=1.  
TGRAFX.  
RETURN,TGRAFX,GRF,HEADER,SYSNUM,TEKLIB.  
LIBRARY.
```

```

      PROGRAM PPP(INPUT,OUTPUT,HEADER,DUM,PP,TAPE1C=HEADER,TAPE1I=PP,
      &TAPE2=OUTPUT,TAPE5=INPUT,TAPE13=DUM)
C*****
C      PARAMETER PACKAGE PROGRAM
C      DIRECTS THE EXECUTION
C      OF THE CHOSEN
C      SUBROUTINE
C*****

      COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,IPP,IDUM
      COMMON/DB/DBNAME(4)
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
      &SYSSET(4),SB1SET(4),SB2SET(4),
      &SYSNAM(10,4),SUB1NM(20,4),SUB2NM(20,21,4),
      &PARNAM(40,4),NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      &INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
      &RATED,YN,PARNAM
      DATA IEND/3HEND/,ILST/4HLIST/,ICRTE/4HCRTE/,
      &IMNIN/4HMININ/

C
C      TEKTRONIX PAGING
      WRITE(ITYOUT,1)
      WRITE(ITYOUT,4)
      WRITE(ITYOUT,2)
      READ(ITTYIN,3)IANSR

C
100  LINE=34
      CALL PAGER (LINE,0)
      IF(IANSR.EQ.IEND) GO TO 1000
      IF(IANSR.EQ.ICRTE) CALL CREATE

C
C *** READ IN /HDR1/ ***
C
      REWIND IHDR
      CALL READR1
      IF (IANSR.EQ.IMNIN) CALL MANUAL
      IF (IANSR.EQ.ILST) CALL LIST
      WRITE(ITYOUT,2)
      READ(ITTYIN,3)IANSR
      GO TO 100

C
C *** FORMAT STATEMENTS ***
C
1    FORMAT (1X//,1X,70(1H*),/,1X,1H*,22X,25HPARAMETER PACKAGE PROGRAM,
      & 21X,1H*,/,1X,70(1H*),//)
2    FORMAT (1X,10HCOMMAND-->)
3    FORMAT(A4)
4    FORMAT (1X,19HCOMMANDS AVAILABLE:/,5X,
      & 28HCRTE - CREATES FILE OF NAMES,/,5X,
      & 49HMININ - ALLOWS MANUAL ENTERING OF PARAMETER VALUES,/,5X,
      & 43HLIST - LISTS OUT NAMES AND PARAMETER VALUES,/,5X,
      & 19HEND - ENDS PROGRAM,//)
1000 CONTINUE
      STOP

```

```

END
C
C*****
BLOCK DATA
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,IPP,IDUM
DATA ITTYIN,ITYOUT,IHDR,IPP,IDUM/5,2,10,11,13/
END
C*****
C
C
C*****
SUBROUTINE CREATE
C
C      -CREATES FILE OF NAMES, HEADER
C
C*****
C
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,IPP,IDUM
COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARN,
&SYSSET(4),SB1SET(4),SB2SET(4),
&SYSNAM(10,4),SUB1NM(20,4),SUB2NM(20,21,4),
&PARNAM(40,4),NUM(40),RATED(40),YN(40),INVERT(40),
&INDEX
C
COMMON /DB/ DBNAME(4)
INTEGER DBNAME,SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,
&SUB1NM,SUB2NM,PARNAM,RATED,YN
DATA SW1/0/,SW2/0/
DATA INO/2HNO/,IYES/3HYES/
C
C *** READ IN DATA BASE NAMES ***
WRITE(ITYOUT,1)
READ(ITTYIN,2)(DBNAME(M),M=1,4)
C *** READ IN NUMBER OF SYSTEMS ***
WRITE(ITYOUT,3)
READ(ITTYIN,4)NSYS
C *** READ IN NUMBER OF LEVEL 1 SUBSYSTEMS ***
WRITE(ITYOUT,5)
READ(ITTYIN,4)NSUB1
IF(NSUB1.EQ.0)SW1=1
C *** READ IN NUMBER OF LEVEL 2 SUBSYSTEMS ***
WRITE(ITYOUT,6)
READ(ITTYIN,4)NSUB2
IF(NSUB2.EQ.0)SW2=1
C *** READ IN NUMBER OF PARAMETERS ***
WRITE(ITYOUT,7)
READ(ITTYIN,4)NPARN
C *** READ IN SYSTEM SETNAME AND SYSTEM NAMES ***
WRITE(ITYOUT,8)
READ(ITTYIN,2)(SYSSET(M),M=1,4)
DO 100 I=1,NSYS
WRITE(ITYOUT,9)(SYSSET(M),M=1,4),I
READ(ITTYIN,2)(SYSNAM(I,M),M=1,4)
100 CONTINUE

```

```

LINE=34
CALL PAGER (LINE,0)
IF(SW1.EQ.1)GO TO 210
C *** READ IN LEVEL 1 SUBSYSTEM SETNAME AND SUBSYSTEM NAMES ***
WRITE(ITTYOUT,10)
READ(ITTYIN,2)(SB1SET(M),M=1,4)
DO 200 I=1,NSUB1
    WRITE (ITYOUT,11) (SB1SET(M),M=1,4),I
    READ(ITTYIN,2)(SUB1NM(I,M),M=1,4)
200    CONTINUE
IF(SW2.EQ.1) GO TO 210
C *** READ IN LEVEL 2 SUBSYSTEM SETNAME AND SUBSYSTEM NAMES ***
WRITE(ITTYOUT,12)
READ(ITTYIN,2)(SB2SET(M),M=1,4)
DO 300 I=1,NSUB1
    DO 300 I1=1,NSUB2
        WRITE (ITYOUT,13) (SB2SET(M),M=1,4),
        & (SUB1NM(I,M),M=1,4),I1
        READ (ITTYIN,2) (SUB2NM(I,I1,M),M=1,4)
300    CONTINUE
LINE=34
CALL PAGER (LINE,0)
C *** READ IN PARAMETER NAMES ***
210    DO 400 I=1,NPARM
        WRITE (ITYOUT,14) I
        READ(ITTYIN,2)(PARNAM(I,M),M=1,4)
400    CONTINUE
LINE=34
CALL PAGER (LINE,0)
DO 501 I=1,NPARM
    RATED(I)=0
    YN(I)=0
    NUM(I)=0
    INVERT(I)=0
501    CONTINUE
C *** WHAT TYPE PARAMETER AND DOES IT NEED TO BE INVERTED ***
DO 500 I=1,NPARM
111    CALL PAGER (LINE,2)
        WRITE(ITTYOUT,15)(PARNAM(I,M),M=1,4)
        READ(ITTYIN,16)NANSR
        LINE=LINE+2
        IF(NANSR.EQ.INO) GO TO 515
        NUM(I)=1
        GO TO 550
515    CALL PAGER (LINE,2)
        WRITE(ITTYOUT,17) (PARNAM(I,M),M=1,4)
        READ(ITTYIN,16)KANSR
        LINE=LINE+2
        IF(KANSR.EQ.INO) GO TO 520
        RATED(I)=1
        GO TO 560
520    CALL PAGER (LINE,2)
        WRITE(ITTYOUT,18) (PARNAM(I,M),M=1,4)
        READ(ITTYIN,16)LANSR

```

```

LINE=LINE+2
  IF(LANSR.EQ.INO) GO TO 525
  YN(I)=1
  GO TO 560
525  CALL PAGER (LINE,2)
     WRITE(ITYOUT,19)
LINE=LINE+2
  GO TO 111
550  CALL PAGER (LINE,4)
     WRITE(ITYOUT,20)(PARNAM(I,M),M=1,4)
     READ(ITYIN,16)MANSR
     IF(MANSR.EQ.IYES)INVERT(I)=1
LINE=LINE+4
560  MANSR=0
     KANSR=0
     LANSR=0
     MANSR=0
500  CONTINUE
C
C *** INPUT INTO HEADER ***
C
  REWIND IHDR
  WRITE(IHDR,23)(DBNAME(M),M=1,4)
  IF(SW1.EQ.1.AND.SW2.EQ.1)SYSDSN=1
  IF(SW1.EQ.0.AND.SW2.EQ.1)SYSDSN=2
  IF(SW1.EQ.0.AND.SW2.EQ.0)SYSDSN=3
  WRITE(IHDR,21)SYSDSN
  WRITE(IHDR,22)NSYS
  WRITE(IHDR,22)NSUB1
  WRITE(IHDR,22)NSUB2
  WRITE(IHDR,22)NPARM
  WRITE(IHDR,23)(SYSSET(M),M=1,4)
  IF(SYSDSN.GE.2)WRITE(IHDR,23)(SB1SET(M),M=1,4)
  IF(SYSDSN.EQ.3)WRITE(IHDR,23)(SB2SET(M),M=1,4)
  DO 600 I=1,NSYS
    WRITE(IHDR,23)(SYSNAM(I,J),J=1,4)
600  CONTINUE
    IF(SW1.EQ.1) GO TO 605
    DO 700 I=1,NSUB1
      WRITE(IHDR,23)(SUB1NM(I,J),J=1,4)
700  CONTINUE
      IF(SW2.EQ.1) GO TO 605
      DO 800 I=1,NSUB1
        DO 800 J=1,NSUB2
          WRITE(IHDR,23)(SUB2NM(I,J,M),M=1,4)
800  CONTINUE
605  DO 900 I=1,NPARM
      WRITE(IHDR,24)(PARNAM(I,J),J=1,4),NUM(I),RATED(I),YN(I),
&  INVERT(I)
900  CONTINUE
C
C *** FORMAT STATEMENTS ***
C
1    FORMAT(1X,35HENTER DATA BASE NAME (MAX 16 CHARS)/)

```

```

2  FORMAT(4A4)
3  FORMAT(1X,34HENTER NO. OF SYSTEMS (2 DIGIT NO.))/)
4  FORMAT(I2)
5  FORMAT(1X,45HENTER NO. OF LEVEL 1 SUBSYSTEMS (2 DIGIT NO.))/)
6  FORMAT(1X,45HENTER NO. OF LEVEL 2 SUBSYSTEMS (2 DIGIT NO.))/)
7  FORMAT(1X,37HENTER NO. OF PARAMETERS (2 DIGIT NO.))/)
8  FORMAT(1X,37HENTER SYSTEM SETNAME (MAX. 16 CHARS.))/)
9  FORMAT(1X,15HENTER NAME FOR ,4A4,5H NO. ,I2,16H (MAX. 16 CHARS))/)
10 FORMAT(1X,47HENTER LEVEL 1 SUBSYSTEM SETNAME (MAX. 16 CHARS))/)
11 FORMAT(1X,14HENTER NAME FOR,1X,4A4,6H, NO. ,I2,2X,
    & 16H(MAX. 16 CHARS.))/)
12 FORMAT(1X,38HENTER LEVEL 2 SUBSYSTEM SETNAME (MAX. ,
    & 9H16CHARS.))/)
13 FORMAT(1X,14HENTER NAME FOR,1X,4A4,3H , ,4A4,6H, NO. ,I2,2X,
    & 16H(MAX. 16 CHARS.))/)
14 FORMAT(1X,28HENTER NAME FOR PARAMETER NO.,I2,/)
15 FORMAT(1X,3HIS ,4A4,33H A NUMERICAL PARAMETER ? (YES/NO))/)
16 FORMAT(A3)
17 FORMAT (1X,3HIS ,4A4,29H A RATED PARAMETER ? (YES/NO))/)
18 FORMAT(1X 3HIS ,4A4,35H A QUALITATIVE PARAMETER ? (YES/NO))/)
19 FORMAT(1X,42HERRGR -- PARAMETER MUST BE CLASSIFIED AS A,
    & 3CHNUMERIC, RATED, OR QUALITATIVE/1X,12HPARAMETER - ,
    & 9HTRY AGAIN/)
20 FORMAT (1X,45HFOR NUMERICAL PARAMETERS, THE LOWEST VALUE IS,/,
    & 1X,28HCONSIDERED THE "BEST" VALUE.,/,1X,
    & 32H DO YOU WISH TO INVERT THIS FOR ,4A4,11H ? (YES/NO))/)
21 FORMAT(I1)
22 FORMAT(I2)
23 FORMAT(4A4)
24 FORMAT(4A4,I1,I1,I1,I1)
    RETURN
    END

```

```

C
C*****
C      SUBROUTINE READR1
C
C      -READS IN HEADER FILE
C
C*****
C
C      COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,IPP,IDUM
C      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARN,SYSSET(4),
C      & SB1SET(4),SB2SET(4),SYSNAM(10,4),SUB1NM(20,4),SUB2NM(20,21,4),
C      & PARNAM(40,4),NUM(40),RATED(40),YN(40),INVERT(40),INDEX
C      COMMON/DB/DBNAME(4)
C      INTEGER DBNAME
C      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
C      & PARNAM,YPARNAM
C
C      *** READ IN HEADER ***
C      RE=IND IHDR
C
C      READ(IHDR,3)(DBNAME(M),M=1,4)
C      READ(IHDR,1)SYSDSN

```

```

READ(IHDR,2)NSYS
READ(IHDR,2)NSUB1
READ(IHDR,2)NSUB2
READ(IHDR,2)NPARM
READ(IHDR,3)(SYSSET(I),I=1,4)
IF(SYSDSN.GE.2)READ(IHDR,3)(SB1SET(I),I=1,4)
IF(SYSDSN.EQ.3)READ(IHDR,3)(SB2SET(I),I=1,4)
DO 100 I=1,NSYS
READ(IHDR,3)(SYSNAM(I,J),J=1,4)
100 CONTINUE
IF(NSUB1.EQ.0) GO TO 400
DO 200 I=1,NSUB1
READ(IHDR,3)(SUB1NM(I,J),J=1,4)
200 CONTINUE
IF(NSUB2.EQ.0) GO TO 400
DO 300 K=1,NSUB1
DO 301 I=1,NSUB2
READ(IHDR,3)(SUB2NM(K,I,J),J=1,4)
301 CONTINUE
300 CONTINUE
400 CONTINUE
DO 500 I=1,NPARM
READ(IHDR,4)(PARNAM(I,J),J=1,4),NUM(I),RATED(I),YN(I),INVERT(I)
500 CONTINUE
C
C *** CALCULATE INDEX ***
C
IF(NSUB1.EQ.0)GO TO 800
IF(NSUB2.EQ.0)GO TO 900
INDEX=NSUB1*NSUB2
GO TO 1000
800 INDEX=1
GO TO 1000
900 INDEX=NSUB1
1000 CONTINUE
C
C *** FORMAT STATEMENTS ***
C
1 FORMAT (I1)
2 FORMAT (I2)
3 FORMAT (4A4)
4 FORMAT (4A4,I1,I1,I1,I1)
RETURN
END
C
C
C*****
SUBROUTINE MANUAL
C
C - ALLOWS ENTERING OF PARAMETER VALUES, CREATES FILE PP
C
C*****
C
COMMON /UNITNO/ ITTYIN,ITYOUT,IHDR,IPP,IDUM

```

```

COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
& SYSSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),SUB1NM(20,4),
& SUB2NM(20,21,4),PARNAM(40,4),NUM(40),RATED(40),YN(40),
& INVERT(40),INDEX
DIMENSION PARAM(85,10)
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
& PARNAM,RATED,YN
DATA IBLANK/1H /

C
  REWIND IPP
  LINE=0

C
  GO TO (1000,2000,3000),SYSDSN

C
1000 DO 100 I=1,NPARM
      CALL PAGER (LINE,4)
      WRITE (ITYOUT,4) (PARNAM(I,M),M=1,4)
      IF (NUM(I).EQ.1) WRITE (ITYOUT,1)
      IF (RATED(I).EQ.1) WRITE (ITYOUT,2)
      IF (YN(I).EQ.1) WRITE (ITYOUT,3)
      LINE=LINE+4
      IF (RATED(I).EQ.1) LINE=LINE+1
      DO 150 J=1,NSYS
          CALL PAGER (LINE,2)
          WRITE (ITYOUT,5) (SYSNAM(J,M),M=1,4)
          READ (ITTYIN,*) PARAM(1,J)
          LINE=LINE+2
150    CONTINUE
      WRITE (IPP,*) (PARAM(1,J),J=1,NSYS)
100    CONTINUE
      GO TO 9000

C
C
2000 DO 200 I=1,NPARM
      CALL PAGER (LINE,4)
      WRITE (ITYOUT,4) (PARNAM(I,M),M=1,4)
      IF (YN(I).EQ.1) WRITE (ITYOUT,3)
      IF (RATED(I).EQ.1) WRITE (ITYOUT,2)
      IF (NUM(I).EQ.1) WRITE (ITYOUT,1)
      LINE=LINE+4
      IF (RATED(I).EQ.1) LINE=LINE+1
      DO 250 K=1,INDEX
          DO 250 J=1,NSYS
              CALL PAGER (LINE,2)
              WRITE (ITYOUT,5) (SYSNAM(J,M),M=1,4),
& (SUB1NM(K,M),M=1,4)
              READ (ITTYIN,*) PARAM(K,J)
              LINE=LINE+2
250          CONTINUE
          DO 260 K=1,INDEX
              WRITE (IPP,*) (PARAM(K,J),J=1,NSYS)
260          CONTINUE
200    CONTINUE
      GO TO 9000

```



```

C
C
3000 DO 300 I=1,NPARN
      CALL PAGER (LINE,4)
      WRITE (ITYOUT,4) (PARNAM(I,M),M=1,4)
      IF (NUM(I).EQ.1) WRITE (ITYOUT,1)
      IF (PATED(I).EQ.1) WRITE (ITYOUT,2)
      IF (YN(I).EQ.1) WRITE (ITYOUT,3)
      LINE=LINE+4
      IF (RATED(I).EQ.1) LINE=LINE+1
      DO 350 J=1,NSYS
      IND=0
        DO 350 K=1,NSUB1
          DO 350 L=1,NSUB2
            CALL PAGER (LINE,2)
            WRITE (ITYOUT,6) (SYSNAM(J,M),M=1,4),
      & (SUB1NM(K,M),M=1,4), (SUB2NM(K,L,M),M=1,4)
            IND=IND+1
            READ (ITTYIN,*) PARAM (IND,J)
            LINE=LINE+2
350    CONTINUE
      DO 360 K=1,INDEX
        WRITE (IPP,*) (PARAM(K,JJ),JJ=1,NSYS)
360    CONTINUE
300    CONTINUE
9000 CONTINUE
C
C*** FORMAT STATEMENTS ***
C
1    FORMAT (1H ,44HTHIS IS A NUMERIC PARAMETER - ENTER A NUMBER/)
2    FORMAT (1H ,44HTHIS IS A RATED PARAMETER - RATE AS FOLLOWS:/,
      & 1X,48H1-EXCELLENT, 2-GOOD, 3-FAIR, 4-POOR, 5-VERY POOR,/)
3    FORMAT (1H ,50HTHIS IS A QUALITATIVE PARAMETER - ENTER 1 FOR YES,
      & 9H 0 FOR NO,/)
4    FORMAT (1H ,4A4,/)
5    FORMAT (1X,4A4,4H -- ,4A4,/)
6    FORMAT (1X,4A4,4H -- ,4A4,4H -- ,4A4,/)
C
      RETURN
      END
C
C*****
      SUBROUTINE LIST
C
C      - PRINTS OUT PARAMETERS AND THEIR VALUES
C
C*****
C
      COMMON /UNITNO/ ITTYIN,ITYOUT,IHDR,IPP,IDUM
      COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARN,SYSSET(4),SB1SET(4),
      & SB2SET(4),SYSNAM(10,4),SUB1NM(20,4),SUB2NM(20,21,4),
      & PARNAM(40,4),NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
      & SUB2NM,RATED,YN,PARNAM

```

