ADAPTATION OF SYSTEM X COMPUTER-ASSISTED PROJECT MANAGEMENT EXERCISES FOR USE AT THE NAVAL POSTGRADUATE SCHOOL

Herbert Henry Joseph Nicholson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ADAPTATION OF SYSTEM X COMPUTER-ASSISTED PROJECT MANAGEMENT EXERCISES FOR USE AT THE NAVAL POSTGRADUATE SCHOOL

by

Herbert Henry Joseph Nicholson

Thesis Advisor: R. M. Hanna

June 1973

Adaptation of System X Computer-Assisted

Project Management Exercises For Use at

the Naval Postgraduate School

by

Herbert Henry Joseph Nicholson
Commander, United States Navy
B.S., Naval Postgraduate School, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

ABSTRACT

System X is a series of project management oriented case studies supported by an interactive time-sharing computer program simulating the analysis and evaluation of a hypothetical surface-to-surface guided missile system acquisition program. The computer-assisted exercises operate from baseline data supplied by a data base, along with system parameters set by the user, in order to compute various deterministic statistics regarding system performance and system costs. The system parameter values may be readily changed and the computations repeated. A desired result may be obtained by repeated iterations of the process.

TABLE OF CONTENTS

LIST OF TABLES

I. INTRODUCTION

System X is a management training device based on the life cycle of a hypothetical surface-to-surface tactical missile system. A series of exercises construct situations similar to those facing Department of Defense weapon system project managers. The exercises are related to management decision-making problems encountered in organization, financial management, contract administration, technical performance, requirements changes, and personnel management. The set of related case studies is an approach to the problem of developing, within the Department of Defense, qualified and competent managers in the field of weapon system acquisition [Ref. 1]. Each case builds on the previous case in order to provide a continuous evolution in the life cycle of the missile system. The case method is designed to provide the student user with a learning tool that he may use to derive an appreciation of the effect of managerial decisions on the solution of problems in the weapons system acquisition process. The use of a "real world" scenario in the cases provides a management training technique applicable for use either in conjunction with appropriate courses in a management curriculum or as a laboratory exercise used as a supplement to an overall curriculum in systems acquisition management.

Development of System X commenced in 1967 as a result of a contract to Peat, Marwick Management Systems Company for the former

Defense Weapons System Management Center, Wright-Patterson Air Force Base. The development of the system was completed in July 1972 for use in the five-month course in Systems Management conducted at the Defense Systems Management School, Fort Belvoir, Virginia.

II. OVERALL DESCRIPTION OF SYSTEM X

System X is a continuum of 30 exercises or case studies which encompass the life cycle of the CONQUEROR surface-to-surface missile system. The missile system is comprised of an amphibious vehicle, transporter, and the surface-to-surface tactical missile. The case studies present situations during five phases of the life cycle: conceptual activities, development, production, deployment, and retirement. Of the 30 exercises, 15 make use of a computer operating in a real-time interactive mode with the user, in order that he may manipulate the large amounts of data contained in a data base maintained within computer memory. The list of exercise titles and those which are computer-assisted are shown in Table I. The remainder of this thesis will concern itself only with the computer-assisted exercises.

The computer programs have been modified for use with an IBM System/360 Model 67 Time Sharing System using the CP/CMS (Control Program/Cambridge Monitor System) set of control and service programs, interactive with an IBM 2741 Communication Terminal. The purpose of CP/CMS is to create an environment in which many users can simultaneously carry out a wide range of data processing applications on a single computing system. In addition, each user can initiate, monitor, and terminate his particular application by carrying on a command/response type dialog or conversation with the system. CP/CMS consists of two

SYSTEM X EXERCISES

	EXERCISE NUMBER	TITLE
Conceptual Activities		
	1	Mission Analysis
	2	Measure of Effectiveness and Economy
*	3	Sensitivity Analysis
*	4	Contingency Analysis
*	5	Systems Requirement
	6	Establishment of a Program Office
	7	System Specification
	8	Procurement Concept
	9	Project Master Plan
	10	Validation Phase
	11	S/PO Management Information System
*	12	System Engineering
*	13	System Support Concepts
	14	System Test & Support Demonstration
Development		
*	15	Structuring Incentives
	16	Source Selection
*	17	Incentive Contract Negotiation

* computer assisted exercise

TABLE I

EXERCISE NUMBER		TITLE
	18	Development Decision
	19	Contractor Management Information System
*	20	Technical Problem
*	21	Engineering Change Proposal
*	22	Production Decision
Production		
*	23	Training Requirements
	24	Production Control Systems
*	25	Reprogramming
	26	Second Source Decision
*	27	Field Support Decision
*	28	Operational Availability and Logistic Support
Deployment		
*	29	Modification Decision
Retirement		
	30	Transitional Activities

* computer assisted exercise

TABLE I (cont.)

major components: a control program (CP-67) and a monitor (CMS). The control program creates the time-sharing part of the environment, which enables many users to simultaneously perform work. The monitor creates the conversational part of the environment, which enables a user to directly monitor his work by conversing with the system [Ref. 2].

All exercises require the use of a System X command and control program, a number of data sets, and a number of analytical models. The purpose of the command and control programs is to direct the processing sequence of the analytical models, establish the exercise data bases, and to provide for interface commands between the user and the analytical models.

There are three such command and control programs used, depending on the exercise desired. They have been named SYSTEM, INCENT, and PROGPL. The SYSTEM program is used for command and control of exercises 3, 4, 5, 12, 13, 20, 23, 27, 28, and 29. The INCENT program is used for command and control of the incentive contracting models used in exercises 15 and 17. The PROGPL program is used for command and control of the program planning models utilized with exercises 21, 22, and 25. A more detailed description of the command and control programs will be given in the next section.

The analytical models receive input from the data base which can be altered at the computer terminal. When satisfactory alteration of the data is completed, an EXECUTE command is issued allowing the analytical models to be executed in the order prescribed by the exercise. Following

the execution of the analytical models, the program returns to the command level. The user can then select output reports. The output reports are printed at the computer terminal by a report generator using the data stored in the data base.

The programs used for the computer-assisted exercises make use of two types of input data files: baseline data files and reports data files. The baseline data files contain data which are read into the program COMMON arrays for use by the analytical models. More than one baseline data file may be used by an exercise. The input reports data files, containing reports formats information, cross references of variables position, and editing information are combined into one random access output report file by a control program subroutine. A more detailed description of each of the above files will be given in the next section.

III. DESCRIPTION AND COMPOSITION OF COMPUTER PROGRAMS

A. CONTROL PROGRAMS

These programs control the processing sequence and act as an interface between the user and the computational models. The control programs are arranged into a main driver program and a number of subroutines to perform specialized procedures. All subroutine calls and execution are transparent to the user.

1. SYSTEM Program

After loading the SYSTEM program into main memory and start of execution, the user is prompted for the exercise number desired. The exercise number is then validated. If valid, the position of the exercise number in a list is used as an index to determine the number of baseline input data files and the input reports data files to be read. A message will be printed if the exercise number is not one of the valid numbers in the list and the user will be prompted for another number. Subroutine DATAX is executed to read baseline input data files required. The files required for each exercise are contained in Appendices A and B.

The data is saved in COMMON arrays. The input data files have a special format in order that various types of data may be combined into a single file. Data types recognized are floating point decimal, exponential, and alphanumeric data. Messages indicating successful reading of the data files are output to the user's terminal.

The user is then prompted for a decision reply to indicate whether or not a detailed list of variables is desired. A response of YES will cause a list of parameters which the user may change to be printed. Any other response sets a switch to suppress printing of the variables.

Subroutine MAKERM is then called to prepare the reports data file for later use in the printing of exercise reports. The input to the subroutine consists of three reports data file types: (1) Formats Data, (2) Reports Cross-reference Data and (3) Editing Data. The input Reports Data Files read are written on a random access file which has been created by the computer operating system. The output consists of the Reports Data File which contains a copy of the data in each of the input data files. The first record of the random access file contains three values indicating to the control program the starting record of the three types of data. A message indicating successful reading of the input Reports Data Files and creation of the random access file is output to the user's terminal.

A call to Subroutine COMMAN is then made. Inputs to this subroutine are mainly in the form of commands. These commands enable the user to control the actions of the computer programs. Certain commands, such as DISPLAY and REPORT, cause the reading of input from the Reports Data File and the output of information on the terminal. The subroutine also performs initialization actions. Additional subroutines called by Subroutine COMMAN provide for analysis of the user's command and to provide for the necessary actions dictated by the user's command.

Examples of valid commands are REPORT, CHANGE, DISPLAY, and EXECUTE. Detailed information concerning the commands is available in Ref. 3.

A command of EXECUTE causes return to the SYSTEM program which then calls the necessary analytical models required for the particular exercise, in the prescribed order of execution. The program is terminated by use of the command STOP. A listing of the SYSTEM program is shown as Appendix C.

2. INCENT Program

A control program similar to that used in the SYSTEM program is used with slight modifications as the Incentive Contract Negotiation program. This program, INCENT, is used with exercises 15 and 17. Files required for this program are shown in Appendices A and B. A list of variable parameters for these exercises is printed automatically, eliminating the requirement for a response from the user for such a listing. Minimum, maximum, and present value of the variable parameter may be determined by use of the DISPLAY command. Only the present value of the parameter may be changed by the use of the CHANGE command. The new value is required to be within the range indicated for that parameter.

Subroutine RUN calls the analytical model-subroutine ICM. A non-standard return is provided to enable the program to bypass the printing of a requested report if Subroutine ICM discovers an illegal set of conditions in the user's parameter settings. Additional detailed information concerning the program is available in Ref. 3. A listing of the INCENT program is shown as Appendix D.

3. PROGPL Program

The third command and control program, PROGPL, is similar in operation to the previous two. This program is used with exercises 21, 22, and 25. The main program is similar to that found in the SYSTEM program. A new subroutine, NETWRK, is included to provide initialization for the Program Evaluation and Review Technique (PERT) network. Two different networks are provided and are initialized from the baseline data base. The Engineering Change Proposal model, which forms the analytical model for exercise 21, is executed by a call to Subroutine PROCOS. Display or change of the current network may be accomplished by the command DISPLAY NET or CHANGE NET respectively. The use of the RESET command resets the network to its original value. Subroutine DATE provides for conversion of dates from calendar form to serial dates and vice versa. Provision is made for leap years. There is provision for a total of 200 activities in the PERT network. The report available is a specialized report known as an "Activity Schedule Report." Several format options are provided which are a representation of the PERT network or a part of the network. The user may request all activites be printed, only the critical path activities, or specify particular activities to be printed.

For exercises 22 and 25, execution is accomplished by a call to the Production Planning model, COSMOD. This model is similar in nature to those analytical models found with the SYSTEM program.

Additional detailed information concerning the program is available in Ref. 3. A listing of the PROGPL program is shown as Appendix E.

B. ANALYTICAL MODELS

There are ten different analytical models currently associated with the System X exercises. A brief description of the models and the command and control program for which it is available will be given below. A list of analytical models required for each command and control program is shown in Appendix F.

1. Mission Simulation Model

The primary functions of the Mission Simulation model (MISSIM) are to evaluate the effectiveness of the vehicle, transporter, and missile (VTM) against each type of target and to evaluate the expected VTM storage/launch unit and missile requirements for each type of target that is engaged. One VTM unit consists of 6 launchers, 40 missiles, 40 warheads, and associated support equipment. The effectiveness and requirements are calculated on a per target basis. Overall levels of VTM requirements and effectiveness are subsequently calculated by the Force Structure Effectiveness model. A detailed description of the model is contained in Refs. 4 and 5.

2. Force Structure Effectiveness Model

The primary purpose of the Force Structure Effectiveness model is to extend the results of the Mission Simulation model in order to determine force level requirements for VTM units. This information is

subsequently used in the Life Cycle Cost model to produce system life cycle costs. The detailed description of the model is contained in Refs. 6 and 7.

3. Logistics Model

The Logistics model (LOG) computes a series of logistics requirements which are used to determine support costs using the Life Cycle Cost model. The model receives most of its input from the baseline data base, and from the output of the Force Structure Effectiveness model. In the model sequence, the position of the Logistics model is fixed, executing after the Force Structure Effectiveness model and before the Life Cycle Cost model. The model is divided into a series of modules, each of which computes a separate logistics support requirement for the missile system. Some of the model input is general to the extent that it is used in more than one module. Other input is module dependent, being peculiar to only one module. Additional information concerning the model is contained in Refs. 8 and 9.

4. Life Cycle Cost Model

The purpose of the Life Cycle Cost model (LIFE) is to calculate and display the time-phased cost elements of research and development, procurement, and operation and maintenance for the system life cycle. The model uses parametric cost estimating relationships, learning curves, and other cost factors which are sensitive to parameters contained in the System X case studies. The model utilizes escalating factors which cause the cost elements to increase at specific points in the life of the project.

The Life Cycle Cost model is never executed by itself in the exercises. Rather, it depends upon the output of the other models either directly or indirectly. Depending upon the exercise, the specific models to be executed are called by the control program. In the calling sequence, however, the Life Cycle Cost model is always executed last, either after the Logistics Model or the Force Structure Effectiveness model.

The model is both resource-oriented (facilities and manpower) and functionally oriented (training and support equipment). The three classes of cost are broken down further into subcategories taking into consideration the orientations mentioned above. The model calculates, for the three major cost categories, discounted life-cycle costs as well as the undiscounted cost. The detailed description of the model is contained in Refs. 10 and 11.

5. Availability Model

The Availability model (AVAIL) computes operational availability for the missile system and its sub-systems where operational availability is defined as the probability that the system (or sub-system) will be capable of operating at or above its required level of performance if called upon to do so at a random point in time [Ref. 12]. Availability, as it is addressed in this model, is degraded by operational time lost due to preventive maintenance, corrective maintenance, and overhauls.

The model receives all its input from the baseline data set and does not depend on the output of any other model. In the model sequence, the Availability model is run before the Mission Simulation model. Its

order in the sequence with regard to the Capability and Reliability models is unimportant. The availabilities computed by the model are used as input by the Force Structure Effectiveness model and the Life Cycle Cost model. A more complete description of the model and its parameters is contained in Refs. 12 and 13.

6. Capability Model

The purpose of the Capability model (CAPAB) is to provide realistic performance measurements (capabilities) of the missile and its transporter. The Capability model is subdivided into four major submodels which interact with each other.

The Missile Performance submodel is the largest and most complicated of the individual submodels. This submodel simulates the flight of the missile from launch to target, gathering performance statistics during the missile's flight. The missile's thrust profile is divided into three basic segments: boost phase, sustain phase, and glide phase.

The missile is directed toward the target through an inertial guidance system which may be assisted by laser-beam illumination of the target. During the flight of the missile, the Guidance System submodel detects deviations from the optimal trajectory and attempts to make the corresponding flight corrections. The inertial guidance system can be designed to provide a certain level of predefined targeting accuracy.

The transporter of the Conqueror missile is an integral part of the overall missile delivery system. The Transporter Speed and Distance submodel defines the vehicle cruise speed, swim speed, and cruise distance.

The Reaction Time submodel provides for a summation of the individual sub-system reaction times in terms of both initial firing and retargeting, and repeat firings. A more complete description of the parameters involved in the submodels and the equations required for calculation of the output values is contained in Refs. 14 and 15.

7. Reliability Model

The Reliability model (RELIB) computes reliability for the Conqueror missile. The missile target engagement sequence can be divided into three steps: pre-ignition, ignition, and in-flight. Reliabilities associated with the last two steps, ignition and in-flight, are addressed in this model. Pre-ignition reliability is computed as missile availability in the Availability model. The Reliability model computes: (1) missile in-flight reliability at various specified ranges, up to and including maximum range, (2) guidance subsystem in-flight reliability at maximum range, (3) propulsion subsystem in-flight reliability at maximum range, and (4) missile propulsion subsystem ignition reliability.

The Reliability model is exercised, in sequence, after the Capability model, from which time of flight data are obtained. The computed output of the Reliability model can be used directly, for example, to examine the sensitivity of life cycle cost to changes in missile subsystem mean time between in-flight failures. The output is also used as input to the Mission Simulation model, where missile reliability impacts upon the effectiveness of the missile when engaging targets. A further description of the model is contained in Refs. 16 and 17.

8. Contractor Incentive Contract Model

The purpose of the Contractor Incentive Contract model (ICM) is to perform a contractor oriented analysis of a proposed incentive contract. This analysis is done with contractor information not normally available to the government including: contractor profit strategy, contractor estimated cost to produce, and cost-performance curves on each performance parameter which is incentivized in the contract.

Utilizing this contractor supplied information, the basic incentive contract, as defined by specific parameter values supplied by the student user or instructor/monitor, is pursued according to one of five contractor oriented profit strategies: (1) the maximization of incentive fee in dollars, (2) the maximization of incentive fee as a percent of sales, (3) the maximization of performance for a given incentive fee, (4) the maximization of performance for a given fee percentage of sales (where sales is the sum of development cost and the total incentive fee), and (5) the maximization of performance for a given cost overrun. Evaluative information is supplied to the user in the form of output reports. These include costs, performance achieved, fees or profits, and the relative value of additional incentive fee on that parameter. Ten individual parameters are available as potential parameters for incentivation. The user may define which, if any, of the parameters for which he may desire to establish incentive values. The relationship between fee and performance is established by specifying the incentive fee value at minimum,

maximum, and target performance values. A linear relationship is used for the share-lines connecting target performance with the maximum and minimum values of performance.

For each of the incentivized parameters, the contractor is assumed to have knowledge of the performance value he may obtain by investing additional development dollars in the parameter. This is referred to as the contractor's cost-performance curve. The contractor must meet the specification value in the contract for all parameters. This specification value is referred to as the minimum performance point. The contractor's cost to obtain the minimum performance point is included in the lump sum known as the "cost to develop a minimum performance system." The cost-perfomance curve for the individually incentivized parameters is not defined at performance points less than the minimum performance value. From the minimum performance value, additional development dollars spent on each incentivized parameter will result in increased performance levels up to a maximum achievable performance level. At this maximum level, the contractor believes that additional development dollars will not produce any significant increase in performance. Hence, the cost-performance curve is asymptotic to the maximum performance level line. The determination of how the contractor will trade-off receivable fees is referred to as his profit strategy.

The initial baseline values for the incentive fees have all been set to zero, in effect, indicating that no parameters have been incentivized. The minimum acceptable performance has been established for each

parameter. Unless the parameter is incentivized, the contractor will produce the minimum performance value for that parameter. Minimum performance points are fixed and may not be varied during the exercise. Target performance points may be set at values within the defined ranges. Additional information concerning contract structure, contractor's structure, parameters, and graphs of the multiple incentive contract share-lines are contained in Refs. 18 and 19.

9. Program Planning Model

The Program Planning model is divided into two independent submodels. One submodel, the Schedule submodel, is PERT oriented and performs network calculations to produce time scheduling information. The user may utilize the scheduling submodel to evaluate alternative production networks encountered in the project management of the Conqueror missile system program. The Cost submodel is oriented to the production planning process and consolidates the planning costs associated with the various management decisions and provides cost reports to the user. This submodel allows the user to aggregate the many varied cost inputs from the production process to produce information-oriented reports for use in managing the production program.

The Schedule submodel (PROCOS) takes, as input, activities in a PERT-oriented network and computes, as output, the early start date, late start date, early finish date, late finish date, and the slack for each of the activities in the network. The user is not allowed to alter the activity arrangement or sequence in the network. He may,

however, alter activity durations. The basis of the network calculations is a topological sort. The critical path for the network selected is indicated by those activities which have the smallest (or largest negative) slack. Those activities on the critical path are indicated on the output reports with an asterisk.

Subroutine DATE is required to convert dates from standard notation into a serial date required for mathematical manipulation. This conversion is reversible for use in the output reports. The serial date is relative to a given zero-date which has been selected as 1 January 1950.

Three reports are available in exercise 21. The use of the command REPORT ALL will list all activities in the current network. The command REPORT CRITICAL will list all activities on the critical path in chronological order. The command REPORT n, where n represents one or more activities, allows the printing of only the selected activites.

For exercises 22 and 25, the Program Planning model uses the second independent submodel, COSMOD. In the process of production planning, the manager is faced with a variety of problems which focus upon three major types of decisions. These are: (1) production go-ahead date, (2) the delivery date of the first production unit, and (3) the procurement quantity of the units over the time frame of the production phase. An error checking routine is used to check to make sure no production orders are requested in years prior to one year after the Go-ahead Decision was given and that the production capacity is not exceeded by the production order quantity. If an error of this nature is found, an error message

is printed at the terminal indicating the current production go-ahead date and which production orders violate the production capacity constraint for each year. In order to assist in making the above decisions, the submodel accumulates costs, for later use in various output reports, in the three major categories of Research, Development, Test and Evaluation (RDT&E), Production, and Operation and Maintenance.

The unit costs exhibit an inflationary trend over the years. There is, however, a "learning curve" associated with the production of the items which will allow fewer people and facilities to produce the same quantity of output. In later production years, the interaction of the learning curve and inflation allow optimal production of lot sizes smaller than the lot sizes in the initial production years.

Both exercise 22, the Production Decision exercise, and exercise 25, the Reprogramming exercise, use the Cost submodel as the analytical model. However, in the System X time frame, the baseline cost values used with exercise 25 have significantly changed. At this point in the life cycle, most RDT&E and certain production funds have been expended. The majority of the production engineering cost and Long Lead Time Item costs have been obligated. These items will no longer be affected by any production decisions. Procurement costs have been increased by six percent.

There are 86 parameters available to the user. Parameter 1 refers to the Production Start-up Date. Parameter 2 is the number of months between the Production Start-up Date and the first unit delivery.

The remainder of the parameters, 3 through 86, refer to the unit quantity of the six possible hardware end items to be produced in a given year. Additional detailed material concerning the Program Planning models, including tables of topological sequence numbers for the PERT networks, category cost matrices, and baseline parameter values are contained in Refs. 20 and 21.

C. DATA FILES

The input data files contain the baseline data values used as input through the COMMON arrays by the analytical models and the input reports data files which form the basis for the random access reports data file used in the printing of the various output reports.

1. Baseline Data File

One or more baseline data files are read into the computer program COMMON arrays before any analytical models are executed. The files contain initialization data values in several different formats: floating point decimal, alphanumeric, and exponential formats. After initialization of the master array, V, the values from the baseline data set are read into specific locations in the V array. Only a portion of the locations in the V array have baseline data values assigned. The remainder are reserved for values entered by the user or those values computed during execution of the analytical models. The baseline data values are therefore inviolate and allow resetting of all values to the original baseline values by use of the command RESET.

2. Input Reports Data Files

These files contain three types of data files: (1) Reports Formats Data Files, (2) Reports Cross-reference Data Files (also known as Variables Data Files), and (3) Editing Data Files. The formats used for printing the reports of an exercise are contained in the Reports Formats Data File. The format for each report are placed in the file in the sequence in which they will be used to print the report. A record placed after the final format for each report containing the characters END signify the end of the report formats.

