

AD-A035 753

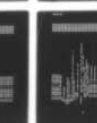
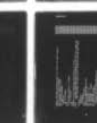
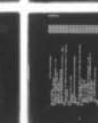
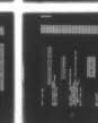
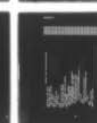
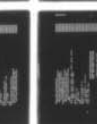
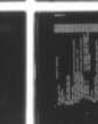
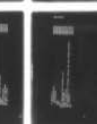
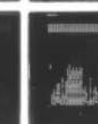
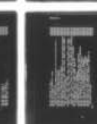
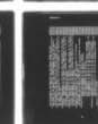
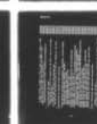
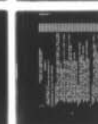
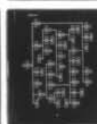
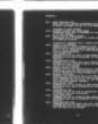
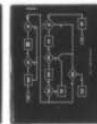
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/6 5/1
JOINT TEST FORCE SIMULATION.(U)

DEC 76 J P EGGERS
6OR/SM/76D-7

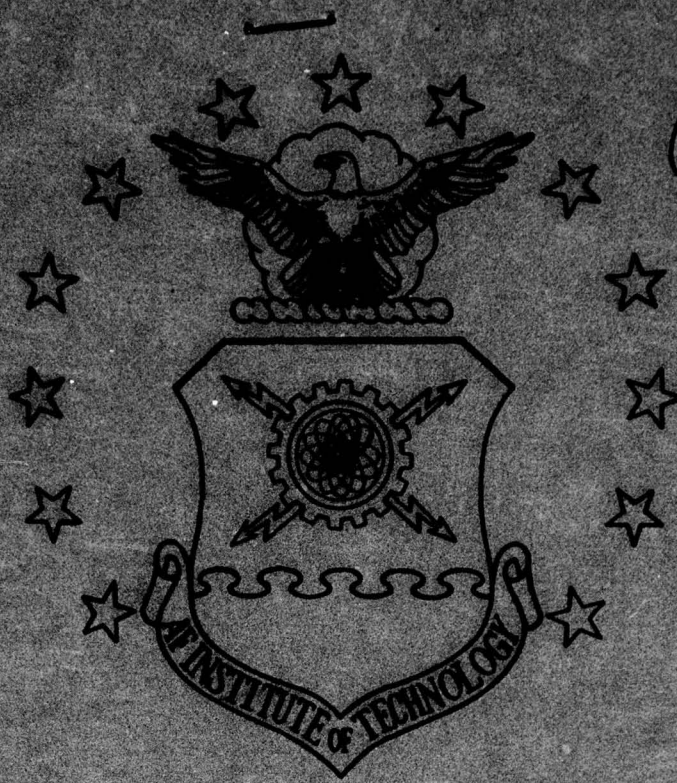
UNCLASSIFIED

NL

1 OF 2
AD
A035753



ADA 035753



UNITED STATES AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC
RECEIVED
FEB 18 1977
A

11 Dec 76

12 183p.

6 JOINT TEST FORCE
SIMULATION.

9 Masters' THESIS,

14 GOR/SM/76D-7

10 James P. Eggers
Captain USAF

Approved for public release; distribution unlimited.

DDC
RECEIVED
FEB 18 1977
RECEIVED

012 225

mt

A

JOINT TEST FORCE
SIMULATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

James P. Eggers, B.S.

Captain USAF

Graduate Operations Research

December 1976

Approved for public release; distribution unlimited

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DOC	Diff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. CODE/SPECIAL
A	

Preface

This research project is intended to provide the members of the 3246 Test Wing at Eglin AFB, Florida, with a computer simulation model which will be helpful in estimating personnel requirements and work completion dates for a joint test force.

Appreciation and gratitude are extended to Mr. Kenneth Mathiasmeier, of the 3246 Test Wing, for his interest and cooperation in determining the model requirements. His knowledge of the organization of a test was valuable in obtaining the necessary background to develop the generalized event-oriented simulation model of work activities.

Appreciation is extended to Mr. Charles Etheridge and Ms. Pam Martin of CACI, Inc., for their expertise and understanding while learning to use SIMSCRIPT II.5.

Thanks is also extended to Mr. William Drake of AFLC/XRS for his technical advice on programming a simulation model of this magnitude. His suggestions were followed without exception and proved invaluable in time savings.

Special thanks are due to Major Charles McNichols and Colonel Ronald Luhs of the AFIT Department of Systems Management. As thesis advisor and reader they provided guidance, objectivity, and expertise.

Most sincere gratitude is extended to my wife, Martha, for her encouragement throughout this endeavor and her patience during the numerous long nights required to develop this model.


JAMES P. EGGERS

Contents

	<u>Page</u>
Preface.....	ii
List of Figures.....	iv
Abstract.....	v
I. Introduction.....	1
Background.....	1
Current Knowledge.....	4
Previous Work.....	5
Statement of Problem.....	6
Scope.....	6
II. Simulation Model.....	8
General Concept.....	8
Capabilities and Limitations.....	10
Language Selection.....	12
Model Input.....	13
Model Description.....	17
Validation.....	36
III. Illustrative Example.....	38
Description of the Test.....	38
Data Input.....	38
Discussion of Output.....	47
IV. Conclusions and Extension of Research.....	49
Bibliography.....	51
Appendix A. Definition of Terms.....	53
Appendix B. Flow Chart.....	60
Appendix C. Program Listing.....	66
Appendix D. Data Input Instructions.....	157
Appendix E. Sample Data.....	163
Appendix F. Sample Output.....	167
Vita.....	174

List of Figures


<u>Figure</u>		<u>Page</u>
1	Test Cycle.....	2
2	Simplified Flow-Chart.....	9


Abstract

This study presents the background, development, and use of a generalized event-oriented simulation model of work activities. The particular question addressed in this research effort was: Can an event-oriented simulation model be developed which will be useful in estimating work completion dates and personnel requirements using the information presented in a work directive? The program is written in SIMSCRIPT II.5 and is adaptable to a particular joint test force by using free-format data inputs.

Any test force which is responsible for the performance of activities may be studied using this model. Any number of activities, missions, test phases, workers, and levels of workers may be defined through the input data. Detailed input instructions are provided to tailor the model to the analyst's requirements. Numerous options are available to include: worker skill factors, mission reduction or cancellation probabilities, activity performance capability with reduced personnel, reports, parametric studies, and replications of the entire data package.

The function of each routine and event is discussed and a complete program listing is appended. A sample study is provided to illustrate many of the significant features.



I. Introduction

Background

The 3246th Test Wing, Eglin AFB, Florida, is the primary Armament Development and Test Center (ADTC) organization which provides the overall management for developmental testing. It determines ADTC test workload, identifies resources needed to accomplish testing, conducts and reports those tests for which it has been designated the responsible test organization, and supports those tests for which it has been designated the participating organization. It receives test requests from other AFSC organizations, USAF commands, DoD departments, U.S. government agencies, and allied government's national defense forces.

When a test request is received from one of these organizations, the first phase of five phases in a test cycle (see Figure 1) is initiated. Upon initial contact by a test requester, a programming engineer is designated to conduct the initial test planning. After informal contact between the test requester and the programming engineer, the test requester sends a formal request to the Test Wing; and the programming engineer prepares an initial contact summary. The Engineering Test Directorate then assigns a test analyst/test design engineer and a test engineer to design a proposed test.