```

C      REWIND IPP
C
      LIM=18
      IF (NSYS.LE.9) LIM=LIM-3
      IF (NSYS.LE.6) LIM=LIM-3
      IF (NSYS.LE.3) LIM=LIM-3
C
      GO TO (1000,2000,3000), SYSDSN
C
1000  CONTINUE
      II=1
      DO 100 I=1,NPARN
      LINE=34
      IF (I.NE.1) CALL PAGER (LINE,0)
        WRITE (ITYOUT,2) (PARNAM(I,M),M=1,4)
        CALL WRTOUT (I,II)
      WRITE (ITYOUT,4)
100   CONTINUE
C
      RETURN
C
2000  CONTINUE
      DO 200 I=1,NPARN
      LINE=34
      CALL PAGER (LINE,0)
        WRITE (ITYOUT,2) (PARNAM(I,M),M=1,4)
      LINE=LINE+5
      DO 210 II=1,INDEX
      CALL PAGER (LINE,LIM)
        IF (LINE.EQ.0) WRITE (ITYOUT,1)
        IF (LINE.EQ.0) LINE=1
        WRITE (ITYOUT,3) (SUBINM(II,M),M=1,4)
        CALL WRTOUT (I,II)
        WRITE (ITYOUT,4)
      LINE=LINE+LIM
210   CONTINUE
200   CONTINUE
C
      RETURN
C
3000  CONTINUE
      LIM=LIM+5
      DO 300 I=1,NPARN
      LINE=34
      CALL PAGER (LINE,0)
        WRITE (ITYOUT,2) (PARNAM(I,M),M=1,4)
      LINE=LINE+5
      INDX=0
      DO 310 J=1,NSUB1
      CALL PAGER (LINE,LIM)
        IF (LINE.EQ.0) WRITE (ITYOUT,1)
        IF (LINE.EQ.0) LINE=1

```

```

        WRITE (ITYOUT,3) (SUB1NM(J,M),M=1,4)
        DO 320 K=1,NSUB2
            WRITE (ITYOUT,3) (SUB2NM(J,K,M),M=1,4)
            INDX=INDX+1
            II=INDX
            CALL WRTOUT(I,II)
            WRITE (ITYOUT,4)
            LINE=LINE+LIM
320     CONTINUE
310     CONTINUE
300     CONTINUE
C
C
        RETURN
C
C*** FORMAT STATEMENTS ***
C
1     FORMAT (1X,70(1H*))
2     FORMAT (1X,70(1H*)/1X,1H*,68X,1H*/1X,1H*,26X,4A4,26X,1H*/
C 1X,1H*,68X,1H*/1X,70(1H*))
3     FORMAT (1X,1H*,68X,1H*/1X,1H*,26X,4A4,26X,1H*/
C 1X,1H*,68X,1H*/1X,70(1H*))
4     FORMAT (1X,1H*,68X,1H*/1X,70(1H*))
C
        END
C
C
C*****
        SUBROUTINE WRTOUT (I,II)
C
C        - PERFORMS PART OF LISTING CHORES
C*****
C
        COMMON /UNITNO/ ITTYIN,ITYOUT,IHDO,IPP,IDUM
        COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(4),
C SB1SET(4),SB2SET(4),SYSNAM(10,4),SUB1NM(20,4),
C SUB2NM(20,21,4),PARNAM(40,4),NUM(40),RATED(40),
C YN(40),INVERT(40),INDEX
        DIMENSION PARAM(85,10),IQUALT(5,3),IPARAM(10)
        INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
C SUB2NM,RATED,YN,PARNAM,PARAM1(10)
C
        DATA IQUALT(1,1)/4HEXCE/,IQUALT(1,2)/4HLEN/,IQUALT(1,3)/4HT /
        DATA IQUALT(2,1)/4HGOOD/,IQUALT(2,2)/4H /,IQUALT(2,3)/4H /
        DATA IQUALT(3,1)/4HFAIR/,IQUALT(3,2)/4H /,IQUALT(3,3)/4H /
        DATA IQUALT(4,1)/4HPOOR/,IQUALT(4,2)/4H /,IQUALT(4,3)/4H /
        DATA IQUALT(5,1)/4HVEPY/,IQUALT(5,2)/4H POO/,IQUALT(5,3)/4HP /
        DATA IYES/3HYES/,INO/2HNO/
C
        READ (IPP,*) (PARAM(II,MT),MT=1,NSYS)
        DO 9000 J=1,NSYS
            IPARAM(J)=PARAM(II,J)
9000    CONTINUE

```

```

      GO TO (1000,1100,1200,1300,1400,1500,1600,1700,1800,1900),NSYS
C
1000  RETURN
C
1100  IF (NUM(I).EQ.1) GO TO 1110
      IF (RATED(I).EQ.1) GO TO 1120
      IF (YN(I).EQ.1) GO TO 1130
      RETURN
1110  WRITE (ITYOUT,1) ((SYSNAM(M,MM),MM=1,4),M=1,NSYS),
      & (PARAM(II,ML),ML=1,NSYS)
      RETURN
1120  WRITE (ITYOUT,2) ((SYSNAM(M,MM),MM=1,4),M=1,NSYS),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,NSYS)
      RETURN
1130  DO 1131 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1131  CONTINUE
      WRITE (ITYOUT,3) ((SYSNAM(M,MM),MM=1,4),M=1,NSYS),
      & (PARAM1(L),L=1,NSYS)
      RETURN
C
1200  IF (NUM(I).EQ.1) GO TO 1210
      IF (RATED(I).EQ.1) GO TO 1220
      IF (YN(I).EQ.1) GO TO 1230
      RETURN
1210  WRITE (ITYOUT,4) ((SYSNAM(M,MM),MM=1,4),M=1,NSYS),
      & (PARAM(II,ML),ML=1,NSYS)
      RETURN
1220  WRITE (ITYOUT,5) ((SYSNAM(M,MM),MM=1,4),M=1,NSYS),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,NSYS)
      RETURN
1230  DO 1231 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1231  CONTINUE
      WRITE (ITYOUT,6) ((SYSNAM(M,MM),MM=1,4),M=1,NSYS),
      & (PARAM1(L),L=1,NSYS)
      RETURN
C
1300  IF (NUM(I).EQ.1) GO TO 1310
      IF (RATED(I).EQ.1) GO TO 1320
      IF (YN(I).EQ.1) GO TO 1330
      RETURN
1310  WRITE (ITYOUT,7) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM(II,ML),ML=1,3),(SYSNAM(4,MM),MM=1,4),
      & PAPAM(II,4)
      RETURN
1320  WRITE (ITYOUT,8) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      & (SYSNAM(4,MM),MM=1,4),(IQUALT(IPARAM(4),J),J=1,3)
      RETURN
1330  DO 1331 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO

```

```

        IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1331  CONTINUE
      WRITE (ITYOUT,9) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM1(L),L=1,3), (SYSNAM(4,MM),MM=1,4),
      & PARAM1(4)
      RETURN
C
1400  IF (NUM(I).EQ.1) GO TO 1410
      IF (RATED(I).EQ.1) GO TO 1420
      IF (YN(I).EQ.1) GO TO 1430
      RETURN
1410  WRITE (ITYOUT,10) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM(II,ML),ML=1,3), ((SYSNAM(M,MM),MM=1,4),M=4,5),
      & (PARAM(II,ML),ML=4,5)
      RETURN
1420  WRITE (ITYOUT,11) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      & ((SYSNAM(M,MM),MM=1,4),M=4,5),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=4,5)
      RETURN
1430  DO 1431 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1431  CONTINUE
      WRITE (ITYOUT,12) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM1(L),L=1,3), ((SYSNAM(M,MM),MM=1,4),M=4,5),
      & (PARAM1(L),L=4,5)
      RETURN
C
1500  IF (NUM(I).EQ.1) GO TO 1510
      IF (RATED(I).EQ.1) GO TO 1520
      IF (YN(I).EQ.1) GO TO 1530
      RETURN
1510  WRITE (ITYOUT,13) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM(II,ML),ML=1,3), ((SYSNAM(M,MM),MM=1,4),M=4,6),
      & (PARAM(II,ML),ML=4,6)
      RETURN
1520  WRITE (ITYOUT,14) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      & ((SYSNAM(M,MM),MM=1,4),M=4,6),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=4,6)
      RETURN
1530  DO 1531 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1531  CONTINUE
      WRITE (ITYOUT,15) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM1(L),L=1,3), ((SYSNAM(M,MM),MM=1,4),M=4,6),
      & (PARAM1(L),L=4,6)
      RETURN
C
1600  IF (NUM(I).EQ.1) GO TO 1610
      IF (RATED(I).EQ.1) GO TO 1620
      IF (YN(I).EQ.1) GO TO 1630

```

```

      RETURN
1610 WRITE (ITYOUT,16) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM(II,ML),ML=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      & (PARAM(II,ML),ML=4,6), (SYSNAM(7,MM),MM=1,4),
      & PARAM(II,7)
      RETURN
1620 WRITE (ITYOUT,17) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      & ((SYSNAM(M,MM),MM=1,4),M=4,6),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=4,6),
      & (SYSNAM(7,MM),MM=1,4), (IQUALT(IPARAM(7),J),J=1,3)
      RETURN
1630 DO 1631 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1631 CONTINUE
      WRITE (ITYOUT,18) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM1(L),L=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      & (PARAM1(L),L=4,6), (SYSNAM(7,MM),MM=1,4),
      & PARAM1(7)
      RETURN
C
1700 IF (NUM(I).EQ.1) GO TO 1710
      IF (RATED(I).EQ.1) GO TO 1720
      IF (YN(I).EQ.1) GO TO 1730
      RETURN
1710 WRITE (ITYOUT,19) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM(II,ML),ML=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      & (PARAM(II,ML),ML=4,6),((SYSNAM(M,MM),MM=1,4),M=7,8),
      & (PARAM(II,ML),ML=7,8)
      RETURN
1720 WRITE (ITYOUT,20) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      & ((SYSNAM(M,MM),MM=1,4),M=4,6),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=4,6),
      & ((SYSNAM(M,MM),MM=1,4),M=7,8),
      & ((IQUALT(IPARAM(ML),J),J=1,3),ML=7,8)
      RETURN
1730 DO 1731 L=1,NSYS
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
1731 CONTINUE
      WRITE (ITYOUT,21) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM1(L),L=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      & (PARAM1(L),L=4,6),((SYSNAM(M,MM),MM=1,4),M=7,8),
      & (PARAM1(L),L=7,8)
      RETURN
C
1800 IF (NUM(I).EQ.1) GO TO 1810
      IF (RATED(I).EQ.1) GO TO 1820
      IF (YN(I).EQ.1) GO TO 1830
      RETURN
1810 WRITE (ITYOUT,22) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      & (PARAM(II,ML),ML=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),

```

```

      C (PARAM(II,ML),ML=4,6),((SYSNAM(M,MM),MM=1,4),M=7,9),
      C (PARAM(II,ML),ML=7,9)
      RETURN
1820  WRITE (ITYOUT,23) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      C ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      C ((SYSNAM(M,MM),MM=1,4),M=4,6),
      C ((IQUALT(IPARAM(ML),J),J=1,3),ML=4,6),
      C ((SYSNAM(M,MM),MM=1,4),M=7,9),
      C ((IQUALT(IPARAM(ML),J),J=1,3),ML=7,9)
      RETURN
1830  DO 1831 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1831  CONTINUE
      WRITE (ITYOUT,24) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      C (PARAM1(L),L=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      C (PARAM1(L),L=4,6),((SYSNAM(M,MM),MM=1,4),M=7,9),
      C (PARAM1(L),L=7,9)
      RETURN
C
1900  IF (NUM(I).EQ.1) GO TO 1910
      IF (RATED(I).EQ.1) GO TO 1920
      IF (YN(I).EQ.1) GO TO 1930
      RETURN
1910  WRITE (ITYOUT,25) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      C (PARAM(II,ML),ML=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      C (PARAM(II,ML),ML=4,6),((SYSNAM(M,MM),MM=1,4),M=7,9),
      C (PARAM(II,ML),ML=7,9),((SYSNAM(IG,MM),MM=1,4),
      C PARAM(II,IG)
      RETURN
1920  WRITE (ITYOUT,26) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      C ((IQUALT(IPARAM(ML),J),J=1,3),ML=1,3),
      C ((SYSNAM(M,MM),MM=1,4),M=4,6),
      C ((IQUALT(IPARAM(ML),J),J=1,3),ML=4,6),
      C ((SYSNAM(M,MM),MM=1,4),M=7,9),
      C ((IQUALT(IPARAM(ML),J),J=1,3),ML=7,9),
      C (SYSNAM(IG,MM),MM=1,4),((IQUALT(IPARAM(IG),J),J=1,3)
      RETURN
1930  DO 1931 L=1,NSYS
      IF (PARAM(II,L).EQ.0) PARAM1(L)=INO
      IF (PARAM(II,L).EQ.1) PARAM1(L)=IYES
1931  CONTINUE
      WRITE (ITYOUT,27) ((SYSNAM(M,MM),MM=1,4),M=1,3),
      C (PARAM1(L),L=1,3),((SYSNAM(M,MM),MM=1,4),M=4,6),
      C (PARAM1(L),L=4,6),((SYSNAM(M,MM),MM=1,4),M=7,9),
      C (PARAM1(L),L=7,9),((SYSNAM(IG,MM),MM=1,4),
      C PARAM1(IG)
      RETURN
C
C
C *** FORMAT STATEMENTS ***
C
1  FORMAT (1X,1H*,2(12X,4A4),12X,1H=/,1X,1H*,2(12X,F16.3),12X,1H*)
2  FORMAT (1X,1H*,2(12X,4A4),12X,1H=/,1X,1H*,14X,3A4,16X,3A4,

```

```

C 14X,1H*)
3  FORMAT (1X,1H*,2(12X,4A4),12X,1H*/,1X,1H*,18X,A3,25X,A3,19X,1H*)
4  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*)
5  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*)
6  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*)
7  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,1H*/,1X,1H*,26X,F16.3,26X,1H*)
8  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,1H*/,1X,1H*,28X,3A4,28X,1H*)
9  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*,
C /,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,1H*/,1X,1H*,
C 26X,4A4,26X,1H*/,1X,1H*,32X,A3,33X,1H*)
10  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,1X,
C 1H*,68X,1H*/,1X,1H*,2(12X,4A4),12X,1H*/,1X,1H*,2(12X,F16.3),
C 12X,1H*)
11  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,2(12X,4A4),12X,1H*/,1X,1H*,14X,3A4,16X,
C 3A4,14X,1H*)
12  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*,
C /,1X,1H*,68X,1H*/,1X,1H*,2(12X,4A4),12X,1H*/,1X,1H*,18X,
C A3,25X,A3,19X,1H*)
13  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,1X,
C 1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),
C 5X,1H*)
14  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),
C 3A4,7X,1H*)
15  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),
C A3,12X,1H*)
16  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,1X,
C 1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,
C 1H*/,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,1H*/,1X,1H*,26X,
C F16.3,26X,1H*)
17  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,
C 7X,1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
C 7X,2(3A4,9X),3A4,7X,1H*/,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,
C 26X,1H*/,1X,1H*,28X,3A4,28X,1H*)
18  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),
C A3,12X,1H*/,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,1H*/,1X,1H*,
C 32X,A3,33X,1H*)
19  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,1X,
C 1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),
C 5X,1H*/,1X,1H*,68X,1H*/,1X,1H*,2(12X,4A4),12X,1H*/,1X,1H*,
C 2(12X,F16.3),12X,1H*)
20  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),
C 3A4,7X,1H*/,1X,1H*,68X,1H*/,1X,1H*,2(12X,4A4),12X,1H*/,1X,
C 1H*,14X,3A4,16X,3A4,14X,1H*)
21  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*/,
C 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,
C 12X,1H*/,1X,1H*,68X,1H*/,1X,1H*,2(12X,4A4),12X,1H*/,1X,1H*,
C 18X,A3,25X,A3,19X,1H*)

```



```

22  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,1X,
    & 1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,
    & 1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
    & 3(5X,F16.3),5X,1H*)
23  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*/,
    & 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),
    & 3A4,7X,1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
    & 7X,2(3A4,9X),3A4,7X,1H*)
24  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*/,
    & 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),
    & A3,12X,1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
    & 11X,2(A3,18X),A3,12X,1H*)
25  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,1H*/,1X,
    & 1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,3(5X,F16.3),5X,
    & 1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
    & 3(5X,F16.3),5X,1H*/,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,1H*/,
    & 1X,1H*,26X,F16.3,26X,1H*)
26  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),3A4,7X,1H*/,
    & 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,7X,2(3A4,9X),
    & 3A4,7X,1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
    & 7X,2(3A4,9X),3A4,7X,1H*/,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,26X,
    & 1H*/,1X,1H*,28X,3A4,28X,1H*)
27  FORMAT (1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),A3,12X,1H*/,
    & 1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,11X,2(A3,18X),
    & A3,12X,1H*/,1X,1H*,68X,1H*/,1X,1H*,3(5X,4A4),5X,1H*/,1X,1H*,
    & 11X,2(A3,18X),A3,12X,1H*/,1X,1H*,68X,1H*/,1X,1H*,26X,4A4,
    & 26X,1H*/,1X,1H*,32X,A3,33X,1H*)

```

END

```

C *****
C *****
C

```

SUBROUTINE PAGER (LINE,LIMIT)

TEKTRONIX PAGING. REMOVE IF NOT USING A TEKTRONIX TERMINAL.

IF (LINE+LIMIT.LT.34) RETURN

LINE=0

PAUSE "TYPE GO FOR NEXT PAGE"

RETURN

END

```

      PROGRAM UPPP (INPUT,OUTPUT,HDR,DUM1,DUM2,UPP,TAPE10=HDR,
& TAPE12=UPP,TAPE2=OUTPUT,TAPES=INPUT,TAPE13=DUM1,TAPE14=DUM2)
C*****
C**
C*          USER'S PREFERENCE PACKAGE PROGRAM
C* THIS PROGRAM CREATES THE USER'S PREFERENCE PACKAGE FOR SSMP
C**
C*****
C
      COMMON /UNITNO/ITTYIN,ITYOUT,IHDR,IUPP,IDUM1,IDUM2
      COMMON /HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
& SYSSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),
& SUB1NM(10,4),SUB2NM(10,10,4),PARNAM(40,4),
& NUM(40), RATED(40),YN(40),INVERT(40),INDEX
      COMMON /RTNG/ RATING(50,40)
      COMMON /HDR2/ NSCEN,SCNDSC(25,15)
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
& PARNAM,SCNDSC,RATED,YN
C
C
      DATA IEND/3HEND/,ICRTE/4HCRTE/,IADD/3HADD/,ILIST/4HLIST/
C
      NPAGE=0
      LINE=34
      CALL PAGER(LINE,NPAGE,0)
      REWIND IUPP
      REWIND IHDR
      REWIND IDUM1
      REWIND IDUM2
      ITERN=0
C *** PRINT OUT HEADING ***
      WRITE(ITYOUT,4)
C *** PRINT OUT NOTE ON COMMANDS ***
      WRITE (ITYOUT,1)
C *** READ IN /HDP1/ ***
      CALL READR1
C *** ASK FOR COMMAND ***
      WRITE (ITYOUT,2)
      READ (ITTYIN,3) IANSR
100  IF (IANSR.EQ.IEND) GO TO 1000
      IF (IANSR.EQ.ICRTE) CALL CREATE
C *** READ IN /HDP2/ ***
      IF(IANSR.NE.ICRTE.AND.ITERN.EQ.0)CALL READR2
      IF (IANSR.EQ.IADD) CALL ADD
      IF (IANSR.EQ.ILIST) CALL LIST
      WRITE (ITYOUT,2)
      READ (ITTYIN,3) IANSR
      ITERN=1
      GO TO 100
C
C
C *** FORMAT STATEMENTS ***
C
1    FORMAT (1X,48HNOTE: THE COMMANDS CRTE & ADD CAN BE ISSUED ONLY,

```

& 12H ONCE DURING/1X,3SHA SESSION & YOU CANNOT ISSUE A LIST,
 & 26H BEFOPE YOU CREATE THAT FILE//)

```

C
2  FORMAT (1X,10HCOMMAND-->)
3  FORMAT (A4)
4  FORMAT(1H1,72(1H*),/1X,19X,33HUSER'S PREFERENCE PACKAGE PROGRAM,
& /,1X,72(1H*)//)

```

```

C
C
1000  CONTINUE
      STOP
      END

```

```

C
C*****
C

```

```

      BLOCK DATA
      -INITIALIZES UNIT NUMBERS FOR I/O

```

```

C
C*****
COMMON /UNITNO/ ITTYIN,ITYOUT,IHDR,IUPP,ICUM1,ICUM2
DATA ITTYIN,ITYOUT,IHDR,IUPP,IDUM1,IDUM2/5,2,10,12,13,14/
END

```