The Reports Cross-reference Data File is used in conjunction with the Reports Formats File. The file contains data which indicate to the control program which master array location values are to be printed in each report and the appropriate formats record in the Reports Format File to be used.

The Editing Data File is used to supply information for the parameters which may be changed by the user. One record is used for each changeable parameter. Each record contains four data elements: parameter description, minimum permitted value of the parameter, maximum permitted value of the parameter, and the position of the parameter value in the master array. This data file is referenced by the command and control program each time the user attempts to change the value of one of the parameters. The new value is compared with the permitted maximum and minimum values and rejected if it is outside the permitted

range. The last record of each edits data file contains the characters END. Complete documentation concerning the data base is contained in Refs. 22 through 28.

IV. SYSTEM IMPLEMENTATION

The requirement for conversion of the computer program was due to the incompatibility of various commands in the original version, used in conjunction with a General Electric Model GE-635 computer, with the commands required by the IBM System/360 computer. The differences in FORTRAN language options allowed by the compilers used with the General Electric and IBM systems also required changes in order to make the programs compatible for use on the Naval Postgraduate School's IBM System/360 computer. A discussion of the overall types of changes along with specific examples of changes required is made below.

In the original version, the programs for a typical exercise were arranged in the form of a series of overlays, which are logically independent sections of the entire program. Each section could then be brought into computer memory as required. In the IBM System/360, as installed at the Naval Postgraduate School, each program using the system is treated as a sequence of 4096-byte units called "pages." By dividing programs into pages, processor storage can be allocated in page (4096-byte) increments. This eliminates the requirement for the program to be divided into logically independent sections. The programs were converted into main programs and associated subroutines called either by a main program or by another of the subroutines used by the main program.

In the original version, the General Electric GE-635 FORTRAN language compiler allowed each overlay to either return to the main program or for control to be passed to another overlay by the use of a CHAIN statement. All CHAIN statements have been converted to a RETURN statement, causing a return of control to the calling main program or subroutine. In the Naval Postgraduate School IBM System/360 FORTRAN compiler, initialization of variables is not automatically performed. It is, therefore, imperative that all variables be initialized in some manner prior to their use. The dimension of arrays whose values are passed as an argument in a subroutine call statement must be equivalent in both the calling program and the subroutine.

One major problem with the original version was the amount of disk storage space required to store the 43 data files required to provide the data base. The Baseline Data Files comprised the largest volume of records required to be stored. Originally, the Baseline Data Files were comprised of 5102 80-character records. By a conversion of the Baseline Data Files into a format which allows a number of data values to be used in each 80-character record, a reduction of over 82 percent, to 909 80-character records, in total record length was achieved.

The computer Operating System Library contains a number of subroutines which are available for use under CMS. The DEFINE subroutine defines FORTRAN disk files that may be accessed randomly and makes a correspondence between a CMS file and a FORTRAN logical unit number. Before each READ or WRITE (or PRINT) statement, the record number must

be set to the record desired. After the execution of the READ or WRITE statement, the record number is automatically incremeted to point to the next record in the file [Ref. 2]. This Operating System Library subroutine is used to form the random access file, FT09F001, used as the output reports data file. Upon the user command STOP, a call is made to two other Operating System Library subroutines; the ERASE subroutine which removes the random access file, FT09F001, from the user's directory and to the EXIT subroutine which causes termination of the program.

Prior to conducting a computer-assisted exercise the user should familiarize himself with the material contained in the appropriate Student Exercise Booklet, Computer Operations Guide, and Computer User's Guide for the particular exercise to be conducted. Instructors/Monitors will find additional material concerning the case contained in the Instructor's Guide Booklet for the particular exercise. Appendix G presents typing conventions and sample exercise commands, reports and messages.

V. USE OF EXERCISES IN CONJUNCTION WITH CURRICULA

The case study method and management game simulation are used extensively at the Naval Postgraduate School as well as other colleges and universities. The System X project management exercises encompass both the case study and the simulation methods. The exercises consider the project manager from the viewpoint of the management scientist who utilizes mathematics, models, and computers to aid him in making optimum decisions. The exercises allow a management student who has studied the principles involved in project management to apply those principles to the decision-making involved in a hypothetical project. The case study method utilized is augmented through the use of the computer which serves as an aid to the user in order to manage the data base. The computer develops reports, makes computations, and makes data comparisons. Instructional methods and materials must include not only the cases themselves and the use of the computer but also must include lectures, seminars, discussions, and workshops.

There are a number of alternatives for use of the set of exercises comprising the life-cycle of the project. The first of these alternatives is to use the System X exercises to formulate a Directed Study course conducted over a one academic quarter time frame. This course could be conducted on either a Pass/Fail basis or as a graded course. The use of the exercises as an unstructured learning experience would lend itself

to the use of the Pass/Fail system of grading, whereas conducting the course on a graded basis would require a subjective grade to be determined by the instructor team. Within the framework of a Directed Study, the exercises may be handled in a number of ways. The simplest of these is to use the exercises as an acquisition overview with the missile system scenario as a "typical" weapon system acquisition. In this case, the computer would receive only limited use in exploring the alternative decisions available in each case. In its most elementary form, this would require only the use of the baseline data values and a limited number of iterations, changing only a limited number of parameter values in order to arrive at the general changing trend of the output values rather than continuing the process to determine a near optimum solution. This may be expanded as required to illustrate the principles and problems involved at each stage of the acquisition process.

A second alternative is to utilize specific exercises or groups of exercises in conjunction with appropriate courses throughout the curriculum. This could be accomplished during the actual period of a particular course to illustrate or provide examples of situations related to the material for which instruction is being provided. Alternatively, selected exercises could be conducted at the beginning of an academic quarter in lieu of course instruction normally conducted at that time. This alternative has the disadvantage of tending to compress the material scheduled to be taught in that quarter into a shorter time frame. If exercises are to be conducted in addition to regularly scheduled material, consideration

to assignment of laboratory-type credit for the time expended should be made. Table II depicts the relevancy of the various System X exercises for applicability to courses within the Systems Acquisition Management curriculum, as contained in the Naval Postgraduate School catalog for 1972-1974.

Thirdly, the set of exercises could be used as the basis for a group thesis project simulating the project management roles required in the operation of a project office managing the acquisition of a major weapon system. A disadvantage to this replacement of the individual thesis is that it eliminates the student's option on selection of a thesis topic. In addition, considerable modification of the scenario and cases would be required to fit the Navy environment in a manner justifying the experience as a thesis project.

It is possible to use some of the System X case exercises in other than a quantitative way. A number of the case exercises, particularly those which are not computer-assisted, may lend themselves as cases in behavioral or management theory studies. The selection of these type cases from the entire package may be found useful in studies involving organizational behavior and human factors. This relationship is shown in Table II. A synopsis of prior case material to the one selected may be required in order for the student to have sufficient background on which to evaluate the selected case. A discussion of the impact of recent political, economic, technological, and social factors on the case may be worthwhile within the selected courses of instruction.

TABLE OF RELEVANCY TO SYSTEMS
ACQUISITION MANAGEMENT CURRICULUM

NPS COURSE NUMBER*	RELEVANT EXERCISE NUMBERS
MN 4145	1, 2, 3, 4, 5
SM 3301	9, 18
SM 3302	6, 8, 9, 10, 18, 25, 30
SM 3304	18, 25
SM 3305	11, 19, 21, 24
SM 4301	7, 8, 10, 12, 14, 20, 21, 22, 29
SM 4302	9, 10
SM 4303	8, 15, 17
SM 4304	16, 26
SM 4305	13, 23, 25, 27, 28

* Naval Postgraduate School Catalog for 1972-1974.

TABLE II

There are possibilities for use of the System X case studies outside of the Systems Acquisition Management curriculum. An examination of the relevancy of the exercises to other curricula should be the subject of further investigation.

VI. CONCLUSIONS AND RECOMMENDATIONS

Based on the alternatives contained in the previous section, it is considered that the project management exercises which comprise System X are suitable for use within a management curriculum by students at the Naval Postgraduate School. Within the Systems Acquisition Management curriculum the following recommendations are made for use of the System X exercises.

The System X exercises should be used within the Systems Acquisition Management curriculum to formulate a Directed Study course to be graded on a Pass/Fail basis. The course should be conducted over at least a one academic quarter time frame in the latter third of the curriculum. The exercises to be conducted should include as a minimum those which concern themselves with the Conceptual, Development, and Production phases of the system life cycle as shown in Table I. The course must be supported by a team of instructors/monitors knowledgeable and experienced in project management to allow a wide spectrum of background, expertise, and opinion to be available to the student.

In the event that the above course of action is impracticable, it is alternatively recommended that selected exercises be used in conjunction with existing courses in the curriculum. Individual cases may be found useful in conjunction with courses in Project Management, Systems Analysis, Project Information Systems, Systems Engineering Management,

Procurement Planning and Negotiation, and Logistic Support. The relevancy of individual exercises to courses within the Systems Acquisition Management curriculum is contained in Table II.

APPENDIX A. EXERCISE INPUT BASELINE DATA FILES

EXERCISE NUMBER	FILETYPE IDENTIFIER			
3	FT03F001	FT03F002		
4	FT03F001	FT03F002		
5	FT03F001	FT03F002		
12	FT03F001	FT03F002		
13	FT03F001	FT03F002	FT03F003	
15				FT10F001
17				FT10F001
20	FT03F001	FT03F002		
21				FT10F002
22				FT10F003
23	FT03F001	FT03F002	FT03F003	
25				FT10F004
27	FT03F001	FT03F002		
28	FT03F001	FT03F002		
29	FT03F001	FT03F002		

APPENDIX B. EXERCISE INPUT REPORTS DATA FILES

EXERCISE NUMBER	FILETYPE IDENTIFIER		
3	FT02F001	FT02F002	FT02F003
4	FT02F001	FT02F002	FT02F003
5	FT02F001	FT02F002	FT02F003
12	FT02F004	FT02F005	FT02F006
13	FT02F007	FT02F008	FT02F009
15	FT01F001	FT01F002	FT01F003
17	FT01F001	FT01F002	FT01F003
20	FT02F010	FT02F011	FT02F012
21	FT04F001	FT04F002	FT04F003
22	FT04F004	FT04F005	FT04F006
23	FT02F013	FT02F014	FT02F015
25	FT04F007	FT04F008	FT04F009
27	FT02F016	FT02F017	FT02F018
28	FT02F019	FT02F020	FT02F021
29	FT02F022	FT02F023	FT02F024