A test planning conference is scheduled as a first step in drafting the work directive. The conferees determine the scope of the test, review the test objectives, and may discuss the Statement of Capability which must be prepared for accepted tests. The expected results of the conference include an outline of the test, an initial testing schedule,

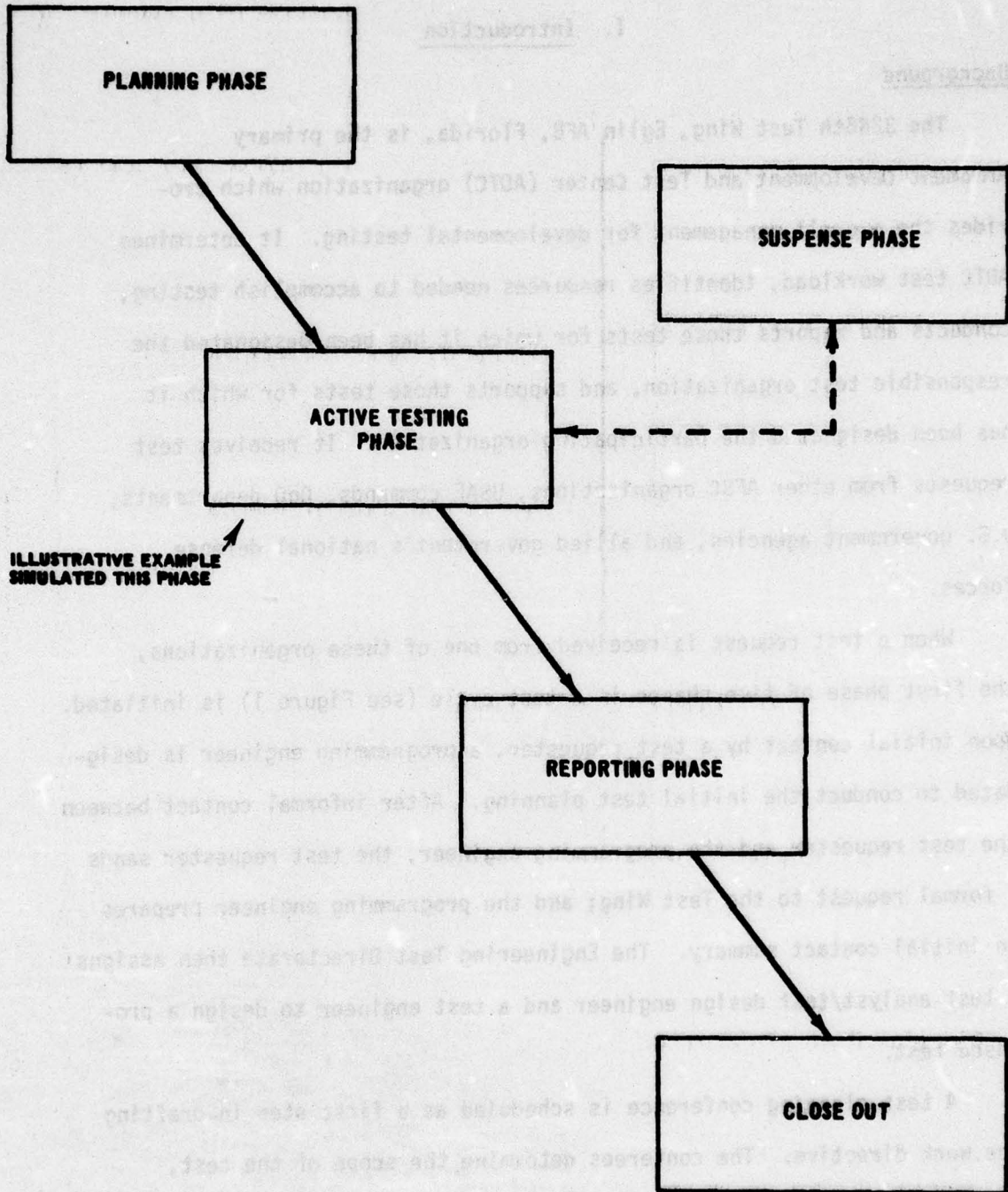


Fig. 1. Test Cycle

required facilities, and a designation of key personnel.

The test design team prepares the method of test annex which explains how the test is to be conducted and controlled, what data are to be collected, and how the data will be analyzed. Specifically, they must refine the test objectives, determine assessment criteria, identify required resources, determine environmental consequences, describe test control procedures, and estimate the data reduction and analysis effort.

After receiving the method of test, technical annexes, and support annexes, the programming engineer completes the work directive. When signed by the ADTC Commander, this directive orders the execution of a test by ADTC assigned personnel. Included in the directive are the objectives, schedule, initial priority, key personnel, and participating agencies. At this time, the planning requirements have been completed.

After completion of the planning phase several activities must be completed prior to the first mission. The test engineer must insure that the test items are available; checklists have been approved; working space for the test item contractor has been provided; and that test-bed aircraft, support aircraft, instrumentation, and radars are available for scheduling during the period of planned testing. He is then ready to submit a mission request.

When a test enters the active phase, the test engineer must be prepared to revise his testing schedule to use alternate equipment, personnel, or facilities rather than accept a delay. Delays can be expected for failures of the test item or test resources, and for scheduling conflicts because of higher priority missions. Available alternatives to scheduling delays include off-hour scheduling, accepting alternate

missions, and attempting piggyback missions with tests of higher priority. If it is determined that delays will extend the completion date, an amendment to the work directive must be initiated and a revised schedule for use of test support resources must be determined.

Tests may enter a suspense phase if a delay of at least 30 days is anticipated. A valid reason for suspending active testing might be a requirement for intermediate data reduction to help determine follow-on procedures. Secondly, the quantity of data collected may be so excessive that it cannot be properly analyzed. It may be necessary to implement improved data reduction techniques.

The reporting cycle begins on the day following the last mission, and normally 60 days are allowed to complete this cycle. Since technical reports are the chief product of the Test Wing, guidelines to include a schedule have been developed. Final data reduction and analysis is completed by the 14th day, and the draft must be completed by the 26th day. Coordination, editing, and review are completed by the 55th day; and then the Wing Commander has until the 60th day for review, approval, and signature.

The final phase of the test cycle is closeout and normally lasts from 45 to 60 days. During this period all cost accounting adjustments are finalized and letters of termination or completion are sent to all test support and other interested agencies. Thus, five phases of a test have been highlighted (Ref 3:1-44).

Current Knowledge

A recurring management problem is the allocation of scarce resources among alternative ends (Ref 15:4). Test forces which study, test, and

report the results of various Air Force projects are competing for the scarce resources of men and equipment. Inflation and current budgetary policies have effectively reduced the amount of resources available for allocation within the Air Force. Therefore, more sophisticated management tools are necessary to consider the range of available alternatives and to determine which alternative most effectively produces the desired results.

In the early 1960's simulation models, which use digital computers to provide insight into complex processes, were designed and developed. The purpose of a computer simulation is to gather information on the characteristics of a system in order to provide a method of predicting the dynamic characteristics of an organization and consequently to provide a basis for making decisions. According to Mr. Richard L. VanHorn, "Simulation offers the most flexible and realistic representation for complex problems of any quantitative technique" (Ref 18:256). Also according to George S. Fishman and Phillip J. Kiviat, "Simulation made it possible to carry out fully integrated system analyses which were generally far too complex to be carried out analytically" (Ref 6:2).

Previous Work

A Defense Documentation Center search submitted for simulation test facilities, simulation facilities, and shop scheduling systems produced no indication that a study like this one has been accomplished. A study for a similar organization, the 4950th Test Wing at WPAFB, Ohio, entitled A Multi-Project Scheduling Program, was completed in February, 1974. The problem which was modeled in the current study differs significantly from the multi-project scheduling program. The current study considered

the effects on completion dates of changing test force composition for a single project. The previous study provided the expected schedule for cycling numerous tests through an existing structure. Discussions with members of the 3246th Test Wing at Eglin AFB, Florida, also indicate that no similar studies have been performed for their organization.