```

C
C*****

```

```

      SUBROUTINE PEADR1
      -THIS MODULE READS IN THE HEADER FILE AND FILLS THE
      COMMON BLOCKS /HDR1/ AND /DB/

```

```

C
C*****
C

```

```

COMMON /UNITNO/ ITTYIN,ITYOUT,IHDR,IUPP,ICUM1,ICUM2
COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARN,SYSSET(4),
& SB1SET(4),SB2SET(4),SYSNAM(10,4),SUB1NM(10,4),
& SUB2NM(10,10,4),PARNAM(40,4),NUM(40),RATED(40),YN(40),
& INVERT(40),INDEX
COMMON/DB/DBNAME(4)
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,DBNAME

```

```

C
      REWIND IHDR

```

```

C
      READ(IHDR,3)(DBNAME(M),M=1,4)
      READ (IHDR,1) SYSDSN
      READ (IHDR,2) NSYS
      READ (IHDR,2) NSUB1
      READ (IHDR,2) NSUB2
      READ (IHDR,2) NPARN
      READ (IHDR,3) (SYSSET(I),I=1,4)
      IF (SYSDSN.GE.2) READ (IHDR,3) (SB1SET(I),I=1,4)
      IF (SYSDSN.EG.3) READ (IHDR,3) (SB2SET(I),I=1,4)

```

```

C
      DO 100 I=1,NSYS
        READ (IHDR,3) (SYSNAM(I,J),J=1,4)
100  CONTINUE

```

```

C      IF (NSUB1.EQ.0) GO TO 400
C
C      DO 200 I=1,NSUB1
C        READ (IHDR,3) (SUB1NM(I,J),J=1,4)
200    CONTINUE
C
C      IF (NSUB2.EQ.0) GO TO 400
C
C      DO 300 K=1,NSUB1
C        DO 301 I=1,NSUB2
C          READ (IHDR,3) (SUB2NM(K,I,J),J=1,4)
301    CONTINUE
300    CONTINUE
C
C      400 CONTINUE
C
C      DO 500 I=1,NPARM
C        READ (IHDR,4) (PARNAM(I,J),J=1,4),NUM(I),RATED(I),YN(I),
C        & INVERT(I)
C
C      500 CONTINUE
C
C      *** FORMAT STATEMENTS ***
C
C      1  FORMAT (I1)
C      2  FORMAT (I2)
C      3  FORMAT (4A4)
C      4  FORMAT (4A4,I1,I1,I1,I1)
C
C      RETURN
C      END
C
C
C *****
C      SUBROUTINE CREATE
C      -THIS MODULE ALLOWS THE USER TO INPUT THE SCENARIO
C      DESCRIPTIONS AND NUMBER OF SCENARIOS AND TO WRITE
C      OUT THE NEW HEADER FILE, THEN INPUT RATING VALUES.
C *****
C
C      COMMON /UNITNO/ ITTYIN,ITYOUT,IHDR,IUPP,IDUM1,IDUM2
C      COMMON /HDR1/  SYSOSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(4),
C      & SB1SET(4),SB2SET(4),SYSNAM(10,4),SUB1NM(10,4),SUB2NM(10,10,4),
C      & PARNAM(40,4),NUM(40),RATED(40),YN(40),INVERT(40),INDEX
C      COMMON /HDR2/ NSCEN,SCNDSC(25,15)
C      COMMON /RTNG/ RATING(50,40)
C      INTEGER SYSOSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
C      & SUB2NM,PARNAM,PATED,YN,SCNDSC
C
C      *** NO. OF SCENARIOS ***
C      WRITE (ITYOUT,1)
C      READ (ITTYIN,*) NSCEN

```

```

C*** SCENARIO DESCRIPTIONS ***
DO 100 I=1,NSCEN
    WRITE (ITYOUT,3) I
    READ (ITTYIN,4) (SCNDSC(I,M),M=1,15)
100 CONTINUE
C
C*** WRITE OUT /HDR1/ ON TO NEW FILE
    CALL WRTR
C*** PERFORM PAIRED COMPARISON ***
    CALL COMPAR(1)
C
C *** FORMAT STATEMENTS ***
1    FORMAT (1X,22HENTER NO. OF SCENARIOS/)
3    FORMAT (1X,43HENTER SCENARIO DESCRIPTION FOR SCENARIO NO.,12/)
4    FORMAT (15A4)
    RETURN
    END
C
C
C*****
C
    SUBROUTINE WRTR
C        -THIS MODULE WRITES OUT THE NEW HEADER FILE WITH THE
C        NEW SCENARIO INFORMATION.
C*****
C
C    COMMON /UNITNO/ ITTYIN,ITYOUT,IHDR,IUPP,IDUM1,IDUM2
C    COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARN,
C    & SYSSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),
C    & SUP1NM(10,4),SUB2NM(10,10,4),PARNAM(40,4),
C    & NUM(40),RATED(40),YN(40),INVERT(40),INDEX
C    COMMON/HDR2/NSCEN,SCNDSC(25,15)
C    COMMON/DB/DBNAME(4)
C
C    INTEGER SCNDSC,DBNAME
C
C    INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,
C    & SUP1NM,SUB2NM,PARNAM,RATED,YN
C
C    REWIND IDUM1
C
C    WRITE(IDUM1,3)(DBNAME(M),M=1,4)
C *** CASE I,II, OR III ***
C    WRITE (IDUM1,1) SYSDSN
C *** NUMBER OF SYSTEMS ***
C    WRITE (IDUM1,2) NSYS
C *** NUMBER OF LEVEL 1 SUBSYSTEMS ***
C    WRITE (IDUM1,2) NSUB1
C *** NUMBER OF LEVEL 2 SUBSYSTEMS ***
C    WRITE (IDUM1,2) NSUB2
C *** NUMBER OF PARAMETERS ***
C    WRITE (IDUM1,2) NPARN
C *** SYSTEM SETNAMES ***

```

```

WRITE (IDUM1,3) (SYSSET(I),I=1,4)
IF (SYSDSN.GE.2) WRITE (IDUM1,3) (SP1SET(I),I=1,4)
IF (SYSDSN.EQ.3) WRITE (IDUM1,3) (SB2SET(I),I=1,4)
C
C *** SYSTEM NAMES ***
DO 100 I=1,NSYS
  WRITE(IDUM1,3)(SYSNAM(I,J),J=1,4)
100 CONTINUE
C
  IF(NSUB1.EQ.0)GO TO 400
C
C *** LEVEL 1 SUBSYSTEM NAMES ***
DO 200 I=1,NSUB1
  WRITE(IDUM1,3)(SUB1NM(I,J),J=1,4)
200 CONTINUE
C
  IF (NSUB2.EQ.0) GO TO 400
C
C *** LEVEL 2 SUBSYSTEM NAMES ***
DO 300 K=1,NSUB1
  DO 301 L=1,NSUB2
    WRITE(IDUM1,3)(SUB2NM(K,L,J),J=1,4)
301 CONTINUE
300 CONTINUE
C
400 CONTINUE
C
C *** PARAMETER NAMES ***
DO 500 I=1,NPARAM
  WRITE (IDUM1,4) (PARNAM(I,J),J=1,4),NUM(I),RATED(I),YN(I),
& INVERT(I)
500 CONTINUE
C
C *** NUMBER OF SCENARIOS ***
WRITE(IDUM1,2)NSCEN
C
C *** SCENARIO DESCRIPTIONS ***
DO 600 I=1,NSCEN
  WRITE(IDUM1,5)(SCNDSC(I,M),M=1,15)
600 CONTINUE
C*** FORMAT STATEMENTS ***
C
1  FORMAT (I1)
2  FORMAT (I2)
3  FORMAT (4A4)
4  FORMAT (4A4,I1,I1,I1,I1)
5  FORMAT (15A4)
C
  RETURN
  END
C
C
C*****
C

```

```

SUBROUTINE COMPAN
C      -THIS MODULE ALLOWS THE USER TO INPUT THE PARAMETER
C      RATINGS AND/OR PERFORM THE PAIRED COMPARISON ANALYSIS
C
C *****
C
COMMON /UNITNO/ ITTYIN,ITYOUT,IHDP,IUPP,ICUM1,ICUM2
COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
& SYSSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),
& SUB1NM(10,4),SUB2NM(10,4),PARNAM(40,4),
& NUM(40),PATED(40),YN(40),INVERT(40),INDEX
COMMON /HDR2/ NSCEN,SCNDSC(25,15)
COMMON /RTNG/ RATING(50,40)

C
DIMENSION ICOUNT(40),IORDER(40)
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,PATED,YN,SCNDSC

C
DATA ICHARA/1HA/,ICHARB/1HB/
DATA IYES/3HYES/

C
C USER GIVEN CHOICE OF ENTERING RATINGS IMMEDIATELY
C OR PERFORMING THE COMPARISON ANALYSIS
WRITE (ITYOUT,8)
READ (ITTYIN,9) IANSR
NPAGE=0
LINE=34
CALL PAGER(LINE,NPAGE,0)
IF (IANSR.EQ.IYES) GO TO 500
DO 100 J=1,NSCEN
  DO 110 KK=1,NPARM
    ICOUNT(KK)=0
110  CONTINUE
C
C *** PRINT OUT SCENARIO DESCRIPTIONS ***
WRITE (ITYOUT,1) (SCNDSC(J,M),M=1,15)
LINE=2
DO 200 I=1,NPARM
  DO 210 I1=1,NPARM
    IF (I1.LE.I) GO TO 210
C *** PRINT OUT PARAMETER NAMES ***
    NPAGE=0
    CALL PAGER(LINE,NPAGE,3)
    WRITE (ITYOUT,2) ICHARA,(PARNAM(I,M),M=1,4),ICHARB,
& (PARNAM(I1,M1),M1=1,4)
    READ (ITTYIN,3) ICHAP
    LINE=LINE+2
C *** ADD UP THE NUMBER OF TIMES CHOSEN ***
    IF (ICHAP.EQ.ICHARA) ICOUNT(I)=ICOUNT(I)+1
    IF (ICHAP.EQ.ICHARB) ICOUNT(I1)=ICOUNT(I1)+1
210  CONTINUE
200  CONTINUE
C

```

```

C ORDER THE PARAMETERS BASED ON THE NUMBER OF TIMES
C CHOSEN (HIGH TO LOW)
      CALL LARGE (NPARM,ICOUNT,IORDER)
      WRITE (ITYOUT,4)
C PRINT OUT PARAMETERS IN ORDER WITH THE NUMBER OF TIMES CHOSEN
C AND READ IN RATINGS
      DO 300 I=1,NPARM
        K=IORDER(I)
        WRITE (ITYOUT,5) (PARNAM(K,M),M=1,4),ICOUNT(K)
        READ (ITTYIN,6) RATING (J,K)
300    CONTINUE
C
100    CONTINUE
C
C *** WRITE OUT UPP DATA FILE ***
      DO 400 K=1,NSCEN
        DO 400 I=1,NPARM
          WRITE(IDUM2,7)RATING(K,I)
400    CONTINUE
      REWIND IDUM2
      RETURN
500    DO 600 J=1,NSCEN
      WRITE(ITYOUT,1)(SCNDESC(J,M),M=1,15)
        DO 700 I=1,NPARM
          WRITE(ITYOUT,10)(PARNAM(I,M),M=1,4)
          READ(ITTYIN,6)RATING(J,I)
700    CONTINUE
      NPAGE=0
      LINE=34
      CALL PAGER(LINE,NPAGE,C)
600    CONTINUE
        DO 800 K=1,NSCEN
          DO 800 I=1,NPARM
            WRITE(IDUM2,7)RATING(K,I)
800    CONTINUE
      REWIND IDUM2
      RETURN
C
C*** FORMAT STATEMENTS ***
C
1      FORMAT (10X,15A4)
2      FORMAT (1X,1X,A1,4H -- ,4A4,3H / ,A1,4H -- ,4A4,/)
3      FORMAT (A1)
4      FORMAT (1X,/,1X,43H RATE THE PARAMETERS WITH VALUES FROM 0 TO 1,/)
5      FORMAT (1X,/,1X,4A4,9H (CHOSEN ,I2,7H TIMES),/)
6      FORMAT (F5.3)
7      FORMAT (1X,F5.3)
8      FORMAT(1X,52H DO YOU WISH TO SKIP THE PAIRED COMPARISON ? (YES/NO),
C /)
9      FORMAT(A3)
10     FORMAT(1X,4A4,/)
C
      END
C

```



```

C
C *****
C
      SUBROUTINE ADD
C      -THIS ROUTINE ALLOWS THE USER TO ADD SCENARIOS
C
C *****
C
      COMMON /UNITNO/ ITTYIN,ITYOUT,INH0,IUPP,ICUM1,ICUM2
      COMMON /HDR1/ SYSDSN,NSYS,NSUP1,NSUB2,NPARN,
& SYSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),
& SUB1NM(10,4),SUB2NM(10,10,4),PARNAM(40,4),
& NUM(40),PATED(40),YN(40),INVERT(40),INDEX
      COMMON /HDR2/ NSCEN,SCNDESC(25,15)
      COMMON /RTNG/ RATING(50,40)
C
      INTEGER SYSDSN,SYSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,PATED,YN,SCNDESC
C
C *** NO. OF SCENARIOS ADDING ***
C
      WRITE (ITYOUT,1)
      READ (ITTYIN,*) ISCEN
      K=ISCEN
      NOLD=NSCEN + 1
C *** READ IN NEW SCENARIO DESCRIPTION ***
      DO 100 I=1,K
      L=I + NSCEN
      WRITE (ITYOUT,3) L
      READ(ITTYIN,4) (SCNDESC(L,M),M=1,15)
100 CONTINUE
C
      NSCEN=ISCEN + NSCFN
C *** WRITE OUT HEADED FILE ***
      CALL WRTR
C *** ENABLE USER TO ENTER PARAMETER RATINGS ***
      CALL COMPAR(NOLD)
C
C *** FORMAT STATEMENTS ***
C
1      FORMAT (1X,39HDO MANY SCENARIOS DO YOU WISH TO ADD ?/)
2      FORMAT (I2)
3      FORMAT (1X,43HENTER SCENARIO DESCRIPTION FOR SCENARIO NO.,I2,/)
4      FORMAT (15A4)
C
      RETURN
      END
C
C
C *****
C      SUBROUTINE LARGE (NPARN,ICOUNT,IORDER)
C      - THIS ROUTINE FINDS THE LARGEST NUMBER OF TIMES A
C      PARAMETER WAS CHOSEN DURING THE PAIRED COMPARISON.
C

```

```

C*****
C
      DIMENSION ICOUNT(40),ICNT(40),IORDER(40)
C
C *** TRANSFER NUMBERS TO NEW ARRAY ***
      DO 100 I=1,NPARM
        ICNT(I)=ICOUNT(I)
100    CONTINUE
C
      DO 200 K=1,NPARM
        BIG=-1
        INDICE=0
C *** CHECK FOR LARGEST VALUE ***
        DO 300 I=1,NPARM
          IF (ICNT(I).GT.BIG) GO TO 305
          GO TO 300
305      BIG=ICNT(I)
          INDICE=I
300    CONTINUE
          IF (INDICE.EQ.0) INDICE=K
C ONCE LARGEST VALUE FOUND, MAKE -1 SO NEXT HIGHEST IS FOUND
          ICNT(INDICE)=-1
C KEEP TRACK OF ORDER (HIGH TO LOW) OF PARAMETERS
          IORDER(K)=INDICE
200    CONTINUE
C
      RETURN
      END
C
C
C*****
C
      SUBROUTINE LIST
C      -THIS ROUTINE ALLOWS THE USER TO PRINT THE PARAMETER
C      RATINGS FOR EACH SCENARIO PLUS THE AVERAGE RATING
C      FOR EACH PARAMETER.
C
C*****
C
      COMMON /UNITNO/ ITTYIN,ITYOUT,IHDP,IUPP,ICUM1,ICUM2
      COMMON /HDR1/ SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
& SYSSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),
& SUB1NM(10,4),SUB2NM(10,10,4),PARNAM(40,4),
& NUM(40),PATED(40),YN(40),INVERT(40),INDEX
      COMMON /HDR2/ NSCEN,SCNDESC(25,15)
      COMMON /RTNG/ RATING(50,40)
      COMMON /CB/DPNAME(4)
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,PATED,YN,SCNDESC,DPNAME
      DIMENSION SUM(50),AVE(50),PSUM(40),PAVE(40),IAVENM(15)
      DATA SUM/50*0/,AVE/50*0/,PAVE/40*0/,PSUM/40*0/
      DATA IAVENM/4H      ,4H      ,4H      AV,4HFRAG,4HE V1,4HLUES,4H FCG,
& 4H EAC,4HH PA,4HPAME,4HTER ,4H      ,4H      ,4H      ,4H      /
C

```

```

C *** SUM UP RATINGS FOR AVERAGE ***
DO 200 I=1,NSCEN
  DO 250 J=1,NPARN
    SUM(I)=SUM(I) + RATING(I,J)
250  CONTINUE
    AVE(I)=SUM(I)/NPARN
200  CONTINUE
C *** AVERAGE PARAMETER RATING ***
C
  DO 300 J=1,NPARN
    PSUM(J)=0
    DO 350 I=1,NSCEN
      PSUM(J) = PSUM(J) + RATING(I,J)
350  CONTINUE
    RSCEN=NSCEN
    PAVE(J)=PSUM(J)/RSCEN
300  CONTINUE
C *** WRITE HEADING ***
C
  DO 100 I=1,NSCEN
    NPAGE=1
    WRITE(ITYOUT,1)(DBNAME(L99),L99=1,4),SYSDSN,
& (SCNDESC(I,M),M=1,15),I
C *** WRITE PARAMETER NAME AND RATING ***
    DO 150 J=1,NPARN
      WRITE(ITYOUT,2)(PARNAM(J,M),M=1,4),RATING(I,J)
150  CONTINUE
    LINE=34
    CALL PAGER(LINE,NPAGE,C)
100  CONTINUE
C *** WRITE HEADING FOR AVERAGE PARAMETER RATINGS ***
  N = NSCEN + 1
  WRITE(ITYOUT,1)(DBNAME(L99),L99=1,4),SYSDSN,(IAVENH(MM),MM=1,15),
& N
C *** WRITE PARAMETER NAMES AND AVERAGE RATINGS ***
  DO 400 J=1,NPARN
    WRITE(ITYOUT,2)(PARNAM(J,M),M=1,4),PAVE(J)
400  CONTINUE
    NPAGE=0
    LINE=34
    CALL PAGER(LINE,NPAGE,C)
C
C *** FORMAT STATEMENTS ***
C
1  FORMAT(1H,/,1X,72(1H*),1X,5X,4A4,5X,5HCASE,11,3X,
& 25HUSER'S PREFERENCE PACKAGE,/,1X,15A4,5X,5HPAGE,
& 12,/,1X,72(1H*))
2  FORMAT(1X,/,1X,20X,4A4,5X,F5.3)
3  FORMAT(1X,/,1X,20X,16(1H-),5X,5(1H-),/,1X,20X,
& 13HAVERAGE VALUE,6X,F5.3)
C
  RETURN
END
C

```

```

C *****
C
C      SUBROUTINE READR2
C          - THIS ROUTINE IS USED WHEN THE USER WISHES TO
C            ADD MORE SCENARIOS
C *****
C
C      COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,IUPP,IDUM1,IDUM2
C      COMMON/HDR2/NSCEN,SCNDSC(25,15)
C      COMMON/RTNG/RATING(50,40)
C      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUP2,NPARN,
C      & SYSSET(4),SB1SET(4),SB2SET(4),SYSNAM(10,4),
C      & SUB1NM(10,4),SUB2NM(10,10,4),PARNAM(40,4),
C      & NUM(40),RATED(40),YN(40),INVERT(40),INDEX
C
C      INTEGER SCNDSC
C      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
C      & PARNAM,RATED,YN
C *** READ NUMBER OF SCENARIOS ***
C      READ(IHDR,1)NSCEN
C *** READ SCENARIO DESCRIPTIONS ***
C      DO 100 I=1,NSCEN
C          READ(IHDR,2)(SCNDSC(I,M),M=1,15)
100      CONTINUE
C *** READ RATINGS ***
C      DO 200 I=1,NSCEN
C          DO 200 J=1,NPARN
C              READ(IUPP,3)RATING(I,J)
200      CONTINUE
C
C
C *** FORMAT STATEMENTS ****
C
1      FORMAT(I2)
2      FORMAT(15A4)
3      FORMAT(F5.3)
C
C      RETURN
C      END
C
C *****
C *****
C
C      SUBROUTINE PAGEP(LINE,NPAGE,LIMIT)
C      IF((LINE+LIMIT).LT.34)RETURN
C      PAUSE"TYPE GO FOR NEXT PAGE"
C      LINE=0
C      NPAGE=NPAGE+1
C      RETURN
C      END

```