APPENDIX C

COMMAND AND CONTROL PROGRAM (SYSTEM)

```
C     *********************
C     * SYSTEM PROGRAM    *
C     *********************
C        MAIN PROGRAM TO DRIVE EXERCISES
C        3, 4, 5, 12, 13, 20, 23, 27, 28, 29
C
C        READS DATA BASE AND CALLS OTHER MODELS
C        CONTAINS - MAIN PROGRAM
C                   SUBROUTINE DATAX-READS DATA FILES
C                   SUBROUTINE MAKERM-READS REPORT FILES
C                   SUBROUTINE COMMAN IS THE COMMAND MODULE
C
      COMMON REPORT,NX,INX
      COMMON V(4500),VERB,AUTO,NPCSFM,NPCSVR,NPCSEC,REPSW
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC
C
C        NEXERS CONTAINS EXERCISE NUMBERS FOR THIS DRIVER
C        REPNAM CONTAINS THE REPORT FILE NUMBER
C        BASNAM CONTAINS THE NUMBER OF BASE DATA FILE
C
      INTEGER NEXERS(10)/3,4,5,12,13,20,23,27,28,29/,
     1 BASNAM(10)/2,2,2,2,3,2,3,3,2,2/,
     2 REPNAM(10)/1,1,1,2,3,4,5,6,7,8/
      INTEGER REPORT,FILE,CH1SW,CH2SW
      DATA AN/'YES '/
      LOGICAL VERB,AUTO
      REPORT=9
      IVARCN=0
C *** INITIALIZE ARRAY V
      DO 3 I=1,4500
    3 V(I)=I
      KOUNT=0
   54 IF(KOUNT.GT.5)GOTO 99
C *** READ EXERCISE NUMBER
      PRINT 4
    4 FORMAT(' EXERCISE NUMBER ?')
      READ(5,5) EXNO
    5 FORMAT(F5.2)
      NX=EXNO
C *** CHECK FOR VALID EXERCISE NUMBER
      DO 6 I=1,10
      IF(NX.EQ.NEXERS(I)) GOTO 8
    6 CONTINUE
```

```
      PRINT 7,NX                                                        SYS00440
    7 FORMAT(' ** INVALID EXERCISE NUMBER - ',I5/)                      SYS00450
      KOUNT=KOUNT+1                                                     SYS00460
      GOTO 54                                                           SYS00470
    8 INX=I                                                             SYS00480
      NERR=0                                                            SYS00490
      VERB=.TRUE.                                                       SYS00500
C *** FILE IS NUMBER OF BASE FILE TO READ                                SYS00510
      FILE=BASNAM(I)                                                    SYS00520
      CALL DATAX(FILE)                                                  SYS00530
      V(1)=NX                                                           SYS00540
   13 CONTINUE                                                          SYS00550
      CH1SW=1                                                           SYS00560
      CH2SW=1                                                           SYS00570
      PRINT 16                                                          SYS00580
   16 FORMAT(/' DO YOU WANT TO SEE THE LIST OF VARIABLES',              SYS00590
     1' (YES OR NO) ?'/)                                                SYS00600
      READ(5,116)ANS                                                    SYS00610
  116 FORMAT(A4)                                                        SYS00620
      IF(ANS.EQ.AN)GOTO 14                                              SYS00630
      CH1SW=2                                                           SYS00640
      CH2SW=2                                                           SYS00650
C *** NREP IS NUMBER OF REPORT FILE TO READ                             SYS00660
   14 NREP=REPNAM(I)-1                                                  SYS00670
      CALL MAKEFM(NREP)                                                 SYS00680
   99 CALL COMMAN                                                       SYS00690
      GOTO(345,345,345,120,130,200,130,130,130,290),INX                 SYS00700
C *** FOR EXERCISES 3, 4, 5                                             SYS00710
  345 CALL MISSIM                                                       SYS00720
      CALL FORCE                                                        SYS00730
      CALL LIFE                                                         SYS00740
      GOTO 99                                                           SYS00750
C *** FOR EXERCISE 12                                                   SYS00760
  120 CALL CAPAB                                                        SYS00770
      CONTINUE                                                          SYS00780
      CALL RELIB                                                        SYS00790
      CALL AVAIL                                                        SYS00800
      GOTO 99                                                           SYS00810
C *** FOR EXERCISES 13, 23, 27, 28                                      SYS00820
  130 CALL AVAIL                                                        SYS00830
      CALL MISSIM                                                       SYS00840
      CALL FORCE                                                        SYS00850
      CALL LOG                                                          SYS00860
      CALL LIFE                                                         SYS00870
      GOTO 99                                                           SYS00880
C *** FOR EXERCISE 20                                                   SYS00890
  200 CALL CAPAB                                                        SYS00900
      CALL RELIB                                                        SYS00910
```

```
      GOTO 130
C *** FOR EXERCISE 29
  290 CALL MISSIM
      CALL FORCE
      GOTO 99
      END


      SUBROUTINE MAKERM(IUN)
C     **********************
C     **********************
      COMMON REPORT
      COMMON LIN
      DIMENSION FORM(16),IV(16),FMAT(10),FV(2)
      LIN=1
      CALL DEFINE(9,'FILE    ','FT09F001',LIN,80)
      IF(IUN.EQ.0)GOTO 10
      IEN=IUN*3
      IDD=0
    5 READ(2,11,END=6)
      GOTO 5
    6 IDD=IDD+1
      IF(IDD.NE.IEN) GOTO 5
   10 NPOSFM=2
      LIN=2
    8 READ(2,11,END=20)FORM
   11 FORMAT(1X,16A4)
      WRITE(9) FORM
      GO TO 8
   20 NPOSVR=LIN
   26 READ(2,24,END=60) IV
   24 FORMAT(16I5)
      WRITE(9) IV
      GO TO 26
   60 NPOSED=LIN
   61 READ(2,62,END=70) FMAT,FV,ID
   62 FORMAT(1X,10A4,2F10.2,I5)
      WRITE(9) FMAT,FV,ID
      GO TO 61
   70 LIN=1
      WRITE(9) NPOSFM,NPOSVR,NPOSED
      PRINT 71
   71 FORMAT(/' CREATED REPORT FILE FT09F001')
      RETURN
      END
```

```
      SUBROUTINE DATAX(IFL)
C     ********************
C     ********************
      COMMON REPORT,NX,INX
      COMMON V(4500),VERB,AUTC,NPCSFM,NPCSVR,NPCSED,REPSW
     1,CH1SW,CH2SW,NERR,IPAR,AMIN,AMAX,AINC,IVARCN
      DIMENSION FMT(4)
      IEF=0
  100 LAST=0
      N=1
  200 READ(3,219,END=333) ISKIP,IC,FMT
  219 FORMAT(2I5,4A4)
      IF(ISKIP.EQ.0)GOTO 250
      N=N+LAST
      LAST=ISKIP
  250 IF(IC.EQ.0)GOTO 200
      READ(3,FMT) (NPOS,V(NPOS+N-1),II=1,IC)
      GO TO 200
  333 IEF=IEF+1
      PRINT 55,IEF
   55 FORMAT(/' BASE DATA READ FROM FILE FT03F00',I1)
      IF(IEF.LT.IFL) GO TO 100
      RETURN
      END


      SUBROUTINE COMMAN
C     ****************
C     ****************
C      CONTAINS LOGIC FOR USER DIALOGUE
C       SUB-ELEMENTS ARE:
C                     XTRACT SUBROUTINE
C                     REPRT SUBROUTINE
C                     CHA1 SUBROUTINE
C                     CHA2 SUBROUTINE
C                     SET SUBROUTINE
      COMMON REPORT,NX,INX
      COMMON V(4500),VERB,AUTC,NPCSFM,NPCSVR,NPCSED,REPSW
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC
      COMMON LIN
      DIMENSION LIST1(50),LIST2(10),LINE(72)
      INTEGER REPORT,FILE,CH1SW,CH2SW
      LOGICAL VERB,AUTC
      CALL DEFINE(9,'FILE    ','FT09F001',LIN,80)
      CONFIG=V(2)
      IF(CH1SW.EQ.3.OR.CH2SW.EQ.3)GOTO 15
C *** FIRST TIME THRU- INITIALIZE -
C *** INITIALIZE VERBOSE/BRIEF SWITCH
```

```
      VERB=.TRUE.                                                       SYS01810
C *** INIT. ALTO SWITCH PER EXERCISE                                    SYS01820
      GOTO(6,7,7,7,7,7,7,7,7,7),INX                                     SYS01830
    6 ALTO=.TRUE.                                                       SYS01840
      GOTO 8                                                            SYS01850
    7 ALTO=.FALSE.                                                      SYS01860
    8 GOTO(10,9,9,10,9,10,9,9,9,9),INX                                  SYS01870
C *** SET CONFIGURATION                                                 SYS01880
    9 CONFIG=V(2)                                                       SYS01890
      CALL CONFG(CONFIG)                                                SYS01900
   10 PRINT 120,V(1)                                                    SYS01910
      LIN=1                                                             SYS01920
      READ(REPORT)NPOSFM,NPOSVR,NPOSEC                                  SYS01930
C *** PROCESS EDITS FILE WITH INITIAL CALL TO CHANGE SUBS               SYS01940
C *** ARGUMENTS ARE NOT USED IN FIRST CALL                              SYS01950
      GOTO(12,13,12,12,12,12,12,12,12,12),INX                           SYS01960
C *** READ EDITS FILE(S) AND SAVE PARAMETER POINTERS                    SYS01970
   12 CALL CHA1(NFNC,L1,LIST1)                                          SYS01980
      IF(INX.NE.3)GOTO 15                                               SYS01990
   13 CALL CHA2(NFNC,L1,LIST1)                                          SYS02000
C *** NORMAL ENTRY                                                      SYS02010
C *** REQUEST USER COMMAND                                              SYS02020
   15 CONTINUE                                                          SYS02030
      IF (IVARCN.NE.0) CALL VARY                                        SYS02040
      IF(NERR.EQ.5)PRINT 11                                             SYS02050
   11 FORMAT(/' AW C:MON YOU HAVE MADE FIVE INPUT ERRORS.'/             SYS02060
     1  ' IF SYSTEM X WERE OPERATIONAL YOU WOULD HAVE WIPED OUT'/       SYS02070
     1  ' MONTREAL, CHICAGO AND EAST MCKEESPORT PENNSYLVANIA='/)        SYS02080
      IF(NERR.EQ.5)NERR=6                                               SYS02090
      PRINT 14                                                          SYS02100
   14 FORMAT(/' COMMAND ?'/)                                            SYS02110
C *** READ USER COMMAND                                                 SYS02120
   17 READ 16, LINE                                                     SYS02130
   16 FORMAT(72A1)                                                      SYS02140
      CALL XTRACT(LINE,72,L1,LIST1,L2,LIST2,NFNC,NREP,&15)              SYS02150
      IF(NFNC.EQ.0)GOTO 17                                              SYS02160
      GOTO (20,25,30,35,40,45,50,55,60,70,70,75,80),NFNC                SYS02170
C *** REPORT OPTION ***                                                 SYS02180
   20 IF(NREP.LT.1)GOTO 15                                              SYS02190
C *** PRINT REPORT HEADING                                              SYS02200
      PRINT 120,V(1)                                                    SYS02210
      GOTO(21,22,23,224,225,228,230,231,232,233),INX                    SYS02220
C *** EXERCISE 3                                                        SYS02230
   21 PRINT 121                                                         SYS02240
      GOTO 24                                                           SYS02250
C *** EXERCISE 4                                                        SYS02260
   22 PRINT 122                                                         SYS02270
      GOTO 24                                                           SYS02280
```

```
C *** EXERCISE 5                                                        SYS02290
   223 PRINT 123                                                        SYS02300
       GOTO 24                                                          SYS02310
C *** EXERCISE 12                                                       SYS02320
   224 PRINT 124                                                        SYS02330
       GOTO 24                                                          SYS02340
C *** EXERCISE 13                                                       SYS02350
   225 PRINT 125                                                        SYS02360
       GOTO 24                                                          SYS02370
C *** EXERCISE 20                                                       SYS02380
   228 PRINT 128                                                        SYS02390
       GOTO 24                                                          SYS02400
C *** EXERCISE 23                                                       SYS02410
   230 PRINT 130                                                        SYS02420
       GOTO 24                                                          SYS02430
C *** EXERCISE 27                                                       SYS02440
   231 PRINT 131                                                        SYS02450
       GOTO 24                                                          SYS02460
C *** EXERCISE 28                                                       SYS02470
   232 PRINT 132                                                        SYS02480
       GOTO 24                                                          SYS02490
C *** EXERCISE 29                                                       SYS02500
   233 PRINT 133                                                        SYS02510
       GOTO 24                                                          SYS02520
C *** PRINT REPORT WITH REPORT SUBROUTINE                                SYS02530
    24 CALL REPRT(NREP)                                                 SYS02540
       GOTO 15                                                          SYS02550
C *** REPORT HEADINGS                                                   SYS02560
   120 FORMAT(//' EXERCISE ',F3.0)                                      SYS02570
   121 FORMAT(/' SENSITIVITY ANALYSIS.'/1X,43(1H-)/)                    SYS02580
   122 FORMAT(/' CONTINGENCY ANALYSIS.'/1X,43(1H-)/)                    SYS02590
   123 FORMAT(/' SYSTEM REQUIREMENTS.'/1X,43(1H-)/)                     SYS02600
   124 FORMAT(/' SYSTEMS ENGINEERING.'/1X,43(1H-)/)                     SYS02610
   125 FORMAT(/' SYSTEM SUPPORT CONCEPTS.'/1X,43(1H-)/)                 SYS02620
   128 FORMAT(/' TECHNICAL PROBLEMS.'/1X,43(1H-)/)                      SYS02630
   130 FORMAT(/' TRAINING REQUIREMENTS.'/1X,43(1H-)/)                   SYS02640
   131 FORMAT(/' FIELD SUPPORT DECISION.'/1X,43(1H-))                   SYS02650
   132 FORMAT(/' OPERATIONAL AVAILABILITY AND LOGISTIC SUPPORT.',       SYS02660
     1        /1X,63(1H-)/)                                             SYS02670
   133 FORMAT(/' MODIFICATION DECISION.'                                SYS02680
     1              /1X,43(1H-)/)                                       SYS02690
C                                                                       SYS02700
C *** CHANGE OPTION ***                                                 SYS02710
    25 GOTO(26,27,28,26,26,26,26,26,26,26),INX                          SYS02720
    26 CALL CHA1(NFNC,L1,LIST1)                                         SYS02730
       GOTO 15                                                          SYS02740
    27 CALL CHA2(NFNC,L1,LIST1)                                         SYS02750
       IF(NFNC.EQ.6)GOTO 15                                             SYS02760
```

```
      DO 271 I=1,L1                                                     SYS02770
      IF(LIST1(I).NE.1) GOTO 271                                        SYS02780
      IF(CONFIG.EQ.V(2))GOTO 15                                         SYS02790
      CONFIG=V(2)                                                       SYS02800
      CALL CONFG(CONFIG)                                                SYS02810
      GOTO 15                                                           SYS02820
  271 CONTINUE                                                          SYS02830
      GOTO 15                                                           SYS02840
   28 CONTINUE                                                          SYS02850
      IF(NREP.NE.3.AND.NREP.NE.16)PRINT 29                              SYS02860
   29 FORMAT(/' USE C OR T TO SPECIFY PARAMETERS.'/)                    SYS02870
      IF(NREP.EQ.3) CALL CHA1(NFNC,L1,LIST1)                            SYS02880
      IF(NREP.EQ.16) CALL CHA2(NFNC,L1,LIST1)                           SYS02890
      GOTO 15                                                           SYS02900
C *** STOP OPTION ***                                                   SYS02910
   30 STOP                                                              SYS02920
C *** VERBOSE OPTION ***                                                SYS02930
   35 VERB=.TRUE.                                                       SYS02940
      PRINT 36                                                          SYS02950
   36 FORMAT(' VERBOSE MODE')                                           SYS02960
      GOTO 15                                                           SYS02970
C *** BRIEF OPTION ***                                                  SYS02980
C *** GO BRIEF                                                          SYS02990
   40 VERB=.FALSE.                                                      SYS03000
      PRINT 41                                                          SYS03010
   41 FORMAT(' BRIEF MODE')                                             SYS03020
      GOTO 15                                                           SYS03030
C *** DISPLAY OPTION *** (USES CHANGE SUBROUTINES)                      SYS03040
   45 GOTO(26,27,28,26,26,26,26,26,26,26),INX                           SYS03050
C *** MANUAL OPTION ***                                                 SYS03060
   50 CONTINUE                                                          SYS03070
      AUTO=.FALSE.                                                      SYS03080
      GOTO 15                                                           SYS03090
C *** AUTO OPTION ***                                                   SYS03100
   55 CONTINUE                                                          SYS03110
      AUTO=.TRUE.                                                       SYS03120
      GOTO 15                                                           SYS03130
C *** RUN OPTION ***                                                    SYS03140
   60 CONTINUE                                                          SYS03150
      IF(INX.EQ.2.OR.INX.EQ.3)CALL THREAT                               SYS03160
      RETURN                                                            SYS03170
C *** SET OR SEE OPTION ***                                             SYS03180
   70 CALL SET(NFNC,L1,LIST1)                                           SYS03190
      GOTO 15                                                           SYS03200
C *** VARY OPTION ***                                                   SYS03210
   75 CONTINUE                                                          SYS03220
      IPAR=LIST1(1)                                                     SYS03230
      CALL VARY                                                         SYS03240
```

```
      GO TO 15
C *** RESET OPTION ***
   80 GOTO(81,84,82,82,82,82,82,82,82,82),INX
C *** NO RESET IN EXERCISE 3
   81 PRINT 83
   83 FORMAT(/' INVALID FIELD - RES'/)
      NERR=NERR+1
      GOTO 15
   82 CONTINUE
      IF(NREP.NE.16)CALL CHA1(NFNC,L1,LIST1)
      IF(INX.NE.3)GOTO 15
   84 CONTINUE
      IF(NREP.NE.3) CALL CHA2(NFNC,L1,LIST1)
      IF(CONFIG.EQ.V(2)) GO TO 15
      CONFIG=V(2)
      CALL CONFG(CONFIG)
      GOTO 15
      END


      SUBROUTINE XTRACT(IN,NN,LV,LVAR,NL,LIST,NFNC,NREP,*)
C     ***********************************************************
C     ***********************************************************
C *** SUBROUTINE TO ANALYZE USERS COMMAND MESSAGE
      COMMON REPORT,NX,INX
      COMMON V(4500),VERB,AUTO,NPCSFM,NPCSVR,NPCSED,REPSW
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC
      DIMENSION LETA(35),LETB(35),LETC(35)
      DIMENSION LIST(10),LVAR(50)
      LOGICAL ALPHA,FOUND,SUM,THR,TOT
      LOGICAL YEAR
      DIMENSION IN(72),NDIG(10)
      INTEGER CH1SW,CH2SW
      DATA NDIG/'0','1','2','3','4','5','6','7','8','9'/
      DATA ISP/' '/,ICM/','/
      DATA LETA/1HB,1HB,1HT,1HY,1HA,1HR,1HC,1HS,1HV,1HB,
     1          1HD,1HM,1HA,1HE,1HS,1HS,1HV,1HR,1HB,1HA,
     1          1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
     1          1HL,1HB,1HB,1HB,1HT/
      DATA LETB/1HL,1HL,1HH,1HE,1HB,1HE,1HH,1HT,1HE,1HR,
     1          1HI,1HA,1HU,1HX,1HE,1HE,1HA,1HE,1HL,1H ,
     1          1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     1          1H ,1HL,1HL,1HL,1H /
      DATA LETC/1HK,1HK,1HR,1HA,1HC,1HP,1HA,1HC,1HR,1HI,
     1          1HS,1HN,1HT,1HE,1HT,1HE,1HR,1HS,1HK,1H ,
     1         1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     1         1H ,1HK,1HK,1HK,1H /
C *** INITIALIZE ***
```

```
    2 ALPHA=.FALSE.                                                    SYS03710
      FOUND=.FALSE.                                                    SYS03720
      THR=.FALSE.                                                      SYS03730
      YEAR=.FALSE.                                                     SYS03740
      NL=0                                                             SYS03750
      LV=0                                                             SYS03760
      NVAL=0                                                           SYS03770
C *** N IS SIZE OF LETA,LETB,LETC ARRAYS                               SYS03780
      N=35                                                             SYS03790
      NSP=0                                                            SYS03800
      NFNC=0                                                           SYS03810
      NREP=0                                                           SYS03820
    5 CONTINUE                                                         SYS03830
   10 FORMAT(72A1)                                                     SYS03840
C *** REPEAT LOOP FOR EACH INPUT CHARACTER                              SYS03850
      DO 200 I=1,NN                                                    SYS03860
C *** CHECK FOR SPACE                                                  SYS03870
      IF(IN(I).NE.ISP) GOTO 15                                         SYS03880
      NSP=NSP+1                                                        SYS03890
      IF(NSP.GT.5) GOTO 300                                            SYS03900
      IF(.NOT.FOUND) GOTO 200                                          SYS03910
      GOTO 50                                                          SYS03920
C *** CHECK FOR COMMA                                                  SYS03930
   15 IF(IN(I).NE.ICM) GOTO 20                                         SYS03940
      GOTO 50                                                          SYS03950
   20 NSP=0                                                            SYS03960
C *** IS CHARACTER NUMERIC                                             SYS03970
      DO 25 J=1,10                                                     SYS03980
      IF(IN(I).NE.NDIG(J)) GOTO 25                                     SYS03990
C *** YES ,ARE WE SAVING ALPHA FIELD                                   SYS04000
      IF(ALPHA) GOTO 200                                               SYS04010
C *** NO, IS IT FIRST CHAR IN FIELD                                    SYS04020
      IF(.NOT.FOUND)IFST=I                                             SYS04030
      FOUND=.TRUE.                                                     SYS04040
C *** ADD VALUE INTO NVAL                                              SYS04050
      NVAL=NVAL*10+J-1                                                 SYS04060
      GOTO 200                                                         SYS04070
   25 CONTINUE                                                         SYS04080
C *** CHAR MUST BE ALPHA                                               SYS04090
C *** IF FIRST IN FIELD INITIALIZE POINTERS                            SYS04100
   30 IF(FOUND) GOTO 35                                                SYS04110
      FOUND=.TRUE.                                                     SYS04120
      ALPHA=.TRUE.                                                     SYS04130
      IFST=I                                                           SYS04140
      GOTO 200                                                         SYS04150
C *** IF NOT FIRST CHAR FIELD MUST BE TYPE ALPHA                       SYS04160
   35 IF(.NOT.ALPHA) GOTO 151                                          SYS04170
      GOTO 200                                                         SYS04180
```

```
C *** END OF FIELD ***                                                    SYS04190
C *** IF ALPHA GO IDENTIFY,IF NUMERIC SAVE IN LIST                         SYS04200
   50 IF(ALPHA) GOTO 100                                                  SYS04210
C *** NUMERIC FIELD ***                                                    SYS04220
C *** DECIDE WHICH LIST TO SAVE IT IN                                      SYS04230
C *** YEAR=.TRUE. MEANS SECOND LIST                                        SYS04240
      IF(.NOT.YEAR) GOTO 80                                               SYS04250
C *** SAVE IN SECOND LIST                                                  SYS04260
C *** IF THR IS SET INSERT INCLUSIVE VALUES                                SYS04270
      IF(.NOT.THR) GOTO 70                                                SYS04280
      K=LIST(NL)+1                                                        SYS04290
      IF(K.EQ.1)GOTO 151                                                  SYS04300
      IF(NVAL.LT.K) GOTO 151                                              SYS04310
      DO 60 J=K,NVAL                                                      SYS04320
      NL=NL+1                                                             SYS04330
      LIST(NL)=J                                                          SYS04340
   60 CONTINUE                                                            SYS04350
      THR=.FALSE.                                                         SYS04360
      GOTO 190                                                            SYS04370
C *** THR IS NOT SET, SAVE SINGLE VALUE                                    SYS04380
   70 NL=NL+1                                                             SYS04390
      LIST(NL)=NVAL                                                       SYS04400
      GOTO 190                                                            SYS04410
C *** SAVE IN FIRST LIST                                                   SYS04420
C *** IF THR IS SET INSERT INCLUSIVE VALUES                                SYS04430
   80 IF(.NOT.THR) GOTO 85                                                SYS04440
      K=LVAR(LV)+1                                                        SYS04450
      IF(K.EQ.1)GOTO 151                                                  SYS04460
      IF(NVAL.LT.K) GOTO 151                                              SYS04470
      DO 83 J=K,NVAL                                                      SYS04480
      IF(LV.EQ.50) GOTO 300                                               SYS04490
      LV=LV+1                                                             SYS04500
   83 LVAR(LV)=J                                                          SYS04510
      THR=.FALSE.                                                         SYS04520
      GOTO 190                                                            SYS04530
C *** THR IS NOT SET,SAVE SINGLE VALUE                                     SYS04540
   85 LV=LV+1                                                             SYS04550
      LVAR(LV)=NVAL                                                       SYS04560
      IF(LV.EQ.50) GOTO 300                                               SYS04570
      GOTO 190                                                            SYS04580
C *** FIELD IS ALPHA ***                                                   SYS04590
C *** IDENTIFY                                                             SYS04600
  100 DO 150 J=1,N                                                        SYS04610
      IF(IN(IFST).NE.LETA(J)) GOTO 150                                    SYS04620
      IF(LETB(J).EQ.ISP.AND.I.EQ.IFST+1)GOTO 99                           SYS04630
      IF(IN(IFST+1).NE.LETB(J)) GOTO 150                                  SYS04640
      IF(IN(IFST+2).NE.LETC(J)) GOTO 150                                  SYS04650
   99 CONTINUE                                                            SYS04660
```

```
      II=J
      IF(J.GT.5.AND.J.LE.19) II=6
      IF(J.GT.19) II=7
      GOTO(101,102,103,104,105,106,107),II
  101 CONTINUE
      GOTO 190
  102 CONTINUE
      GOTO 190
  103 THR=.TRUE.
      GOTO 190
  104 CONTINUE
      THR=.FALSE.
      GOTO 190
  105 PRINT 1105
 1105 FORMAT(' LINE ABORTED')
      RETURN 1
C *** SET OPTION INDICATOR ***
  106 NFNC=J-5
      GOTO 190
C *** SET REPORT INDICATOR ***
  107 NREP=J-19
      IF(J.EQ.26)YEAR=.FALSE.
      GOTO 190
  150 CONTINUE
  151 PRINT 152,IN(IFST),IN(IFST+1),IN(IFST+2)
  152 FORMAT(' INVALID FIELD- ',3A1)
      NERR=NERR+1
      RETURN 1
  190 ALPHA=.FALSE.
      FOUND=.FALSE.
      NVAL=0
  200 CONTINUE
  300 RETURN
      END


      SUBROUTINE REPRT(N)
C     ******************
C     ******************
      COMMON REPORT,NX,INX
      COMMON V(4500),VERB,AUTO,NPCSFM,NPCSVR,NPCSEQ,REPSW
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC
      COMMON LIN
      DIMENSION FM(16),NVARS(16)
      INTEGER REPORT,FILE,CH1SW,CH2SW
      LOGICAL VERB,AUTO
      CALL DEFINE (9,'FILE    ','FT09F001',LIN,80)
      DATA END/3HEND/
```

```
C *** TEST REPORT NUMBER FOR VALIDITY
      IF(N.GT.0.AND.N.LT.21)GOTO 1
      PRINT 61
      RETURN
C ***GET RECORD CONTAINING POSITION OF FORMATS AND VARIABLES
    1 IREK=NPOSVR+N-1
      LIN=IREK
      READ(REPORT) NVARS
      IR=NVARS(1)
      IV=NVARS(2)
C *** INITIALIZE POINTER TO FORMAT RECORD
      IF(IR.LT.1) GOTO 60
      NF=IR+NPOSFM-1
C *** INITIALIZE POINTER TO VARIABLES RECORD
      LV=IV+NPOSVR-1
C *** READ FIRST VARIABLES RECORD
      LIN=LV
    5 READ(REPORT) NVARS
C *** GET POSITION OF RECORD IN REPORT FILE
      LINE=NVARS(1)+NPOSFM-1
      GOTO 7
    6 NF=NF+1
C *** READ NEXT FORMAT RECORD
    7 LIN=NF
      READ(REPORT) FM
C *** IS THIS END OF REPORT
      IF(FM(1).EQ.END)GOTO 50
C *** ARE THERE VARIABLES WITH THIS FORMAT