Statement of Problem

An increasing number of complex tests in the 3246th Test Wing, Eglin AFB, Florida, are being performed by dedicating personnel to a joint test force (JTF). The Test Wing requested the development of an event-oriented simulation of work activities performed by the JTF and support personnel. By adapting this simulation to a current JTF, the director can estimate work completion dates, personnel requirements, and the relationship between the number of assigned personnel and the work completion dates.

The particular question addressed in this research effort was: Can an event-oriented simulation model be developed which will be useful in estimating work completion dates and personnel requirements using the information presented in a work directive?

Scope

The purpose of this research was to develop a generalized event-oriented simulation model of work activities. This simulation is adaptable to a particular joint test force. Using information presented in the work directive for a test, it provides insight in estimating personnel requirements and work completion dates. Since the inputs to the simulation will be extracted from the work directive, the phases following the planning phase can be simulated.

This simulation will enable joint test force directors to estimate

the overall personnel requirements of the test force relative to a specified completion date. Secondly, they will be able to determine the relative effects of assigning various numbers and levels of workers to a test. Through the use of simulation, they will be able to identify potential problems, and experiment with various personnel configurations and work rules. Thus, by using the simulation model suitable alternatives may be found which will avoid or alleviate an indicated problem.

Since modeling resources for the entire Eglin AFB complex is beyond the scope of this study, probabilities to determine the availability of resources and activities which are not under the control of the joint test force director were subjectively assessed by the point of contact within ADTC.

II. Simulation Model

General Concept

To understand a simulation model one must note the relationship between the concepts being simulated and the simulation model. In order to determine the relationship between the number of workers assigned to a JTF and the resulting completion date, the following two items are of primary importance: What (activity) must be performed? Who (worker) will perform the activity? A simplified flow-chart showing an application of the basic concept of the simulation model is diagrammed in figure 2. A complete flow-chart for the simulation model is listed in Appendix B. For illustrative purposes, the active testing phase of a test cycle was simulated.

Activity/Mission/Test Phase. In order to satisfy the requirements of the test objectives developed during the planning phase, certain actions (activities) must be accomplished. The work directive specifies a sequence of actions which must be completed to satisfy each objective. These sequences of actions are called missions. Frequently, a mission must be completed several times or with slight variations in order to satisfy the objectives. In the simulation model, this completion of similar missions is called a test phase. A test phase specifies the number of similar missions which must be completed.

An activity is the basic building block in the actual test as well as in the simulation model. A common characteristic of all activities is that a quantity of time must elapse prior to completion. An analyst will define the activities to match the level of detailed desired

for modeling a test. In the illustrative example, 14 activities were defined. At this selected level of detail, "conduct an air mission" was designed as an activity. This portion of a mission could have been modeled at a greater level of detail by defining activities such as drive to the hangar, become airborne, fly to the range, make twelve passes over a target, etc.

Using this simulation model, a test can be simulated at whatever level of detail that the inputs can be accurately determined. The same test can be simulated at different aggregate levels depending on the desired accuracy of the output and the level of effort that one is willing to expend in developing a data package for the simulation model.

Worker. The second major modeling concept is: Who (worker) will perform the activity? Different levels of workers are defined to accommodate the idea that certain activities are accomplished by only a certain type of person. In the simulation model workers may be stereotyped into different levels of workers. The analyst may relate these levels to such items as job titles, AFSC's, etc. Each worker may be assigned a skill factor for each activity. This factor adjusts the activity time to reflect the realization that all people do not perform an activity in the same length of time. These two concepts form the basis of the model. Definitions are provided in Appendix A to clarify the meaning of the various simulation terms that are used in discussing the model.

Capabilities and Limitations

Capabilities. Any test force which is responsible for the perform-

ance of activities may be studies using the model developed in this research effort. Any number of activities, missions, test phases, and workers may be defined through the input data. Activities may be grouped in any order to represent a mission in a particular test phase. Additional flexibility is provided by allowing worker requirements, activity sequencing, and mission priorities to be defined through the input data. Situations in which there are no similarities between missions can be represented at the expense of the additional data inputs to separately describe each mission.

Three worker level preferences may be specified for each activity. The selection of members from these levels affects the activity time, by averaging the skill factors of the assigned workers. After team assignments have been made, the remaining qualified workers are assigned to approved activities on a partial-team basis. The selected activity time is increased in proportion to the number of workers required. Activity times may be specified for each test phase and, consequently, reduce the number of separately defined activities. Any number of cancellation factors may be entered for each mission, and each activity has the capability of reducing the mission effectiveness. Numerous options are listed under Options in the Model Input section.

Limitations. A simulation model is an effective method for studying complex systems. However, a model should not be used without a thorough understanding of the meaning of each required input. The model does not refuse to give answers simply because the inputs are inaccurate or unreasonable. Since there is a certain amount of

abstraction in collecting or constructing any data package, there exists a possibility of biasing the results. In a stochastic model, the sensitivity of these inputs should be tested to insure that the estimates do not seriously bias the results.

Language Selection

Selection of a simulation language is determined primarily by the nature of the problem, and secondly by the accessibility of the various languages. Different languages are designed to model systems involving a flow of discrete items. The problem considered is one of discrete items. Discrete-model languages perform the following functions: advance the simulated time, search for the next scheduled event, place a new event on the event file, and test for status matches to trigger conditional events (Ref 5:142).

The two most widely used event-oriented simulation languages are General Purpose Simulation System (GPSS) and SIMSCRIPT. GPSS automatically performs the following functions: collects data describing a model's simulated behavior, prints summaries of the data at the end of the simulation, and provides the format statements indicating how the data summaries are to be displayed. GPSS has incorporated much of the underlying logic of discrete event simulation, and thereby permits a model builder to use the language without understanding its internal logic. Use of a high level language in this manner invariably results in model invalidities which are frequently subtle in nature, and therefore are difficult to detect (Ref 16:vii).

SIMSCRIPT is a general language which can do everything that can

be done in FORTRAN. The model builder uses named entities to represent objects in the system and has complete freedom to define the properties of these entities. SIMSCRIPT also has an extensive report-generating feature which displays the selected output in a prescribed format. Although SIMSCRIPT performs the basic functions of event-oriented languages, it requires an analyst to translate the problem formulation into appropriate model representation and provide some of the programming details. (Ref 17:214)

SIMSCRIPT II.5 was selected for this study for several reasons. According to George S. Fishman and Phillip J. Kiviat, "Very minor simplifying assumptions can generate output processes whose behavior differs considerably from what is desired." (Ref 6:14) Since SIMSCRIPT II.5 is a full programming language, the number of assumptions necessary in modeling the test force were minimal. Also since every statement in every routine is written by the programmer, there is no guesswork involved as to the function of a particular segment of the program. Finally, the extensive report generating feature of SIMSCRIPT II.5 permitted formatted reports tailored to meet the requirements of weekly, monthly, and final reports.

Model Input

Instructions. Detailed data input instructions are listed in Appendix D. However, prior to answering these questions, the basic structure of a test must be determined.

Activities. First the required activities must be identified from the required resources for each mission and the test objectives.

Criteria used to determine the number of different kinds of activities that need to be identified is generally based on the required equipment and the degree of difficulty. Specific questions that must be considered include:

1. Will the activity be performed during normal duty hours?
2. What type of distribution will describe the time required to perform the activity?
3. What levels of workers can perform this activity?
4. Can the activity be expected to reduce the mission effectiveness?
5. By what percentage is the activity expected to reduce the mission effectiveness?

After defining these basic characteristics of the activities, the activities are further defined by entering the time function parameters and the number of workers required in each test phase. Thus, the data input requirements have been reduced since the basic activity data are not repeated for each test phase.