```

OVLPLAY(SSM,0,0)
PROGRAM SSM(INPUT=512,OUTPUT=512,HEADER=512,PP=512,UPP=512,
& VPARAM=512,SYSNUM,TAPE5=INPUT,TAPE2=OUTPUT,TAPE10=HEADER,
& TAPE12=PP,TAPE13=UPP,TAPE14=VPARM,TAPE15=SYSNUM)

```

```

*****

```

MAIN OVERLAY

```

DIRECTS THE EXECUTION OF THE OTHER LEVELS OF OVERLAYS &
SUBROUTINES

```

```

*****

```

```

COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARAM,
&      SYSSET(4),SB1SET(4),SB2SET(4),
&      SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),
&      PARNAM(40,4),
&      NUM(40),RATED(40),YN(40),INVERT(40),INDEX
COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARAM,ISYS
COMMON/PARAM1/NOPPM,CHPRM(40),ABV(40),BELW(40),
& RLOWR(40),HIR(40),RTD(40),IRANGE(40),IS=NO(40)
COMMON/PARAM3/MATRIX(40,7,10)
COMMON/PARAM2/ELIM(7,10,40)
COMMON/RTNG/RATING(40)
COMMON/PARAM4/SYSNRM(85,10)
COMMON/RTG/PTING(40)
COMMON/DB/DBNAME(4)
INTEGER SCNDSC
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,PARNAM,
& RATED,YN,ELIM,MATRIX,CHPRM,DBNAME

```

```

100 REWIND IPP
    REWIND IPPH
    REWIND IUPP
    REWIND IUPPH

```

```

*** READ IN FILE WITH HEADER IN IT FOR PARAMETER PACKAGE

```

```

CALL OVERLAY(3HSSM,1,0,0)

```

```

*** USER CHOOSES SCENERIO OR INPUTS HIS OWN RATINGS

```

```

CALL OVERLAY(3HSSM,2,0,0)

```

```

*** PRINT OUT RATINGS TABLE ***

```

```

CALL RATOUT

```

```

*** CHOOSE THE PARAMETERS WANTED IN THE DECISION PROCESS

```

```

CALL OVERLAY(3HSSM,3,0,0)

```

```

C
C *** ELIMINATE SYSTEMS THAT DO NOT QUALIFY OR ARE NOT WANTED
C
C     CALL OVERLAY(3HSSM,4,0,0)
C
C *** INVERTS DATA SO THAT THE LOWEST VALUES CONSTITUTE THE BEST
C     VALUE
C
C     CALL OVERLAY(3HSSM,5,0,0)
C
C *** CALCULATE SYSTEM VALUES & DETERMINE THE OPTIMUM SYSTEM
C
C     CALL OVERLAY(3HSSM,6,0,0)
C
C *** DETERMINE SENSITIVITY & UNCERTAINTY ANALYSIS
C
C     CALL ANLYS(IEXEC)
C     IF(IEXEC.EQ.1)GO TO 100
C     STOP
C     END
C *****
C *****
C
C     SUBROUTINE ANLYS(IEXEC)
C
C *****
C
C     THIS ROUTINE ALLOWS THE USER TO ADD ANY LOGIC NEEDED TO ANALYZE
C     THE DECISION & TO ALLOW THE USER TO RERUN THE MODEL
C
C *****
C
C     COMMON/PTG/RTING(40)
C     COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
C     COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
C     COMMON/HDR1/SYSDSN,NSYS,NSUP1,NSUB2,NPARM,
C     & SYSSET(4),SB1SET(4),SB2SET(4),
C     & SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
C     & NUM(40),RATED(40),YN(40),INVERT(40),INDEX
C     INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
C     & PARNAM,RATED,YN,SCNDSC
C     DATA INO/2HNO/
C
C     IEXEC=0
C     WRITE(ITYOUT,1)
C     READ(ITYIN,2)IANSR
C     IF(IANSR.EQ.INO)GO TO 9000
C     IEXEC=1
C 9000 RETURN
C
C *** FORMAT STATEMENTS ***
C
C 1     FORMAT(1X,/,1X,26HDO YOU WISH TO EXECUTE THE,
C     & 25H PROGRAM AGAIN ? (YES/NO)/)

```

```

2      FORMAT(A3)
      END

C
C *****
C *****
C
      SUBROUTINE RATOUT
C
C *****
C
      THIS ROUTINE PRINTS THE RATINGS CHOSEN IN A TABLE
C
C *****
C
      COMMON/PTG/RTING(40)
      COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
      & SYSSET(4),SB1SET(4),SB2SET(4),
      & SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
      & NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/UNITNO/ITYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
      & PARNAM,RATED,YN,SCNDSC
C
C      TEKTRONIX PAGING: TAKE OUT IF NOT USING TEKTRONIX TERMINAL OR
C      GRAPHICS
C      LINE=34
C      NPAGE=0
C      CALL PAGER(LINE,NPAGE,C)
C      WRITE(ITYOUT,1)(SCNDSC(ICH0IC,M),M=1,15)
C
C      DO 100 I=1,NPARM
C          WRITE(ITYOUT,2)(PARNAM(I,M),M=1,4),RTING(I)
100    CONTINUE
C
C      WRITE(ITYOUT,3)
C
C      RETURN
C
C *** FORMAT STATEMENTS ***
C
1      FORMAT(1X,/,15X,15A4,/,21X,38(1H*),/21X,1H*,5X,
      & 13H PARAMETERS,8X,1H*,10H RATINGS *,/,1X,20X,38(1H*))
2      FORMAT(21X,1H*,5X,4A4,5X,1H*,2X,F5.3,2X,1H*)
3      FORMAT(21X,36(1H*),/)
      END
C
C *****
C *****
C
      BLOCK DATA
C
C *****
C

```

```

C      ALLOWS THE USER TO CHANGE UNIT NUMBERS FOR I/O MORE EASILY
C
C *****
C      COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
C      DATA ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS/5,2,10,10,12,
C      & 13,14,15/
C      END
C *****
C *****
C
C      SUBROUTINE PAGER(LINE,NPAGE,LIMIT)
C      IF((LINE+LIMIT).LT.34) RETURN
C      PAUSE "TYPE GO FOR NEXT PAGE"
C      LINE=0
C      NPAGE=NPAGE+1
C      RETURN
C      END
C
C *****
C *****
C
C      OVERLAY(SSM,1,G)
C      PROGRAM RPP
C
C *****
C
C      THIS ROUTINE READS IN THE INFORMATION THAT DESCRIBES THE
C      PARAMETER PACKAGE AND PLACES IT IN COMMON STATEMENT /HDR1/
C
C *****
C      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
C      &          SYSET(4),SB1SET(4),SB2SET(4),
C      &          SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),
C      &          PARNAM(40,4),
C      &          NUM(40),RATED(40),YN(40),INVERT(40),INDEX
C      COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
C      COMMON/DB/DBNAME(4)
C      COMMON/HDR2/NSCEN,SCNOSC(25,15),INDEX1,ICHOIC
C      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
C      & PARNAM,RATED,YN,DBNAME,SCNOSC
C
C      *** READ IN DATA BASE NAME FOR PARAMETER PACKAGE ***
C
C      READ (IPPH,3)(DBNAME(M),M=1,4)
C
C      *** READ IN SYSTEM DESIGN
C
C      READ(IPPH,1)SYSDSN
C
C      *** READ NO. OF SYSTEMS,SUBSYSTEMS,PARAMETER
C
C      READ(IPPH,2)NSYS
C      READ(IPPH,2)NSUB1
C      READ(IPPH,2)NSUB2

```



```

      READ(IPPH,2)NPARM
      IF(NSYS.EQ.C)GO TO 600
C
C *** READ IN SETNAMES
C
      READ(IPPH,3)(SYSSET(I),I=1,4)
      IF(SYSDSN.GE.2)READ(IPPH,3)(SB1SET(I),I=1,4)
      IF(SYSDSN.EQ.3)READ(IPPH,3)(SB2SET(I),I=1,4)
C
C *** READ IN SYSTEM NAMES
C
      DO 100 I=1,NSYS
      READ(IPPH,3)(SYSNAM(I,J),J=1,4)
100   CONTINUE
      IF(NSUB1.EQ.0)GO TO 400
C
C *** READ IN SUBSYSTEM LEVEL 1 NAMES
C
      DO 200 I=1,NSUB1
      READ (IPPH,3)(SUB1NM(I,J),J=1,4)
200   CONTINUE
      IF(NSUB2.EQ.0)GO TO 400
C *** READ IN SUBSYSTEM LEVEL 2 NAMES
      DO 300 K=1,NSUB1
      DO 301 I=1,NSUB2
      READ (IPPH,3)(SUB2NM(K,I,J),J=1,4)
301   CONTINUE
300   CONTINUE
C
400   CONTINUE
C
C *** READ IN PARAMETER NAMES & PARAMETER CHARACTERISTICS
C
      DO 500 I=1,NPARM
      READ(IPPH,4)(PARNAM(I,J),J=1,4),NUM(I),RATED(I),YN(I),
& INVERT(I)
500   CONTINUE
C
      GO TO 700
C
600   WRITE(ITYOUT,5)
      CALL EXIT
C
C *** CALCULATE INDEX FOR POSITIONING FILES ***
C
700   CONTINUE
      IF(NSUB1.EQ.0)GO TO 800
      IF(NSUB2.EQ.0)GO TO 900
      INDEX=NSUB1*NSUB2
      GO TO 1000
800   INDEX=1
      GO TO 1000
900   INDEX=NSUB1
1000  CONTINUE

```

```

C
C *** CALL RUPP ***
C
C     CALL OVERLAY(3HSSM,1,1,0)
C
C *** FORMAT STATEMENTS ***
C
C     1     FORMAT(I1)
C     2     FORMAT(I2)
C     3     FORMAT(4A4)
C     4     FORMAT(4A4,I1,I1,I1,I1)
C     5     FORMAT(1X,37HERROR IN HEADER -- NO. OF SYSTEMS = 0)
C
C     END
C
C *****
C *****
C
C     OVERLAY(SSM,1,1)
C     PROGRAM RUPP
C
C *****
C     THIS ROUTINE READS IN THE INFORMATION THAT DESCRIBES THE
C     USER PREFERENCE PACKAGE AND PLACES IT IN COMMON STATEMENT
C     /HDR2/
C *****
C
C     COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
C     COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
C     INTEGER SCNDSC
C
C *** READ IN NO. OF SCENERIOS
C
C     READ(IUPPH,1)NSCEN
C
C     IF(NSCEN.EQ.0)WRITE(ITYOUT,2)
C
C *** READ IN SCENERIO DESCRIPTION
C
C     DO 100 I=1,NSCEN
C         READ(IUPPH,3)(SCNDSC(I,J),J=1,15)
100  CONTINUE
C
C *** FORMAT STATEMENTS ***
C
C     1     FORMAT(I2)
C     2     FORMAT(49H ERROR IN USER PREFERENCE PACKAGE - NO. OF SCENAR,
C     3     & 6HIOS =0)
C     3     FORMAT(15A4)
C     END
C
C *****
C *****
C

```

```

      OVEPLAY(SSM,2,0)
      PROGRAM SCNSEL

C
C *****
C
C      THIS ROUTINE READS IN THE RATINGS ARRAY EITHER BY THE USER'S
C      PREFERENCE PACKAGE OR BY THE USER'S OWN RATINGS
C
C *****
C
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
      &      SYSSET(4),SB1SET(4),SB2SET(4),
      &      SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),
      &      PARNAM(40,4),
      &      NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
      COMMON/UNITNO/ITYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
      COMMON/RTNG/RATING(40)
      COMMON/RTG/RTING(40)
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,PARNAM,
      & RATED,YN,SCNDSC
      DATA IYES/3HYES/

C
C *** WRITE OUT TITLE ***
C
C      TEKTRONIX PAGING: TAKE OUT IF NOT USING TEKTRONIX TERMINAL OR
C      GRAPHICS
      LINE=34
      NPAGE=0
      CALL PAGER(LINE,NPAGE,0)
      WRITE(ITYOUT,15)

C
C *** WRITE OUT AVAILABLE PARAMETERS ***
C
      WRITE(ITYOUT,13)
      DO 199 I=1,NPARM
        WRITE(ITYOUT,14)I,(PARNAM(I,J),J=1,4)
199  CONTINUE

C
C *** INPUT OWN RATINGS ? ***
C
C      TEKTRONIX PAGING
      LINE=34
      NPAGE=0
      CALL PAGER(LINE,NPAGE,0)
      WRITE(ITYOUT,1)
      READ(ITYIN,8)IANS
      IF(IANS.EQ.IYES)GO TO 100
      WRITE(ITYOUT,11)

C
C *** PRINT OUT SCENARIO DESCRIPTIONS ***
C
      DO 200 I=1,NSCEN
        WRITE(ITYOUT,10)I,(SCNDSC(I,J),J=1,15)

```

```

200  CONTINUE
C
C *** READ IN CHOICE OF SCENERIC
C
      WRITE(ITYOUT,9)
      READ(ITYIN,*)ICHOIC
      IF(ICHOIC.EQ.C.OR.ICHOIC.GT.NSCEN)WRITE(ITYOUT,16)
      IF(ICHOIC.EQ.1)GO TO 250
      INDEX1=(ICHOIC-1)*NPARM
C
      GO TO 260
C
C *** READ IN RATINGS FROM USERS' PREFERENCE PACKAGE ***
C
250  DO 300 I=1,NPARM
      READ(IUPP,5)RATING(I)
300  CONTINUE
C
      GO TO 1000
C
260  DO 310 I=1,INDEX1
      READ (IUPP,6)IDUM
310  CONTINUE
C
      DO 400 I=1,NPARM
      READ(IUPP,5)RATING(I)
400  CONTINUE
C
      GO TO 1000
C
100  CONTINUE
C
C *** READ IN USER'S OWN RATINGS ***
C
      TEKTRONIX PAGING
      NPAGE=C
      LINE=34
      ICHOIC=1
      WRITE(ITYOUT,2)
      READ(ITYIN,3)(SCNOSC(1,J),J=1,15)
      CALL PAGER(LINE,NPAGE,C)
      WRITE(ITYOUT,17)
      DO 500 I=1,NPARM
      WRITE(ITYOUT,12)(PARNAM(I,J),J=1,4)
      READ(ITYIN,*)RATING(I)
      IF(RATING(I).LT.C.OR.RATING(I).GT.1)WRITE(ITYOUT,16)
500  CONTINUE
C
1000 DO 1001 L=1,NPARM
      RTING(L)=RATING(L)
1001 CONTINUE
C
C *** FORMAT STATEMENTS ***
C

```

```

1  FORMAT(1X,/,1X,,40HDO YOU WISH TO ENTER YOUR OWN PATINGS ? ,
& 8H(YES/NO)/)
2  FORMAT(1X,30HPLEASE ENTER YOUR SCENARIO DES,
& 24HDESCRIPTION,LIMIT=60 CHARS./)
3  FORMAT(15A4)
4  FORMAT(I2)
5  FORMAT(1X,F5.3)
6  FORMAT(A1)
7  FORMAT(4A4)
8  FORMAT(A3)
9  FORMAT(1X,/,1X,30HENTER NO OF CHOICE OF SCENARIO,/)
10 FORMAT(6X,I2,4H -- ,15A4)
11 FORMAT(1X,/,1X,19HSCENARIOS AVAILABLE,/)
12 FORMAT(1X,4A4,/)
13 FORMAT(1X,20HLIST OF AVAILABLE PARAMETERS,/)
14 FORMAT(1X,I2,4H -- ,4A4)
15 FORMAT(1H1,/,1X,72(1H*),/,20X,30HSYSTEM SELECTION MODEL PROGRAM,
& //,1X,72(1H*),/)
16 FORMAT(1X,/,1X,
& 50H ***-ERROR-*** BAD INPUT--STOP EXECUTION & RESTART,/)
17 FORMAT(1X,/,1X,
& 45HPLEASE ENTER PATINGS (0-1) FOR EACH PARAMETER,/)
END

```

```

C
C *****
C *****
C

```

```

OVERLAY(SSM,3,0)
PROGRAM PARSEL

```

```

C
C *****
C
C READS IN PARAMETERS WANTED AND THEIR RESPECTIVE RANGES THAT
C ARE TO BE INCLUDED IN THE DECISION PROCESS
C
C *****
C

```

```

COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARN,
& SYSET(4),SB1SET(4),SB2SET(4),
& SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),
& PARNAM(40,4)
& ,NUM(40),RATED(40),YN(40),INVERT(40),INDEX
COMMON/PARAM1/NOPRM,CHPRM(40),ABV(40),BELW(40),RLOWR(40),
& HIP(40),RTD(40),IRANGE(40),ISLNO(40)
COMMON/UNITNO/ITYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARN,ISYS
INTEGER SYSDSN,SYSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
& PATED,YN,PARNAM
INTEGER CHPRM
DATA INO/2HNO/

```

```

C
C *** INITIALIZE TO ZERO ***
C
CC 190 KHY=1,40
ABV(KHY)=0

```

```

        BELW(KHY)=0
        RLOWR(KHY)=0
        HIR(KHY)=0
        RTD(KHY)=0
        IPANGE(KHY)=0
        CHPPM(KHY)=0
        ISWNO(KHY)=0
199    CONTINUE
C
C *** NO. OF PARAMETERS CHOSEN ***
C
C    TEKTRONIX PAGING
        LINE=34
        NPAGE=0
        CALL PAGER(LINE,NPAGE,C)
        WRITE(ITYOUT,3)
        READ(ITYIN,*)NOPRM
        IF(NOPRM.LE.0.OR.NOPRM.GT.NPARM)WRITE(ITYOUT,18)
        WRITE(ITYOUT,16)
C
C *** READ IN CHOICE OF PARAMETERS
        DO 200 I=1,NOPRM
            WRITE(ITYOUT,17)
            READ(ITYIN,*)CHPRM(I)
            IF(CHPRM(I).LE.0.OR.CHPRM(I).GT.NPARM)WRITE(ITYOUT,18)
200    CONTINUE
C
C *** READ PAGES OR LIMITS ON INDIVIDUAL PARAMETERS, IF ANY
C
C    TEKTRONIX PAGING
        NPAGE=0
        LINE=34
        CALL PAGER(LINE,NPAGE,C)
        DO 300 I=1,NOPRM
            K=CHPRM(I)
            LINE=34
            NPAGE=0
            IF(I.EQ.5.OR.I.EQ.10.OR.I.EQ.15.OR.I.EQ.20.OR.I.EQ.25.OR.I.EQ.30.
& .OR.I.EQ.35.OR.I.EQ.40)CALL PAGER(LINE,NPAGE,C)
            WRITE(ITYOUT,5)(PARNAM(K,J),J=1,4)
            IF(NUM(K).EQ.1)GO TO 1000
            IF(RATED(K).EQ.1)GO TO 2000
            IF(YN(K).EQ.1)WRITE(ITYOUT,15)
300    CONTINUE
C
        GO TO 9190
C
C *** SPECIFY A LIMIT ? ***
C
1000    WRITE(ITYOUT,6)
        READ(ITYIN,7)IANS
        IF(IANS.EQ.INO)ISWNO(I)=1
        IF(IANS.EQ.INO)GO TO 300
C

```