      IF(LINE.EQ.NF) GOTO 10
      PRINT FM
      GOTO 6
   10 NC=NVARS(2)+2
      PRINT FM,(V(NVARS(J)),J=3,NC)
      LV=LV+1
C *** READ NEXT VARIABLES RECORD
      LIN=LV
      READ(REPORT) NVARS
C *** GET POSITION OF NEXT RECORD IN REPORT FILE
      LINE=NVARS(1)+NPOSFM-1
      IF(LINE.NE.NF) GOTO 6
      GOTO 10
   50 CONTINUE
      RETURN
C *** ERROR
   60 PRINT 61
   61 FORMAT(/' REPORT UNKNOWN'/)
      GOTO 50
      END
```

```
      SUBROUTINE CHAI(N,L,LIS)                                                  SYS05610
C     ************************                                                  SYS05620
C     ************************                                                  SYS05630
      COMMON REPORT,NX,INX                                                      SYS05640
      COMMON V(4500),VERB,AUTO,NPCSFM,NPCSVR,NPCSED,REPSW                       SYS05650
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC                              SYS05660
      COMMON LIN                                                                SYS05670
      DIMENSION RANGE(2),F(10),FDUM(1),IDUM(1)                                  SYS05680
      DIMENSION LIS(50)                                                         SYS05690
      INTEGER REPORT,FILE,CH1SW,CH2SW                                           SYS05700
      LOGICAL VERB,AUTO                                                         SYS05710
      DATA END/3HEND/                                                           SYS05720
      IVSTT=4200                                                                SYS05730
      IDSTT=4250                                                                SYS05740
      IF (CH1SW.EQ.3) GOTO 5                                                    SYS05750
    1 FORMAT(//' THE STUDENT MAY VARY THE FOLLOWING SYSTEM PARAMETERS')         SYS05760
      NUM=1                                                                     SYS05770
    2 NREK=NUM+NPCSED-1                                                         SYS05780
      LIN=NREK                                                                  SYS05790
      READ(REPORT) F,RANGE,ID                                                   SYS05800
      IF(F(1).EQ.END)GOTO 3                                                     SYS05810
      V(NUM+IVSTT)=V(ID)                                                        SYS05820
      V(NUM+IDSTT)=ID                                                           SYS05830
      NUM=NUM+1                                                                 SYS05840
      GOTO 2                                                                    SYS05850
    3 NUM=NUM-1                                                                 SYS05860
      V(4200)=NUM                                                               SYS05870
      IF(INX.EQ.5)PRINT 70                                                      SYS05880
      IF (CH1SW.EQ.2) GOTO 50                                                   SYS05890
      IF(INX.NE.5)PRINT 1                                                       SYS05900
   70 FORMAT(//' ENTER VALUES USING THE FOLLOWING MAINTENANCE CATEG',           SYS05910
     1 'ORY  CODES'/' ( 0. =NOT REQUIRED)'/' ( 1. =ORGANIZATION)'               SYS05920
     2  /' ( 2. =INTERMEDIATE)'/' ( 3. =DEPOT        )' )                       SYS05930
      N=6                                                                       SYS05940
      L=NUM                                                                     SYS05950
      DO 4 I=1,NUM                                                              SYS05960
    4 LIS(I)=I                                                                  SYS05970
      GOTO 10                                                                   SYS05980
    5 CONTINUE                                                                  SYS05990
      NUM=V(4200)                                                               SYS06000
      IF(N.EQ.13)GOTO 6                                                         SYS06010
      IF(.NOT.AUTO)GOTO 10                                                      SYS06020
      IF(N.EQ.6)GOTO 10                                                         SYS06030
    6 DO 7 I=1,NUM                                                              SYS06040
      SAV=V(I+IVSTT)                                                            SYS06050
      NED=V(I+IDSTT)                                                            SYS06060
```

```
    7 V(NED)=SAV                                                          SYS06070
      PRINT 8                                                             SYS06080
    8 FORMAT(/' SYSTEM PARAMETERS RESET'/)                                SYS06090
      IF(N.EQ.13) GOTO 50                                                 SYS06100
   10 IF(L.EQ.0) GOTO 60                                                  SYS06110
      IF (CHISW.EQ.1) GOTO 101                                            SYS06120
      IF(V(1).EQ.3.0.AND.N.NE.6)L=1                                       SYS06130
  101 CONTINUE                                                            SYS06140
      DO 40 I=1,L                                                         SYS06150
      NC=LIS(I)                                                           SYS06160
      NODUM=NC                                                            SYS06170
      IF(V(1).EQ.29..AND.NO.GT.1)NODUM=2                                  SYS06180
      IF(V(1).NE.5..AND.CHISW.EQ.1)PRINT 11,NODUM                         SYS06190
      IF(V(1).EQ.5..AND.CHISW.EQ.1)PRINT 13,NO                            SYS06200
      IF (CHISW.EQ.3) PRINT 12                                            SYS06210
   12 FORMAT(/)                                                           SYS06220
      IF(NC.LT.1.OR.NO.GT.NUM) GOTO 60                                    SYS06230
   11 FORMAT(//' PARAMETER ',I3)                                          SYS06240
   13 FORMAT(//' PARAMETER  C ',I3)                                       SYS06250
      NAB=V(NC+IDSTT)                                                     SYS06260
C *** GET EDIT INFO.                                                       SYS06270
      NREK=NC+NPCSED-1                                                    SYS06280
      IF(CHISW.NE.3)GOTO 14                                               SYS06290
      NED=V(IDSTT+1)                                                      SYS06300
      IF(INX.EQ.10.AND.NC.EQ.2)NREK=NO+V(NED)+NPCSED-2                    SYS06310
   14 IF(INX.EQ.10.AND.NC.GT.1)NO=2                                       SYS06320
      LIN=NREK                                                            SYS06330
      READ(REPORT) F,RANGE                                                SYS06340
      IF(.NOT.VERB)GOTO 18                                                SYS06350
      PRINT F                                                             SYS06360
      PRINT 15,RANGE,NO,V(NAB)                                            SYS06370
   15 FORMAT(' MIN=',F9.3,'  MAX=',F9.3,'  PRESENT VALUE(',I2,            SYS06380
     1       ')=',F9.3)                                                   SYS06390
      GOTO 20                                                             SYS06400
   18 PRINT 19,NO,V(NAB)                                                  SYS06410
   19 FORMAT(' PRESENT VALUE(',I2,')',F9.3)                               SYS06420
C *** IF CHANGE REQUEST NEW VALUE                                          SYS06430
   20 IF(N.EQ.6)GOTO 40                                                   SYS06440
      PRINT 22                                                            SYS06450
   22 FORMAT(' ENTER NEW VALUE')                                          SYS06460
      READ(5,24)VAL                                                       SYS06470
   24 FORMAT(F15.3)                                                       SYS06480
      IF(VAL.GE.RANGE(1).AND.VAL.LE.RANGE(2))GOTO 30                      SYS06490
      IF(VERB)PRINT 26                                                    SYS06500
      IF(.NOT.VERB)PRINT 27,RANGE                                         SYS06510
   26 FORMAT(' VALUE OUTSIDE RANGE. RETRY.')                              SYS06520
   27 FORMAT(' PERMITTED RANGE OF VALUE IS ',F10.2,' TO ',F10.2)          SYS06530
      NERR=NERR+1                                                         SYS06540
```

```
      GOTO 50                                                           SYS06550
   30 V(NAB)=VAL                                                        SYS06560
      IF(INX.NE.10)GOTO 40                                              SYS06570
      IF(NC.NE.1)GOTO 40                                                SYS06580
      NED=V(IDSTT+2)                                                    SYS06590
      SAV=V(IVSTT+2)                                                    SYS06600
      V(NED)=SAV                                                        SYS06610
      PRINT 32                                                          SYS06620
   32 FORMAT(/' MAXIMUM RANGE RESET')                                   SYS06630
   40 CONTINUE                                                          SYS06640
   50 CHISW=3                                                           SYS06650
      RETURN                                                            SYS06660
   60 PRINT 61,NUM                                                      SYS06670
   61 FORMAT(' VALID ITEMS ARE 1 THRU ',I3/)                            SYS06680
      NERR=NERR+1                                                       SYS06690
      GOTO 50                                                           SYS06700
      END                                                               SYS06710


      SUBROUTINE SET(N,L,LIS)                                           SYS06720
C     ***********************                                           SYS06730
C     ***********************                                           SYS06740
      COMMON REPORT,NX,INX                                              SYS06750
      COMMON V(4500),VERB,AUTC,NPCSFM,NPCSVR,NPCSED,REPSW               SYS06760
     1,NERR,IVARCN,IPAR,AMIN,AMAX,AINC                                  SYS06770
      DIMENSION LIS(50)                                                 SYS06780
      IF(L.EQ.0) RETURN                                                 SYS06790
      IF(L.GT.50)L=50                                                   SYS06800
      DO 30  I=1,L                                                      SYS06810
      NO=LIS(I)                                                         SYS06820
      IF(NO.LT.1.OR.NO.GT.4500)GOTO 30                                  SYS06830
      PRINT 5,NO,V(NO)                                                  SYS06840
    5 FORMAT(' ITEM(',I4,')=',F15.3)                                    SYS06850
      IF(N.EQ.11) GOTO 30                                               SYS06860
      PRINT 6                                                           SYS06870
    6 FORMAT(' NEW VALUE')                                              SYS06880
      READ(5,7)VAL                                                      SYS06890
      V(NO)=VAL                                                         SYS06900
    7 FORMAT(F12.3)                                                     SYS06910
   30 CONTINUE                                                          SYS06920
      RETURN                                                            SYS06930
      END                                                               SYS06940
```

```
      SUBROUTINE CHA2(N,L,LIS)                                          SYS06950
C     ************************                                          SYS06960
C     ************************                                          SYS06970
      COMMON REPORT,NX,INX                                              SYS06980
      COMMON V(4500),VERB,AUTO,NPCSFM,NPCSVR,NPCSED,REPSW.              SYS06990
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC                      SYS07000
      COMMON LIN                                                        SYS07010
      DIMENSION RANGE(2),F(10),FDUM(1),IDUM(1)                          SYS07020
      DIMENSION LIS(16)                                                 SYS07030
      INTEGER REPORT,FILE,CH1SW,CH2SW                                   SYS07040
      LOGICAL VERB,AUTO                                                 SYS07050
      DATA END/3HEND/                                                   SYS07060
      NUM=V(4220)                                                       SYS07070
      IVSTT=4300                                                        SYS07080
      IDSTT=4320                                                        SYS07090
  543 FORMAT(2F10.2)                                                    SYS07100
      IF (CH2SW.EQ.3) GOTO 5                                            SYS07110
    1 FORMAT(//' THE STUDENT MAY VARY THE FOLLOWING THREAT PARAMETERS') SYS07120
      NUM=1                                                             SYS07130
C *** READ RECORDS FOLLOWING THE 9 FOR CONFIG HANDLED BY CHA1           SYS07140
    2 NREK=NUM+NPCSED-1 +10                                             SYS07150
      LIN=NREK                                                          SYS07160
      READ(REPORT) F,RANGE,ID                                           SYS07170
      IF(F(1).EQ.END)GOTO 3                                             SYS07180
      V(NUM+IVSTT)=V(ID)                                                SYS07190
      V(NUM+IDSTT)=ID                                                   SYS07200
      NUM=NUM+1                                                         SYS07210
      GOTO 2                                                            SYS07220
    3 NUM=NUM-1                                                         SYS07230
  234 FORMAT(F10.2)                                                     SYS07240
      V(4300)=NUM                                                       SYS07250
      IF(CH2SW.EQ.2)GOTO 50                                             SYS07260
      PRINT 1                                                           SYS07270
      N=6                                                               SYS07280
      L=NUM                                                             SYS07290
      DO 4 I=1,NUM                                                      SYS07300
    4 LIS(I)=I                                                          SYS07310
      V(3)=1.0                                                          SYS07320
      GOTO 10                                                           SYS07330
    5 NUM=V(4300)                                                       SYS07340
      IF(N.EQ.13) GOTO 6                                                SYS07350
      IF(.NOT.AUTO)GOTO 10                                              SYS07360
      IF(N.EQ.6)GOTO 10                                                 SYS07370
    6 DO 7 I=1,NUM                                                      SYS07380
      SAV=V(I+IVSTT)                                                    SYS07390
      NED=V(I+IDSTT)                                                    SYS07400
    7 V(NED)=SAV                                                        SYS07410
C *** RESET THREAT INDICATOR                                            SYS07420
```

```
      V(3)=1.0                                                          SYS07430
      PRINT 8                                                           SYS07440
    8 FORMAT(/' THREAT PARAMETERS RESET'/)                              SYS07450
      IF(N.EQ.13)GOTO 50                                                SYS07460
   10 IF(L.EQ.0)GOTO 60                                                 SYS07470
      DO 40 I=1,L                                                       SYS07480
      NO=LIS(I)                                                         SYS07490
      IF (CH2SW.EQ.3) GOTO 110                                          SYS07500
      IF(V(1).EQ.4.0)PRINT 11,NO                                        SYS07510
      IF(V(1).EQ.5.0.AND.NO.NE.1)PRINT 13,NO                            SYS07520
  110 CONTINUE                                                          SYS07530
      IF(NO.LT.1.OR.NO.GT.16) GOTO 60                                   SYS07540
   11 FORMAT(//' PARAMETER ',I3)                                        SYS07550
   13 FORMAT(//' PARAMETER  T ',I3)                                     SYS07560
C *** CHECK CHANGES FOR CONSISTENCE WITH THREAT                         SYS07570
      IF (CH2SW.EQ.1) GOTO 17                                           SYS07580
      IF(NO.EQ.1.AND.V(1).EQ.5.0)GOTO 70                                SYS07590
      IF(N.EQ.6) GOTO 17                                                SYS07600
C *** GET PRESENT THREAT INDICATOR                                      SYS07610
      NTH=V(3)                                                          SYS07620
      GOTO(12,14,16),NTH                                                SYS07630
C *** THREAT IS NOM - CHANGE TO 2 OR 3 DEP. ON GROUP                    SYS07640
   12 IF(NO.GT.1.AND.NO.LT.12) V(3)=2.0                                 SYS07650
      IF(NO.GT.11)V(3)=3.0                                              SYS07660
      GOTO 17                                                           SYS07670
C *** RESTRICT CHANGES TO 2 THRU 11                                     SYS07680
   14 IF(NO.NE.1.AND.NO.GT.11)GOTO 65                                   SYS07690
      GOTO 17                                                           SYS07700
C *** RESTRICT CHANGES TO 12 THR 16                                     SYS07710
   16 IF(NO.NE.1.AND.NO.LT.12) GOTO 65                                  SYS07720
   17 CONTINUE                                                          SYS07730
      IF(NO.EQ.1.AND.V(1).EQ.5.0)GOTO 40                                SYS07740
      NAB=V(NO+ICSTT)                                                   SYS07750
C *** GET EDIT INFO.                                                    SYS07760
      NREK=NO+NPUSED-1+10                                               SYS07770
      LIN=NREK                                                          SYS07780
      READ(REPORT) F,RANGE                                              SYS07790
      IF(.NOT.VERB)GOTO 18                                              SYS07800
      PRINT F                                                           SYS07810
      PRINT 15,RANGE,NO,V(NAB)                                          SYS07820
   15 FORMAT(' MIN=',F9.3,'  MAX=',F9.3,'  PRESENT VALUE(',I2,          SYS07830
     1          ')=',F9.3)                                              SYS07840
      GOTO 20                                                           SYS07850
   18 PRINT 19,NO,V(NAB)                                                SYS07860
   19 FORMAT(' PRESENT VALUE(',I2,')',F9.3)                             SYS07870
C *** IF CHANGE REQUEST NEW VALUE                                       SYS07880
   20 IF(N.EQ.6)GOTO 40                                                 SYS07890
      PRINT 22                                                          SYS07900
```

```
22 FORMAT(' ENTER NEW VALUE')
   READ(5,24)VAL
24 FORMAT(F15.3)
   IF(VAL.GE.RANGE(1).AND.VAL.LE.RANGE(2))GOTO 30
   IF(VERB)PRINT 26
   IF(.NOT.VERB)PRINT 27,RANGE
26 FORMAT(' VALUE OUTSIDE RANGE. RETRY.')
27 FORMAT(' PERMITTED RANGE OF VALUE IS ',F10.2,' TO ',F10.2)
   NERR=NERR+1
   GOTO 50
30 V(NAB)=VAL
40 CONTINUE
50 CH2SW=3
55 RETURN
60 IF(V(1).EQ.5.0)PRINT 61,NUM
61 FORMAT(' VALID ITEMS ARE 2 THRU',I4/)
   IF(V(1).EQ.4.0) PRINT 62,NUM
62 FORMAT(' VALID ITEMS ARE 1 THRU',I4/)
   NERR=NERR+1
   GOTO 50
65 PRINT 66
66 FORMAT(' CHANGES INCONSISTENT. CHANGE 2 THRU 11, OR 12 THRU 16
  1 '/' USE RESET BETWEEN GROUPS'//)
   NERR=NERR+1
   GOTO 50
70 PRINT 71
71 FORMAT(/' USE SYSTEM PARAMETERS TO CHANGE CONFIGURATION')
   NERR=NERR+1
   GOTO 50
   END


   SUBROUTINE CONFG(C)
C  ******************
C  ******************
   COMMON REPORT,NX,INX
   COMMON V(4500),VERB,AUTO,NPCSFM,NPCSVR,NPCSED,REPSW
  1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC
   INTEGER CH1SW,CH2SW
   DIMENSION NPOS(9)
   DIMENSION NSTT(4)
   DATA NPCS/60,109,108,107,59,57,308,309,310/
   DATA NSTT/3770,3780,3790,3800/
   N=C
   IF(N.GT.0.AND.N.LT.4)GOTO 5
   NERR=NERR+1
   PRINT 4
 4 FORMAT(/' INVALID CONFIGURATION - VALUE SET TO 1.0'/)
```

```
      N=1                                                               SYS08370
      C=1.0                                                             SYS08380
    5 CONTINUE                                                          SYS08390
      NSTART=NSTT(N)                                                    SYS08400
      GOTO(6,6,6,35,35,6,35,35,35,35),INX                               SYS08410
    6 GOTO(10,20,30),N                                                  SYS08420
   10 CONTINUE                                                          SYS08430
      PRINT 11                                                          SYS08440
   11 FORMAT(/' CONFIGURATION 1.'/)                                     SYS08450
      GOTO 35                                                           SYS08460
   20 CONTINUE                                                          SYS08470
      PRINT 21                                                          SYS08480
   21 FORMAT(/' CONFIGURATION 2.'/)                                     SYS08490
      GOTO 35                                                           SYS08500
   30 CONTINUE                                                          SYS08510
      PRINT 31                                                          SYS08520
   31 FORMAT(/' CONFIGURATION 3.'/)                                     SYS08530
   35 DO 40 I=1,9                                                       SYS08540
   40 V(NPCS(I))=V(NSTART+I)                                            SYS08550
      RETURN                                                            SYS08560
      END                                                               SYS08570


      SUBROUTINE VARY                                                   SYS08580
C     ***************                                                   SYS08590
C     ***************                                                   SYS08600
      COMMON REPORT,NX,INX                                              SYS08610
      COMMON V(4500),VERB,AUTC,NPCSFM,NPCSVR,NPCSED,REPSW               SYS08620
     1,CH1SW,CH2SW,NERR,IVARCN,IPAR,AMIN,AMAX,AINC                      SYS08630
      COMMON LIN                                                        SYS08640
      INTEGER REPORT,FILE,CH1SW,CH2SW                                   SYS08650
      LOGICAL ALTO                                                      SYS08660
      DIMENSION LINE(72),LIS(50),LDUM(1),RANGE(2),F(10)                 SYS08670
    1 FORMAT(F15.3)                                                     SYS08680
    2 FORMAT(/,' SYSTEM PARAMETERS RESET ',//)                          SYS08690
    3 FORMAT(/,' INVALID PARAMETER, RETRY',/)                           SYS08700
    4 FORMAT(/,' MIN=',F9.3,' MAX=',F9.3,' PRESENT VALUE (',I2,         SYS08710
     1        ') =',F9.3)                                               SYS08720
    5 FORMAT(/,' ENTER INITIAL VALUE FOR PARAMETER ',I2,/)              SYS08730
    6 FORMAT(/,' VALUE OUTSIDE RANGE, RETRY',/,' PERMITTED RANGE IS ',  SYS08740
     1        F10.2,'  TO ',F10.2,/)                                    SYS08750
    7 FORMAT(/,' ENTER FINAL VALUE FOR PARAMETER ',I2,/)                SYS08760
    8 FORMAT(/,' ENTER INCREMENT (MAX OF 10 ITERATIONS)',/)             SYS08770
    9 FORMAT(/,' ERROR, MAX VALUE LESS THAN MIN, RETRY',/)              SYS08780
   10 FORMAT(                                                           SYS08790
     1    /,50(1H-),///,2X,' PARAMETER ',I2,' VALUE',5X,                SYS08800
     1    ' LIFE CYCLE COST',4X,' DISCOUNTED LIFE CYCLE COST',//)       SYS08810
   11 FORMAT(/,F15.3,10X,4X,F10.0,10X,4X,F10.0)                         SYS08820
```

```
12 FORMAT(//,' EXERCISE ',F3.0)
13 FORMAT(/,' SENSITIVITY ANALYSIS.' /1X,43(1H-)/)
14 FORMAT(////,6X,' COST VS ')
15 FORMAT(/,' VARY COMMAND USED ONLY IN EXERCISE 3',/)
   IDSTT=4250
   IVSTT=4200
   IF(V(1).EQ.3.)GOTO 70
   PRINT 15
   GO TO 1000
70 CONTINUE
   ALPAR = V(IPAR + IDSTT)
   IF (IVARCN.NE.0) GO TO 200
   NUM = V(4200)
   IF(.NOT.AUTO) GO TO 100
80 DO 90 I=1,NUM
   SAV = V(I+IVSTT)
   NED= V(I+IDSTT)
90 V(NED) = SAV
   PRINT 2
100 CONTINUE
   IF((IPAR.GE.1).AND.(IPAR.LE.NUM)) GO TO 110
   PRINT 3
   GO TO 1000
110 NAB=V(IPAR+IDSTT)
   NREK=IPAR+NPCSED - 1
   LIN=NREK
   READ (REPORT) F,RANGE
   PRINT F
120 PRINT 4,RANGE,IPAR,V(NAB)
130 PRINT 5,IPAR
   READ(5,1)AMIN
   IF ((AMIN.GE.RANGE(1)).AND.(AMIN.LE.RANGE(2))) GO TO 140
   PRINT 6,RANGE
   GO TO 130
140 PRINT 7,IPAR
   READ(5,1)AMAX
   IF((AMIN.GE.RANGE(1)).AND.(AMAX.LE.RANGE(2))) GO TO 150
   PRINT 6,RANGE
   GO TO 140
150 CONTINUE
   IF(AMIN.LE.AMAX) GO TO 160
   PRINT 9
   GO TO 120
160 PRINT 8
   READ(5,1)AINC
   PRINT 12,V(1)
   PRINT 13
   PRINT 14
```

```
      PRINT F                                                           SYS09310
      PRINT 10,IPAR                                                     SYS09320
      GO TO 210                                                         SYS09330
  200 CONTINUE                                                          SYS09340
      PRINT 11,V(ADPAR),V(1534),V(1535)                                 SYS09350
      IF(IVARCN.GT.10) GO TO 1000                                       SYS09360
  210 CONTINUE                                                          SYS09370
      AM = AMIN + (IVARCN * AINC)                                       SYS09380
      IVARCN = IVARCN + 1                                               SYS09390
      IF (AM.GT.AMAX) GO TO 1000                                        SYS09400
      V(ADPAR)=AM                                                       SYS09410
      GOTO 200                                                          SYS09420
 1000 IVARCN=0                                                          SYS09430
      RETURN                                                            SYS09440
      END                                                               SYS09450

      SUBROUTINE THREAT                                                 SYS09460
C     *****************                                                 SYS09470
C     *****************                                                 SYS09480
      COMMON REPORT,NX,INX                                              SYS09490
      COMMON V(4500)                                                    SYS09500
      DIMENSION TD(10),CEPI(8),FAC(8)                                   SYS09510
      DATA TD/300.,300.,20.,60.,200.,200.,180.,30.,100.,130./           SYS09520
      DATA CEPI/33.3,50.,66.6,83.25,100.,116.56,133.2,150./             SYS09530
      DATA FAC/0.,.04,.08,.15,.25,.30,.34,.38/                          SYS09540
      DO 5 I=1,10                                                       SYS09550
    5 V(1675+I)=TD(I)                                                   SYS09560
      CEP=1.1774*(((V(308)**2)+(V(309)**2))**.5)                        SYS09570
      IF=0                                                              SYS09580
      IS=0                                                              SYS09590
      FA=1.                                                             SYS09600
      DO 10 I=1,8                                                       SYS09610
      IF(CEP.LE.CEPI(I)) GO TO 20                                       SYS09620
   10 IS=I                                                              SYS09630
      IF=IS                                                             SYS09640
      GO TO 25                                                          SYS09650
   20 IF=I                                                              SYS09660
      IF(IS.EQ.0) IS=1                                                  SYS09670
   25 DELT=CEPI(IF)-CEPI(IS)                                            SYS09680
      DELV=CEP-CEPI(IS)                                                 SYS09690
      IF(DELT.LE..0) GO TO 30                                           SYS09700
      FA=FA+DELV/DELT                                                   SYS09710
   30 TFA=FAC(IS)*FA                                                    SYS09720
      DO 50 I=1,5                                                       SYS09730
      IF(V(2513+I).GE..95) GO TO 50                                     SYS09740
      IF(V(2513+I).GE..80) GO TO 51                                     SYS09750
      IF(TFA.LE..05) TFA=.05                                            SYS09760
```

```
      TEMP=.95-V(2513+I)                                                SYS09770
      IF(TEMP.GT..35) TEMP=.35                                          SYS09780
      TEMP=TEMP-.15                                                     SYS09790
      TEMP=1.+.2*(TEMP/.20)                                             SYS09800
      GO TO 55                                                          SYS09810
   51 TEMP=.95-V(2513+I)                                                SYS09820
      TEMP=TEMP/.15                                                     SYS09830
   55 V(1675+I)=V(1675+I)+V(1675+I)*TFA*TEMP                            SYS09840
   50 CONTINUE                                                          SYS09850
      IF(V(2).EQ.2.) GO TO 60                                           SYS09860
      IF(V(2125).LE. 25.) GO TO 70                                      SYS09870
      TEMP=V(2125)-25.                                                  SYS09880
      IF(TEMP.GT.20.) TEMP=20.                                          SYS09890
      V(1680)=V(1680)+70.*(TEMP/15.)                                    SYS09900
      GOTO 70                                                           SYS09910
   60 V(1685)=200.                                                      SYS09920
   70 RETURN                                                            SYS09930
      END                                                               SYS09940
```