Missions. Secondly, based on similarities in activity requirements and timing requirements, missions must be constructed for each test phase. Normally, missions in a test phase consist of a standard sequence of activities; however, minor differences can be accommodated by using external event notices to create a mission. External missions select a subset of the standard activity sequence, and also permit an activity to be added to the sequence by entering the required information. By using this feature, the number of test phases required

may be reduced.

Workers. Thirdly, the categories or level of workers must be determined based on the requirements for each activity. The actual number of workers in each level and their skill factors in performing each activity can be varied for each simulation run. The workers are a primary variable in determining the completion date. To minimize data input, the default of the same skill factor for all workers is recommended for the initial runs.

Program Variables. After the basic structure is determined, entries regarding quantities, times, probabilities, priorities, programming options, and programming control variables are entered. Names of workers and activities should be no more than six characters.

Activity Test Phase Codes (ATPC). Activity test phase codes determine the order in which activities will be performed. They are composed of two segments and an optional prefix to provide priorities between missions of different test phases. The first segment corresponds to the test phase numbers which are numbered chronologically from 1 to the total number of test phases. The second segment indicates the order in which activities will be performed within the test phase. All codes must have the same number of digits.

Mission Priorities. Normally it is anticipated that the priorities between missions of different test phases are implied priorities corresponding to the chronological order in which they are scheduled. In this situation no prefix is added to the test phase

number in the activity test phase code. To explicitly assign priorities between missions in different test phases, all activity test phase codes must be prefixed with a priority number. (Low number indicates the highest priority.)

For example, the first activity to be performed in each of the first four test phases would be coded 11, 21, 31, and 41. To assign a higher priority to test phase 3 missions, the ATPC's would be prefixed as follows: 211 221, 131, 241.

Empirical Distributions. Empirical distributions may be entered for activity times which cannot be accurately described by one of the ten provided distributions. It is recommended that the use of this feature be minimized. According to George S. Fishman and Philip J. Kiviat, "Sampling from a theoretical rather than empirical distribution is generally considered preferable, since it exposes a simulated system to the universe of possible stimuli rather than merely to those that have occurred in the past" (Ref 6:20).

Options. Numerous yes/no options are provided in the input data to tailor the model to the analyst's requirements. Depending on the total length of the test and the details required, weekly and monthly reports may be omitted. The features which reduce mission effectiveness and provide mission cancellations may also be deleted. The user can specify the number of workdays per week, and also may elect to work priority missions on non-duty days. On initial runs the same-skill-factor option, which assigns a value of 1 to every worker for each activity, may be used. External events may be used

to schedule a specific mission on a certain day or to terminate the simulation on a specific date. Finally, replications of the entire data package may be run either without resetting the random number generators or by entering new seed values for the random number generators.

Model Description

This section describes the function of each routine and event in the simulation model. The descriptions are presented in the same order as they are listed in the program in Appendix C. This order also corresponds to the general sequence in which the events and routines are used when conducting a mission. The reader should scan this section during the first reading, and then refer to a particular routine or event description to better understand some of the ideas presented later. If more detail is required to clarify a point, the reader is directed to the program listing in Appendix C.

PREAMBLE. The PREAMBLE declares the static system structure of the model, using an entity-attribute-set framework (Ref 12:193-207); declares the events that compose the dynamic structure; defines special properties of the two structures, such as the mode of attributes, the ranking of sets, and priority order of events; and specifies data-collection and analysis tasks. This set of global declarations describes to the compiler the system being simulated and the data-collection and analysis statements.

The static system structure includes the permanent entities, empirical functions, temporary entities, and the system-owned sets.

The permanent entities are TEST.PHASE, KIND.OF.ACTIVITY, LEVEL.OF.WORKER, and WORKER. A TEST.PHASE represents a requirement that a certain number of similar missions be completed. Its attributes record information regarding numbers of missions. A KIND.OF.ACTIVITY is created for every different type of activity, and has attributes which identify the parameters of each activity. A LEVEL.OF.WORKER is created for each category of workers and is used to gather utilization statistics. A WORKER is created to represent each person on the joint test force. Its attributes identify the person, his capabilities, and his current status. Compound permanent entities are TEST.PHASE/KIND.OF.ACTIVITY and WORKER/KIND.OF.ACTIVITY: activity duration times in each phase, and the worker skill factors for each activity are stored in the compound entities.

Three types of empirical functions can be entered for activity times which cannot be characterized by one of the ten provided standard functions. When using an empirical function, a sample value is selected by generating a random number, matching it against the probability entries, and thus selecting the activity time. INTERGER.STEP.FUNCTION and REEL.STEP.FUNCTION select in a step-like manner a real or integer value. CONTINUOUS.FUNCTION samples with linear interpolation between real sample values. Sampling is performed by successively comparing a random probability value to the cumulative probability value until a value is found that is greater than or equal to it. The associated value or the interpolated value between the associated value and the value preceding it is returned as the sample

value for step and continuous functions respectively.

The temporary entities are MISSION, ACTIVITY, AND XFACTOR. When a MISSION is created as a result of an internal event notice, the sequence of activities is copied from the set called STRUCTURE of the designated TEST.PHASE, STRUCTURE(TEST.PHASE), into the set called ROUTING, which is owned by the newly created MISSION, ROUTING(MISSION). If the MISSION is created as a result of an external event notice, the selected subset of activities is copied into ROUTING(MISSION). Additionally, activity codes which are not a part of the standard sequence may be entered with the required attributes. When such an entry is made, a warning, along with the necessary information to verify the input, is printed.

Each XFACTOR represents a reason for which a scheduled mission could be cancelled. These factors are used to represent items which could be expected to affect both standard and special missions. Examples are cancellation because of a higher priority mission and unsuitable weather.

The system owns two sets in which missions are filed until certain conditions exist. The members of the set called QUEUE are all the missions that are waiting for workers to perform the next activity. Missions are filed in the QUEUE on a first in, first out basis ordered by low priority code, then by low activity code, and then by low notification time of the impending activity. The set called WAIT.FOR.MISSION.TIME has as its members all missions for which all pre-mission activities have been completed prior to the scheduled active mission time. Missions are filed in WAIT.FOR.

MISSION.TIME on a first in, first out basis ordered by low notification time.

The dynamic system structure describes the time-dependent behavior of the system being simulated through the use of event notices. An event is an instant in time at which an activity starts or stops; it is executed in zero simulated time; and it changes the state of the system.

The event notices provide the proper sequencing of activities so that the order of performing activities in the model corresponds to the system which is being modeled. The periodic events are discussed in the sections entitled MAIN and REPORTS. Other events are dependent on the state of the simulation for scheduling and are discussed in the order that they are listed in the program. The capability of scheduling a special mission or of ending the simulation on a specific date from outside the simulation model is provided through the use of external event notices.

The Data Collection and Analysis declarations specify which statistics are to be gathered during the simulation. These statistical measurements are the principal output of the simulation experiments. Through the use of several ACCUMULATE and TALLY statements in the preamble, the operating program is free of data collection and data reduction statements. The ACCUMULATE statements gather time dependent statistics on mission waiting time and worker utilization. The TALLY statements compute various statistics associated with mission duration times.

The miscellaneous declarations define several abbreviations and global variables used in the program. Numerous functions are defined

in this section which are used for simulation control.

MAIN. The MAIN routine provides overall simulation experiment control. It calls two programmer-written routines to initialize the static system state, and it schedules events for the system-provided timing routine. The START.SIMULATION statement removes the first scheduled event from the system file of scheduled events and starts the simulation by transferring control to it. Tracing of event notices is provided during the tracing period by calling MYE.TRACE.

Eventually, either by running out of data or by programmer action, control passes from the timing routine to the statement following the START.SIMULATION statement. If there is no more data, the simulation is terminated. If there is more data, the system is initialized for another run. If more than one run of the entire data package is requested, routine REPLICATION is called to reset the simulation model.