```

C *** ABOVE,BELOW, OR IN BETWEEN ? ***
C
      WRITE(ITYOUT,8)
      READ(ITYIN,9)IRNG
      IF(IRNG.LE.0.OR.IRNG.GT.3)WRITE(ITYOUT,18)
      IRANGE(I)=IPNG
C
      GO TO(10,20,30),IPNG
C
C *** ABOVE ***
C
10      WRITE(ITYOUT,11)(PARNAM(K,J),J=1,4)
      READ(ITYIN,*)ABV(K)
      BELW(K)=0
      RLOWR(K)=0
      HIR(K)=0
      GO TO 1001
C
C *** BELOW ***
C
20      WRITE(ITYOUT,12)(PARNAM(K,J),J=1,4)
      READ(ITYIN,*)BELW(K)
      ABV(K)=0
      RLOWR(K)=0
      HIR(K)=0
      GO TO 1001
C
C *** IN BETWEEN ****
C
30      WRITE(ITYOUT,12)(PARNAM(K,J),J=1,4)
      READ(ITYIN,*)HIR(K)
      WRITE(ITYOUT,11)(PARNAM(K,J),J=1,4)
      READ(ITYIN,*)RLOWR(K)
      ABV(K)=0
      BELW(K)=0
C
1001 GO TO 300
C
C *** SPECIFY LIMIT ? ***
C
2000 WRITE(ITYOUT,6)
      READ(ITYIN,7)IIANS
      IF(IIANS.EQ.INO)ISWNO(I)=1
      IF(IIANS.EQ.INO)GO TO 300
      WRITE(ITYOUT,14)
      READ(ITYIN,9)IIANS
      IF(IIANS.LE.0.OR.IIANS.GT.5)WRITE(ITYOUT,18)
      RTD(K)=IIANS
      IRANGE(I)=IIANS
      GO TO 300
C
C *** FORMAT STATEMENTS ***
C
3      FORMAT(1X,45HENTER THE NUMBER OF PARAMETERS ? )

```

AD-A119 160

SOUTHEASTERN CENTER FOR ELECTRICAL ENGINEERING EDUCAT--ETC F/G 9/2
ADVANCED TECHNOLOGY MULTIPLE CRITERIA DECISION MODEL.(U)

NOV 81 P J SWEENEY, K B BERNER, J R FRAKER F33615-77-C-2059

UNCLASSIFIED

AFWAL-TR-81-2112

NL

4 OF 4

40 A
1 9100

END
DATE
FILMED
10-82
DTIC


```

      & 24H IN THE DECISION PROCESS/)
4      FORMAT(I2)
5      FORMAT(1X,/,1X,13HPARAMETER -- ,4A4)
6      FORMAT(53H WOULD YOU LIKE TO SPECIFY A LIMIT THAT THE PARAMETER/,
      & 26H HAS TO SATISFY ? (YES/NO)/)
7      FORMAT(A3)
8      FORMAT (1X,37HDO YOU WANT THE PARAMETER TO BE ABOVE,
      & 7H, BELOW,/,1X,34HOR IN BETWEEN THE SPECIFIED LIMIT?,/,
      & 1X,32H(CHOOSE 1, 2, OR 3 RESPECTIVELY),/)
9      FORMAT(I1)
11     FORMAT(1X,4A4,1H>)
12     FORMAT(1X,4A4,1H<)
14     FORMAT(1X,46HTHIS PARAMETER CAN BE PATED AS EXCELLENT,GOOD,,
      & 24HFAIR,POOR, OR VERY POOR,/,1X,26HCHOOSE 1,2,3,4,OR,5,PESP.,/)
15     FORMAT(1X,47HA QUALATITIVE PARAMETER--CANNOT SPECIFY A RANGE)
16     FORMAT(1X,/,1X,38HENTER EACH INTEGER NO. ASSOCIATED WITH,
      & 13H EACH CHOSEN ,/,1X,34HPARAMETER (FROM LOWEST TO HIGHEST),/)
17     FORMAT(1X,18H PARAMETER NO. -- )
18     FORMAT(1X,/,1X,5GH ***-ERROR-BAD INPUT--STOP EXECUTION & RESTART,
      & /)
9190  CONTINUE
      END

```

```

C
C *****
C *****
C

```

```

      OVERLAY(SSM,4,0)
      PROGRAM ELIMIN

```

```

C
C *****
C      THIS ROUTINE ELIMINATES UNWANTED PARAMETERS AND RANGES
C      OF OTHER PAPAMETERS
C *****
C

```

```

      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPAM,
      & SYSSET(4),SB1SET(4),SB2SET(4),
      & SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
      & NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPAM,ISYS
      COMMON/PARAM1/NOPRM,CHPRM(40),ABV(40),BELW(40),RLOWR(40),
      & HIR(40),RTD(40),IRANGE(40),ISNO(40)
      COMMON/PARAM2/ELIM(7,10,40)
      COMMON/HDR2/NSCEN,SCNOSC(25,15),INDEX1,ICHOCIC
      COMMON/DB/DBNAME(4)
      DIMENSION PARAM(85,10)
      DIMENSION DUM(10)
      INTEGER MATRIX,ELIM,CHPRM
      INTEGER SYSDSN,SYSSET,SB2SET,SYSNAM,SUB1NM,SUB2NM,PARNAM,
      & RATED,YN,DBNAME,SCNOSC

```

```

C
C *** INITIALIZE MATRIX TO 1 ***
C
      DO 995 KMN2=1,NPAM
      DO 995 KMN=1,INDEX

```

```

      DO 995 KMN1=1,NSYS
        MATRIX(KMN2,KMN,KMN1)=1
995  CONTINUE
C
C *** INITIALIZE ELIM TO 0 ***
C
      DO 997 KJU=1,INDEX
      DO 997 KJU1=1,NSYS
      DO 997 KJU2=1,NPARM
        ELIM(KJU,KJU1,KJU2)=0
997  CONTINUE
      REWIND IUPP
      REWIND IPP
      K=0
C
C *** FIND PARAMETERS CHOSEN ***
C
      DO 200 I=1,NPARM
      DO 205 M=1,NOPRM
        IF(I.EQ.CHPRM(M))GO TO 210
205  CONTINUE
C
C *** INDEX FILE TO CORRECT PARAMETER ***
C
      DO 215 LJ=1,INDEX
      READ(IPP,*)(DUM(KLK),KLK=1,NSYS)
215  CONTINUE
C
C *** ELIMINATE PARAMETER ***
C
      DO 220 NL=1,INDEX
      DO 220 NLL=1,NSYS
        MATRIX(I,NL,NLL)=0
220  CONTINUE
200  CONTINUE
C
C *** TEST TO SEE IF ALL SYSTEMS HAVE BEEN ELIMINATED ***
C
      SW2=0
      DO 599 L=1,NOPRM
      DO 599 LL=1,INDEX
      DO 599 LLL=1,NSYS
        K=CHPRM(L)
        IF(MATRIX(K,LL,LLL).EQ.0)SW2=1
599  CONTINUE
      IF(SW2.EQ.0)GO TO 9190
      SW1=0
      DO 555 L=1,INDEX
      DO 555 LL=1,NSYS
      DO 555 LLL=1,NPARM
        IF(MATRIX(LL,L,LL).EQ.1)SW1=1
555  CONTINUE
C
      LINE=34

```

```

        NPAGE=C
        CALL PAGER(LINE,NPAGE,C)
        WRITE(ITYOUT,5)(DBNAME(KK),KK=1,4),SYSDSN,
C (SCNDSC(ICH0IC,M),M=1,15),1
        IF(S=1.EQ.0)WRITE(ITYOUT,4)
C
C *** CALL ELIMINATION TABLE ***
C
        CALL OVERLAY(3HSSM,4,1)
        RE=IND IPP
C
C *** IF ALL SYSTEMS HAVE BEEN ELIMINATED DON'T PRINT UTILITY
C VALUE TABLE ***
C
        IF(S=1.EQ.0)CALL EXIT
        GO TO 9190
C
C *** READ IN PARAMETER DATA ***
C
210   DO 300 J=1,INDEX
        READ(IPP,*)(PARAM(J,KL),KL=1,NSYS)
300   CONTINUE
        K=K+1
C
C *** TEST FOR WHICH TYPE OF PARAMETER & WHAT RANGE, IF ANY ***
C
        IF(RATED(I).EQ.1)GO TO 101
        IF(YN(I).EQ.1)GO TO 102
        IF(ABV(I).EQ.0.AND.BELW(I).EQ.0.AND.HIR(I).EQ.0.AND.
C RLOWR(I).EQ.0)GO TO 103
        IF(ABV(I).EQ.0.AND.BELW(I).EQ.0)GO TO 104
        IF(ABV(I).EQ.0)GO TO 105
C
C *** NUMERICAL PARAMETER -- RANGE IS ABOVE ***
C
        DO 301 L=1,INDEX
        DO 301 LL=1,NSYS
        IVBIN=C
        DO 302 MM=1,NOPRM
        NKI=CHPRM(MM)
        IF(MATRIX(NKI,L,LL).EQ.0.AND.ISWNO(K).EQ.0)GO TO 303
302   CONTINUE
        IF(IVBIN.EQ.1)GO TO 301
        IF(ISWNO(K).EQ.1)GO TO 301
        IF(PARAM(L,LL).GE.ABV(I))GO TO 304
        DO 305 LM=1,NOPRM
        KII=CHPRM(LM)
C
C *** PARAMETER IS ELIMINATED ***
C
        MATPIX(KII,L,LL)=0
305   CONTINUE
        ELIM(L,LL,K)=I
        GO TO 301

```

```

C
C *** PARAMETER IS ALLOWED TO REMAIN ***
C
304 MATRIX(I,L,LL)=1
301 CONTINUE
GO TO 200

C
C *** TEST TO SEE IF PARAMETER WOULD BE ELIMINATED ANYHOW ***
C
303 IF(PARAM(L,LL).LT.ABV(I))ELIM(L,LL,K)=I
MATRIX(I,L,LL)=0
IVBIN=1
GO TO 302

C
C *** NUMERICAL PARAMETER -- RANGE IS BELOW ***
C
105 DO 401 L=1,INDEX
DO 401 LL=1,NSYS
IVBIN=0
DO 402 MM=1,NOPRM
NKI=CHPPM(MM)
IF(MATRIX(NKI,L,LL).EQ.0.AND.ISWNO(K).EQ.0)GO TO 403
CONTINUE
402 IF(IVBIN.EQ.1)GO TO 401
IF(ISWNO(K).EQ.1)GO TO 401
IF(PAPAM(L,LL).LE.BELW(I))GO TO 404
DO 405 LM=1,NOPRM
KII=CHPPM(LM)
MATRIX(KII,L,LL)=0
405 CONTINUE
ELIM(L,LL,K)=I
GO TO 401
404 MATRIX(I,L,LL)=1
401 CONTINUE
GOTO 200
403 IF(PARAM(L,LL).GT.BELW(I))ELIM(L,LL,K)=I
MATRIX(I,L,LL)=0
IVBIN=1
GO TO 402

C
C *** NUMERICAL PARAMETER -- RANGE IS IN BETWEEN ***
C
104 DO 501 L=1,INDEX
DO 501 LL=1,NSYS
IVBIN=0
DO 502 MM=1,NOPRM
NKI=CHPPM(MM)
IF(MATRIX(NKI,L,LL).EQ.0.AND.ISWNO(K).EQ.0)GO TO 503
CONTINUE
502 IF(IVBIN.EQ.1)GO TO 501
IF(ISWNO(K).EQ.1)GO TO 501
IF(PARAM(L,LL).GE.RLOWR(I).AND.PAPAM(L,LL).LE.HIR(I))GO TO 504
DO 505 LM=1,NOPRM
KII=CHPPM(LM)

```

```

      MATRIX(KII,L,LL)=0
505  CONTINUE
      ELIM(L,LL,K)=I
      GO TO 501
504  MATRIX(I,L,LL)=1
501  CONTINUE
      GO TO 200
503  IF(PARAM(L,LL).LT.RLOWR(I).OR.PARAM(L,LL).GT.HIP(I))
      & ELIM(L,LL,K)=I
      MATRIX(I,L,LL)=0
      IVBIN=1
      GO TO 502
C
103  GO TO 200
C
C *** QUALITATIVE PARAMETER ***
C
102  DO 601 L=1,INDEX
      DO 601 LL=1,NSYS
      IVBIN=0
      DO 602 MM=1,NOPRM
      NKI=CHPRM(MM)
      IF(MATRIX(NKI,L,LL).EQ.0.AND.ISWNO(K).EQ.0)GO TO 603
602  CONTINUE
      IF(IVBIN.EQ.1)GO TO 601
      IF(PARAM(L,LL).EQ.1)GO TO 604
      DO 605 LM=1,NOPRM
      KII=CHPRM(LM)
      MATRIX(KII,L,LL)=0
605  CONTINUE
      ELIM(L,LL,K)=I
      GO TO 601
604  MATRIX(I,L,LL)=1
601  CONTINUE
      GO TO 200
603  IF(PARAM(L,LL).NE.1)ELIM(L,LL,K)=I
      MATRIX(I,L,LL)=0
      IVBIN=1
      GO TO 602
C
C *** RATED PARAMETER ***
C
101  IF(PTD(I).EQ.0)GO TO 200
      DO 701 L=1,INDEX
      DO 701 LL=1,NSYS
      IVBIN=0
      DO 702 MM=1,NOPRM
      NKI=CHPRM(MM)
      IF(MATRIX(NKI,L,LL).EQ.0.AND.ISWNO(K).EQ.0)GO TO 703
702  CONTINUE
      IF(IVBIN.EQ.1)GO TO 701
      IF(ISWNO(K).EQ.1)GO TO 701
      IF(PARAM(L,LL).LE.RTD(I))GO TO 704
      DO 705 LM=1,NOPRM

```

```

      KII=CHPRM(LM)
      MATPIX(KII,L,LL)=0
705  CONTINUE
      ELIM(L,LL,K)=I
      GO TO 701
704  MATRIX(I,L,LL)=1
701  CONTINUE
      GO TO 200
703  IF(PAPAM(L,LL).GT.RTD(I))ELIM(L,LL,K)=I
      MATRIX(I,L,LL)=0
      IVBIN=1
      GO TO 702

C
C *** FORMAT STATEMENTS ***
C
1    FORMAT(1X,I1)
2    FORMAT(A1)
3    FORMAT(I1)
4    FORMAT(1X,/,20X,32HALL SYSTEMS HAVE BEEN ELIMINATED)
5    FORMAT(1H ,72(1H*)/1X,5X,4A4,5X,5HCASE ,I1,3X,
C    17HELIMINATION TABLE,/,1X,15A4,5X,5HPAGE ,I2,/,
C    1X,72(1H*))
9190 CONTINUE
      END

C
C
C
C *****
C *****
C
      OVERLAY(SSM,4,1)
      PROGRAM TABLE

C
C *****
C
C    THIS ROUTINE FINDS WHICH SYSTEMS HAVE BEEN ELIMINATED
C    FROM THE DECISION PROCESS AND PRINTS THE REASONS WHY IT
C    WAS ELIMINATED.
C
C *****
C
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
      &SYSSET(4),SB1SET(4),SB2SET(4),
      &SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
      &NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/PARAM1/NOPRM,CHPRM(40),ABV(40),BELW(40),PLOWR(40),
      &HIP(40),RTD(40),IRANGE(40),ISNO(40)
      COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
      COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
      COMMON/PARAM2/ELIM(7,10,40)
      COMMON/OB/DBNAME(4)
      INTEGER ICUALT(5,3),ELIM,DBNAME,SCNDSC
      INTEGER CHPRM,SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
      &SUB2NM,RATED,YN

```

```

DATA IQUALT(1,1)/4HEXCE/,IQUALT(1,2)/4HLEN/,IQUALT(1,3)/4HT /
DATA IQUALT(2,1)/4HG00D/,IQUALT(2,2)/4H /,IQUALT(2,3)/4H /
DATA IQUALT(3,1)/4HFAIR/,IQUALT(3,2)/4H /,IQUALT(3,3)/4H /
DATA IQUALT(4,1)/4HPOOR/,IQUALT(4,2)/4H /,IQUALT(4,3)/4H /
DATA IQUALT(5,1)/4HVERY/,IQUALT(5,2)/4H POO/,IQUALT(5,3)/4HR /
WRITE(ITYOUT,101)
WRITE(ITYOUT,102)(SCNDSC(ICH0IC,I),I=1,15)

C
C *** GO TO THE CORRECT CASE ***
C
GO TO (1000,2000,3000),SYSDSN

C
C *** CASE 1 ***
C
1000 WRITE(ITYOUT,103)(SYSSET(I),I=1,4)
      LINE=14
      NPAGE=1
      DO 1100 L=1,NSYS
        LL=1
        DO 1200 LLL=1,NOPRM
          IF(ELIM(LL,L,LLL).EQ.0)GO TO 1200

C
C *** PAGING ***
C
CALL PAGER (LINE,NPAGE,5)
IF(LINE.EQ.0) WRITE (ITYOUT,1) (DBNAME(M),M=1,4),SYSDSN,
C (SCNDSC(ICH0IC,M),M=1,15),NPAGE
IF(LINE.EQ.0) LINE=4
      K=ELIM(LL,L,LLL)

C
C *** GO TO CORRECT PARAMETER ***
C
      IF(NUM(K).EQ.1)GO TO 1110
      IF(RATED(K).EQ.1)GO TO 1120
      IF(YN(K).EQ.1)GO TO 1130
1200 CONTINUE
1100 CONTINUE
GO TO 9190

C
C *** NUMERICAL PARAMETERS ***
C
1110 GO TO (1111,1112,1113),IRANGE(LL)
1111 WRITE(ITYOUT,104)(SYSNAM(L,M),M=1,4),(PARNAM(K,M),M=1,4),ABV(K)
      LINE=LINE+2
      GO TO 1200
1112 WRITE(ITYOUT,105)(SYSNAM(L,M),M=1,4),(PARNAM(K,M),M=1,4),
      & BELW(K)
      LINE=LINE+2
      GO TO 1200
1113 WRITE(ITYOUT,106)(SYSNAM(L,M),M=1,4),(PARNAM(K,M),M=1,4),
      & PLOWP(K),(PARNAM(K,M),M=1,4),HIR(K)
      LINE=LINE+5
      GO TO 1200

C

```

```

C *** RATED PARAMETERS ***
C
1120      JJ=IRANGE(LLL)
          WRITE(ITYOUT,107)(SYSNAM(L,M),M=1,4),(PARNAM(K,M),M=1,4),
C          (IQUALT(JJ,M),M=1,3)
          LINE=LINE+2
          GO TO 1200

C
C *** QUALITATIVE PARAMETERS ***
C
1130      WRITE(ITYOUT,108)(SYSNAM(L,M),M=1,4),(PARNAM(K,M),M=1,4)
          LINE=LINE+2
          GO TO 1200

C
C *** CASE 2 ***
C
2000      WRITE(ITYOUT,109)(SYSSET(I),I=1,4),(SELSET(I),I=1,4)
          LINE=14
          NPAGE=1
          DO 2100 L=1,NSYS
            DO 2200 LL=1,INDEX
              DO 2300 LLL=1,NOPRM
                IF(ELIM(LL,L,LLL).EQ.0)GO TO 2300

C
C *** PAGING ***
C
          CALL PAGER (LINE,NPAGE,5)
          IF (LINE.EQ.0) WRITE (ITYOUT,1) (DBNAME(M),M=1,4),SYSDSN,
C          (SCNDSC(ICH0IC,MM),MM=1,15),NPAGE
          IF (LINE.EQ.0) LINE=4
                  K=ELIM(LL,L,LLL)
                  IF(NUM(K).EQ.1)GO TO 2110
                  IF(RATED(K).EQ.1)GO TO 2120
                  IF (YN(K).EQ.1)GO TO 2130

2300      CONTINUE
2200      CONTINUE
2100      CONTINUE
          GO TO 9190

C
C *** NUMERICAL PARAMETERS ***
C
2110      GO TO (2111,2112,2113),IRANGE(LLL)
2111      WRITE(ITYOUT,110)(SYSNAM(L,M),M=1,4),(SUB1NM(LL,M),M=1,4),
C          (PARNAM(K,M),M=1,4),ABV(K)
          LINE=LINE+2
          GO TO 2300
2112      WRITE(ITYOUT,111)(SYSNAM(L,M),M=1,4),(SUB1NM(LL,M),M=1,4),
C          (PARNAM(K,M),M=1,4),BELW(K)
          LINE=LINE+2
          GO TO 2300
2113      WRITE(ITYOUT,112)(SYSNAM(L,M),M=1,4),(SUB1NM(LL,M),M=1,4),
C          (PARNAM(K,M),M=1,4),RLOWR(K),(PARNAM(K,M),M=1,4),HIR(K)
          LINE=LINE+5
          GO TO 2300

```