APPENDIX D

COMMAND AND CONTROL PROGRAM (INCENT)

```
C     ******************
C     * INCENT PROGRAM *
C     ******************
      COMMON V(4500),VERB,AUTC,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW
     1     ,NERR,NX
      DIMENSION LIST1(66),LIST2(10),LINE(72)
      LOGICAL VERB,AUTO,CHASW
      DIMENSION YEAR(10)
      DATA YEAR/4H1976,4H1977,4H1978,4H1979,4H1980,
     1          4H1981,4H1982,4H1983,4H1984,4H1985/
      COMMON/CA/LIN
      VERB=.TRUE.
      AUTO=.FALSE.
      CHASW=.TRUE.
      DO 5 I=1,4500
    5 V(I)=I
      DO 6 I=1,10
    6 V(I+40)=YEAR(I)
      CALL MAKER
      CALL DATA(10)
      LIN=1
      READ(9,224)NPOSFM,NPOSVR,NPOSED
  224 FORMAT(16I5)
      PRINT 9998
 9998 FORMAT(' EXERCISE NUMBER? (15. OR 17.)')
      READ 9997,EXNO
 9997 FORMAT(F10.0)
      V(1)=EXNO
      CALL START
      CH1SW=2
   15 CONTINUE
      PRINT 14
   14 FORMAT(/' COMMAND ?')
   17 READ 16, LINE
   16 FORMAT(72A1)
      CALL XTRACT(LINE,72,L1,LIST1,L2,LIST2,NFNC,NREP,&15)
      IF(NFNC.EQ.0)GOTO 17
      GOTO(20,25,30,35,40,45,50,55,65,60,60,15,70),NFNC
   20 IF(NREP.LT.1)GOTO 15
      IF(CHASW) CALL RUN(&15)
      IF(V(1).EQ.17.)GOTO 23
      PRINT 21,V(1)
   21 FORMAT(/' EXERCISE ',F3.0//
```

```
      1 ' STRUCTURING INCENTIVES'
      1 /1X,43('-'))
        GOTO 24
   23 PRINT 22,V(1)
   22 FORMAT(/' EXERCISE ',F3.0//
      1 ' INCENTIVE CONTRACT NEGOTIATION'
      1 /1X,52('-'))
   24 CONTINUE
      CALL REPRT(NREP)
      CHASW=.FALSE.
      GOTO 15
   25 CALL CHANGE(NFNC,L1,LIST1,NREP)
      CHASW = .TRUE.
      GOTO 15
   30 CALL ERASE('FILE    ','FT09F001')
      CALL EXIT
C *** GO VERBOSE
   35 VERB=.TRUE.
      PRINT 36
   36 FORMAT(' VERBOSE MODE')
      GOTO 15
C *** GO BRIEF
   40 VERB=.FALSE.
      PRINT 41
   41 FORMAT(' BRIEF MODE')
      GOTO 15
   45 CALL CHANGE(NFNC,L1,LIST1,NREP)
      GOTO 15
   50 CONTINUE
      ALTC=.FALSE.
      GOTO 15
   55 CONTINUE
      ALTC=.TRUE.
      GOTO 15
C *** SEE OR SET VALUES IN  'V'
   60 CALL SET(NFNC,L1,LIST1)
      GOTO 15
   65 CHASW=.FALSE.
      CALL RUN($15)
      GOTO 15
C *** RESET
   70 CALL RESET
      GOTO 15
      END
```

```
      SUBROUTINE MAKER
C     ****************
C     ****************
      DIMENSION FORM(16),IV(16),FMAT(10),FV(2)
      CALL DEFINE(9,'FILE    ','FT09F001',LIN,80)
      COMMON/DA/LIN
      LIN=2
      NPOSFM=2
    8 CONTINUE
      READ(1,11,END=20)FORM
   11 FORMAT(1X,16A4)
      WRITE(9,11)FORM
      GO TO 8
   20 NPOSVR=LIN
   26 READ(1,24,END=60) IV
   24 FORMAT(    16I5)
      WRITE(9,24)IV
      GOTO 26
   60 NPOSED=LIN
   61 READ(1,62,END=70) FMAT,FV,ID
   62 FORMAT(1X,10A4,2F10.2,I5)
      WRITE(9,62)FMAT,FV,ID
      GO TO 61
   70 LIN=1
      WRITE(9,24)NPOSFM,NPOSVR,NPOSED
      PRINT 71
   71 FORMAT(' CREATED REPORT FILE FT09F001')
      RETURN
      END


      SUBROUTINE START
C     ****************
C     ****************
      COMMON V(4500),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW
     1    ,NERR,NX
      DIMENSION F(10),RANGE(2)
      DATA END/3HEND/
      COMMON/DA/LIN
      ISTRT=4200
      JSTRT=4300
      NUM=1
    2 NREK=NUM+NPOSED-1
      LIN=NREK
      READ(9,62)F,RANGE,ID
   62 FORMAT(1X,10A4,2F10.2,I5)
      IF(F(1).EQ.END) GOTO 3
      V(NUM+ISTRT)=V(ID)
```

```
      V(NUM+JSTRT)=ID                                                   INCO1340
      NUM=NUM+1                                                         INCO1350
      GOTO 2                                                            INCO1360
    3 NUM=NUM-1                                                         INCO1370
      V(ISTRT)=NUM                                                      INCO1380
      PRINT 10                                                          INCO1390
   10 FORMAT(//' CONTRACT DATA.'                                        INCO1400
     1 /' PERFORMANCE AND/OR FEE (P OR F)' )                            INCO1410
      PRINT 11                                                          INCO1420
   11 FORMAT( '        ( 1) TOTAL DEVELOPMENT COST'/                    INCO1430
     1 '          ( 2) RANGE ERROR'/                                    INCO1440
     1 '          ( 3) DEFLECTION ERROR'/                               INCO1450
     1 '          ( 4) AVAILABILITY'/                                   INCO1460
     1 '          ( 5) RELIABILITY'/                                    INCO1470
     1 '          ( 6) MISSILE WEIGHT'/                                 INCO1480
     1 '          ( 7) MAXIMUM RANGE'/                                  INCO1490
     1 '          ( 8) SWIM SPEED'/                                     INCO1500
     1 '          ( 9) CRUISE SPEED'/                                   INCO1510
     1 '          (10) CRUISE RADIUS'/                                  INCO1520
     1 '          (11) SCHEDULE'/                                       INCO1530
     1/' CONTRACTOR DATA (C).'/6X,'( 1) CONTRACTOR STRATEGY(3,4,5)'/    INCO1540
     1 '          ( 2) STRATEGY GIVEN'/                                 INCO1550
     1 '          ( 3) MINIMUM SYSTEM DEVELOPMENT COST'///)             INCO1560
      RETURN                                                            INCO1570
      END                                                               INCO1580


      SUBROUTINE REPRT(N)                                               INCO1590
C     ******************                                                INCO1600
C     ******************                                                INCO1610
      COMMON V(4500),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW         INCO1620
     1  ,NERR,NX                                                        INCO1630
      DIMENSION FM(16),NVARS(16)                                        INCO1640
      DIMENSION IR(20),IV(20)                                           INCO1650
      LOGICAL VERB,AUTO                                                 INCO1660
      DATA END/3HEND/                                                   INCO1670
      COMMON/DA/LIN                                                     INCO1680
C *** READ FIRST 20 RECORDS CONTAINING REPORT POSITIONS                 INCO1690
      DO 3 I=1,20                                                       INCO1700
      IREK=I+NPOSVR-1                                                   INCO1710
      LIN=IREK                                                          INCO1720
      READ(9,24)IR(I),IV(I)                                             INCO1730
   24 FORMAT(16I5)                                                      INCO1740
    3 CONTINUE                                                          INCO1750
    4 CONTINUE                                                          INCO1760
C *** INITIALIZE POINTER TO FORMAT RECORD                               INCO1770
      IF(IR(N).LT.1)GOTO 60                                             INCO1780
      NF=IR(N)+NPOSFM-1                                                 INCO1790
```

```
C *** INITIALIZE POINTER TO VARIABLES RECORD                                    INC01800
      LV=IV(N)+NPOSVR-1                                                         INC01810
C *** READ FIRST VARIABLES RECORD                                                INC01820
      LIN=LV                                                                    INC01830
      READ(9,24)NVARS                                                           INC01840
C *** GET POSITION OF RECORD IN REPORT FILE                                      INC01850
      LINE=NVARS(1)+NPOSFM-1                                                    INC01860
      GOTO 7                                                                    INC01870
    6 NF=NF+1                                                                   INC01880
C *** READ NEXT FORMAT RECORD                                                    INC01890
    7 LIN=NF                                                                    INC01900
      READ(9,11)FM                                                              INC01910
   11 FORMAT(1X,16A4)                                                           INC01920
C *** IS THIS END OF REPORT                                                      INC01930
      IF(FM(1).EQ.END)GOTO 50                                                   INC01940
      NERR=NERR+1                                                               INC01950
C *** ARE THERE VARIABLES WITH THIS FORMAT                                       INC01960
      IF(LINE.EQ.NF) GOTO 10                                                    INC01970
      PRINT FM                                                                  INC01980
      GOTO 6                                                                    INC01990
   10 NC=NVARS(2)+2                                                             INC02000
      PRINT FM,(V(NVARS(J)),J=3,NC)                                             INC02010
      LV=LV+1                                                                   INC02020
C *** READ NEXT VARIABLES RECORD                                                 INC02030
      LIN=LV                                                                    INC02040
      READ(9,24)NVARS                                                           INC02050
C *** GET POSITION OF NEXT RECORD IN REPORT FILE                                 INC02060
      LINE=NVARS(1)+NPOSFM-1                                                    INC02070
      IF(LINE.NE.NF) GOTO 6                                                     INC02080
      GOTO 10                                                                   INC02090
   50 RETURN                                                                    INC02100
C *** ERROR                                                                      INC02110
   60 PRINT 61                                                                  INC02120
   61 FORMAT(/' REPORT UNKNOWN'/)                                               INC02130
      GOTO 50                                                                   INC02140
      END                                                                       INC02150


      SUBROUTINE CHANGE(N,L,LIS,NREP)                                           INC02160
C     ********************************                                          INC02170
C     ********************************                                          INC02180
      COMMON V(4500),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW                 INC02190
     1      ,NERR,NX                                                            INC02200
      DIMENSION RANGE(2),F(10),LIS(66)                                          INC02210
      LOGICAL VERB,AUTO                                                         INC02220
      REAL TP(6)/'FEE ','F   ','PER ','P   ','CON ','C   '/                     INC02230
      COMMON/DA/LIN                                                             INC02240
C *** INITIALIZE                                                                 INC02250
```

```
      ISTRT=4200                                                        INC02260
      JSTRT=4300                                                        INC02270
      NUM=V(ISTRT)                                                      INC02280
      NPER=13                                                           INC02290
      NFEE=6                                                            INC02300
      NCON=3                                                            INC02310
      NC1=0                                                             INC02320
      NC2=0                                                             INC02330
C *** IS LIST EMPTY                                                      INC02340
      IF(L.LE.0)GOTO 60                                                 INC02350
C *** REPEAT FOR EACH PAR                                                INC02360
    3 DO 40 I=1,L                                                       INC02370
      NO=LIS(I)                                                         INC02380
      IF(NO.LT.1.OR.NO.GT.12) GOTO 60                                   INC02390
      IF(NREP.EQ.NCON)GOTO 5                                            INC02400
      IF(NREP.EQ.NPER)GOTO 5                                            INC02410
      IF(NREP.EQ.NFEE)GOTO 5                                            INC02420
      GOTO 70                                                           INC02430
    5 CONTINUE                                                          INC02440
      IF(NREP.EQ.NFEE.AND.NO.GT.11)GOTO 60                              INC02450
      IF(NREP.EQ.NPER.AND.NO.GT.11) GOTO 60                             INC02460
      IF(NREP.EQ.NCON.AND.NO.GT.3) GOTO 60                              INC02470
      GOTO 10                                                           INC02480
   70 PRINT 71                                                          INC02490
   71 FORMAT(' FEE(F),PERFORMANCE(P), OR CONTRACTOR(C)')                 INC02500
      READ 1,TYP                                                        INC02510
    1 FORMAT(A4)                                                        INC02520
      NREP=0                                                            INC02530
      IF(TYP.EQ.TP(1).OR.TYP.EQ.TP(2)) NREP=NFEE                        INC02540
      IF(TYP.EQ.TP(3).OR.TYP.EQ.TP(4)) NREP=NPER                        INC02550
      IF(TYP.EQ.TP(5).OR.TYP.EQ.TP(6)) NREP=NCON                        INC02560
      IF(NREP.EQ.0)GOTO 50                                              INC02570
      GOTO 5                                                            INC02580
   10 NP=0                                                              INC02590
      IF(NREP.EQ.NFEE) NP=33                                            INC02600
      IF(NREP.EQ.NCON)NP=66                                             INC02610
C *** REPEAT FOR MIN,TGT,MAX                                             INC02620
      NZZZ=11                                                           INC02630
      IF(NREP.EQ.NCON)NZZZ=35                                           INC02640
      DO 30 NZ=1,33,NZZZ                                                INC02650
      NREK=NPOSED+NO+NP+NZ-2                                            INC02660
      LIN=NREK                                                          INC02670
      READ(9,62)F,RANGE                                                 INC02680
   62 FORMAT(1X,10A4,2F10.2,I5)                                         INC02690
      IF(N.EQ.6)PRINT 18                                                INC02700
   18 FORMAT(/)                                                         INC02710
      PRINT F                                                           INC02720
      NPOS=JSTRT+NO+NP+NZ-1                                             INC02730
```

```
      NAB=V(NPOS)
      IF(VERB)PRINT 20,RANGE,V(NAB)
      IF(.NOT.VERB)PRINT 21,V(NAB)
   20 FORMAT(' MIN VALUE=',F8.2,' MAX VALUE=',F8.2,' PRESENT VALUE=',
     1      F8.2/)
   21 FORMAT(' PRESENT VALUE=',F8.2/)
      IF(N.EQ.6)GOTO 30
      PRINT 26
   26 FORMAT(' ENTER NEW VALUE')
      READ 2,VAL
    2 FORMAT(F8.2)
      IF(VAL.LT.RANGE(1).OR.VAL.GT.RANGE(2))GOTO 85
      V(NAB)=VAL
      IF(NREP.EQ.NCON.AND.NO.EQ.1.AND.VAL.GT.2.0)NC1=VAL
      IF(NREP.EQ.NCON.AND.NO.EQ.2)NC2=1
   30 CONTINUE
   40 CONTINUE
      IF(NC1.EQ.0)GOTO 50
      IF(NC2.GT.0)GOTO 50
      GOTO(50,50,42,44,46),NC1
   42 PRINT 43
      GOTO 48
   44 PRINT 45
      GOTO 48
   46 PRINT 47
      GOTO 48
   43 FORMAT(/' RESET STRATEGY GIVEN TO FEE VALUE'/)
   45 FORMAT(/' RESET STRATEGY GIVEN TO FEE %'/)
   47 FORMAT(/' RESET STRATEGY GIVEN TO OVER-RUN VALUE'/)
   48 L=1
      LIS(1)=2
      GOTO 3
   50 RETURN
   60 PRINT 61
   61 FORMAT(' INVALID PARAMETER NUMBER')
      GOTO 50
   85 PRINT 86
   86 FORMAT(' VALUE IS OUTSIDE LIMITS,RETRY')
      GOTO 50
      END


      SUBROUTINE DATA(CFILE)
C     **********************
C     **********************
C *** SUBROUTINE TO READ BASELINE DATA FROM -- CFILE
      COMMON V(4500),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW
     1     ,NERR,NX
```

```
      INTEGER DFILE
      DATA NBLK/4H    /
      DATA NREAL/4HREAL/,NETYPE/4HETYP/,NALPHA/4HALPH/
      LAST=0
      NTYPE=1
    5 GOTO(10,14,18),NTYPE
C *** READ A RECORD FROM DFILE
   10 READ(DFILE,11,END=50) NAME,NSIZE,NPOS,VAL
   11 FORMAT(1X,A4,2X,I5,I5,F15.3)
      GOTO 20
   14 READ(DFILE,15,END=50)NAME,NSIZE,NPOS,VAL
   15 FORMAT(1X,A4,2X,I5,I5,E15.7)
      GOTO 20
   18 READ(DFILE,19,END=50) NAME,NSIZE,NPOS,VAL
   19 FORMAT(1X,A4,2X,I5,I5,1X,A4)
   20 IF(NAME.NE.NBLK)GOTO 25
C *** IF NEW ARRAY GO SAVE SIZE AND UPDATE ARRAY POINTER
C *** SAVE VALUE IN 'V'
      V(NPOS+N-1)=VAL
      GOTO 5
C *** NEW ARRAY - IF FIRST INITIALIZE N
   25 IF(LAST.NE.0)GOTO 30
      N=1
      LAST=NSIZE
      GOTO 5
   30 IF(NAME.NE.NREAL)GOTO 32
      NTYPE=1
      GOTO 5
   32 IF(NAME.NE.NETYPE)GOTO 34
      NTYPE=2
      GOTO 5
   34 IF(NAME.NE.NALPHA)GOTO 40
      NTYPE=3
      GOTO 5
   40 N=N+LAST
      LAST=NSIZE
      GOTO 5
C *** END OF FILE ON DFILE
   50 CONTINUE
C *** PRINT MESSAGE
      PRINT 55
   55 FORMAT(/' BASE DATA READ FROM FILE'/)
      RETURN
      END
```

```
      SUBROUTINE XTRACT(IN,NN,LV,LVAR,NL,LIST,NFNC,NREP,*)
C     ******************************************************
C     ******************************************************
      COMMON V(4500),VERB,AUTC,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW
     1 ,NERR,NX
      DIMENSION LETA(35),LETB(35),LETC(35)
      DIMENSION LIST(10),LVAR(50)
      LOGICAL ALPHA,FOUND,THR,YEAR
      DIMENSION IN(72),NDIG(10)
      DATA NDIG/'0','1','2','3','4','5','6','7','8','9'/
      DATA ISP/' '/,ICM/','/
      DATA LETA/1HB,1HB,1HT,1HY,1HA,1HR,1HC,1HS,1HV,1HB,
     1          1HD,1HM,1HA,1HE,1HS,1HS,1HB,1HR,1HW,1HA,
     1          1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
     1          1HL,1HP,1HE,1HB,1HT/
      DATA LETB/1HL,1HL,1HH,1HE,1HB,1HE,1HH,1HT,1HE,1HR,
     1          1HI,1HA,1HU,1HX,1HE,1HE,1HL,1HE,1HE,1H ,
     1          1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     1          1H ,1H ,1HL,1HL,1H /
      DATA LETC/1HK,1HK,1HR,1HA,1HO,1HP,1HA,1HO,1HR,1HI,
     1          1HS,1HN,1HT,1HE,1HT,1HE,1HK,1HS,1HR,1H ,
     1         1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     1         1H ,1H ,1HK,1HK,1H /
    2 ALPHA=.FALSE.
      FOUND=.FALSE.
      THR=.FALSE.
      YEAR=.FALSE.
      NL=0
      LV=0
      NVAL=0
C *** N IS SIZE OF LETA,LETB,LETC ARRAYS
      N=35
      NSP=0
      NFNC=0
      NREP=0
    5 CONTINUE
   10 FORMAT(72A1)
      DO 200 I=1,NN
      IF(IN(I).NE.ISP) GOTO 15
      NSP=NSP+1
      IF(NSP.GT.5) GOTO 300
      IF(.NOT.FOUND) GOTO 200
      GOTO 50
   15 IF(IN(I).NE.ICM) GOTO 20
      GOTO 50
   20 NSP=0
      DO 25 J=1,10
      IF(IN(I).NE.NDIG(J)) GOTO 25
```

```
      IF(ALPHA) GOTO 200                                               INC04120
      IF(.NOT.FOUND)IFST=I                                             INC04130
      FOUND=.TRUE.                                                     INC04140
      NVAL=NVAL*10+J-1                                                 INC04150
      GOTO 200                                                         INC04160
   25 CONTINUE                                                         INC04170
C *** MUST BE ALPHA                                                     INC04180
   30 IF(FOUND) GOTO 35                                                INC04190
      FOUND=.TRUE.                                                     INC04200
      ALPHA=.TRUE.                                                     INC04210
      IFST=I                                                           INC04220
      GOTO 200                                                         INC04230
   35 IF(.NOT.ALPHA) GOTO 151                                          INC04240
      GOTO 200                                                         INC04250
   50 IF(ALPHA) GOTO 100                                               INC04260
      IF(.NOT.YEAR) GOTO 80                                            INC04270
      IF(.NOT.THR) GOTO 70                                             INC04280
      K=LIST(NL)+1                                                     INC04290
      IF(NVAL.LT.K) GOTO 151                                           INC04300
      DO 60 J=K,NVAL                                                   INC04310
      NL=NL+1                                                          INC04320
      LIST(NL)=J                                                       INC04330
   60 CONTINUE                                                         INC04340
      THR=.FALSE.                                                      INC04350
      GOTO 190                                                         INC04360
   70 NL=NL+1                                                          INC04370
      LIST(NL)=NVAL                                                    INC04380
      GOTO 190                                                         INC04390
   80 IF(.NOT.THR) GOTO 85                                             INC04400
      K=LVAR(LV)+1                                                     INC04410
      IF(NVAL.LT.K) GOTO 151                                           INC04420
      DO 83 J=K,NVAL                                                   INC04430
      IF(LV.EQ.50) GOTO 300                                            INC04440
      LV=LV+1                                                          INC04450
   83 LVAR(LV)=J                                                       INC04460
      THR=.FALSE.                                                      INC04470
      GOTO 190                                                         INC04480
   85 LV=LV+1                                                          INC04490
      LVAR(LV)=NVAL                                                    INC04500
      IF(LV.EQ.50) GOTO 300                                            INC04510
      GOTO 190                                                         INC04520
  100 DO 150 J=1,N                                                     INC04530
      IF(IN(IFST).NE.LETA(J)) GOTO 150                                 INC04540
      IF(LETB(J).EQ.ISP.AND.I.EQ.IFST+1)GOTO 99                        INC04550
      IF(IN(IFST+1).NE.LETB(J)) GOTO 150                               INC04560
      IF(IN(IFST+2).NE.LETC(J)) GOTO 150                               INC04570
   99 CONTINUE                                                         INC04580
      II=J                                                             INC04590
```

```
      IF(J.GT.5.AND.J.LE.19) II=6                                       INC04600
      IF(J.GT.19) II=7                                                  INC04610
      GOTO(101,102,103,104,105,106,107),II                              INC04620
  101 CONTINUE                                                          INC04630
      GOTO 190                                                          INC04640
  102 CONTINUE                                                          INC04650
      GOTO 190                                                          INC04660
  103 THR=.TRUE.                                                        INC04670
      GOTO 190                                                          INC04680
  104 YEAR=.TRUE.                                                       INC04690
      THR=.FALSE.                                                       INC04700
      GOTO 190                                                          INC04710
  105 PRINT 1105                                                        INC04720
 1105 FORMAT(' LINE ABORTED')                                           INC04730
      RETURN 1                                                          INC04740
  106 NFNC=J-5                                                          INC04750
      GOTO 190                                                          INC04760
  107 NREP=J-19                                                         INC04770
      IF(J.EQ.26)YEAR=.FALSE.                                           INC04780
      GOTO 190                                                          INC04790
  150 CONTINUE                                                          INC04800
  151 PRINT 152,IN(IFST),IN(IFST+1),IN(IFST+2)                          INC04810
  152 FORMAT(' INVALID FIELD- ',3A1)                                    INC04820
      NERR=NERR+1                                                       INC04830
      RETURN 1                                                          INC04840
  190 ALPHA=.FALSE.                                                     INC04850
      FOUND=.FALSE.                                                     INC04860
      NVAL=0                                                            INC04870
  200 CONTINUE                                                          INC04880
  300 RETURN                                                            INC04890
      END                                                               INC04900


      SUBROUTINE SET(N,L,LIS)                                           INC04910
C     ************************                                          INC04920
C     ************************                                          INC04930
      COMMON  V(4500),VERB,AUTC,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW        INC04940
     1    ,NERR,NX                                                      INC04950
      DIMENSION LIS(50)                                                 INC04960
      IF(L.LT.1)GOTO 40                                                 INC04970
      IF(L.LE.50) GOTO 3                                                INC04980
      PRINT 2                                                           INC04990
    2 FORMAT(' MAXIMUM NUMBER OF ITEMS IS 50')                          INC05000
      L=50                                                              INC05010
    3 CONTINUE                                                          INC05020
      DO 30  I=1,L                                                      INC05030
      NC=LIS(I)                                                         INC05040
      PRINT 5,V(NC)                                                     INC05050
```

```
    5 FORMAT(' VALUE=',F12.3)
      IF(N.EQ.11) GOTO 30
      PRINT 6
    6 FORMAT(' NEW VALUE=')
      READ 7,VAL
      V(NO)=VAL
    7 FORMAT(F12.3)
   30 CONTINUE
   40 RETURN
      END

      SUBROUTINE RUN(*)
C     ****************
C     ****************
      CALL ICM(N)
      IF (N .EQ. 1) RETURN1
      RETURN
      END

      SUBROUTINE RESET
C     ****************
C     ****************
      COMMON V(4500),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,CH1SW,CH2SW
     1    ,NERR,NX
      ISTRT=4200
      JSTRT=4300
      NUM=V(ISTRT)
      DO 10 I=1,NUM
      NAB=V(I+JSTRT)
      V(NAB)=V(I+ISTRT)
   10 CONTINUE
      PRINT 15
   15 FORMAT(' ALL PARAMETERS RESET'/)
      RETURN
      END
```