In MAIN several SCHEDULE statements specify those events which will occur periodically throughout the simulation. When a SCHEDULE statement is executed, an event notice of the specified type is put in a file ordered by scheduled time; when the simulation clock advanced to this time, the event is executed. The events BEGIN.WORKING, TAKE.BREAK, and EO.DUTY.HOURS call routines to assign or remove workers from the activities of the current missions. Depending on the priority of the mission and current simulation time, worker status is changed to reflect the current conditions. The first CONFIRMED.MISSION in test phase 1 is scheduled for 0800 on the first day of the simulation. The weekly report and monthly reports are optional and will be discussed along with the final report in the reports section. The routines INITIALIZE,

REPLICATION, XWREPORT, and XMREPORT are released when the data indicates that they are not required, and thereby provide additional memory to run the simulation.

INITIALIZE. The INITIALIZE routine, which is called in MAIN, reads all data that will remain constant for the duration of the simulation run. These data include information relevant to the different test phases, missions, activities, time functions, and simulation control variables. The date reference is established by calling the system routine ORIGIN.R. Storage is allocated for the entities by using the CREATE statement, and then the various attributes are read. A "master copy" of a mission in each test phase is created by sequentially filing activities in the set owned by each test phase. The correct flow of input data is checked at several points by inserting the character string "OK". If the data input is incorrect, the run terminates; and an error message is printed which indicates where the error was detected. Data which are correctly entered is then printed for verification and reference when analyzing the simulation output. Upon return to MAIN for the final replication of the entire data package, the space occupied by the routine is released and may be used for dynamic storage such as creating and scheduling temporary entities and event notices.

RUN.INITIALIZATION. The RUN.INITIALIZATION routine is called in MAIN for each run and facilitates parametric studies. Input data for each current run are used with the background data read in INITIALIZE to provide the data package for the current simulation run. The data include information relevant to the workers, levels of workers, skill factors, and the number of days required between missions. The correct flow in input

data is again checked and printed as in INITIALIZE. the space occupied by this routine is not released and is used for each run until the data are exhausted.

BEGIN WORKING. The event BEGIN.WORKING is scheduled at the beginning of each workday and at the end of each break. It changes each worker's status to idle if he is on break or off-duty, and too busy if he is working overtime. It then calls the routine CHECK.QUEUE to see if there are any missions to be performed.

CHECK.QUEUE. The routine CHECK.QUEUE is called from the BEGIN.WORKING, CONFIRMED.MISSION, EO.ACTIVITY, ACCOMPLISH.PRI.MISSION, EO.DUTYHOURS, and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. Every mission which has completed its premission activities prior to mission time is removed from QUEUE and filed in WAIT.FOR.MISSION.TIME. If the current simulation time is during nonduty hours, either after duty hours or on the weekend, the routine OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS is called. If the check is made during duty hours, ATTEND.TO.MISSION is called for each mission for which there is a sufficient number of qualified workers. During normal duty hours, BUILD.PARTIAL.TEAMS is then called to try to assign the remaining workers.

CONFIRMED.MISSION. The event CONFIRMED.MISSION is notification of a scheduled mission. It is initially scheduled for 0800 of the first day of the simulation and is subsequently scheduled as directed in the input data. When notification is received, DISBAND.PARTIAL.TEAMS is called to free team members for new assignments. Notification of an internally scheduled mission is accepted during nonduty hours, but not on nonduty days (the number of duty days is a user input). Rejected missions are

rescheduled for the same time the following day. Externally scheduled missions are accepted at any time.

If the event is scheduled internally, a mission is created with the standard activities, effectiveness reductions, and cancellation factors for a mission in the indicated test phase. If the notification was received through external event notice, the indicated subset of activities and any additional activities are read as data input. A mission is then created with these required activities and the associated effectiveness reduction and cancellation factors. For each new mission VALIDATE.THE.NUMBER.OF.WORKERS.REQUIRED, DETERMINE.PREMISSION.CANCELLATION, and CALCULATE.MISSION.EFFECTIVENESS are called. If the mission is cancelled, BUILD.PARTIAL.TEAMS is called. If the priority of the first activity is 1, an ACCOMPLISH.PRI.MISSION is scheduled. Otherwise, depending on the simulation time, ATTEND.TO.MISSION or OVERTIME.ACTIVITIES.FOR.OVERTIME.MISSIONS is called. BUILD.PARTIAL.TEAMS is called to try to assign the remaining workers. LOOK.AHEAD.COMPLETE.MISSIONS is called to determine the test phase of the next mission, followed by MISSION.SCHEDULE.

MSN.SCHEDULE. The routine MSN.SCHEDULE is called by CONFIRMED.MISSION to schedule the arrival of the next mission. If the number of days between missions is less than 1 and the computed time is within normal duty hours, the mission is scheduled for that time. If the computed time is after duty hours, the mission is scheduled for the beginning of duty hours on the next duty day. If the number of days between missions is 1 or more, the mission is scheduled in that number of days at a time selected randomly from the user-provided function for the time

of day.

VALIDATE.NUMBER.OF.WORKERS.REQUESTED. The routine VALIDATE.NUMBER.OF.WORKERS.REQUESTED is called by CONFIRMED.MISSION. It checks each activity in a mission to insure that the number of workers requested in each level have been provided in the simulation. If the request is greater than the number of qualified workers in the simulation, an error message is printed with information describing the current state of the simulation. The simulation is then terminated. This check is necessary to preclude generating a mission which could never be completed because its requirements could never be met. This error in input data is normally encountered when conducting parametric studies.

DETERMINE.PREMISSION.CANCELLATION. DETERMINE.PREMISSION.CANCELLATION selects a random probability for each cancellation factor of a mission. If the selected random probability is less than the corresponding factor, the mission cancellation code indicator is set equal to 1.

CALCULATE.MISSION.EFFECTIVENESS. A similar function is performed by the routine CALCULATE.MISSION.EFFECTIVENESS. For each activity of a mission, a random probability is selected and compared to the percent of missions affected. If the selected value is less than this percentage, current mission effectiveness is then reduced using a uniform distribution between the upper and lower limits indicated for each activity. If the total reduction exceeds 100 percent, an informational message is printed and the cancellation code indicator is set equal to 1. This action was programmed to prevent a possible situation in which the workers would always be busy but never complete a productive mission. It is assumed that, if this observation were made at an installation,

corrective action would be taken. Upon return to CONFIRMED.MISSION, missions with cancel code equal to 1 are cancelled and a new mission is scheduled.

ATTEND.TO.MISSION. The routine ATTEND.TO.MISSION is called only during normal duty hours by the CONFIRMED.MISSION, ACCOMPLISH.PRI.MISSION, or CHECK.QUEUE. Every priority mission which has completed its premission activities prior to mission time is removed from the QUEUE and is filed in WAIT.FOR.MISSION.TIME until its mission time (time of notification plus a user provided mission preparation time). For all other missions in which the number of available workers qualified to perform its next activity exceeds the number of workers required for that activity, the routine ALLOCATION is called.

SUM.WORKERS. In determining the number of workers currently available to perform an activity, the routine SUM.WORKERS considers all combinations of worker level preferences. Depending on the input arguments, a combination of the four sections is used.

The first section determines the number of idle workers. If the activity has a priority 1, the fourth section adds the additional number of qualified workers who are not working on another priority 1 activity.

The second section determines the number of qualified workers who are not working. This section is primarily used for priority 1 missions which arrive during break times. Since workers who are busy during a break are by definition working on a priority 1 mission, the qualifications of the remaining workers are tested.