```

C *** RATED PARAMETERS ***
2120      JJ=IRANGE(LL)
2121      WRITE(ITCUT,113)(SYSNAM(L,M),M=1,4),(SUB1NM(LL,M),M=1,4),
      &      (PARNAM(K,M),M=1,4),(ICUALT(JJ,M),M=1,3)
      LINE=LINE+2
      GO TO 2300
C *** QUALITATIVE PARAMETERS ***
2130      WRITE(ITCUT,114)(SYSNAM(L,M),M=1,4),(SUB1NM(LL,M),M=1,4),
      &      (PARNAM(K,M),M=1,4)
      LINE=LINE+2
      GO TO 2300
C
C *** CASE 3 ***
C
3000      WRITE(ITCUT,115)(SYSSET(I),I=1,4),(SB1SET(I),I=1,4),(SP2SET(I),
      &      I=1,4)
      LINE=14
      NPAGE=1
      DO 3100 L=1,NSYS
      INDX=0
      DO 3200 L1=1,NSUB1
      DO 3300 LL=1,NSUB2
      INDX=INDX+1
      J=INDX
      DO 3400 LLL=1,NOPRM
      IF(ELIM(J,L,LLL).EQ.0)GO TO 3400
C
C *** PAGING ***
C
      CALL PAGER (LINE,NPAGE,5)
      IF (LINE.EQ.0) WRITE (ITCUT,1) (CBNAME(M),
      &      M=1,4),SYSDSN,(SCNDSC(ICOIC,M),M=1,15),
      &      NPAGE
      IF (LINE.EQ.0) LINE=4
      K=ELIM(J,L,LLL)
      IF(NUM(K).EQ.1)GO TO 3110
      IF(RATED(K).EQ.1)GO TO 3120
      IF(YN(K).EQ.1)GO TO 3130
3400      CONTINUE
3300      CONTINUE
3200      CONTINUE
3100      CONTINUE
      GO TO 9190
C
C *** NUMERICAL PARAMETERS ***
C
3110      GO TO (3111,3112,3113),IRANGE(LL)
3111      WRITE(ITCUT,116)(SYSNAM(L,M),M=1,4),(SUB1NM(L1,M),M=1,4),
      &      (SUB2NM(L1,LL,M),M=1,4),(PARNAM(K,M),M=1,4),ABV(K)
      LINE=LINE+2
      GO TO 3400
3112      WRITE(ITCUT,117)(SYSNAM(L,M),M=1,4),(SUB1NM(L1,M),M=1,4),
      &      (SUB2NM(L1,LL,M),M=1,4),(PARNAM(K,M),M=1,4),BELOW(K)
      LINE=LINE+2

```

```

      GO TO 3400
3113  WRITE(ITYOUT,118)(SYSNAM(L,M),M=1,4),(SUB1NM(L1,M),M=1,4),
      & (SUB2NM(L1,LL,M),M=1,4),(PARNAM(K,M),M=1,4),RLOWP(K),
      & (PARNAM(K,M),M=1,4),HIR(K)
      LINE=LINE+5
      GO TO 3400
C
C *** RATED PARAMETERS ***
C
3120  JJ=IRANGE(LLL)
      WRITE(ITYOUT,119)(SYSNAM(L,M),M=1,4),(SUB1NM(L1,M),M=1,4),
      & (SUB2NM(L1,LL,M),M=1,4),(PARNAM(K,M),M=1,4),
      & (IQUALT(JJ,M),M=1,3)
      LINE=LINE+2
      GO TO 3400
C
C *** QUALITATIVE PARAMETERS ***
C
3130  WRITE(ITYOUT,120)(SYSNAM(L,M),M=1,4),(SUB1NM(L1,M),M=1,4),
      & (SUB2NM(L1,LL,M),M=1,4),(PARNAM(K,M),M=1,4)
      LINE=LINE+2
      GO TO 3400
C
C *** FORMAT STATEMENTS ***
C
1     FORMAT(1H1,72(1H*))/1X,5X,4A4,5X,5HCASE ,11,3X,
      & 17HELIMINATION TABLE,/,1X,15A4,5X,5HPAGE ,12,/,
      & 1X,72(1H*))
101   FORMAT (/30X,17HELIMINATION TABLE/)
102   FORMAT (10X,15A4,/)
103   FORMAT (11X,4A4,10X,24HREASONS FOR ELIMINATION ,/11X,
      & 16(1H-),10X,23(1H-)/)
104   FORMAT(1X,10X,4A4,10X,4A4,1X,1H<,F12.3)
105   FORMAT(1X,10X,4A4,10X,4A4,1X,1H>,F12.3)
106   FORMAT(1X,10X,4A4,10X,4A4,1H<,F12.3/1X,54X,2HOR/,1X,36X,
      & 4A4,1H>,F12.3)
107   FORMAT(1X,10X,4A4,10X,4A4,1H<,3A4)
108   FORMAT(1X,10X,4A4,10X,4A4)
109   FORMAT (1X,4A4,5X,4A4,5X,23HREASONS FOR ELIMINATION,/,
      & 1X,16(1H-),5X,16(1H-),5X,23(1H-)/)
110   FORMAT(1X,4A4,5X,4A4,5X,4A4,1X,1H<,F12.3)
111   FORMAT(1X,4A4,5X,4A4,5X,4A4,1X,1H>,F12.3)
112   FORMAT(1X,4A4,5X,4A4,5X,4A4,1H<,F12.3,/1X,6CX,2HOR/,1X,42X,
      & 4A4,1H>,F12.3)
113   FORMAT(1X,4A4,5X,4A4,5X,4A4,1X,1H<,1X,3A4)
114   FORMAT(1X,4A4,5X,4A4,5X,4A4)
115   FORMAT (57X,11HREASONS FOR,/,1X,4A4,2X,4A4,2X,4A4,6X,
      & 11HELIMINATION,/,1X,4(16(1H-),2X)/)
116   FORMAT(1X,4A4,2X,4A4,2X,4A4,2X,4A4,1X,1H<,F12.3)
117   FORMAT(1X,4A4,2X,4A4,2X,4A4,2X,4A4,1X,1H>,F12.3)
118   FORMAT(1X,4A4,2X,4A4,2X,4A4,2X,4A4,1H<,F12.3/1X,59X,2HCO/,
      & ,1X,54X,4A4,1H>,F12.3)
119   FORMAT(1X,4A4,2X,4A4,2X,4A4,2X,4A4,2H<,3A4)
120   FORMAT(1X,4A4,2X,4A4,2X,4A4,2X,4A4)

```

919C CONTINUE
END

```
C
C *****
C *****
C
      OVERLAY(SSM,5,0)
      PROGRAM INVRT
C *****
C THIS ROUTINE INVERTS PARAMETER DATA SO THAT LOW VALUES
C CONSTITUTE THE BEST UTILITY
C *****
C
      COMMON/HDP1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
      & SYSSET(4),SB1SET(4),SB2SET(4),
      & SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
      & NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
      DIMENSION DUM(10),PARAM(85,10),VPARAM(85,10)
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,  SUB1NM,PARNAM,
      & RATED,YN,SUB2NM
C
      REWIND IPP
      REWIND IVPARM
C
C *** INITIALIZE VPARAM TO ZERO ***
C
      DO 50 J=1,INDEX
      DO 50 JJ=1,NSYS
      VPARAM(J,JJ)=0
50    CONTINUE
C
C *** READ IN PARAMETER DATA ***
C
      DO 100 I=1,NPARM
      IF(INVERT(I).EQ.0)GO TO 101
C
      DO 300 K=1,INDEX
      READ(IPP,*)(PARAM(K,KK),KK=1,NSYS)
300    CONTINUE
C
C *** FIND LARGEST VALUE FOR EACH SYSTEM ***
C
      DO 400 KK=1,NSYS
      BIG=0.0
      DO 450 K=1,INDEX
      IF(PARAM(K,KK).GT.BIG)BIG=PARAM(K,KK)
450    CONTINUE
C
C *** INVERT PARAMETER VALUES ***
C
      DO 460 L=1,INDEX
      IF(PARAM(L,KK).EQ.0)GO TO 460
      VPARAM(L,KK)=BIG/PARAM(L,KK)
```

```

460  CONTINUE
C
400  CONTINUE
C
C *** OUTPUT PARAMETER VALUES ***
C
      DO 500 LI=1,INDEX
        WRITE(IVPARM,*)(VPARAM(LI,LL1),LL1=1,NSYS)
500  CONTINUE
C
100  CONTINUE
C
      GO TO 9190
C
C *** IF PARAMETERS VALUES DON'T NEED INVERTING WRITE OUT
C *** PARAMETER VALUES ***
101  DO 200 J=1,INDEX
      READ(IPP,*)(PARAM(J,KLK),KLK=1,NSYS)
      WRITE(IVPARM,*)(PARAM(J,KMK),KMK=1,NSYS)
200  CONTINUE
      GO TO 100
C
C *** FORMAT STATEMENTS ***
C
1    FORMAT(A1)
9190 CONTINUE
      END
C
C *****
C *****
C
      OVERLAY(SSM,6,0)
      PROGRAM UTILT
C
C *****
C
      THIS ROUTINE DETERMINES THE SYSTEM VALUE OR UTILITY
      FOR EACH SYSTEM
C
C *****
C
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
& SYSSET(4),SB1SET(4),SB2SET(4),
& SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
& NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/UNITNO/ITTYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
      COMMON/RTNG/RATING(40)
      COMMON/PARAM3/MATRIX(40,7,10)
      COMMON/PARAM1/NOPEM,CHPRM(40),ABV(40),BELW(40),PLCWP(40),
& HIP(40),RTD(40),IRANGE(40),IS.NO(40)
      COMMON/PARAM4/SYSNRM(85,10)
      COMMON/DB/DBNAME(4)
      DIMENSION PARAM(85,10)
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET      E      1-      CE

```

```

      E PAPNAM,RATED,YN
      INTEGER CHPRM,DBNAME
      DATA IYES/3MYES/
C
C *** INITIALIZE SYSNRM ***
C
      REWIND IVPARM
      DO 50 J=1,INDEX
      DO 50 JJ=1,NSYS
      SYSNRM(J,JJ)=0
50    CONTINUE
C
C *** FIND TOTAL SUM OF RATINGS ***
C
      SUMM=0
      DO 800 JI=1,NOPRM
      KP=CHPRM(JI)
      SUMM=PATING(KP) + SUMM
800   CONTINUE
C
C *** NORMALIZE RATINGS ***
C
      DO 850 LJ=1,NOPRM
      KP=CHPRM(LJ)
      RATING(KP)=PATING(KP)/SUMM
850   CONTINUE
C
C *** READ IN INVERTED PARAMETERS ***
C
      DO 100 I=1,NPARM
      DO 110 II=1,INDEX
      READ(IVPARM,*)(PARAM(II,III),III=1,NSYS)
110   CONTINUE
C
C *** ZERO OUT PARAMETERS AND FIND THE LARGEST & SMALLEST
C *** VALUES AMONG THE SYSTEMS
C
      DO 120 II=1,INDEX
      BIG=0.0
      SMALL = 1E20
      DO 130 III=1,NSYS
      PARAM(II,III)=MATRIX(I,II,III)*PARAM(II,III)
C
      IF(PARAM(II,III).EQ.0.0)GO TO 130
      IF(PARAM(II,III).LT.SMALL)SMALL=PARAM(II,III)
      IF(PARAM(II,III).GT.BIG)BIG=PARAM(II,III)
130   CONTINUE
C
C *** CALCULATE UTILITY ***
C
      DO 140 III=1,NSYS
      IF(PARAM(II,III).EQ.0.0)GO TO 140
      IF(BIG.EQ.SMALL)UTIL=1
      IF(BIG.EQ.SMALL)GO TO 909

```

```

          UTIL=((BIG-PARAM(II,III))/(BIG-SMALL))
909      PARAM(II,III)=UTIL
C
C *** CALCULATE RATED UTILITY ***
C
      PARAM(II,III)=PARAM(II,III)*RATING(I)
C
C *** SUM RATED UTILITIES TO FIND SYSTEM VALUE ***
C
      IF(PARAM(II,III).EQ.0)GO TO 140
      SYSNRM(II,III)=PARAM(II,III) + SYSNRM(II,III)
140     CONTINUE
C
C *** NORMALIZE THE SYSTEM VALUE ***
C
C
120     CONTINUE
100     CONTINUE
C
C *** PRINT RESULTS IN A TABLE ***
C
      LINE=34
      NPAGE=0
      CALL PAGER(LINE,NPAGE,0)
      WRITE(ITYOUT,2)
      READ(ITYIN,3)IANSR
      IF(IANSR.EQ.IYES)CALL OVERLAY(3HSSM,6,1)
      DO 1000 I=1,INDEX
        WRITE(ISYS,*)(SYSNRM(I,J),J=1,NSYS)
1000    CONTINUE
C
C *** FORMAT STATEMENTS ***
C
1      FORMAT(I1)
2      FORMAT(//,1X,44HDO YOU WISH TO SEE TABULAR OUTPUT ? (YES/NO),
      & /)
3      FORMAT(A3)
      END
C
C *****
C *****
C
      OVERLAY(SSM,6,1)
      PPROGRAM OUTPUT
C
C *****
C
      THIS ROUTINE GIVES THE USER A CHOICE OF EITHER TEKTRONIX
      GRAPHICS OR TABULAR OUTPUT. THE OUTPUT CONSISTS OF SYSTEM
      VALUES THAT HAVE DETERMINED WHICH SYSTEM IS OPTIMUM
C
C *****
C
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARN,

```

```

& SYSSFT(4),SB1SET(4),SB2SET(4),
& SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
& NUM(40),RATED(40),YN(40),INVERT(40),INDEX
COMMON/UNITNO/ITYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPAR*,ISYS
COMMON/PARAM4/SYSNRM(85,10)
COMMON/DB/DBNAME(4)
COMMON/HDR2/NSCEN,SCNDSC(25,15),INDEX1,ICHOIC
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
& PARNAM,RATED,YN,DBNAME,SCNDSC
DIMENSION ICHAR(10),IOPT(10)
DATA ICHAR/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ/
DATA IBLANK/1H /

C
LINE=34
NPAGE=0
CALL PAGER(LINE,NPAGE,0)

C
WRITE(ITYOUT,11)(DBNAME(L99),L99=1,4),SYSDSN,
& (SCNDSC(ICHOIC,M),M=1,15),NPAGE
WRITE(ITYOUT,1)
WRITE(ITYOUT,4)(DBNAME(M),M=1,4)
WRITE(ITYOUT,5)(SYSSET(M),M=1,4)
LINE=9
DO 1100 I=1,NSYS
    WRITE(ITYOUT,15)ICHA(I),(SYSNAM(I,M),M=1,4)
    LINE=LINE+1
1100 CONTINUE
C
GO TO (1000,2000,3000),SYSDSN

C
C *** CASE 1 ***
C
1000 CONTINUE
BIG=0.0
J=1
DO 1010 I=1,NSYS
    IF(SYSNRM(J,I).GT.BIG)BIG=SYSNRM(J,I)
1010 CONTINUE
C
DO 1090 KU=1,10
    IOPT(KU)=IBLANK
1090 CONTINUE
C
ICOUNT=0
C
DO 1020 I=1,NSYS
    IF(BIG.NE.SYSNRM(J,I))GO TO 1020
C
ICOUNT=ICOUNT + 1
IOPT(ICOUNT)=ICHA(I)
1020 CONTINUE
C
IF(BIG.NE.0)GO TO 1030
WRITE(ITYOUT,6)

```

```

      GO TO 9190
C
C   TEKTRONIX PAGING
1030 CALL PAGER(LINE,NPAGE,6)
      IF (LINE.EQ.0) WRITE (ITYOUT,11) (DBNAME(M),M=1,4),SYSDSN,
      & (SCNDSC(ICH0IC,M),M=1,15),NPAGE
      IF (LINE.EQ.0) LINE=4
      CALL WRTE(IOPT,J)
      LINE=LINE+6
      GO TO 9190
C
C *** CASE 2 ***
C
2000 CONTINUE
C
      NCALL =0
      DO 2011 J=1,NSUB1
      CALL PAGER (LINE,NPAGE,10)
      IF (LINE.EQ.0) WRITE (ITYOUT,11) (DBNAME(M),M=1,4),SYSDSN,
      & (SCNDSC(ICH0IC,M),M=1,15),NPAGE
      IF (LINE.EQ.0) LINE=4
      WRITE(ITYOUT,7)(SUB1NM(J,M),M=1,4)
      LINE=LINE+4
      BIG=0.0
      DO 2010 I=1,NSYS
      IF(SYSNRM(J,I).GT.BIG)BIG=SYSNRM(J,I)
2010 CONTINUE
C
      DO 2090 KU=1,10
      IOPT(KU)=IBLANK
2090 CONTINUE
C
      ICOUNT=0
C
      DO 2020 I=1,NSYS
      IF(BIG.NE.SYSNRM(J,I))GO TO 2020
      ICOUNT=ICOUNT + 1
      IOPT(ICOUNT)=ICHAR(I)
2020 CONTINUE
C
      IF(BIG.NE.0)GO TO 2030
      WRITE(ITYOUT,6)
      GO TO 2011
C
2030 CONTINUE
C
      CALL WRTE(IOPT,J)
      LINE=LINE+6
C
2011 CONTINUE
      GO TO 9190
C
C *** CASE 3 ***
C

```



```

3000 CONTINUE
INDXXX=0
BIG=0.0
NPAGE=1
DO 3011 K=1,NSUB1
    LINE=LINE+4
    WRITE (ITYOUT,7) (SUB1NM(K,M),M=1,4)
DO 3012 KK=1,NSUB2
    CALL PAGER (LINE,NPAGE,9)
    IF (LINE.EQ.0) WRITE (ITYOUT,11) (DBNAME(M),M=1,4),SYSDSN,
E    (SCNDSC(ICH01C,M),M=1,15),NPAGE
    IF (LINE.EQ.0) LINE=4
3001    WRITE(ITYOUT,8)(SUB2NM(K,KK,M),M=1,4)
    LINE=LINE+3
    INDXXX=INDXXX+1
    J=INDXXX
    BIG=0.0
    DO 3010 I=1,NSYS
        IF(SYSNRM(J,I).GT.BIG)BIG=SYSNRM(J,I)
3010 CONTINUE
C
DO 3090 KU=1,10
    IOPT(KU)=IBLANK
3090 CONTINUE
C
ICOUNT=0
C
DO 3020 I=1,NSYS
    IF(BIG.NE.SYSNRM(J,I))GO TO 3020
    ICOUNT=ICOUNT+1
    IOPT(ICOUNT)=ICHAP(I)
3020 CONTINUE
C
IF(BIG.NE.0)GO TO 3030
LINE=LINE+2
WRITE(ITYOUT,6)
GO TO 3012
C
3030 CONTINUE
C
CALL WRTE(IOPT,J)
LINE= LINE+6
3012 CONTINUE
3011 CONTINUE
C
C *** FORMAT STATEMENTS ***
C
1    FORMAT(1X,/)
3    FORMAT(I1)
4    FORMAT(1X,T25,*UTILITY VALUES FOR *,4A4/)
5    FORMAT(1X,T32,4A4/)
15   FORMAT(1X,T32,A1,* -- *,4A4)
6    FORMAT(1X,*ALL SYSTEMS HAVE BEEN ELIMINATED *)
7    FORMAT(1X,/,1X,4H*** ,4A4,4H ***/,1X,16(1H-))

```

```

8      FORMAT(6X,4A4,/6X,16(1H-)/)
9      FORMAT(1X,/,1X,70(1H*)/,1X,70(1H*)/)
10     FORMAT(1X,/,1X,
E 5JH ***-ERROR-*** BAD INPUT--STOP EXECUTION & PESTART,/)
11     FORMAT(1H,72(1H*)/1X,5X,4A4,5X,5HCASE,11,3X,
E 20UTILITY VALUES TABLE,/1X,15A4,5X,5HPAGE,12,/,1X,
E 72(1H*))
9190  CONTINUE
      END

C
C *****
C *****
C
      SUBROUTINE WRTE(IOPT,J)

C
C *****
C
      CONTAINS THE WRITE STATEMENTS FOR THE UTILITY VALUE TABLE
C
C *****
C
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,
E  SYSSET(4),SB1SET(4),SB2SET(4),
E  SYSNAM(10,4),SUB1NM(20,4),SUB2NM(10,10,4),PARNAM(40,4),
E  NUM(40),RATED(40),YN(40),INVERT(40),INDEX
      COMMON/PARAM4/SYSNRM(85,10)
      COMMON/UNITNO/ITYIN,ITYOUT,IPPH,IUPPH,IPP,IUPP,IVPARM,ISYS
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,PARNAM,
E  RATED,YN
      DIMENSION ICHAR(10),IOPT(10)
      DATA ICHAR/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ/