APPENDIX E

COMMAND AND CONTROL PROGRAM (PROGPL)

```
C    **************************
C    * PROGRAM PLANNING MODEL *
C    **************************
C *** MODEL FOR EXERCISES 21,22, AND 25
C *** READS DATA BASE AND CALLS OTHER SUBROUTINES
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,REPSW,CH1SW
     1,CH2SW,NERR
      INTEGER NEX(3)/21,22,25/,BAS(3)/21,22,25/,CFILE
      LOGICAL VERB,AUTO
   55 PRINT 4
    4 FORMAT(' EXERCISE NUMBER ?')
      READ(5,5) EXNO
    5 FORMAT(F5.2)
C *** INITIALIZE FIRST TIME THROUGH
      NERR=0
      VERB=.TRUE.
      DO 3 I=1,4500
    3 V(I)=0.0
      NX=EXNO
      DO 6 I=1,3
      IF(NX.EQ.NEX(I)) GOTO 8
    6 CONTINUE
      PRINT 7,NX
    7 FORMAT(' ** INVALID EXERCISE NUMBER - ',I5/)
      GO TO 55
C *** READ DATA FILES
    8 NBAS=I+1
      CALL DATAY(NBAS)
      CH1SW=2
      CH2SW=2
C *** SET NAME OF REPORT FILE
C *** GET REPORT FILE NAME
   14 NREP=I-1
      CALL MAKER(NREP)
      IF(I.EQ.1) CALL COMM21
      IF(I.GT.1) CALL COMM22
      END
```

```
      SUBROUTINE DATAY(IFL)
C     *********************
C     *********************
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NPOSEC,REPSW,CH1SW
     1,CH2SW,NERR
      REAL FMT(4)
      N=1
      LAST=0
      IF(IFL.EQ.1) GO TO 5
      IENS=1
    2 READ(10,219,END=1)
      GO TO 2
    1 IENS=IENS+1
      IF(IENS.NE.IFL) GO TO 2
    5 READ(10,219,END=333)ISKIP,IC,FMT
  219 FORMAT(2I5,4A4)
      N=N+LAST
      LAST=ISKIP
      IF(IC.EQ.0) GO TO 5
      READ(10,FMT)(NPOS,V(NPOS+N-1),II=1,IC)
      GO TO 5
  333 WRITE(6,55) IFL
   55 FORMAT(/' BASE DATA READ FROM FILE FT10F00',I1/)
      RETURN
      END

      SUBROUTINE MAKER(IUN)
C     *********************
C     *********************
      DIMENSION FORM(16),IV(16),FMAT(10),FV(2)
      CALL DEFINE(9,'FILE    ','FT09F001',LIN,80)
      COMMON/DA/LIN
      LIN=2
      IF(IUN.EQ.0) GO TO 10
      IEN=IUN*3
      IDD=0
    5 READ(4,11,END=6)
      GO TO 5
    6 IDD=IDD+1
      IF(IDD.NE.IEN) GO TO 5
   10 NPOSFM=2
    8 READ(4,11,END=20) FORM
   11 FORMAT(1X,16A4)
      WRITE(9,11) FORM
      GO TO 8
   20 NPOSVR=LIN
   26 READ(4,24,END=60) IV
```

```
   24 FORMAT(16I5)                                                   PRO00840
      WRITE(9,24) IV                                                 PRO00850
      GO TO 26                                                       PRO00860
   60 NPOSED=LIN                                                     PRO00870
   61 READ(4,62,END=70) FMAT,FV,ID                                   PRO00880
   62 FORMAT(1X,10A4,2F10.2,I5)                                      PRO00890
      WRITE(9,62) FMAT,FV,ID                                         PRO00900
      GO TO 61                                                       PRO00910
   70 LIN=1                                                          PRO00920
      WRITE(9,24) NPOSFM,NPOSVR,NPOSED                               PRO00930
      WRITE(6,71)                                                    PRO00940
   71 FORMAT(' CREATED REPORT FILE FT09F001')                        PRO00950
      RETURN                                                         PRO00960
      END                                                            PRO00970

      SUBROUTINE SET(N,L,LIS)                                        PRO00980
C     ***********************                                        PRO00990
C     ***********************                                        PRO01000
      COMMON NX,V(6965),VERB,AUTC,NPOSFM,NPOSVR,NPOSED,REPSW,CH1SW   PRO01010
     1,CH2SW,NERR                                                    PRO01020
      DIMENSION LIS(250)                                             PRO01030
      IF(L.EQ.0) RETURN                                              PRO01040
      IF(L.GT.50)L=50                                                PRO01050
      NNN=6965                                                       PRO01060
      IF(V(1).GT.21.) NNN=4500                                       PRO01070
      DO 30  I=1,L                                                   PRO01080
      NO=LIS(I)                                                      PRO01090
      IF(NO.LT.1.OR.NO.GT.NNN )GOTO 30                               PRO01100
      PRINT 5,NO,V(NO)                                               PRO01110
    5 FORMAT(' ITEM(',I4,')=',F15.3)                                 PRO01120
      IF(N.EQ.11) GOTO 30                                            PRO01130
      PRINT 6                                                        PRO01140
    6 FORMAT(' NEW VALUE')                                           PRO01150
      READ 7,VAL                                                     PRO01160
      V(NO)=VAL                                                      PRO01170
    7 FORMAT(F12.3)                                                  PRO01180
   30 CONTINUE                                                       PRO01190
      RETURN                                                         PRO01200
      END                                                            PRO01210

      SUBROUTINE DATE (MM,ID,IY)                                     PRO01220
C     *************************                                      PRO01230
C     *************************                                      PRO01240
C *** SUBROUTINE FOR CONVERSION MM DD YY TO SERIAL N OR BACK         PRO01250
      DIMENSION MAB(12)                                              PRO01260
      DIMENSION LEN(12)                                              PRO01270
```

```
      DATA MAB/0,31,59,90,120,151,181,212,243,273,304,334/          PRC01280
      DATA LEN/31,28,31,30,31,30,31,31,30,31,30,31/                 PRO01290
      IF (ID .EQ. 0) GO TO 50                                       PRC01300
C *** CONVERT MM DD YY TO SERIAL FROM 1/1/50                         PRO01310
      N = MAB(MM) + ID + (IY - 50) * 365                            PRC01320
      M = (N + 669) / 1460                                          PRC01330
      J = IY - 4 * ((IY - 48)/4) -48                                PRC01340
      IF (J .NE. 0) GO TO 6                                         PRC01350
      IF (MM .NE. 3) GO TO 6                                        PRC01360
      IF (ID .NE. 1) GO TO 6                                        PRO01370
    5 M = M + 1                                                     PRC01380
    6 CONTINUE                                                      PRC01390
      MM = N + M                                                    PRC01400
      RETURN                                                        PRC01410
C *** CONVERT SERIAL DATE TO MM DD YY                               PRC01420
C *** FIRST UPDATE SATS. & SUNDAYS TO MONDAY                        PRC01430
   50 CONTINUE                                                      PRC01440
      NYR=365                                                       PRC01450
      NDAYS=MM                                                      PRC01460
C *** FIRST LEAP-YEAR AFTER 1950 IS 1952                            PRC01470
      LEAP=52                                                       PRC01480
      IY=50                                                         PRC01490
C *** LP IS=1 FOR LEAP YEARS, =0 FOR NON LEAP YEARS                 PROC1500
   56 LP=0                                                          PRC01510
C *** NOW GET YEAR                                                  PRC01520
   57 IF(NDAYS.LE.NYR+LP)GOTO 65                                    PRC01530
      NDAYS=NDAYS-(NYR+LP)                                          PRC01540
      IY=IY+1                                                       PRC01550
      IF(IY.NE.LEAP)GOTO 56                                         PRCC1560
      LEAP=LEAP+4                                                   PRC01570
      LP=1                                                          PRCC1580
      GOTO 57                                                       PRCC1590
C *** SET LENGTH OF FEB AND GET MONTH                               PRC01600
   65 LEN(2)=LP+28                                                  PRC01610
      DO 66 I=1,12                                                  PRCC1620
      IF(NDAYS.LE.LEN(I))GOTO 70                                    PRC01630
      NDAYS=NDAYS-LEN(I)                                            PRCC1640
   66 CONTINUE                                                      PRC01650
C *** SET MM FOR MONTH                                              PRC01660
   70 MM=I                                                          PRC01670
C *** SET DAY (REMAINDER)                                           PRC01680
      ID=NDAYS                                                      PRCC1690
      RETURN                                                        PRO01700
      END                                                           PRC01710
```

```
      SUBROUTINE YTRACT(IN,NN,LV,LVAR,NL,LIST,NFNC,NREP,*)
C     ***************************************************************
C     ***************************************************************
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NFOSED,REFSW,CH1SW
     1,CH2SW,NERR
      DIMENSION LETA(35),LETB(35),LETC(35)
      DIMENSION LIST(10),LVAR(250)
      LOGICAL ALPHA,FOUND,THR,YEAR
      DIMENSION IN(72),NDIG(10)
      DATA NDIG/'0','1','2','3','4','5','6','7','8','9'/
      DATA ISP/' '/,ICM/','/
      DATA ISLSH/1H//
      DATA LETA/1HB,1HB,1HT,1HY,1HA,1HR,1HC,1HS,1HV,1HB,
     1          1HD,1HM,1HA,1HE,1HS,1HS,1HB,1HR,1HB,1HA,
     1          1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
     1          1HL,1HA,1HC,1HB,1HT/
      DATA LETB/1HL,1HL,1HH,1HE,1HB,1HE,1HH,1HT,1HE,1HR,
     1          1HI,1HA,1HU,1HX,1HE,1HE,1HL,1HE,1HL,1H ,
     1          1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     1          1H ,1HL,1HR,1HL,1H /
      DATA LETC/1HK,1HK,1HR,1HA,1HO,1HP,1HA,1HC,1HR,1HI,
     1          1HS,1HN,1HT,1HE,1HT,1HE,1HK,1HS,1HK,1H ,
     1         1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     1         1H ,1HL,1HI,1HK,1H /
      ALPHA=.FALSE.
      FOUND=.FALSE.
      THR=.FALSE.
      YEAR=.FALSE.
      DO 5 I=1,10
    5 LIST(I)=0
      DO 6 I=1,250
    6 LVAR(I)=0
      NL=0
      LV=0
      NVAL=0
C *** N IS SIZE OF LETA,LETB,LETC ARRAYS
      N=35
      NSP=0
      NFNC=0
      NREP=0
      DO 200 I=1,NN
      IF(IN(I).NE.ISP) GOTO 15
      NSP=NSP+1
      IF(NSP.GT.5) GOTO 300
      IF(.NOT.FOUND) GOTO 200
      GOTO 50
   15 IF(IN(I).NE.ICM) GOTO 18
      GOTO 50
```

```
18    IF(IN(I).NE.ISLSH)GOTO 20                                     PRC02200
      GOTO 50                                                       PRC02210
20    NSP=0                                                         PRC02220
      DO 25 J=1,10                                                  PRC02230
      IF(IN(I).NE.NDIG(J)) GOTO 25                                  PRC02240
      IF(ALPHA) GOTO 200                                            PRC02250
      IF(.NOT.FOUND)IFST=I                                          PRC02260
      FOUND=.TRUE.                                                  PRC02270
      NVAL=NVAL*10+J-1                                              PRC02280
      GOTO 200                                                      PRC02290
25    CONTINUE                                                      PRC02300
C *** MUST BE ALPHA                                                  PRC02310
30    IF(FOUND) GOTO 35                                             PRC02320
      FOUND=.TRUE.                                                  PRC02330
      ALPHA=.TRUE.                                                  PRC02340
      IFST=I                                                        PRC02350
      GOTO 200                                                      PRC02360
35    IF(.NOT.ALPHA) GOTO 151                                       PRC02370
      GOTO 200                                                      PRC02380
50    IF(ALPHA) GOTO 100                                            PRC02390
      IF(.NOT.YEAR) GOTO 80                                         PRC02400
      IF(.NOT.THR) GOTO 70                                          PRC02410
      K=LIST(NL)+1                                                  PRC02420
      IF(K.EQ.1)GOTO 151                                            PRC02430
      IF(NVAL.LT.K) GOTO 151                                        PRC02440
      DO 60 J=K,NVAL                                                PRC02450
      NL=NL+1                                                       PRC02460
      LIST(NL)=J                                                    PRC02470
60    CONTINUE                                                      PRC02480
      THR=.FALSE.                                                   PRC02490
      GOTO 190                                                      PRC02500
70    NL=NL+1                                                       PRC02510
      LIST(NL)=NVAL                                                 PRC02520
      GOTO 190                                                      PRC02530
80    IF(.NOT.THR) GOTO 85                                          PRC02540
      K=LVAR(LV)+1                                                  PRC02550
      IF(K.EQ.1)GOTO 151                                            PRC02560
      IF(NVAL.LT.K) GOTO 151                                        PRC02570
      DO 83 J=K,NVAL                                                PRC02580
      IF(LV.EQ.50) GOTO 300                                         PRC02590
      LV=LV+1                                                       PRC02600
83    LVAR(LV)=J                                                    PRC02610
      THR=.FALSE.                                                   PRC02620
      GOTO 190                                                      PRC02630
85    LV=LV+1                                                       PRC02640
      LVAR(LV)=NVAL                                                 PRC02650
      IF(LV.EQ.50) GOTO 300                                         PRC02660
      GOTO 190                                                      PRC02670
```

```
  100 DO 150 J=1,N
      IF(IN(IFST).NE.LETA(J)) GOTO 150
      IF(LETB(J).EQ.ISP.AND.I.EQ.IFST+1)GOTO 99
      IF(IN(IFST+1).NE.LETB(J)) GOTO 150
      IF(IN(IFST+2).NE.LETC(J)) GOTO 150
   99 CONTINUE
      II=J
      IF(J.GT.5.AND.J.LE.19) II=6
      IF(J.GT.19) II=7
      GOTO(101,102,103,104,105,106,107),II
  101 CONTINUE
      GOTO 190
  102 CONTINUE
      GOTO 190
  103 THR=.TRUE.
      GOTO 190
  104 CONTINUE
      THR=.FALSE.
      GOTO 190
  105 PRINT 1105
 1105 FORMAT(' LINE ABORTED')
      RETURN 1
  106 NFNC=J-5
      GOTO 190
  107 NREP=J-19
      IF(J.EQ.26)YEAR=.FALSE.
      GOTO 190
  150 CONTINUE
  151 PRINT 152,IN(IFST),IN(IFST+1),IN(IFST+2)
  152 FORMAT(' INVALID FIELD- ',3A1)
      NERR=NERR+1
      RETURN 1
  190 ALPHA=.FALSE.
      FOUND=.FALSE.
      NVAL=0
  200 CONTINUE
  300 RETURN
      END


      SUBROUTINE COMM21
C     *****************
C     *****************
      COMMON NX,V(6965),VERB,AUTO,NPCSFM,NPOSVR,NPCSED,REPSW,CH1SW
     1,CH2SW,NERR
      COMMON/DA/LIN
      DIMENSION LIST1(250),LIST2(10),LINE(72)
      LOGICAL VERB,AUTO,RUNSW
```

```
C *** TEST FOR FIRST TIME THROUGH                                       PRC03140
      IF (CH1SW.EQ.3) GOTO 15                                           PRC03150
      IF (CH2SW.EQ.3) GOTO 15                                           PRC03160
C *** FIRST TIME THRU- INITIALIZE NETWORK NUMBER                         PRC03170
      NCON=V(3)                                                         PRC03180
C *** INITIALIZE VERBOSE/BRIEF SWITCH                                   PRC03190
      VERB=.TRUE.                                                       PRC03200
      RUNSW=.FALSE.                                                     PRC03210
C *** INIT. AUTO SWITCH PER EXERCISE                                    PRC03220
    6 AUTO=.FALSE.                                                      PRC03230
   10 PRINT 120,V(1)                                                    PRC03240
      LIN=1                                                             PRC03250
      READ(9,100)NPOSFM,NPOSVR,NPOSED                                   PRC03260
  100 FORMAT(16I5)                                                      PRC03270
      CALL NETWRK(NCON)                                                 PRC03280
C *** RUN MODEL FIRST TIME THROUGH                                      PRC03290
      GOTO 60                                                           PRC03300
C *** NORMAL ENTRY                                                      PRC03310
C *** REQUEST USER COMMAND                                              PRC03320
   15 CONTINUE                                                          PRC03330
      IF(NERR.EQ.5)PRINT 11                                             PRC03340
   11 FORMAT(/' AW C:MON YOU HAVE MADE FIVE INPUT ERRORS.'/             PRC03350
     1  ' IF SYSTEM X WERE OPERATIONAL YOU WOULD HAVE WIPED OUT'/       PRC03360
     1  ' MONTREAL, CHICAGO AND EAST MCKEESPORT PENNSYLVANIA='/)        PRC03370
      IF(NERR.EQ.5)NERR=6                                               PRC03380
      PRINT 14                                                          PRC03390
   14 FORMAT(//' COMMAND ?')                                            PRC03400
   17 READ 16, LINE                                                     PRC03410
   16 FORMAT(72A1)                                                      PRC03420
      CALL XTRACT(LINE,72,L1,LIST1,L2,LIST2,NFNC,NREP,&15)              PRC03430
      IF(NFNC.EQ.0)GOTO 17                                              PRC03440
      GOTO (20,25,30,35,40,45,50,55,60,70,70,75,80),NFNC                PRC03450
C *** REPORT OPTION.                                                    PRC03460
   20 IF(.NOT.RUNSW)CALL PROCCS                                         PRC03470
      RUNSW=.TRUE.                                                      PRC03480
      IF(NREP.LT.1.AND.L1.GT.0)GOTO 1124                                PRC03490
      PRINT 120,V(1)                                                    PRC03500
   21 PRINT 121                                                         PRC03510
C *** GET REPORT                                                        PRC03520
   24 IF(NREP.GT.12)GOTO 1124                                           PRC03530
      PRINT 22                                                          PRC03540
   22 FORMAT(/' REPORT UNKNOWN')                                        PRC03550
      GOTO 15                                                           PRC03560
 1124 CALL REPNET(NFNC,NREP,L1,LIST1)                                   PRC03570
      GOTO 15                                                           PRC03580
  120 FORMAT(//' EXERCISE ',F3.0)                                       PRC03590
  121 FORMAT(/' ENGINEERING CHANGE PROPOSAL.',5X,A8,1X,A8/              PRC03600
     1        1X,47(1H-))                                               PRC03610
```

```
C *** NFNC=2 CHANGE OPTION                                              PRC03620
   25 IF(NREP.EQ.15)GOTO 27                                             PRC03630
   26 CALL CHA1(NFNC,L1,LIST1)                                          PRC03640
      IF(NFNC.EQ.2)RUNSW=.FALSE.                                        PRC03650
      GOTO 15                                                           PRC03660
   27 IF(NFNC.EQ.6) GOTO 293                                            PRC03670
      PRINT 28,V(3)                                                     PRC03680
   28 FORMAT(/' PRESENT NETWORK=',F3.0/' ENTER NEW VALUE')              PRC03690
  281 READ 2281,V(3)                                                    PRC03700
 2281 FORMAT(F10.0)                                                     PRC03710
      NCON=V(3)                                                         PRC03720
      IF(NCON.GT.0.AND.NCON.LE.V(5))GOTO 291                            PRC03730
      PRINT 29,V(5)                                                     PRC03740
   29 FORMAT(/' ONLY',F4.0,' NETWORKS AVAILABLE'/' ENTER NEW VALUE')    PRC03750
      GOTO 281                                                          PRC03760
  291 CALL NETWRK(NCON)                                                 PRC03770
      RUNSW=.FALSE.                                                     PRC03780
      GOTO 15                                                           PRC03790
  293 PRINT 294,V(3)                                                    PRC03800
  294 FORMAT(/' PRESENT NETWORK=',F3.0)                                 PRC03810
      GOTO 15                                                           PRC03820
C *** NFNC=3 STOP OPTION                                                 PRC03830
   30 CALL ERASE('FILE    ','FT09F001')                                 PRC03840
      CALL EXIT                                                         PRC03850
C *** GO VERBOSE                                                        PRC03860
C *** NFNC=4 VERBOSE OPTION                                             PRC03870
   35 VERB=.TRUE.                                                       PRC03880
      PRINT 36                                                          PRC03890
   36 FORMAT(' VERBOSE MODE')                                           PRC03900
      GOTO 15                                                           PRC03910
C *** NFNC=5 BRIEF OPTION                                               PRC03920
C *** GO BRIEF                                                          PRC03930
   40 VERB=.FALSE.                                                      PRC03940
      PRINT 41                                                          PRC03950
   41 FORMAT(' BRIEF MODE')                                             PRC03960
      GOTO 15                                                           PRC03970
C *** NFNC=6 DISPLAY OPTION (USES CHANGE SUBROUTINE)                    PRC03980
   45 GOTO 25                                                           PRC03990
   50 CONTINUE                                                          PRC04000
C *** NFNC=7 MANUAL OPTION                                              PRC04010
      AUTO=.FALSE.                                                      PRC04020
      GOTO 15                                                           PRC04030
   55 CONTINUE                                                          PRC04040
      AUTO=.TRUE.                                                       PRC04050
      GOTO 15                                                           PRC04060
C *** RUN OPTION NFNC=9                                                 PRC04070
C *** CALL ENGINEERING CHANGE PROPOSAL MODEL                            PRC04080
   60 CONTINUE                                                          PRC04090
```

```
      CALL PROCOS                                                       PRO04100
      RUNSW=.TRUE.                                                      PRO04110
      GOTO 15                                                           PRO04120
C *** SET OR SEE VALUES IN ARRAY V                                      PRO04130
   70 CALL SET(NFNC,L1,LIST1)                                           PRO04140
      GOTO 15                                                           PRO04150
   75 CONTINUE                                                          PRO04160
      GOTO 15                                                           PRO04170
   80 CALL NETWRK(NCON)                                                 PRO04180
      GOTO 15                                                           PRO04190
      END                                                               PRO04200


      SUBROUTINE NETWRK(NCON)                                           PRO04210
C     ***********************                                           PRO04220
C     ***********************                                           PRO04230
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,REPSW,CH1SW      PRO04240
     1,CH2SW,NERR                                                       PRO04250
      DIMENSION NW(3),MW(4)                                             PRO04260
      DATA NW/3965,4965,5965/                                           PRO04270
      DATA MW/10,360,610,860/                                           PRO04280
      NSTART=NW(NCON)                                                   PRO04290
      DO 20 J=1,4                                                       PRO04300
      M=MW(J)                                                           PRO04310
      K=(J-1)*250                                                       PRO04320
      DO 10 I=1,250                                                     PRO04330
      V(I+M)=V(K+I+NSTART)                                              PRO04340
   10 CONTINUE                                                          PRO04350
   20 CONTINUE                                                          PRO04360
      RETURN                                                            PRO04370
      END                                                               PRO04380


      SUBROUTINE REPNET(N,NR,L,LIS)                                     PRO04390
C     ****************************                                      PRO04400
C     ****************************                                      PRO04410
      COMMON NX,CONTRL(10),INDACT(250),STRTEV(50),ENDEV(50)             PRO04420
      COMMON JBEV(250),JEEV(250),TIMEN(250)                             PRO04430
      COMMON ES(250),EC(250),SL(250),CL(250)                            PRO04440
      COMMON STRTTI(50),ENDTI(50)                                       PRO04450
      COMMON SLACK(250),NACTOP(250),CRITP(250)                          PRO04460
      COMMON NACT,STD,ENDATE,NACTL,SSLACK                               PRO04470
      COMMON JEVTCP(250),JACTCP(250),TIMEVT(250)                        PRO04480
      COMMON TIMEVL(250),OTHERS(3000)                                   PRO04490
      REAL INDACT,JBEV,JEEV,NACTOP                                      PRO04500
      REAL JEVTCP,JACTCP,CRITP                                          PRO04510
      DIMENSION LIS(250)                                                PRO04520
      DATA NSTAR/1H*/,NBLK/1H /,NCEE/1HC/                               PRO04530
```

```
      TODAY=CONTRL(2)                                                   PRC04540
      CRIT=1.                                                           PRC04550
      NUMACT = 250                                                      PRC04560
      NCRIT=14                                                          PRC04570
      NALL=13                                                           PRC04580
      PRINT 5,CONTRL(3)                                                 PRO04590
    5 FORMAT(//25X,'ACTIVITY SCHEDULE REPORT'/25X,24(1H-)/              PRC04600
     1  /32X,'NETWORK',F3.0,                                            PRC04610
     1 //13X,2('    EARLIEST  LATEST')/12X,2('     START'),             PRC04620
     1 3X,2('  END DATE')/                                              PRC04630
     1 ' ACT   DURATION  DATE(ES)  DATE(LS)  (EE)      (LE)',6X,'SLACK')PRO04640
C *** ALL ACTIVITIES REQUIRED                                             PRC04650
      IF(NR.NE.NALL)GOTO 10                                             PRC04660
C *** SET LIST                                                          PRC04670
      DO 8 I=1,NUMACT                                                   PRC04680
    8 LIS(I)=I                                                          PRC04690
      L=NUMACT                                                          PRC04700
      GOTO 25                                                           PRC04710
C *** CRITICAL ONLY REQUIRED                                            PRC04720
   10 IF(NR.NE.NCRIT)GOTO 20                                            PRC04730
C *** SET LIST FOR CRITICAL ACTS.                                       PRC04740
      L=0                                                               PRC04750
      DO 14 I=1,NUMACT                                                  PRC04760
      J = NACTOP(I)                                                     PRC04770
      IF (J .EQ. 0) GOTO 14                                             PRC04780
      IF (CRITP(J) .NE. CRIT) GOTO 14                                   PRC04790
      L=L+1                                                             PRC04800
      LIS(L) = J                                                        PRC04810
   14 CONTINUE                                                          PRC04820
      GOTO 25                                                           PRC04830
C *** CHECK REQU. ACTS.                                                 PRC04840
   20 IF(L.LT.1.OR.L.GT.NUMACT)GOTO 60                                  PRO04850
      DO 22 I=1,L                                                       PRC04860
      IF(LIS(I).LT.1.OR.LIS(I).GT.NUMACT) GOTO 60                       PRC04870
   22 CONTINUE                                                          PRC04880
C *** GO THRU LIS PRINTING REQUIRED ACTS.                               PRC04890
C *** FIRST UNPACK DATES                                                PRC04900
   25 DO 30 I=1,L                                                       PRC04910
      NPOS=LIS(I)                                                       PRC04920
      NAC=INDACT(NPOS)                                                  PRC04930
      IF(NAC.LE.0)GOTO 30                                               PRO04940
      NMES=ES(NAC)                                                      PRC04950
      NMLS=SL(NAC)                                                      PRC04960
      NMEE=EC(NAC)                                                      PRC04970
      NMLE=CL(NAC)                                                      PRC04980
      NDES=0                                                            PRC04990
      NDLS=0                                                            PRC05000
      NDEE=0                                                            PRC05010
```

```
      NDLE=0
   26 FORMAT(4I5)
      CALL DATE(NMES,NDES,NYES)
      CALL DATE(NMLS,NDLS,NYLS)
      CALL DATE(NMEE,NDEE,NYEE)