The third section also determines the number of qualified workers who are not working. This section is used for overtime missions which

may be either priority 1 or priority 2 missions. By using the second section and a local variable indicating overtime, the number of qualified workers not working overtime is determined. If the activity has a priority of 1, the fourth section adds the number of qualified workers which are not working on another priority 1 activity. Additions made in the fourth section will be made available as the priority 1 activity preempts lower priority activities to meet its requirements.

In all cases, the routine returns the number of qualified workers which is then compared to the number of workers required to perform an activity. If the required number of workers is available, the routine ALLOCATION is called.

ALLOCATION. The routine ALLOCATION is called by ATTEND.TO.MISSION and assigns qualified workers to a team for activities which are performed during normal duty hours. If the number of required workers is zero, indicating that the manpower requirements are external to the test force, an activity time is selected. Otherwise, team members are assigned in the order of the activity's worker level preference. Selection within a level is based on the least busy time for each worker. If the activity has a priority 1 and there are not enough idle workers to fill the team, lower priority missions are preempted until the required workers are found. After the team assignments are made, an activity time is selected and an EO.ACTIVITY is scheduled. Upon return to the original calling routine, CHECK.QUEUE is called to reassign workers who may have been preempted.

ACCOMPLISH.PR1.MISSION. The event ACCOMPLISH.PR1.MISSION is scheduled for priority 1 activities at a user-supplied length of time after

notification of a scheduled mission. This time represents the required preparation time before conducting the first priority 1 activity in a mission. DISBAND.PARTIAL.TEAMS is called to free team members for new assignments. Dependent upon the current simulation time, ATTEND.TO.MISSION or OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS is called. CHECK.QUEUE is then called to reassign workers if a preempt occurred.

PREEMPT. The routine PREEMPT is called by ALLOCATION and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. PREEMPT selects one non-priority 1 mission to preempt each time it is called and then calls INTERRUPT.AND.FILE.

INTERRUPT.AND.FILE. The routine INTERRUPT.AND.FILE is called by PREEMPT, TAKE.BREAK, and EO.DUTY.HOURS. It removes the workers from the team, calculates the remaining activity time and files the interrupted mission in the QUEUE.

VALUE. Routine VALUE is called by ALLOCATION and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. If the activity has previously been interrupted, it multiplies the continuation time by the average skill factor of the new team members and the partial team factor. It then returns to the calling routine. Otherwise, it selects an activity time using one of the ten provided distributions or a user provided empirical distribution. The arguments for the function are indicated for the activity and test phase. A sample value is selected from the function and this value is multiplied by the average skill factor of the workers assigned to perform the activity and the partial team factor. The weighted time value is returned as the activity time.

LOOK.AHEAD.COMPLETE.MISSIONS. The routine LOOK.AHEAD.COMPLETE.MIS-

SIONS is called by CONFIRMED.MISSION. It sums the final mission effectiveness for all test phase missions which are currently in the simulation. This value is added to the number of completed missions in a test phase and the total is compared to the number of required missions. If the total exceeds the number of required missions, a mission in the next test phase will be scheduled. This routine was developed on the assumption that prior to completing the final activity in a mission, an approximate mission effectiveness can be determined. In a simulation run in which the number of missions in the QUEUE builds or the mission duration is lengthy, an unrealistically excessive number of missions would be scheduled in a test phase. The look ahead routine is provided to allow the simulation model to perform like the observed system.

TAKE.BREAK. The event TAKE.BREAK is scheduled periodically at the user-provided times for coffee and lunch breaks. It puts all idle workers on break status, identifies every non-priority 1 mission in progress, and calls INTERRUPT.AND.FILE for each of these missions.

EO.ACTIVITY. The event EO.ACTIVITY is scheduled by ALLOCATION and OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS for each activity at the simulation time determined in routine VALUE. It calls DISBAND.PARTIAL.TEAMS to free team members for new assignments. It then removes the completed activity from the set of activities required for the mission, ROUTING (MISSION); changes the mission priority and mission code to correspond to the next activity (determines ranking order in QUEUE); and removes the workers from the team (activity). Worker status is then determined based on current simulation time. The mission is then filed in the

QUEUE insuring that low coded activities of all missions are performed first. If this event is called during a weekday or if the user has allowed priority missions to be worked on the weekends, CHECK.QUEUE is called.

EO.DUTY.HOURS. The event DO.DUTY.HOURS is scheduled daily at the user-provided time of day. It calls DISBAND.PARTIAL.TEAMS to stop the associated activities. The status of every idle worker is changed to off-duty, and the status of team members of priority 1 or priority 2 missions is changed to overtime. Regardless of priority, missions with less than 30 minutes remaining on the current activity will be completed on overtime. This action is based on the assumption that a worker will elect to complete a task that is nearly finished so that the following day's efforts can be devoted to new activities. OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS is then called to work any priority missions which are waiting in the QUEUE.

OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS. The routine OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS is called by CONFIRMED.MISSION, CHECK.QUEUE, ACCOMPLISH.PRI.MISSION, and EO.DUTY.HOURS. It is used for missions performed during non-duty hours throughout the work week, and on weekends if the user has allowed priority missions to be worked on the weekends. Every mission which has completed its premission activities prior to mission time is removed from the QUEUE and filed in WAIT.FOR.MISSION.TIME. If the number of required workers is zero, indicating that the manpower requirements are external to the test force, an activity time is selected. Otherwise, team members are assigned in the order of the activity's worker level preference. Selection within a level is based

on the least overtime for each worker. If the activity has a priority 1 and there are not enough off duty workers to fill the team, lower priority missions are preempted until the required workers are found. After the team assignments are made, an activity time is selected and an EO.ACTIVITY is scheduled. If the activity is initiated on a weekend, an informational message is printed regarding that activity. Upon return to the calling routine, CHECK.QUEUE is called to reassign the workers if a preempt occurred.

BUILD.PARTIAL.TEAMS. The routine BUILD.PARTIAL.TEAMS is called by CHECK.QUEUE and CONFIRMED.MISSION after workers have been allocated to the activities. This routine selects those missions whose impending activity can be started with a partial-team selected from the remaining workers. ALLOCATION is called for each of these missions.

DISBAND.PARTIAL.TEAMS. The routine DISBAND.PARTIAL.TEAMS is called by CONFIRMED.MISSION, ACCOMPLISH.PRI.MISSION, TAKE.BREAK, EO.ACTIVITY, AND EO.DUTY.HOURS. These teams are disbanded during breaks and any time a new activity comes available for worker allocation. Thus, activities for which the required number of workers are available will receive worker assignments prior to making any partial allocations. Partial-teams are not permitted to work on activities during breaks or nonduty hours.

COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION. The routine COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION is called by the PREAMBLE prior to destroying the record of a mission. If the cancel code indicator is equal to 1, that value is added to the number of cancelled missions in the test phase. Otherwise the mission duration is computed and the cur-

rent effectiveness level is added to the number of completed missions in the test phase. If the number of completed missions in each test phase is greater than or equal to the number required, an EO.SIMULATION is scheduled.

REPORTS. The events WREPORT and MREPORT are scheduled weekly and monthly, respectively, if requested by the user. Otherwise, these events are not called and the releasable routines XWREPORT and XMREPORT which contain the actual reporting program, are released (events are not releasable). The FINAL.REPORT is always printed at the end of each simulation run.

The same format is followed in all three reports with only minor variations in the information presented. Since there is a variable number of days in a month, certain values are presented only in time units in the monthly report rather than as percentages. The weekly and monthly reports print the cumulative number of completed missions in each test phase. The final report prints the date that each test phase was completed and the ending date of the simulation.

There are three major sections in each report: mission statistics for each test phase, waiting time information for all missions, and worker utilization information. The mission statistics presented for each test phase include the number scheduled, number cancelled, number required, effective productivity, cumulative number completed, average mission duration, standard deviation of duration times, total of duration times, sum of the squares of duration times, and the date the test phase was completed.