C
      GO TO (1101,1102,1103,1104,1105,1106,1107,1108,1109,1110),NSYS

C
1101  RETURN
1102  WRITE(ITYOUT,1)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
      WRITE(ITYOUT,2)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
      RETURN
1103  WRITE(ITYOUT,3)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
      WRITE(ITYOUT,4)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
      RETURN
1104  WRITE(ITYOUT,5)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
      WRITE(ITYOUT,6)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
      RETURN
1105  WRITE(ITYOUT,7)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
      WRITE(ITYOUT,8)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
      RETURN
1106  WRITE(ITYOUT,9)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
      WRITE(ITYOUT,10)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
      RETURN
1107  WRITE(ITYOUT,11)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
      WRITE(ITYOUT,12)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
      RETURN
1108  WRITE(ITYOUT,13)(ICAR(I),I=1,NSYS),(SYSSET(M),M=1,4)

```

```

WRITE(ITYOUT,14)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
RETURN
1109 WRITE(ITYOUT,15)(ICHAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
WRITE(ITYOUT,16)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
RETURN
1110 WRITE(ITYOUT,17)(ICHAR(I),I=1,NSYS),(SYSSET(M),M=1,4)
WRITE(ITYOUT,18)(SYSNRM(J,K),K=1,NSYS),(IOPT(M),M=1,10)
RETURN
C
C ***
C   FORMAT STATEMENTS
C ***
C
1   FORMAT(1X,18X,7HOPTIMUM,/1X,3X,A1,7X,A1,7X,4A4,/)
2   FORMAT(1X,2(F5.3,2X),5X,10A1,/)
3   FORMAT(1X,26X,7HOPTIMUM,/1X,3X,3(A1,6X),4A4,/)
4   FORMAT(1X,3(F5.3,2X),5X,10A1,/)
5   FORMAT(1X,31X,7HOPTIMUM,/1X,3X,4(A1,6X),4A4,/)
6   FORMAT(1X,4(F5.3,2X),5X,10A1,/)
7   FORMAT(1X,38X,7HOPTIMUM,/,1X,3X,5(A1,6X),4A4,/)
8   FORMAT(1X,5(F5.3,2X),5X,10A1,/)
9   FORMAT(1X,46X,7HOPTIMUM,/,1X,3X,6(A1,6X),4A4,/)
10  FORMAT(1X,6(F5.3,2X),5X,10A1,/)
11  FORMAT(1X,54X,7HOPTIMUM,/,1X,3X,7(A1,6X),4A4,/)
12  FORMAT(1X,7(F5.3,2X),5X,10A1,/)
13  FORMAT(1X,62X,7HOPTIMUM,/,1X,3X,8(A1,6X),4A4,/)
14  FORMAT(1X,8(F5.3,2X),5X,10A1,/)
15  FORMAT(1X,70X,7HOPTIMUM,/,1X,3X,9(A1,6X),4A4,/)
16  FORMAT(1X,9(F5.3,2X),5X,10A1,/)
17  FORMAT(1X,72X,7HOPTIMUM,/,1X,3X,10(A1,6X),4A4,/)
18  FORMAT(1X,10(F5.3,2X),5X,10A1,/)
END

```

```

OVERLAY(GRF,D,D)
PROGRAM GRAPHX(INPUT,OUTPUT,HEADER,SYSNUM,TAPE10=HEADER,
& TAPE15=SYSNUM,TAPE2=OUTPUT,TAPE5=INPUT)
C*****
C***
C* THIS PROGRAM DISPLAYS ON GRAPHS, USING THE TEKTRONIX *
C* PLOT-10/TERMINAL CONTROL SYSTEM, THE OUTPUT OF THE *
C* SYSTEM SELECTION MODEL PROGRAM. TWO TYPES OF GRAPHS *
C* CAN BE PRINTED, TIME GRAPHS(WHICH PLOT TIME VERSUS *
C* SYSTEMS) OR SYSTEMS VERSUS THE OPTIMAL VALUES. GRAPHS *
C* CAN VARY FROM GRAPHING TWO TO TEN SYSTEMS. *
C***
C*****
C*****
C***
C* DESCRIPTION OF VARIABLE NAMES *
C* *
C* SYSDSN - INDICATES DESCRIPTION OF THE PROBLEM *
C* 1 - JUST SYSTEMS INCLUDED *
C* 2 - SYSTEMS AND LEVEL ONE SUBSYSTEMS *
C* 3 - SYSTEMS AND BOTH SUBSYSTEMS *
C* NSYS - NUMBER OF SYSTEMS *
C* NSUB1 - NUMBER OF LEVEL ONE SUBSYSTEMS *
C* NSUB2 - LARGEST NO. OF LEVEL 2 SUBSYSTEMS *
C* NPAPH - NUMBER OF PARAMETERS *
C* SYSSET - GENERAL SETNAME FOR THE SYSTEMS *
C* SB1SET - GENERAL SETNAME FOR THE LEVEL ONE *
C* SB2SET - GENERAL SETNAME FOR THE LEVEL TWO *
C* SYSNAM - SYSTEM NAMES *
C* SUB1NM - LEVEL ONE SUBSYSTEM NAMES *
C* SUB2NM - LEVEL TWO SUBSYSTEM NAMES *
C* PARNAM - PARAMETER NAMES *
C***
C*****
C***
C
C*** COMMON STATEMENTS ***
C
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPAPH,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
COMMON/PARAM4/SYSNRM(85,10)
COMMON/GRAF/VAL(7),SYST(8),IYEARS(6),ISYST(15),ISIZE(15),
& YARRAY(11),XAR(4),RSYS,IKEY(5),IONE(4),ITWO(4),ITHREE(4),
& IFOUR(4),IFIVE(4),ISIX(4),ISEVEN(4),IEIGHT(4),ININE(4),
& ITEN(5),SYSNA(2),SB1NME(2),SB2NME(2),ICOLON(4),IBAUD,JTERM
C
C*** INTEGER STATEMENTS ***
C
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,VAL,SYST,SYSNA
INTEGER SB1NME,SB2NME,NSYS2

```

```

DATA INO/2HNO/,IYES/3HYES/
  REWIND ISYS
  REWIND IHDR
C
C*** DOES USER WANT A GRAPH? ***
C
  WRITE (ITYOUT,18)
18  FORMAT (1H ,43HDO YOU WISH TO SEE GRAPHIC OUTPUT? (YES/NO),/)
  READ (ITTYIN,19) IANSR
19  FORMAT (A3)
  IF (IANSR.NE.IYES) GO TO 999
C
C*** GET TEKTRONIX INITT (IBAUD,JTERM) PARAMETERS
  WRITE (ITYOUT,1)
1  FORMAT (1H ,45HENTER CHAR/SEC (I.E., 300 BAUD = 30 CHAR/SEC),/)
  READ (ITTYIN,*) IBAUD
  WRITE (ITYOUT,2)
2  FORMAT (1H ,20HENTER TERMINAL TYPE:,,/ ,1H ,14H1=4010,4012/13,
& / ,1H ,9H2=4014/15,/ ,1H ,30H3=4014/15 WITH GRAPHICS MODULE,/)
  READ (ITTYIN,*) JTERM

  CALL OVERLAY(3HGRF,1,0)

  DO 100 I=1,INDEX
  READ(ISYS,*)(SYSNRM(I,J),J=1,NSYS)
100  CONTINUE

  RSYS=NSYS
C
C*** SETS VALUES FOR X-COORDINATES OF GPAPHS ***
C
  XAR(1)=-1.
  XAR(2)=RSYS
  XAR(3)=1.
  XAR(4)=1.

  IF(SYSDSN.EQ.1) CALL OVERLAY(3HGRF,2,0)
  IF(SYSDSN.EQ.2) CALL OVERLAY(3HGRF,3,0)
  IF(SYSDSN.EQ.3) CALL OVERLAY(3HGRF,4,0)

999  STOP
     END

BLOCK DATA
COMMON/GRAF/VAL(7),SYST(8),IYEARS(6),ISYST(15),ISIZE(15),
& YAPRAY(11),XAR(4),RSYS,IKEY(5),IONE(4),ITWC(4),ITREE(4),
& IFOUR(4),IFIVE(4),ISIX(4),ISEVEN(4),IEIGHT(4),ININE(4),
& ITEN(5),SYSNA(2),SB1NME(2),SB2NME(2),ICOLON(4),IBAUD,JTERM
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
C
C*** DATA FOR GRAPHS WRITTEN IN ASCII ***
C
  DATA IYEARS/5,89,69,65,82,83/
  DATA ISYST/14,83,89,83,84,69,77,32,32,86,65,76,85,69,83/

```

```

DATA ISIZE/14,83,73,90,69,32,80,65,82,65,77,69,84,69,82/
DATA VAL/6,86,65,76,85,69,83/
DATA SYST/7,83,89,83,84,69,77,83/
DATA IKEY/4,75,69,89,58/
DATA ICOLON/3,32,32,58/
DATA IONE/3,49,32,61/
DATA ITWO/3,50,32,61/
DATA ITHREE/3,51,32,61/
DATA IFOUR/3,52,32,61/
DATA IFIVE/3,53,32,61/
DATA ISIX/3,54,32,61/
DATA ISEVEN/3,55,32,61/
DATA IEIGHT/3,56,32,61/
DATA ININE/3,57,32,61/
DATA ITEN/4,49,46,32,61/
DATA ITTYIN,ITYOUT,IHDR,ISYS/2,5,10,15/
END

```

```

OUEPLAY(GRF,1,0)

```

```

C*****
C***
C* THIS OVERLAY READS IN THE NECESSARY DATA, OUTPUT FROM *
C* THE SYSTEM SELECTION MODEL PROGRAM. THE DATA FILES *
C* ARE "HEADER" AND "SYSNUM"; THEY MUST BE ATTACHED BE- *
C* FORE THE PPROGRAM WILL EXECUTE. *
C***
C*****
C***
C PROGRAM READR
C
C*** COMMON STATEMENTS ***
C
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
COMMON/OB/DBNAME(2)
COMMON/PARAM4/SYSNRM(85,10)
C
C *** INTEGER STATEMENTS ***
C
INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,DBNAME
C
C *** READ IN DATA BASE NAME ***
C
READ(IHDR,3)(DBNAME(M),M=1,2)
C
C *** READ IN SYSTEM DESIGN ***
C
READ(IHDR,1)SYSDSN
C
C *** READ IN NO. OF SYSTEMS,SUBSYSTEMS,PARAMETERS ***

```

```

C
  READ(IHDR,2)NSYS
  READ(IHDR,2)NSUB1
  READ(IHDR,2)NSUB2
  READ(IHDR,2)NPARM
  IF(NSYS.EQ.0) GO TO 600

C
C *** READ IN SETNAMES ***
C
  READ(IHDR,3)(SYSSET(I),I=1,2)
  IF(SYSDSN.GE.2)READ(IHDR,3)(SB1SET(I),I=1,2)
  IF(SYSDSN.EQ.3)READ(IHDR,3)(SB2SET(I),I=1,2)

C
C *** READ IN SYSTEM NAMES ***
C
  DO 100 I=1,NSYS
    READ(IHDR,3)(SYSNAM(I,J),J=1,2)
100  CONTINUE
    IF(NSUB1.EQ.0)GO TO 400

C
C *** READ IN SUBSYSTEM LEVEL ONE NAMES ***
C
  DO 200 I=1,NSUB1
    READ(IHDR,3)(SUB1NM(I,J),J=1,2)
200  CONTINUE
    IF(NSUB2.EQ.0) GO TO 400

C
C *** READ SUBSYSTEM LEVEL TWO NAMES ***
C
  DO 300 K=1,NSUB1
    DO 300 I=1,NSUB2
      READ(IHDR,3)(SUB2NM(K,I,J),J=1,2)
300  CONTINUE
C
400  CONTINUE
C
C *** READ IN PARAMETER NAMES AND PARAMETER CHARACTERISTICS ***
C
  DO 500 I=1,NPARM
    READ(IHDR,4)(PARNAM(I,J),J=1,2),NUM(I),RATED(I),YN(I),
& INVERT(I)
500  CONTINUE
    GO TO 700
600  WRITE(ITYOUT,5)
    CALL EXIT

C
C *** CALCULATE INDEX FOR POSITIONING FILES ***
C
700  CONTINUE

    IF(NSUB1.EQ.0) GO TO 800
    IF(NSUB2.EQ.0) GO TO 900
    INDEX=NSUB1*NSUB2

```

```

      GO TO 1000
800   INDEX=1
      GO TO 1000
900   INDEX=NSUB1
1000  CONTINUE
C
C ***FORMAT STATEMENTS ***
C
1     FORMAT(I1)
2     FORMAT(I2)
3     FORMAT(A10,A6)
4     FORMAT(A10,A6,I1,I1,I1,I1)
5     FORMAT(1X,37HERROR IN HEADER -- NO. OF SYSTEMS= 0 )
      END

```

```

      OVEPLAY(GRF,2,0)
C*****
C***
C* THIS OVERLAY PRINTS OUT ONE GRAPH COMPARING ALL THE *
C* SYSTEMS INVOLVED. *
C***
C*****

```

```

      PROGRAM GRAF1
C
C*** COMMON STATEMENTS ***
C
      COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
      COMMON/PARAM4/SYSNRM(85,10)
      COMMON/GRAF/VAL(7),SYST(8),IYEARS(6),ISYST(15),ISIZE(15),
& YARRAY(11),XAR(4),RSYS,IKEY(5),IONE(4),ITWO(4),
& ITHREE(4),IFOUR(4),IFIVE(4),ISIX(4),ISEVEN(4),
& IEIGHT(4),ININE(4),ITEN(5),SYSNA(2),SB1NME(2),SB2NME(2)
& ,ICOLON(4),IBAUD,JTERM

```

```

C
C*** INTEGER STATEMENTS ***
C
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
& PARNAM,RATED,YN,VAL,SYST,SYSNA,NSUM,SB1NME,SB2NME
& ,NSYS
      NSUM=0
      YARRAY(1)=RSYS
      CALL INITT (IBAUD,JTERM)
      CALL TERM (JTERM,4096)
      DO 1100 I=1,NSYS
        II=I+1

```

```

C
C*** SET Y-VALUES FROM SYSTEM VALUES ***
C

```



```

        IR=8
        CALL NOTATE(2200,200,IEIGHT(1),IEIGHT(2))
        IF(IR.GE.NSYS) GO TO 280
        IR=9
        CALL NOTATE(2200,100,ININE(1),ININE(2))
        IF(IR.GE.NSYS) GO TO 280
        IR=10
        CALL NOTATE(2200,C,ITEN(1),ITEN(2))
280    CONTINUE

        DO 212 MSYS=1,NSYS
        IF (MSYS.LE.5) ISYSPL=MSYS
        IF (MSYS.GT.5) ISYSPL=MSYS-5
        CALL MOVABS (1230,400-100*(ISYSPL-1))
        IF (MSYS.GT.5) CALL MOVREL (1200,0)
        DO 313 I=1,2
313    SYSNA(I)=SYSNAM(MSYS,I)
        CALL ACUTST (20,SYSNA)
212    CONTINUE
C
C*** PRINTS SYSTEM SETNAMES ***
C
1101   CALL MOVABS(1000,500)
        CALL ACUTST(20,SYSSET)
C
C*** DELAY COMMAND NEEDS ANY ONE INPUT TO CONTINUE ***
C
        CALL TINPUT(K)

        CALL FINITT(0.,700.)
        END

        OVERLAY(GRF,3,0)
C*****
C***
C* THIS OVERLAY PRINTS OUT GRAPHS OF RATING VALUES VERSUS *
C* SYSTEMS. EACH OF THE POSSIBLE GRAPHS INCLUDES THE NAMES *
C* OF THE FIRST SUBSET AND THE SYSTEM SETNAMES. *
C***
C*****
C***
        PROGRAM GRAF2

C
C*** COMMON STATEMENTS ***
C
        COMMON/UNITNO/ITTYIN,ITYOUT,IHOR,ISYS
        COMMON/HOR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARN,SYSSET(2),
        & SB1SET(2),SB2SET(2),SYSNAM(10,2),SUBINM(20,2),
        & SUP2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
        & YN(40),INVERT(40),INDEX
        COMMON/PARAM4/SYSNRM(85,10)
        COMMON/GRAF/VAL(7),SYST(8),IYEARS(6),ISYST(15),ISIZE(15),

```

```

      & YARRAY(11),XAR(4),RSYS,IKEY(5),IONE(4),ITWC(4),
      & ITHREE(4),IFOUR(4),IFIVE(4),ISIX(4),ISEVEN(4),
      & IEIGHT(4),ININE(4),ITEN(5),SYSNA(2),SB1NME(2),SB2NME(2)
      & ,ICOLON(4),IBAUD,JTERM
C
C*** INTEGER STATEMENTS ***
C
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
      & PARNAM,RATED,YN,VAL,SYST,SYSNA,NSUM,SB1NME,SB2NME
      NSUM=0
      DO 100 NOW=1,NSUB1
        YARRAY(1)=RSYS
        CALL INITT (IBAUD,JTERM)
        CALL TERM (JTERM,4096)
        DO 2100 I=1,NSYS
          II=I+1
C
C*** SET Y-VALUES FROM SYSTEM VALUES ***
C
          YARRAY(II)=SYSNRM(NOW,I)
2100  CONTINUE
C
C*** INITIALIZES TERMINAL AND TERMINAL STATUS AREA ***
C
      CALL BINITT
C
C*** LIMITS OF COORDINATES OF GRAPHS ***
C
      CALL DLMX(0.,RSYS+1.)
      CALL DLMY(0.,1.)
      CALL SLIMY(900,2775)
C
C*** PRINTING OF DATA POINTS ON GRAPHS ***
C
      IDEN=NSYS/3+2
      CALL XDEN(IDEN)
      CALL XFRM(1)
      IBARWD=6000/(3*(RSYS+2.))
      CALL VBARST (8,IBARWD,100)
      CALL CHECK(XAR,YARRAY)
      CALL DSPLAY(XAR,YARRAY)
      CALL FRAME
C
C*** LABELING OF GRAPHS ***
C
      CALL MOVARS(100,2500)
      CALL VLABEL(ISYST(1),ISYST(2))

      CALL NOTATE (1910,600,SYST(1),SYST(2))
      CALL NOTATE (600,500,IKEY(1),IKEY(2))
      IR=1
      CALL NOTATE(1000,400,IONE(1),IONE(2))

```

```

        IF (IR.GE.NSYS)GO TO 290
        IR=2
        CALL NOTATE(1000,300,ITWO(1),ITWO(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=3
        CALL NOTATE(1000,200,ITHREE(1),ITHREE(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=4
        CALL NOTATE(1000,100,IFOUR(1),IFOUR(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=5
        CALL NOTATE(1000,0,IFIVE(1),IFIVE(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=6
        CALL NOTATE(2200,400,ISIX(1),ISIX(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=7
        CALL NOTATE(2200,300,ISEVEN(1),ISEVEN(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=8
        CALL NOTATE(2200,200,IEIGHT(1),IEIGHT(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=9
        CALL NOTATE(2200,100,ININE(1),ININE(2))
        IF (IR.GE.NSYS)GO TO 290
        IR=10
        CALL NOTATE(2200,0,ITEN(1),ITEN(2))
290    CONTINUE

        DO 212 MSYS=1,NSYS
        IF (MSYS.LE.5) ISYSPL=MSYS
        IF (MSYS.GT.5) ISYSPL=MSYS-5
        CALL MOVABS (1230,400-100*(ISYSPL-1))
        IF (MSYS.GT.5) CALL MOVREL (1200,0)
        DO 313 I=1,2
313    SYSNA(I)=SYSNA(MSYS,I)
        CALL AOUTST (20,SYSNA)
212    CONTINUE
C
C*** LABELING SYSTEM SETNAMES ***
C
1131  CALL MOVABS(1000,500)
        CALL AOUTST(20,SYSSET)
C
C*** LABELING LEVEL ONE SUBSYSTEM SETNAMES ***
C
        CALL MOVABS(1200,3150)
        CALL AOUTST(20,S61SET)
        CALL NOTATE(2100,3150,ICOLON(1),ICOLON(2))

        DO 117 IRE=1,2
C
C*** RENAMING OF SUBSYSTEM ONE LEVEL NAMES ***
C

```

```

        SBINME(IRE)=SUB1NM(NOW,IRE)