      CALL DATE(NMLE,NDLE,NYLE)
      NCFLAG=NBLK
      IF(CRITP(NAC).EQ.1.0)NCFLAG=NSTAR
      NCOMP=NBLK
      IF(EC(NAC).LE.TODAY)NCOMP=NCEE
      PRINT 28,NAC,NCOMP,TIMEN(NAC),NMES,NDES,NYES,
     1         NMLS,NDLS,NYLS,  NMEE,NDEE,NYEE,NMLE,NDLE,NYLE,
     1      SLACK(NAC),NCFLAG
   28 FORMAT(I4,A1,F10.0,4(2X,I2,2('/',I2)),F7.0,A1)
   30 CONTINUE
   40 RETURN
C *** ERROR
   60 PRINT 62
   62 FORMAT(/' INVALID ACTIVITY NUMBER'/)
      RETURN
      END


      SUBROUTINE CHA1(N,L,LIS)
C     ************************
C     ************************
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,REPSW,CH1SW
     1,CH2SW,NERR
      DIMENSION RANGE(2),F(10),FDUM(1),IDUM(1)
      DIMENSION LIS(250)
      LOGICAL VERB,AUTO
      RANGE(1)=0.0
      RANGE(2)=100.
      IDSTT=860
      NUM=V(4)
      TODAY=V(2)
      IF(L.LE.0)GOTO 60
      DO 40 I=1,L
      NO=LIS(I)
      IF(NO.LT.1.OR.NO.GT.NUM)GOTO 60
      NAB=NO+IDSTT
   18 IF(VERB)PRINT 19,NO,V(NAB)
      IF(.NOT.VERB)PRINT 23,NO,V(NAB)
   19 FORMAT(' PRESENT VALUE(',I3,')',F9.3)
C *** IF CHANGE REQUEST NEW VALUE
   20 IF(N.EQ.6)GOTO 40
      NEC=V(NO+10)
      IF(V(NEC+1360).LT.TODAY)GOTO 35
```

```
      IF(VERB)PRINT 22,NO                                               PRC05480
   22 FORMAT(/' NEW VALUE(',I3,')=' )                                   PRC05490
   23 FORMAT(' (',I3,')=',F5.0)                                         PRC05500
      READ 24,VAL                                                       PRC05510
   24 FORMAT(F15.3)                                                     PRC05520
      IF(VAL.GE.RANGE(1).AND.VAL.LE.RANGE(2))GOTO 30                    PRC05530
      PRINT 27,RANGE                                                    PRC05540
   27 FORMAT(' PERMITTED RANGE OF VALUE IS ',F5.0,' TO ',F5.0)          PRC05550
      NERR=NERR+1                                                       PRC05560
      GOTO 50                                                           PRC05570
   35 PRINT 36,NC                                                       PRC05580
   36 FORMAT(' ACTIVITY',I4,' IS COMPLETED')                            PRC05590
      PRINT 37                                                          PRC05600
   37 FORMAT(/)                                                         PRC05610
      GOTO 40                                                           PRC05620
   30 V(NAB)=VAL                                                        PRC05630
   40 CONTINUE                                                          PRC05640
   50 CH1SW=3                                                           PRC05650
      RETURN                                                            PRC05660
   60 PRINT 61,NUM                                                      PRC05670
   61 FORMAT(' VALID ITEMS ARE 1 THRU ',I3/)                            PRC05680
      NERR=NERR+1                                                       PRC05690
      GOTO 50                                                           PRC05700
      END                                                               PRC05710

      SUBROUTINE COMM22                                                 PRC05720
C     ****************                                                  PRC05730
C     ****************                                                  PRC05740
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,REPSW,CH1SW      PRC05750
     1,CH2SW,NERR                                                       PRC05760
      COMMON/DA/LIN                                                     PRC05770
      DIMENSION LIST1(250),LIST2(10),LINE(72)                           PRC05780
      LOGICAL VERB,AUTO                                                 PRC05790
C *** TEST FOR FIRST TIME THROUGH                                       PRC05800
      IF (CH1SW.EQ.3) GOTO 15                                           PRC05810
      IF (CH2SW.EQ.3) GOTO 15                                           PRC05820
C     FIRST TIME THRU- INITIALIZE -                                     PRC05830
C *** INITIALIZE VERBOSE/BRIEF SWITCH                                   PRC05840
      VERB=.TRUE.                                                       PRC05850
C *** INIT. AUTO SWITCH PER EXERCISE                                    PRC05860
      AUTO=.FALSE.                                                      PRC05870
   10 PRINT 120,V(1)                                                    PRC05880
      LIN=1                                                             PRC05890
      READ(9,100)NPOSFM,NPOSVR,NPOSED                                   PRC05900
  100 FORMAT(16I5)                                                      PRC05910
C *** PROCESS EDITS FILE WITH INITIAL CALL TO CHANGE SUBS               PRC05920
C *** ARGUMENTS ARE NOT USED IN FIRST CALL                              PRC05930
```

```
C *** READ EDITS FILE(S) AND SAVE PARAMETER POINTERS
   12 CALL CHA2(NFNC,L1,LIST1)
C *** NORMAL ENTRY
C *** REQUEST USER COMMAND
   15 CONTINUE
      IF(NERR.EQ.5)PRINT 11
   11 FORMAT(/' AW C:MON YOU HAVE MADE FIVE INPUT ERRORS.'/
     1  ' IF SYSTEM X WERE OPERATIONAL YOU WOULD HAVE WIPED OUT'/
     1  ' MONTREAL, CHICAGO AND EAST MCKEESPORT PENNSYLVANIA='/)
      IF(NERR.EQ.5)NERR=6
      PRINT 14
   14 FORMAT(//' COMMAND ?')
   17 READ 16, LINE
   16 FORMAT(72A1)
      CALL YTRACT(LINE,72,L1,LIST1,L2,LIST2,NFNC,NREP,&15)
      IF(NFNC.EQ.0)GOTO 17
      GOTO (20,25,30,35,40,45,50,55,60,70,70,75,80),NFNC
C *** REPORT OPTION.
   20 CONTINUE
      PRINT 120,V(1)
      IF(NX.EQ.25)GOTO 22
   21 PRINT 121
      GOTO 29
   22 PRINT 122
      GOTO 29
C *** GET REPORT
   29 CONTINUE
      CALL REPRT(NREP)
      GOTO 15
  120 FORMAT(//' EXERCISE ',F3.0)
  121 FORMAT(/' PRODUCTION PLANNING  ',5X,A8,1X,A8/1X,43(1H-))
  122 FORMAT(/' REPROGRAMMING',15X,A8,1X,A8/1X,43(1H-)/)
C *** NFNC=2 CHANGE OPTION
   25 CALL CHA2(NFNC,L1,LIST1)
      GOTO 15
C *** NFNC=3 STOP OPTION
   30 CALL ERASE('FILE    ','FT09F001')
      CALL EXIT
C *** GO VERBOSE
C *** NFNC=4 VERBOSE OPTION
   35 VERB=.TRUE.
      PRINT 36
   36 FORMAT(' VERBOSE MODE')
      GOTO 15
C *** NFNC=5 BRIEF OPTION
C *** GO BRIEF
   40 VERB=.FALSE.
      PRINT 41
```

```
   41 FORMAT(' BRIEF MODE')                                            PRC06420
      GOTO 15                                                           PRC06430
C *** NFNC=6 DISPLAY OPTION (USES CHANGE SUBROUTINE)                    PRC06440
   45 GOTO 25                                                           PRC06450
   50 CONTINUE                                                          PRC06460
C *** NFNC=7 MANUAL OPTION                                              PRC06470
      AUTO=.FALSE.                                                      PRC06480
      GOTO 15                                                           PRC06490
   55 CONTINUE                                                          PRC06500
      AUTO=.TRUE.                                                       PRC06510
      GOTO 15                                                           PRC06520
C *** RUN OPTION NFNC=9                                                 PRC06530
   60 CONTINUE                                                          PRC06540
      CALL COSMOD                                                       PRC06550
C *** SET OR SEE VALUES IN ARRAY V                                      PRC06560
   70 CALL SET(NFNC,L1,LIST1)                                           PRC06570
      GOTO 15                                                           PRC06580
   75 CONTINUE                                                          PRC06590
      GOTO 15                                                           PRC06600
   80 GOTO 25                                                           PRC06610
      END                                                               PRC06620


      SUBROUTINE REPRT(N)                                               PRC06630
C     ******************                                                PRC06640
C     ******************                                                PRC06650
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NFOSED,REPSW,CH1SW      PRC06660
     1,CH2SW,NERR                                                       PRC06670
      COMMON/DA/LIN                                                     PRC06680
      DIMENSION FM(16),NVARS(16)                                        PRC06690
      LOGICAL VERB,AUTO                                                 PRC06700
      DATA END/3HEND/                                                   PRC06710
      IFLAG=0                                                           PRC06720
C *** TEST REPORT NUMBER FOR VALIDITY                                   PRC06730
      IF(N.GT.0.AND.N.LT.21)GOTO 1                                      PRC06740
      PRINT  61                                                         PRC06750
      RETURN                                                            PRC06760
C *** GET POSITIONS OF FORMATS AND VARIABLES                            PRC06770
    1 IREK=NPOSVR+N-1                                                   PRC06780
      LIN=IREK                                                          PRC06790
      READ(9,103) IR,IV                                                 PRC06800
  103 FORMAT(16I5)                                                      PRC06810
C *** INITIALIZE POINTER TO FORMAT RECORD                               PRC06820
      IF(IR.LT.1)GOTO 60                                                PRC06830
      NF=IR+NPOSFM-1                                                    PRC06840
C *** INITIALIZE POINTER TO VARIABLES RECORD                            PRC06850
      LV=IV+NPOSVR-1                                                    PRC06860
C *** READ FIRST VARIABLES RECORD                                       PRC06870
```

```
      LIN=LV
      READ(9,103) NVARS
C *** GET POSITION OF RECORD IN REPORT FILE
      LINE=NVARS(1)+NPOSFM-1
      GOTO 7
    6 NF=NF+1
C *** READ NEXT FORMAT RECORD
    7 LIN=NF
      READ(9,104) FM
  104 FORMAT(1X,16A4)
C *** IS THIS END OF REPORT
      IF(FM(1).EQ.END)GOTO 50
C *** ARE THERE VARIABLES WITH THIS FORMAT
      IF(LINE.EQ.NF) GOTO 10
      PRINT FM
      GOTO 6
   10 NC=NVARS(2)+2
      IF(IFLAG.EQ.1) GOTO 2
      IFLAG=1
      PRINT FM
    2 PRINT FM,(V(NVARS(J)),J=3,NC)
      LV=LV+1
C *** READ NEXT VARIABLES RECORD
      LIN=LV
      READ(9,103) NVARS
C *** GET POSITION OF NEXT RECORD IN REPORT FILE
      LINE=NVARS(1)+NPOSFM-1
      IF(LINE.NE.NF) GOTO 6
      GOTO 10
   50 RETURN
C *** ERROR
   60 PRINT 61
   61 FORMAT(/' REPORT UNKNOWN'/)
      GOTO 50
      END


      SUBROUTINE CHA2(N,L,LIS)
C     ************************
C     ************************
      COMMON NX,V(6965),VERB,AUTO,NPOSFM,NPOSVR,NPOSED,REPSW,CH1SW
     1,CH2SW,NERR
      COMMON/CA/LIN
      DIMENSION RANGE(2),F(10),FDUM(1),IDUM(1),LDUM(10)
      DIMENSION NDATE(20),LDATE(250)
      DIMENSION LIS(250)
      LOGICAL VERB,AUTO
      DATA END/3HEND/
```

```
      IVSTT=5500
      IDSTT=5600
      IF (CHISW.EQ.3) GOTO 5
      NUM=1
    2 NREK=NUM+NPOSED-1
      LIN=NREK
      READ(9,101) F,RANGE,ID
  101 FORMAT(1X,10A4,2F10.2,I5)
      IF(F(1).EQ.END)GOTO 3
      V(NUM+IVSTT)=V(ID)
      V(NUM+IDSTT)=ID
      NUM=NUM+1
      GOTO 2
    3 NUM=NUM-1
      V(IVSTT)=NUM
      GOTO 50
    5 CONTINUE
      NUM=V(IVSTT)
      IF(N.EQ.13)GOTO 6
      IF(.NOT.AUTO)GOTO 10
      IF(N.EQ.6)GOTO 10
    6 DO 7 I=1,NUM
      SAV=V(I+IVSTT)
      NED=V(I+IDSTT)
    7 V(NED)=SAV
      PRINT 8
    8 FORMAT(/' PARAMETERS RESET'/)
      IF(N.EQ.13) GOTO 50
   10 IF(L.EQ.0) GOTO 60
      DO 40 I=1,L
      NO=LIS(I)
      IF (CHISW.EQ.3) PRINT 12
   12 FORMAT(/)
      IF(NO.LT.1.OR.NO.GT.NUM)GOTO 140
      NAB=V(NO+IDSTT)
C *** GET EDIT INFO.
      NREK=NO+NPOSED-1
      LIN=NREK
      READ(9,102) F,RANGE
  102 FORMAT(1X,10A4,2F10.2)
      IF(NO.EQ.1)GOTO 34
      IF(.NOT.VERB)GOTO 18
      PRINT F
      PRINT 15,RANGE,NO,V(NAB)
   15 FORMAT(' MIN=',F9.3,'  MAX=',F9.3,'  PRESENT VALUE(',I3,
     1       ')=',F9.3)
      GOTO 20
   18 CONTINUE
```

```
      PRINT 17,NO,V(NAB)                                                PRC07820
   17 FORMAT(' VALUE(',I3,')=',F9.3)                                    PRC07830
C *** IF CHANGE REQUEST NEW VALUE                                       PRO07840
   20 IF(N.EQ.6)GOTO 40                                                 PRC07850
      IF(VERB)PRINT 22                                                  PRC07860
   22 FORMAT(' NEW VALUE=')                                             PRC07870
      IF(.NOT.VERB)PRINT 23,NO                                          PRC07880
   23 FORMAT(' (',I3,')')                                               PRC07890
      READ 24,VAL                                                       PRC07900
   24 FORMAT(F15.3)                                                     PRC07910
      IF(VAL.GE.RANGE(1).AND.VAL.LE.RANGE(2))GOTO 30                    PRC07920
      IF(VERB)PRINT 26                                                  PRC07930
      IF(.NOT.VERB)PRINT 27,RANGE                                       PRC07940
   26 FORMAT(' VALUE OUTSIDE RANGE. RETRY.')                            PRO07950
   27 FORMAT(' PERMITTED RANGE OF VALUE IS ',F10.2,' TO ',F10.2)        PRC07960
      NERR=NERR+1                                                       PRC07970
      GOTO 50                                                           PRC07980
   30 V(NAB)=VAL                                                        PRC07990
      IF(NO.NE.2)GOTO 40                                                PRC08000
   32 NS=V(IDSTT+1)                                                     PRO08010
      IMS=V(NS)                                                         PRC08020
      IMS=IMS+VAL*30.4167                                               PRC08030
      V(NS+1)=IMS                                                       PRC08040
      IDS=0                                                             PRC08050
      IYS=0                                                             PRC08060
      CALL DATE(IMS,IDS,IYS)                                            PRC08070
      PRINT 31,IMS,IDS,IYS                                              PRC08080
   31 FORMAT(/' FIRST UNIT DELIVERY DATE ',I2,2('/',I2))                PRO08090
      GOTO 40                                                           PRC08100
   34 IMS=V(NAB)                                                        PRC08110
      IDS=0                                                             PRC08120
      IYS=0                                                             PRC08130
      IMM=RANGE(1)                                                      PRC08140
      IDM=0                                                             PRC08150
      IYM=0                                                             PRC08160
      IMX=RANGE(2)                                                      PRC08170
      IDX=0                                                             PRC08180
      IYX=0                                                             PRC08190
      CALL DATE(IMS,IDS,IYS)                                            PRC08200
      CALL DATE(IMM,IDM,IYM)                                            PRC08210
      CALL DATE(IMX,IDX,IYX)                                            PRC08220
      PRINT F                                                           PRC08230
      PRINT 36,IMM,IDM,IYM,IMX,IDX,IYX,IMS,IDS,IYS                      PRC08240
   36 FORMAT(/' EARLIEST ',I2,2('/',I2),' LATEST ',I2,2('/',I2),        PRO08250
     1    ' PRESENT ',I2,2('/',I2) )                                    PROC8260
      IF(N.EQ.6)GOTO 40                                                 PRC08270
   35 PRINT37                                                           PRC08280
   37 FORMAT(/' ENTER NEW DATE')                                        PRC08290
```

```
      READ 33,NDATE                                                     PRO08300
   33 FORMAT(20A1)                                                      PRO08310
      CALL YTRACT(NDATE,20,LD,LDATE,LX,LDUM,LY,LZ,&35)                  PRO08320
      IF(LD.NE.3)GOTO 35                                                PRO08330
      IMS=LDATE(1)                                                      PRO08340
      IDS=LDATE(2)                                                      PRO08350
      IYS=LDATE(3)                                                      PRO08360
   38 FORMAT(3(I2,1X))                                                  PRO08370
      IF(IMS.GT.0.AND.IMS.LT.13)GOTO 131                                PRO08380
      PRINT 130                                                         PRO08390
  130 FORMAT(/' MONTH INVALID')                                         PRO08400
      GOTO 35                                                           PRO08410
  131 IF(IDS.GT.0.AND.IDS.LE.31)GOTO 134                                PRO08420
      PRINT 132                                                         PRO08430
  132 FORMAT(/' DAY INVALID')                                           PRO08440
      GOTO 35                                                           PRO08450
  134 IF(IYS.GE.75.AND.IMS.LE.79)GOTO136                                PRO08460
      PRINT 135                                                         PRO08470
  135 FORMAT(/' YEAR INVALID')                                          PRO08480
      GOTO 35                                                           PRO08490
  136 CALL DATE(IMS,IDS,IYS)                                            PRO08500
      IF(IMS.LT.RANGE(1).OR.IMS.GT.RANGE(2))GO TO 138                   PRO08510
      V(NAB)=IMS                                                        PRO08520
      VAL=V(2)                                                          PRO08530
      GOTO 32                                                           PRO08540
  138 PRINT 139                                                         PRO08550
  139 FORMAT(/' DATE OUTSIDE RANGE')                                    PRO08560
      GOTO 35                                                           PRO08570
  140 IF(NO.LT.1974.OR.NO.GT.1987)GOTO 60                               PRO08580
      IF(N.EQ.6)GOTO 146                                                PRO08590
      PRINT 141,NO                                                      PRO08600
  141 FORMAT(/' QUANTITY OF ALL ITEMS TO BE PRODUCED IN',I5)            PRO08610
      READ 24,VAL                                                       PRO08620
      IF(VAL.LT.0.0.OR.VAL.GT.48.0)GOTO 145                             PRO08630
      IST=NO-1971                                                       PRO08640
      DO 143 J=IST,91,15                                                PRO08650
  143 V(J+2342)=VAL                                                     PRO08660
      GOTO 40                                                           PRO08670
  145 PRINT 1145                                                        PRO08680
 1145 FORMAT(/' PERMITTED RANGE OF VALUE IS 0.0 TO 48.0')               PRO08690
      GOTO 50                                                           PRO08700
  146 PRINT 147                                                         PRO08710
  147 FORMAT(/' USE INDIVIDUAL DISPLAY FOR ITEM QUANTITIES')            PRO08720
      GOTO 50                                                           PRO08730
   40 CONTINUE                                                          PRO08740
   50 CHISW=3                                                           PRO08750
      RETURN                                                            PRO08760
   60 PRINT 61,NUM                                                      PRO08770
```

```
   61 FORMAT(' VALID ITEMS ARE 1 THRU ',I3,' OR 1974 THRU 1987'/)       PRC08780
      NERR=NERR+1                                                       PRC08790
      GOTO 50                                                           PRC08800
      END                                                               PRC08810


      SUBROUTINE XTRACT(IN,NN,LV,LVAR,NL,LIST,NFNC,NREP,*)              PRO08820
C     ***********************************************************       PRC08830
C     ***********************************************************       PRO08840
      COMMON NX,V(6965),VERB,AUTC,NPCSFM,NPCSVR,NPCSED,REPSW,CH1SW      PRC08850
     1,CH2SW,NERR                                                       PRC08860
      DIMENSION LETA(35),LETB(35),LETC(35)                              PRO08870
      DIMENSION LIST(10),LVAR(250)                                      PRC08880
      LOGICAL ALPHA,FOUND,THR,YEAR                                      PRC08890
      DIMENSION IN(72),NDIG(10)                                         PRC08900
      DATA NDIG/'0','1','2','3','4','5','6','7','8','9'/                PRC08910
      DATA ISP/' '/,ICM/','/                                            PRC08920
      DATA LETA/1HB,1HB,1HT,1HY,1HA,1HR,1HC,1HS,1HV,1HB,                PRC08930
     1          1HD,1HM,1HA,1HE,1HS,1HS,1HB,1HR,1HB,1HA,                PRC08940
     1          1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,                PRO08950
     1          1HL,1HA,1HC,1HN,1HT/                                    PRO08960
      DATA LETB/1HL,1HL,1HH,1HE,1HB,1HE,1HH,1HT,1HE,1HR,                PRC08970
     1          1HI,1HA,1HU,1HX,1HE,1HE,1HL,1HE,1HL,1H ,                PRC08980
     1          1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,                PRC08990
     1          1H ,1HL,1HR,1HE,1H /                                    PRC09000
      DATA LETC/1HK,1HK,1HR,1HA,1HC,1HP,1HA,1HC,1HR,1HI,                PRC09010
     1          1HS,1HN,1HT,1HE,1HT,1HE,1HK,1HS,1HK,1H ,                PRC09020
     1         1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,                 PRC09030
     1         1H ,1HL,1HI,1HT,1H /                                     PRC09040
      ALPHA=.FALSE.                                                     PRC09050
      FOUND=.FALSE.                                                     PRO09060
      THR=.FALSE.                                                       PRC09070
      YEAR=.FALSE.                                                      PRC09080
      DO 1 I=1,10                                                       PRC09090
    1 LIST(I)=0                                                         PRC09100
      DO 3 I=1,250                                                      PRC09110
    3 LVAR(I)=0                                                         PRC09120
      NL=0                                                              PRC09130
      LV=0                                                              PRC09140
      NVAL=0                                                            PRC09150
C *** N IS SIZE OF LETA,LETB,LETC ARRAYS                                PRC09160
      N=35                                                              PRC09170
      NSP=0                                                             PRC09180
      NFNC=0                                                            PRC09190
      NREP=0                                                            PRO09200
      DO 200 I=1,NN                                                     PRC09210
      IF(IN(I).NE.ISP) GOTO 15                                          PRC09220
      NSP=NSP+1                                                         PRC09230
```

```
      IF(NSP.GT.5) GOTO 300                                         PRO09240
      IF(.NOT.FOUND) GOTO 200                                       PRO09250
      GOTO 50                                                       PRO09260
   15 IF(IN(I).NE.ICM) GOTO 20                                      PRO09270
      GOTO 50                                                       PRO09280
   20 NSP=0                                                         PRO09290
      DO 25 J=1,10                                                  PRO09300
      IF(IN(I).NE.NDIG(J)) GOTO 25                                  PRO09310
      IF(ALPHA) GOTO 200                                            PRO09320
      IF(.NOT.FOUND)IFST=I                                          PRO09330
      FOUND=.TRUE.                                                  PRO09340
      NVAL=NVAL*10+J-1                                              PRO09350
      GOTO 200                                                      PRO09360
   25 CONTINUE                                                      PRO09370
C *** MUST BE ALPHA                                                  PRO09380
   30 IF(FOUND) GOTO 35                                             PRO09390
      FOUND=.TRUE.                                                  PRO09400
      ALPHA=.TRUE.                                                  PRO09410
      IFST=I                                                        PRO09420
      GOTO 200                                                      PRO09430
   35 IF(.NOT.ALPHA) GOTO 151                                       PRO09440
      GOTO 200                                                      PRO09450
   50 IF(ALPHA) GOTO 100                                            PRO09460
      IF(.NOT.YEAR) GOTO 80                                         PRO09470
      IF(.NOT.THR) GOTO 70                                          PRO09480
      K=LIST(NL)+1                                                  PRO09490
      IF(K.EQ.1)GOTO 151                                            PRO09500
      IF(NVAL.LT.K) GOTO 151                                        PRO09510
      DO 60 J=K,NVAL                                                PRO09520
      NL=NL+1                                                       PRO09530
      LIST(NL)=J                                                    PRO09540
   60 CONTINUE                                                      PRO09550
      THR=.FALSE.                                                   PRO09560
      GOTO 190                                                      PRO09570
   70 NL=NL+1                                                       PRO09580
      LIST(NL)=NVAL                                                 PRO09590
      GOTO 190                                                      PRO09600
   80 IF(.NOT.THR) GOTO 85                                          PRO09610
      K=LVAR(LV)+1                                                  PRO09620
      IF(K.EQ.1)GOTO 151                                            PRO09630
      IF(NVAL.LT.K) GOTO 151                                        PRO09640
      DO 83 J=K,NVAL                                                PRO09650
      IF(LV.EQ.50) GOTO 300                                         PRO09660
      LV=LV+1                                                       PRO09670
   83 LVAR(LV)=J                                                    PRO09680
      THR=.FALSE.                                                   PRO09690
      GOTO 190                                                      PRO09700
   85 LV=LV+1                                                       PRO09710
```

```
      LVAR(LV)=NVAL                                                     PRC09720
      IF(LV.EQ.50) GOTO 300                                             PRC09730
      GOTO 190                                                          PRC09740
  100 DO 150 J=1,N                                                      PRC09750
      IF(IN(IFST).NE.LETA(J)) GOTO 150                                  PRC09760
      IF(LETB(J).EQ.ISP.AND.I.EQ.IFST+1)GOTO 99                         PRC09770
      IF(IN(IFST+1).NE.LETB(J)) GOTO 150                                PRC09780
      IF(IN(IFST+2).NE.LETC(J)) GOTO 150                                PRC09790
   99 CONTINUE                                                          PRC09800
      II=J                                                              PRC09810
      IF(J.GT.5.AND.J.LE.19) II=6                                       PRC09820
      IF(J.GT.19) II=7                                                  PRC09830
      GOTO(101,102,103,104,105,106,107),II                              PRC09840
  101 CONTINUE                                                          PRC09850
      GOTO 190                                                          PRC09860
  102 CONTINUE                                                          PRC09870
      GOTO 190                                                          PRC09880
  103 THR=.TRUE.                                                        PRC09890
      GOTO 190                                                          PRC09900
  104 CONTINUE                                                          PRC09910
      THR=.FALSE.                                                       PRC09920
      GOTO 190                                                          PRC09930
  105 PRINT 1105                                                        PRC09940
 1105 FORMAT(' LINE ABORTED')                                           PRC09950
      RETURN 1                                                          PRC09960
  106 NFNC=J-5                                                          PRC09970
      GOTO 190                                                          PRC09980
  107 NREP=J-19                                                         PRC09990
      IF(J.EQ.26)YEAR=.FALSE.                                           PRC10000
      GOTO 190                                                          PRC10010
  150 CONTINUE                                                          PRC10020
  151 PRINT 152,IN(IFST),IN(IFST+1),IN(IFST+2)                          PRC10030
  152 FORMAT(' INVALID FIELD- ',3A1)                                    PRC10040
      NERR=NERR+1                                                       PRC10050
      RETURN 1                                                          PRC10060
  190 ALPHA=.FALSE.                                                     PRC10070
      FOUND=.FALSE.                                                     PRC10080
      NVAL=0                                                            PRC10090
  200 CONTINUE                                                          PRC10100
  300 RETURN                                                            PRC10110
      END                                                               PRC10120
```