Waiting time information for all missions is presented in the queue

report. The information presented includes the maximum, minimum, average, standard deviation, and histogram of the number of missions waiting for workers. Also, the total waiting time for all missions is reported. This information is not printed by test phase because there is not necessarily a correlation between the waiting time for missions in a test phase and the requirements of that test phase. Since all missions fill worker requirements from the same groups of workers, overlapping of test phases, differing priorities and numerous other factors may cause missions in a test phase to wait for an untractable variety of reasons. The significant information to be derived from this section is the time frame in which large backlogs of work can be expected and the relative size of the backlog. The cure for these anticipated problem areas is indicated in the final section of the report.

Worker utilization information is presented in the third section of the report. Information includes a histogram of worker's idle time by worker level and the report of each worker's status divided into the categories of idle, busy, break, off duty and overtime. Types of information which can be extracted from this data might include: which worker levels can tolerate or require personnel changes, base figures for estimating the value or cost of these changes, and an indication of the amount of overtime that can be expected with various configurations of the simulation. Thus, this section indicates areas in which changes could be made to alleviate those problem areas highlighted in the first two sections.

LARGEST. The routine LARGEST is called by each of the reports to determine the length of the worker idle time section. It checks all

levels of workers for the maximum number of idle workers in the reporting period and returns the largest value to the calling routine.

MY.TRACE. The routine MY.TRACE is called by every routine and prints information only during the tracing period designated in the data input. It prints the name of the called routine, current simulation time, and the associated mission number.

MYE.TRACE. The routine MYE.TRACE is called by every event and prints information only during the tracing period designated in the data input. It prints the name of the called event, current simulation time, and selected information.

MY.LIST. The routine MY.LIST is called as required throughout the simulation to print the members of sets and the attributes of permanent or temporary entities.

QENTER.INFO, QEXIT.INFO. Each time a mission enters or exits the QUEUE, the routine QENTER.INFO or QEXIT.INFO is called by the PREAMBLE. During the tracing period, these routines then call INFO.PACKAGE to print relevant information.

WENTER.INFO, WEXIT.INFO. Each time a mission enters or exits WAIT.FOR.MISSION.TIME, the routine WENTER.INFO or WEXIT.INFO is called by the PREAMBLE. During the tracing period, these routines then call INFO.PACKAGE to print relevant information.

ERROR.PACKAGE. The routine ERROR.PACKAGE is called whenever a programmer detected error in the simulation is found. These messages are generated independently of the SIMSCRIPT error messages and indicate conditions which may not be detected as errors by the compiler. Four available error messages are printed as required with the appropriate

information to analyze the detected error.

INFO.PACKAGE. The routine INFO.PACKAGE is a collection of statements to list information describing the current state of the simulation. It is called from numerous locations throughout the program.

EO.SIMULATION. The event EO.SIMULATION may be scheduled internally or externally; and subsequently, it schedules the FINAL.REPORT. Internally, it is scheduled by COMPUTE.STAY.TIME.AND.RECORD.CANCELLATION whenever the required number of missions in each test phase has been completed. If the event is scheduled externally, information describing the current state of the simulation is printed prior to scheduling the FINAL.REPORT.

CLEAR. The routine to CLEAR is called by the FINAL.REPORT to reset the state of the simulation for successive experiments. The starting conditions of the model are identical to the previous run to include restoring the seed values of the random number generators. Using this routine, a series of sequential runs with slightly modified versions of selected input can be examined. The principal use is to measure the behavior of alternative inputs in the form of number of workers, skill factors and days between missions. By using this routine, parametric studies can be conducted without repeating an entire data package.

REPLICATION. The routine REPLICATION is called if the user requests more than one run of the entire data package. This routine clears the model structure which was created by the previous set of data. However, seed values of the random number generators are not reset to the starting conditions. Optionally, the user may enter new seed values. This simulation model is designed to study the transient

condition, represented by starting the simulation with no missions in the system and continuing until the required number of missions have been completed. In order to determine a confidence interval, the run must be replicated to obtain a distribution of results. This routine allows multiple replications with different seed values for the random number generators.

Validation

The problem of verifying computer models remains the most elusive methodological problem associated with computer simulations (Ref 13:892). A simulation is only an abstraction of the process under study and because simulations tend to become so complex that they conceal their assumptions and processes (Ref 18:247). Thus, validation is necessary to increase to an acceptable level the confidence that an inference about a simulation process is correct for the actual process. The following steps are necessary in attempting to validate a simulation model (Ref 7:1004):

1. Individually test the subroutines.
2. Review in detail the program logic.
3. Review all system state changes during the test runs under various operating policies.
4. Compare the results of the model to historical data.

Initial checkout runs started with very simple data in the form of 2 activities, 1 mission required, 1 test phase, 1 worker, and constant activity times. The complexity of the data was then increased to verify the different features provided in the program. After validating the model in a strictly deterministic mode, uniform distri-

butions with very small ranges for activity times were added. The results were nearly the same as for the deterministic runs.

The programmer written trace routines were designed specifically to present the information necessary to test the routines and to observe system state changes. Each time an event or routine is called during the tracing period, the appropriate information is printed to show changes in worker status, mission status, or system statistics. If necessary, every mission can be traced through the entire simulation to verify that every routine operates properly under all operating conditions.

After numerous runs with test data constructed to detect any possible errors in logic, a set of data was developed for the Test which was studied to provide guidance in developing the simulation model. Although all phases of the Test have yet to be completed, computer results compare favorably with test phase completion dates, worker utilization, and the time frame of backlog buildups. Thus the model structure was further verified by introducing complex data of an actual Test and by comparison with current historical data.

III. Illustrative Example

Description of Test

A current test, Pave Tack Night Attack Systems/F-4E, is used to illustrate some of the significant features of the simulation model and to show how a joint test force director can adapt this model to a particular test. This test is an integrated Developmental Test and Evaluation/Initial Operational Test and Evaluation (DT&E/IOT&E) effort under the direction of a Joint Test Force (JTF). The major objective of this test is to provide higher headquarters with sufficient information to support a production decision on the Pave Tack Class V Modification to the F-4E aircraft. Sufficient data are required on competing systems to support a source selection decision. (Ref 2:4)

For this example, the DT&E portion was simulated since a significant portion of that test has been completed and was referenced for validation of the simulation results. The estimated active phase of the test is between 25 May 1976 and 28 February 1977. The objectives and required resources were extracted from the work directive with the help of the ADTC point of contact. Pave Tack personnel were consulted to verify and refine the time estimates. A total of 66 missions were simulated.

Data Input

Activities. Fourteen activities that were based on the required resources and the test objectives were identified after analyzing the work directive. These activities are as follows:

1. PMSNPG - Permission preparation and briefings for a ground mission

2. PMSMPA - Pre-mission preparation and briefing for an air mission
3. CGDMSN - Conduct a ground mission
4. CAMSN - Conduct an air mission
5. PSTMNG - Post-mission evaluation for a ground mission
6. PSTMNA - Post-mission debriefing for an air mission
7. DATARD - Data reduction associated with an air mission (No JTF personnel required)
8. DATAAN - Data analysis for an air mission
9. PMSNI - Pre-mission preparation and briefing for an integration test
10. CMSNI - Conduct an air mission for an integration test
11. PSTMNI - Post-mission evaluation for an integration test
12. MSNNEL - Mission performed at Nellis (TDY for JTF personnel)
13. ANTAPE - Analyze tapes
14. ANLOGS - Analyze logs

The definition of these activities can be kept general since activity times and number of required workers are entered for each test phase in which the activity is needed. A defined activity may be used more than once in a single mission if the activity time and required workers remain the same. Effectiveness reduction probabilities and percentages were provided for each activity as subjective inputs by the ADTC point of contact. (See Appendix F Section 1)

Missions/Test Phase. Based on similarities in activity requirements and timing requirements, similar missions were then constructed and grouped into simulation test phases. Test phases are generally based on resource requirements and scheduling difficulties in order to collect statistics on these different areas of interest. Other criteria for

identifying test phases might include suspense dates for a particular group of missions, off-base testing, or different funding sources. (See Appendix F Section 2)

Specifically, the Pave Tack Night Attack Systems/F-4E test was divided into eight phases. The first two phases represent the integration tests which precede the DT&E testing. The last six phases represent the DT&E testing.