117    CONTINUE
C
C*** LABELING FIRST LEVEL SUBSYSTEMS ***
C
        CALL MOVABS(2300,3150)
        CALL AOUTST(20,SBINME)
C
C*** DELAY COMMAND; NEEDS ONE INPUT TO CONTINUE ***
C
        CALL TINPUT(K)

100    CONTINUE
        CALL FINITT(0.,700.)
        END

        OVERLAY(GRF,4,0)
C*****
C***
C* THIS OVERLAY PPINTS 4 GRAPHS OF RATING VALUES VER
C* SUS SYSTEMS. EACH OF THE POSSIBLE GRAPHS INCLUDES
C* THE NAMES OF THE FIRST AND SECOND SUBSETS AND THE
C* SYSTEM SETNAMES AND NAMES.
C***
C*****
C***
        PPROGRAM GRAF3
C
C*** COMMON STATEMENTS ***
C
        COMMON/UNITNO/ITTYIN,ITYOUT,THDR,ISYS
        COMMON/HOP1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
        COMMON/PARAM4/SYSNRM(85,10)
        COMMON/GRAF/VAL(7),SYST(8),IYEARS(6),ISYST(15),ISIZE(15),
& YARRAY(11),XAR(4),RSYS,IKEY(5),IONE(4),ITWO(4),
& ITHREE(4),IFOUR(4),IFIVE(4),ISIX(4),ISEVEN(4),
& IEIGHT(4),ININE(4),ITEN(5),SYSNA(2),SB1NME(2),SB2NME(2)
& ,ICOLON(4),IBAUD,JTERM
C
C*** INTEGER STATEMENTS ***
C
        INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,SUB2NM,
& PARNAM,RATED,YN,VAL,SYST,SB1NME,NSUM,SB2NME,SYSNA
        NSUM=0
        INDX=0
        CALL INITT (IBAUD,JTERM)
        CALL TEPH(JTERM,4096)
        DO 3200 L=1,NSUB1
        DO 3200 LL=1,NSUB2

```

```

      CALL NEWPAG
      INDX=INDX+1
      YAPRAY(1)=RSYS
      DO 3100 I=1,NSYS
        II=I+1
C
C*** SET Y-VALUES FROM SYSTEM VALUES ***
C
      YARRAY(II)=SYSNRH(INDX,I)

3100  CONTINUE
C
C*** INITIALIZES TERMINAL AND TERMINAL STATUS AREA ***
C
      CALL BINITT
C
C*** LIMITS OF COORDINATES OF GRAPHS ***
C
      CALL DLIMX(C.,RSYS+1.)
      IDEN=NSYS/3+2
      CALL XDEN(IDEN)
      CALL XFRM(1)
      IBARWD=6000/(3*(RSYS+2.))
      CALL VBARST (8,IBARWD,100)
      CALL DLIMY(C.,1.)
      CALL SLIMY(900,2775)

C
C*** PRINTING OF DATA POINTS ON GRAPHS ***
C
      CALL CHECK(XAR,YARRAY)
      CALL DSPLAY(XAR,YARRAY)
      CALL FRAME
C
C*** LABELING OF GRAPHS ***
C
      CALL NOTATE(1910,600,SYST(1),SYST(2))
      CALL MOVABS(100,2500)
      CALL VLABEL(ISYST(1),ISYST(2))

      CALL NOTATE(600,500,IKEY(1),IKEY(2))
      IR=1
      CALL NOTATE(1000,400,IONE(1),IONE(2))
      IF(IR.GE.NSYS)GO TO 260
      IR=2
      CALL NOTATE(1000,300,ITWO(1),ITWO(2))
      IF(IR.GE.NSYS)GO TO 260
      IR=3
      CALL NOTATE(1000,200,ITHREE(1),ITHREE(2))
      IF(IR.GE.NSYS)GO TO 260
      IR=4
      CALL NOTATE(1000,100,IFOUR(1),IFOUR(2))
      IF(IR.GE.NSYS)GO TO 260
      IR=5

```

```

CALL NOTATE(1000,0,IFIVE(1),IFIVE(2))
  IF(IR.GE.NSYS)GO TO 260
  IR=6
CALL NOTATE(2200,400,ISIX(1),ISIX(2))
  IF(IR.GE.NSYS)GO TO 260
  IR=7
CALL NOTATE(2200,300,ISEVEN(1),ISEVEN(2))
  IF(IR.GE.NSYS)GO TO 260
  IR=8
CALL NOTATE(2200,200,IEIGHT(1),IEIGHT(2))
  IF(IR.GE.NSYS)GO TO 260
  IR=9
CALL NOTATE(2200,100,ININE(1),ININE(2))
  IF(IR.GE.NSYS)GO TO 260
  IR=10
CALL NOTATE(2200,0,ITEN(1),ITEN(2))
260 CONTINUE

DO 212 MSYS=1,NSYS
  IF (MSYS.LE.5) ISYSPL=MSYS
  IF (MSYS.GT.5) ISYSPL=MSYS-5
  CALL MOVABS (1230,400-100*(ISYSPL-1))
  IF (MSYS.GT.5) CALL MOVREL (1200,0)
  DO 313 I=1,2
313  SYSNA(I)=SYSNAM(MSYS,I)
  CALL AOUTST (20,SYSNA)
212 CONTINUE
1161 CALL MOVABS(1000,500)
  CALL AOUTST(20,SYSSET)
  CALL MOVABS(1200,3160)
  CALL AOUTST(20,SB1SET)
  CALL NOTATE (2100,3160,ICOLON(1),ICOLON(2))
  DO 118 ITE=1,2
  SB1NME(ITE)=SUB1NM(L,ITE)
118 CONTINUE
  CALL MOVABS(2300,3160)
  CALL AOUTST(20,SB1NME)
  CALL MOVABS(1200,3060)
  CALL AOUTST(20,SB2SET)
  CALL NOTATE (2100,3060,ICOLON(1),ICOLON(2))
  DO 450 IER=1,2
  SB2NME(IER)=SUB2NM(L,LL,IER)
450 CONTINUE
  CALL MOVABS(2300,3060)
  CALL AOUTST(20,SB2NME)
  CALL TINPUT(K)
3200 CONTINUE
  CALL FINITT(0.,700.)
END

```

```

      OVERLAY (GRF,0,0)
      PROGRAM GRAPH1(INPUT,OUTPUT,HEADER,SYSNUM,TAPE10=HEADER,
& TAPE15=SYSNUM,TAPE5=INPUT,TAPE2=OUTPUT)
C*****
C***
C* THIS PROGRAM USES DATA FILES HEADER AND SYSNUM TO CREATE A
C* VERTICAL BAR GRAPH OF SYSTEM NUMBERS VERSUS SYSTEM RATING
C* VALUES, OBTAINED FROM THE MAIN PROGRAM SSMP.  NORMAL LIMITS
C* ON PROBLEM SIZE ARE THE SAME AS ALWAYS, AND THE NUMBER OF
C* GRAPHS PRODUCED IS EQUAL TO INDEX.
C***
C*****
C
      COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
      COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
      COMMON/PARAM4/SYSNRM(85,10)
      COMMON /GRAF/ ISYST(15),XARRAY(4),RSYS,IBAUD,JTERM
C
C*** INTEGER STATEMENTS ***
C
      INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,VAL,SYST,SYSNA
      INTEGER SB1NME,SB2NME,NSYS2
      DATA IYES/3HYES/
C
C*** DOES USER WANT TO SEE TIME GRAPH? ***
C
      WRITE (ITYOUT,18)
18      FORMAT (1X,32HDO YOU WISH TO SEE A TIME GRAPH?,/)
      READ (ITTYIN,19) IANSR
19      FORMAT (A3)
      IF (IANSR.NE.IYES) GO TO 999
C
      REWIND IHDR
      REWIND ISYS
C
C*** READ IN DATA FROM HEADER FILE ***
C
      CALL OVERLAY (3HGRF,1,0)
C
C*** GET SYSNUM VALUES INTO SYSNRM ***
C
      DO 100 I=1,INDEX
          READ (ISYS,*) (SYSNRM(I,J),J=1,NSYS)
100      CONTINUE
C
      RSYS=NSYS
      WRITE (ITYOUT,2)
2      FORMAT (1H ,45HENTER CHAR/SEC (I.E., 300 BAUD = 30 CHAR/SEC),/)
      READ (ITTYIN,*) IBAUD
      WRITE (ITYOUT,3)

```

```

3   FORMAT (1H ,2QHENTER TERMINAL TYPE:,,1H ,14H1=4C10,4012/13,
& /,1H ,9H2=4014/15,,1H ,30H3=4014/15 WITH GRAPHICS MODULE,/)
   READ (ITTYIN,*) JTERM
C
C*** IBAUD AND JTERM ARE USED TO CALL INITT IN OVERLAYS ***
C
   IF (SYSDSN.EQ.1) GO TO 1000
   IF (SYSDSN.EQ.2.AND.NSUB1.NE.5) GO TO 1000
   IF (SYSDSN.EQ.3.AND.NSUB2.NE.5) GO TO 1000
C
C*** ERROR CONDITIONS--PROBLEM IS RESTRICTED TO 5 TIMES ***
C
C*** NOW CALL APPROPRIATE OVERLAY ***
C
   IF (SYSDSN.EQ.2) CALL OVERLAY (3HGRF,2,0)
   IF (SYSDSN.EQ.3) CALL OVERLAY (3HGRF,3,0)
C
   GO TO 999
1000 WRITE (ITYOUT,1)
1   FORMAT (1H ,36HPROBLEM UNSUITABLE FOR TIME GRAPHS -,/,1H ,
& 43HMUST HAVE EXACTLY 5 LEVEL 1 OR 2 SUBSYSTEMS)
999  STOP
     END
C
C*****
C***
                                BLOCK DATA
C***
C*****
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
COMMON/PARAM4/SYSNRM(85,10)
COMMON /GRAF/ ISYST(15),XARRAY(4),RSYS,IBAUD,JTERM
C
C*** INTEGER STATEMENTS ***
C
   INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,VAL,SYST,SYSNA
   INTEGER SB1NME,SB2NME,NSYS2
C
   DATA ISYST/14,83,89,83,84,69,77,32,32,86,65,76,25,69,23/
   DATA ITTYIN,ITYOUT,IHDR,ISYS/2,5,10,15/
   DATA XARRAY/-1.,5.,1.,1./
   END
C
   OVERLAY(GRF,1,0)
C*****
C***
C* THIS OVERLAY READS IN THE NECESSARY DATA, OUTPUT FROM *
C* THE SYSTEM SELECTION MODEL PROGRAM. THE DATA FILES *
C* ARE "HEADER" AND "SYSNUM"; THEY MUST BE ATTACHED BE- *

```



```

C* FOR THE PROGRAM WILL EXECUTE.
C***
C*****
C***
    PROGRAM READR
C
C*** COMMON STATEMENTS ***
C
    COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
    COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
    COMMON/DB/DBNAME(2)
    COMMON/PARAM4/SYSNRM(85,10)
C
C *** INTEGER STATEMENTS ***
C
    INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,DBNAME
C
C *** READ IN DATA BASE NAME ***
C
    READ(IHDR,3)(DBNAME(M),M=1,2)
C
C *** READ IN SYSTEM DESIGN ***
C
    READ(IHDR,1)SYSDSN
C
C *** READ IN NO. OF SYSTEMS,SUBSYSTEMS,PARAMETERS ***
C
    READ(IHDR,2)NSYS
    READ(IHDR,2)NSUB1
    READ(IHDR,2)NSUB2
    READ(IHDR,2)NPARM
    IF(NSYS.EQ.0) GO TO 600
C
C *** READ IN SETNAMES ***
C
    READ(IHDR,3)(SYSSFT(I),I=1,2)
    IF(SYSDSN.GE.2)READ(IHDR,3)(SB1SET(I),I=1,2)
    IF(SYSDSN.EQ.3)READ(IHDR,3)(SB2SET(I),I=1,2)
C
C*** READ IN SYSTEM NAMES ***
C
    DO 100 I=1,NSYS
        READ(IHDR,3)(SYSNAM(I,J),J=1,2)
100    CONTINUE
        IF(NSUB1.EQ.0)GO TO 400
C
C *** READ IN SUBSYSTEM LEVEL ONE NAMES ***
C
    DO 200 I=1,NSUB1
        READ(IHDR,3)(SUB1NM(I,J),J=1,2)

```

```

200  CONTINUE
    IF(NSUB2.EQ.0) GO TO 400
C
C *** READ SUBSYSTEM LEVEL TWO NAMES ***
C
    DO 300 K=1,NSUB1
    DO 300 I=1,NSUB2
        READ(IHDR,3)(SUB2NM(K,I,J),J=1,2)
300  CONTINUE
C
400  CONTINUE
C
C *** READ IN PARAMETER NAMES AND PARAMETER CHARACTERISTICS ***
C
    DO 500 I=1,NPARN
        READ(IHDR,4)(PARNAM(I,J),J=1,2),NUM(I),RATED(I),YN(I),
        & INVERT(I)
500  CONTINUE
    GO TO 700
600  WRITE(ITYOUT,5)
    CALL EXIT

C
C *** CALCULATE INDEX FOR POSITIONING FILES ***
C
700  CONTINUE

    IF(NSUB1.EQ.0) GO TO 800
    IF(NSUB2.EQ.0) GO TO 900
    INDEX=NSUB1*NSUB2
    GO TO 1000
800  INDEX=1
    GO TO 1000
900  INDEX=NSUB1
1000 CONTINUE
C
C ***FORMAT STATEMENTS ***
C
1    FORMAT(I1)
2    FORMAT(I2)
3    FORMAT(A10,A6)
4    FORMAT(A10,A6,I1,I1,I1,I1)
5    FORMAT(1X,37HERROR IN HEADER -- NO. OF SYSTEMS= 0 )
    END
C
    OVERLAY (GRF,2,0)
    PROGRAM GRAF2
C*****
C***
C* THIS OVERLAY GRAPHS THE CASE 2 DATA, 1 CURVE FOR EACH SYSTEM,*
C* ON 1 GRAPH AS TIME (THE LEVEL 1 SUBSYSTEM) VS. SYSTEM VALUES.*
C***
C*****
C

```

```

COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
COMMON/HDP1/SYSDSN,NSYS,NSUB1,NSUB2,NPARN,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
COMMON/PARAM4/SYSNRM(85,10)
COMMON /GRAF/ ISYST(15),XARRAY(4),RSYS,IBAUD,JTERM
C
C*** INTEGEF STATEMENTS ***
C
    INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,VAL,SYST,SYSNA
    INTEGER SB1NME,SB2NME,NSYS2
    DIMENSION YARRAY(6),ILABEL(2),SYSNA(2)
C
C*** INITIALIZE TEKTRONIX SYSTEM ***
C
    CALL INITT (IBAUD,JTERM)
    CALL TERM(JTERM,4096)
    CALL BINITT
C
C*** SET LIMITS FOR GRAPHS ***
C
    CALL DLIMX (1.,5.)
    CALL DLIMY (0.,1.)
    CALL SLIMY (900,2800)
    YARRAY(1)=RSYS
C
C*** SET UP Y-VALUES ***
C
    DO 10 ISYSN=1,NSYS
    DO 11 J=1,5
11    YARRAY(J+1)=SYSNRM(J,ISYSN)
C
C*** SET PARAMETERS FOR GRAPH AND DO IT TO IT ***
C
    CALL XLAB(0)
    CALL SYMBL(ISYSN)
    CALL XDEN(1)
    CALL CHECK(XARRAY,YARRAY)
    IF (ISYSN.NE.1) GO TO 2000
C
C*** (ONLY CALL DSPLAY ON 1ST TIME THROUGH,ELSE USE CPLOT ***
C
    CALL DSPLAY (XARRAY,YARRAY)
    GO TO 1999
2000 CALL CPLOT (XARRAY,YARRAY)
1999 CONTINUE
C
C*** NOW OUTPUT SYMBOL AND RESPECTIVE SYSTEM NAME ***
C
    IF (ISYSN.LE.5) ISYSPL=ISYSN
    IF (ISYSN.GT.5) ISYSPL=ISYSN-5
    CALL MOVABS (600,600-100*ISYSPL)

```

```

        IF (ISYSN.GT.5) CALL MOVREL (1600,0)
        CALL SYMOUT (ISYSN,1.)
        CALL MOVREL (100,-30)
        DO 12 I=1,2
12      SYSNA(I)=SYSNAM(ISYSN,I)
        CALL AOUTST (20,SYSNA)
C
10      CONTINUE
C
C*** WRITE TITLE,AXIS LABELS ***
C
        DO 20 I=1,5
            CALL MOVABS (490+(I-1)*750,700)
            DO 21 J=1,2
21          ILABEL(J)=SUB1NM(I,J)
            CALL AOUTST (20,ILABEL)
20      CONTINUE
C
        CALL MOVABS(100,2500)
        CALL VLABEL (ISYST(1),ISYST(2))
        CALL MOVREL (0,-100)
        CALL TINPUT (K)
C
C*** TINPUT WAITS FOR INPUT OF ANY CHARACTER WITH ETX ***
C
        STOP
        END
C
        OVERLAY (GRF,3,0)
        PROGRAM GRAF3
C*****
C***
C* THIS OVERLAY CREATES (NSUB1) GRAPHS OF THE CASE 3 DATA.
C***
C*****
COMMON/UNITNO/ITTYIN,ITYOUT,IHDR,ISYS
COMMON/HDR1/SYSDSN,NSYS,NSUB1,NSUB2,NPARM,SYSSET(2),
& SB1SET(2),SB2SET(2),SYSNAM(10,2),SUB1NM(20,2),
& SUB2NM(10,10,2),PARNAM(40,2),NUM(40),RATED(40),
& YN(40),INVERT(40),INDEX
COMMON/PARAM4/SYSNRM(85,10)
COMMON /GRAF/ ISYST(15),XARRAY(4),RSYS,IBAUD,JTERM
DIMENSION YARRAY(6),ILABEL(2),SYSNA(2),ICOLON(4)
C
C*** INTEGER STATEMENTS ***
C
        INTEGER SYSDSN,SYSSET,SB1SET,SB2SET,SYSNAM,SUB1NM,
& SUB2NM,PARNAM,RATED,YN,VAL,SYST,SYSNA
        INTEGER SB1NME,SB2NME,NSYS2
        DATA ICOLON/3,32,32,58/
C
        DO #OC ITIME=1,NSUB1
            CALL INITT (IBAUD,JTERM)
            CALL BINITT

```

```

C      CALL DLIMY(0.,1.)
      CALL DLIMX(1.,5.)
      CALL SLIMY (900,2800)

C      YARRAY(1)=RSYS
      DO 10 ISYSN=1,NSYS
        DO 11 J=1,5
11      YARRAY (J+1)=SYSNRH(J,ISYSN)
C
      CALL XLAB(0)
      CALL SYMBL(ISYSN)
      CALL CHECK (XARRAY,YARRAY)
      IF (ISYSN.NE.1) GO TO 2000
      CALL DPLAY (XARRAY,YARRAY)
      GO TO 1999
2000   CALL CPLOT (XARRAY,YARRAY)
1999   CONTINUE
C
      IF (ISYSN.LE.5) ISYSPL=ISYSN
      IF (ISYSN.GT.5) ISYSPL=ISYSN-5
      CALL MOVABS (600,600-100*ISYSPL)
      IF (ISYSN.GT.5) CALL MOVREL (1600,0)
      CALL SYMOUT (ISYSN,1.)
      CALL MOVREL (100,-30)
      DO 12 I=1,2
12      SYSNA(I)=SYSNAM(ISYSN,I)
      CALL AOUTST (20,SYSNA)
10     CONTINUE
C
      DO 20 I=1,5
        CALL MOVABS (490+(I-1)*750,700)
        DO 21 J=1,2
21      ILABEL(J)=SUB2NM(ITIME,I,J)
        CALL AOUTST (20,ILABEL)
20     CONTINUE
C
      CALL MOVABS (100,2500)
      CALL VLABEL (ISYST(1),ISYST(2))
      CALL MOVABS (1200,3150)
      CALL AOUTST (20,S81SET)
C
      DO 41 J=1,2
41      ILABEL(J)=SUB1NM(ITIME,J)
      CALL NOTATE (2100,3150,ICOLON(1),ICOLON(2))
      CALL MOVABS (2300,3150)
      CALL AOUTST (20,ILABEL)
800   CALL TINPUT(K)
C
      STOP
      END

```

ATE
LME