APPENDIX F. ANALYTICAL MODELS REQUIRED BY COMMAND AND CONTROL PROGRAMS

CONTROL PROGRAM	ANALYTICAL MODEL TITLE	SUBROUTINE NAME
SYSTEM	Mission Simulation	MISSIM
	Force Structure Effectiveness	FORCE
	Life Cycle Cost	LIFE
	Logistics	LOG
	Availability	AVAIL
	Capability	CAPAB
	Reliability	RELIB
INCENT	Incentive Contract	ICM
PROGPL	Engineering Change Proposal	PROCOS
	Program Planning	COSMOD

APPENDIX G. TYPING CONVENTIONS AND SAMPLE EXERCISE.

Typing Conventions

The following typing conventions apply to the use of all computer-assisted exercises. Input is accepted from the user's terminal only at certain points of the computer exercise. Every user input line must be terminated by pressing the RETURN key. No data transmission of the line typed will take place until the RETURN signal has been received by the computer. If errors are noticed in the line being typed before the RETURN key is pressed, the errors can be corrected by the user. The @ character is interpreted to mean "delete the last character typed"; @@ will delete the last two characters, and so on. Spaces are counted as characters. The BACKSPACE key should not be used. The user may delete the entire current line being typed by typing the ¢ character and pressing the RETURN key. The new line may then be restarted.

If more than one data value is to be typed on a single line, the values must be separated by a comma or a blank character. Data items may not be separated by more than five blank characters. Numeric data may contain only the digits 0-9, an optional preceeding plus or minus sign, and a decimal point. Commas may not be used to group thousands or millions. The comma is interpreted as a delimiter between two data values. All numeric input data to the program requires a decimal point. Alphanumeric data may contain all letters, numbers, and punctuation characters except a blank character, a comma, or the @ and ¢ characters.

Where percentages are requested from the user, they should be typed as whole numbers and not as their fractional equivalents, e. g. six and one-half percent would be entered as 6.5, not as 0.065. The percent sign should not be used.

<u>Program commands</u>

Use of the computer models is initiated by typing in the model name after the log-in procedure has been completed, e.g. SYSTEM, INCENT, or PROGPL. This command will cause the model routines required by the exercise to be loaded into working storage and will provide access to the exercise data files. Operation of the computer models during the exercise is controlled by the user's choice of a number of commands.

When the computer program types the message COMMAND ? on a line, it is at command level. At this point, the user may enter any of the command words allowed by the particular exercise, plus other information that may be required to execute the selected command. After the requested command, with the exception of the command STOP, has been completed, the program will return to the command level.

Examples of various program commands are as follows. It is only necessary to type the first three characters of each command. The remaining characters are optional. Commands may be typed in either upper or lower case letters. Responses will always be printed in upper case letters.

```
system                              (program name)

EXERCISE NUMBER ?

3.                                  (exercise number desired)

BASE DATA READ FROM FILE FT03F001

BASE DATA READ FROM FILE FT03F002

DO YOU WANT TO SEE THE LIST OF VARIABLES (YES OR NO) ?

yes

CREATED REPORT FILE FT09F001

EXERCISE 3.

THE STUDENT MAY VARY THE FOLLOWING SYSTEM PARAMETERS

PARAMETER  1
SYSTEM X AVAILABILITY
MIN=   0.750  MAX=   0.990   PRESENT VALUE( 1)=        0.800

PARAMETER  2
SWIM SPEED (KNOTS)
MIN=   1.000  MAX=   6.000   PRESENT VALUE( 2)=        4.000

PARAMETER  3
LAND CRUISE SPEED (MPH)
MIN=  20.000  MAX= 70,000    PRESENT VALUE( 3)=       50.000

PARAMETER  4
CRUISE RANGE (MILES)
MIN=  50.000  MAX=300.000    PRESENT VALUE( 4)=      200.000

PARAMETER  5
RATE OF FIRE
MIN=   0.200  MAX=   5.000   PRESENT VALUE( 5)=        0.500

                              (the program response has been
                               edited for this example)
```

The command DISPLAY permits the user to obtain the current values of selected exercise input parameters. The command DISPLAY n will cause the magnitude of the n th parameter to be printed at the terminal. Examples of variations of the command are shown below:

```
display 1

SYSTEM X AVAILABILITY
MIN=     0.750    MAX=     0.990    PRESENT VALUE( 1)=        0.800

COMMAND ?

display 2 3

SWIM SPEED (KNOTS)
MIN=     1.000    MAX=     6.000    PRESENT VALUE( 2)=        4.000

LAND CRUISE SPEED (MPH)
MIN=    20.000    MAX=    70.000    PRESENT VALUE( 3)=       50.000

COMMAND ?

dis 4,5

CRUISE RANGE (MILES)
MIN=    50.000    MAX=   300.000    PRESENT VALUE( 4)=      200.000

RATE OF FIRE
MIN=     0.200    MAX=     5.000    PRESENT VALUE( 5)=        0.500

dis 6 thr 8

REACTION TIME (MINS)
MIN=     1.000    MAX=    30.000    PRESENT VALUE( 6)=       10.000

RANGE ERROR
MIN=     5.000    MAX=   200.000    PRESENT VALUE( 7)=       40.000

DEFLECTION ERROR
MIN=     5.000    MAX=   200.000    PRESENT VALUE( 8)=       30.000
```

The command CHANGE n allows the user to revise the current stored value of input parameter n. The program will indicate the current value of the parameter and will request that the user enter the revised value of the parameter. The program also indicates the permitted maximum and minimum values for the particular parameter. A request to change a parameter value outside its stated limits will result in an error message. Examples of variations of the command are shown below:

```
change 1

SYSTEM PARAMETERS RESET


SYSTEM X AVAILABILITY
MIN=     0.750     MAX=     0.990     PRESENT VALUE( 1)=          0.800
ENTER NEW VALUE

.87

COMMAND ?


cha

SYSTEM PARAMETERS RESET

VALID ITEMS ARE 1 THRU    9


change 10

SYSTEM PARAMETERS RESET


VALID ITEMS ARE 1 THRU    9
```

```
COMMAND ?

change 1

SYSTEM PARAMETERS RESET


SYSTEM X AVAILABILITY
MIN=     0.750     MAX=     0.990     PRESENT VALUE( 1)=          0.800
ENTER NEW VALUE

.50
VALUE OUTSIDE RANGE.  RETRY.
```

The EXECUTE command is used to initiate the execution of the analytical models based on the current set of input values. When execution of the analytical models has been completed, the message COMMAND ? will be printed. No inputs may be entered on the terminal console while the analytical models are being executed. An example of the EXECUTE command is shown below:

```
COMMAND ?

execute

COMMAND ?
```

The BRIEF command sets a program switch that causes messages to be printed in an abbreviated format. The VERBOSE command sets the switch causing messages to be printed in a detailed format. The program is initially set to the verbose mode. After a command has been issued to set the program into the brief mode messages will continue to be printed in an abbreviated format until a VERBOSE command is issued.

Examples of the BRIEF and VERBOSE commands is shown below:

```
brief

BRIEF MODE

COMMAND ?

dis 1

PRESENT VALUE( 1)        0.800

COMMAND ?

verbose

VERBOSE MODE

COMMAND ?

dis 1

SYSTEM X AVAILABILITY
MIN=    0.750    MAX=    0.990    PRESENT VALUE( 1)=        0.800
```

The command VARY n allows the user in Exercise 3. to make the System X analytical models perform up to ten iterations of the model cycle, each with a different value for the specified parameter. The program will request an initial value for parameter n. This value is used for the first iteration. A maximum or final value is then requested. The program will then ask for the incremental value, which is the value to be added to the initial value at each iteration. A maximum of ten iterations will be performed, even if the final value is not reached. However, if the final value condition is reached in less than ten iterations, execution will stop. With each iteration a report line is printed showing

the parameter value, Life Cycle Cost, and Discounted Life Cycle Cost.

An example of the VARY command is shown below:

```
vary 4

SYSTEM PARAMETERS RESET

CRUISE RANGE (MILES)

MIN=     50.000   MAX=     300.000  PRESENT VALUE ( 4)  =     200.000

ENTER INITIAL VALUE FOR PARAMETER  4

60.

ENTER FINAL VALUE FOR PARAMETER  4

180.

ENTER INCREMENT (MAX OF 10 ITERATIONS)

30.

      COST VS
CRUISE RANGE (MILES)


-----------------------------------------------------------------
```

PARAMETER 4 VALUE	LIFE CYCLE COST	DISCOUNTED LIFE CYCLE COST
80.000	1767084	791759.
90.000	1767084	791759.
120.000	1767084	791759.
150.000	1767084	791759.
180.000	1767084	791759.

The command REPORT a permits the user to print the standard report with the report identification code a. Varying reports are available depending upon the exercise number being executed. There is a maximum of eleven reports available identified by the letters A through K. A REPORT command, with an appropriate identifier must be typed in for each report selected. An example of the REPORT command is shown below:

```
report k
```

EXERCISE 3.

SENSITIVITY ANALYSIS.

VTM SUMMARY REPORT
($000)

DEPLOYMENT LEVEL (YEAR 10)	230.
TOTAL LIFE CYCLE COST	1767084.
DISCOUNTED LIFE CYCLE COST	791759.
AVERAGE VTM SINGLE-SHOT EFFECTIVENESS	0.419
LAUNCHER SURVIVABILITY (BASED ON 30 DAY WAR)	0.216

The STOP command should be entered when the user desires to terminate the current exercise. After giving the STOP command, a new exercise can be initiated by entering the program name. The STOP

command must be given before the user can logout from the terminal console. An example of the STOP command is shown below:

COMMAND ?

stop

LIST OF REFERENCES

1. Smith, J. R., "The Conqueror Missile System," Defense Management Journal, v. 8, no. 3, p. 80, October 1972.

2. IBM Corporation, Control Program-67/Cambridge Monitor System (CP-67/CMS) Version 3.1, User's Guide.

3. Defense System Management School, System X, Systems Programming Documentation, Command and Control Programs, Fort Belvoir, Virginia.

4. Defense Systems Management School, System X, Mission Simulation Model, Computer Model User's Guide, Fort Belvoir, Virginia.

5. Defense Systems Management School, System X, Mission Simulation Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

6. Defense Systems Management School, System X, Force Structure Effectiveness Model, Computer Model User's Guide, Fort Belvoir, Virginia.

7. Defense Systems Management School, System X, Force Structure Effectiveness Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

8. Defense Systems Management School, System X, Logistics Model, Computer Model User's Guide, Fort Belvoir, Virginia.

9. Defense Systems Management School, System X, Logistics Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

10. Defense Systems Management School, System X, Life Cycle Cost Model, Computer Model User's Guide, Fort Belvoir, Virginia.

11. Defense Systems Management School, System X, Life Cycle Cost Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

12. Defense Systems Management School, System X, Availability Model, Computer Model User's Guide, Fort Belvoir, Virginia.

13. Defense Systems Management School, System X, Availability Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

14. Defense Systems Management School, System X, Capability Model, Computer Model User's Guide, Fort Belvoir, Virginia.

15. Defense Systems Management School, System X, Capability Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

16. Defense Systems Management School, System X, Reliability Model, Computer Model User's Guide, Fort Belvoir, Virginia.

17. Defense Systems Management School, System X, Reliability Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

18. Defense Systems Management School, System X, Contractor Incentive Contract Model, Computer Model User's Guide, Fort Belvoir, Virginia.

19. Defense Systems Management School, System X, Contractor Incentive Contract Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

20. Defense Systems Management School, System X, Program Planning Model, Computer Model User's Guide, Fort Belvoir, Virginia.

21. Defense Systems Management School, System X, Program Planning Model, Computer Model Design and Programmer's Guide, Fort Belvoir, Virginia.

22. Defense Systems Management School, System X, Data Base Specifications and Definitions, Fort Belvoir, Virginia.

23. Defense Systems Management School, System X, Data Base Documentation, PMBAS1 Input Data, Fort Belvoir, Virginia.

24. Defense Systems Management School, System X, Data Base Documentation, PMBAS2 Input Data, Fort Belvoir, Virginia.

25. Defense Systems Management School, System X, Data Base Documentation, PMBAS3 Input Data, Fort Belvoir, Virginia.

26. Defense Systems Management School, System X, Data Base Input, BAS1517, Incentive Contract Model, Fort Belvoir, Virginia.

27. Defense Systems Management School, System X, Base Data Input, BAS2122, BAS21, BAS21A, BAS22, Program Planning Model, Fort Belvoir, Virginia.

28. Defense Systems Management School, System X, Data Base Definition, Program Planning Model, Fort Belvoir, Virginia.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	CDR R. M. Hanna, Code 53Hq Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
4.	CDR H. H. J. Nicholson Weapons Department USS America (CVA-66) FPO, New York, New York 09501	1
5.	Professor M. B. Kline, Code 55Kx Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	Naval Postgraduate School, Code 55 Department of Operations Research and Administrative Sciences Monterey, California 93940	1
7.	Sharon D. Raney W. R. Church Computer Center Naval Postgraduate School Monterey, California 93940	1

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Adaptation of System X Computer-assisted Project Management Exercises For Use at the Naval Postgraduate School

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*
Master's Thesis; June 1973

5. AUTHOR(S) *(First name, middle initial, last name)*

Herbert Henry Joseph Nicholson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1973	114	28

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

System X is a series of project management oriented case studies supported by an interactive time-sharing computer program simulating the analysis and evaluation of a hypothetical surface-to-surface guided missile system acquisition program. The computer-assisted exercises operate from baseline data supplied by a data base, along with system parameters set by the user, in order to compute various deterministic statistics regarding system performance and system costs. The system parameter values may be readily changed and the computations repeated. A desired result may be obtained by repeated iterations of the process.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
NAGEMENT TRAINING						
OJECT MANAGEMENT						
STEM X						
STEMS ACQUISITION SIMULATION						

Thesis
N49
c.1
145280
Nicholson
Adaptation of system
x computer-assisted project management exercises for use at the Naval Postgraduate School.