Phase 1 -- Integration Ground Missions. The first integration test phase represents the six ground missions. Two activities were used for each of these missions. The first activity (PMSNPG) included work associated with mission preparation. This activity is expected to require a substantial amount of preparation since the test engineers (TE) are just becoming familiar with the system. From 360 to 480 minutes were allowed for mission preparation. Three TE's are requested; however, test technicians (TT) or test analysts (TA) may be substituted if the TE's are not available. Also, the activity may be conducted with a partial team if three workers are not available. This activity was assigned a priority 2 since it must be completed the same day it is scheduled. The ground mission can then be conducted as scheduled on the following day.

The second activity (CGDMSN) represents the ground checkout procedures such as sensor alignment and power-on checks of all other equipment. The same number and order of personnel preference is required for both activities of a mission in this phase. An activity time between 120 and 180 minutes was selected for these checkout procedures.

Phase 2 -- Integration Air Missions. The second integration test phase represents the 12 flight test missions. Three activities were used

for each of these missions. The first activity (PMSNI) included the work associated with mission preparation. Since the crews are still in the learning process, an activity time between 480 and 600 minutes is required. Two TE's are involved in this familiarization process, but the activity may be conducted by only one worker.

The second activity (CMSNI) represents flight maneuvers and actions to be performed in checking out the system. An activity time between 120 and 180 minutes is required for two TE's. This activity has the potential of reducing the mission effectiveness; therefore, mission reduction probabilities and factors were entered.

The third activity (PSTMNI) represents a debriefing which follows each flight. A TT joins the two TE's to increase his knowledge about the system. A moderate amount of time was allowed (120 to 180 minutes) because of the lengthy discussions required during this learning period. The three cancellation probabilities entered for this test phase represent aircraft maintenance, test item maintenance, and weather.

Phase 3 -- DT&E Task A Missions. Following the integration phases and prior to the formal start of DT&E, test phase 3 represents the missions required for aircrew checkout and system optimization. The activities required for a mission in this test phase are similar to test phase 2 since there is no data reduction or analysis involved. The first activity (PMSNPA) represents mission preparation. The required activity time (90 to 120 minutes) is less than the previous test phases because crews are gaining proficiency with the equipment. This activity is primarily performed by TE's; however, it may also be performed by either TT's or TA's and with partial teams.

The second activity, conducting an air mission (CAMSN), also requires less time (105 to 135 minutes) than in previous phases because of the learning curve. This activity may only be accomplished by TE's because of the flying requirement. Mission effectiveness reduction probabilities and factors are high because of the increased emphasis on the system and the time frame of the test phase.

The third activity (PSTMNA) requires the same amount of time for debriefing (120 - 180 minutes); however, a TT or a TA is required, in addition to the TE's to assist in taking notes and viewing parts of the video tapes. An analysis activity is not required since the spot checks are only used to verify that the systems are functioning properly. The same mission cancellation factors as were used in test phase 2 are appropriate for missions in this test phase.

Phase 4 -- DT&E Objectives A1 through B2. Test phase 4 represents the missions required to determine system performance and operator capability with various target temperature differentials, weather, illumination, and background conditions. This phase is the largest and most time consuming phase of the test. It requires 22 missions consisting of seven activities each. The first three activities (PMSNPA, CAMSN, PSTMNA) are similar to the activities in test phase 3 except that the time and personnel requirements are increased because all missions consist of two aircraft sorties. The two sorties in each mission are flown during the same day.

The next two activities (ANTAPE, ANLOGS) represent the work necessary to analyze the video tapes from the aircraft and to review the mission logs. This analysis is normally performed by TA's after sortie

completion, but overtime is not required. The analysis may also be performed by TE's, TT's, and partial teams. A small reduction of mission effectiveness probability and factors was entered to represent problems discovered on the tapes.

The sixth activity (DATARD) represents the data reduction performed for each mission by the Computer Sciences Laboratory. The manpower requirements are external to the test force and thus zero was entered for the number of required workers. The activity priority is 1; and the activity time, consequently, represents the length of time before the data will be returned to the JTF. A small reduction of mission effectiveness probability and factors was entered to represent problems discovered during data reduction.

The last activity (DATAAN) can be accomplished after the reduced data is received from the Computer Sciences Laboratory. The activity time is extensive (1440-2160 minutes) since the work involves many hand calculations and graph plottings. The activity is normally performed by TA's but may be performed by TT's, TE's and partial teams. Mission cancellation probabilities were doubled based on the number of sorties.

An external mission was used in this test phase to represent a mission which had slightly different requirements. An additional analysis of tape activity (ANTAPE) was added which required more personnel than the initial tape analysis for the mission. All other activities in the mission remained the same.

Phase 5 -- DT&E Objective B3 through B7. The missions in test phase 5 are similar to test phase 4 missions except fewer test resources are required. Seven missions, each consisting of seven activities, are

required. The first five activities (PMSNPA, CAMSN, PSTMNA, ANTAPE, ANLOGS) are the same as the first five activities of test phase 4 missions except fewer workers are required. This requirement is a result of a reduction in the number of data collection sights.

The last two activities (DATARD, DATAAN) have a lower activity time because there is less data to reduce and analyze. The cancellation probability due to test item malfunction was reduced in this test phase to reflect a stabilizing trend in equipment performance.

Phase 6 -- DT&E Objective B8. Test phase 6 represents the missions conducted by JTF personnel while TDY to Nellis AFB. Two TE's were TDY for a period of 6 days. A priority 1 activity (MSNREL) is used to represent these missions so that these two workers will not be released until the mission has been completed. This action insures that the simulation model does not permit these workers to be assigned to teams at Eglin AFB. The overtime statistic for these workers must be mentally adjusted to compensate for their off-duty hours.

An external event was used in this test phase to represent data analysis at Eglin AFB on the missions conducted at Nellis AFB. The Nellis AFB activity (MSNREL) was deleted and an analysis activity (DATAAN) was added with its required entries. Any defined activity may be added to or deleted from a mission in a test phase. It is illustrative to note that this activity was performed in zero time because the arguments for the activity in the data input were not updated after adding the external event.

Phase 7 -- DT&E Objectives G through T and X and AA. The missions in this test phase represent several individual test objectives which have

common resource requirements. This subjective grouping was based on information presented in the required resource table and the accompanying text. Four missions were required consisting of 5 activities each. The first three activities (PMSNPA, CAMSN, PSTMNA) are similar to the first three activities in test phases 4 and 5, except only one aircraft sortie is required for each mission. The activity times and personnel requirements were reduced accordingly.

The last two activities (DATARD, DATAAN) were judged more complicated than the corresponding activity in test phase 5 because the data had to be cross-correlated between several sources. The number of required workers and the activity times were increased to accommodate this additional workload.

Phase 8 -- DT&E Objective W. Test phase 8 represents the missions required to demonstrate the weapon delivery accuracy of the Pave Tack weapon system. Ten missions, consisting of five activities each, were required. The same basic activities (PMSNPA, CAMSN, PSTMNA, DATARD, DATAAN) are required for a mission in test phase 8 as were used in test phase 7. The required work for four of these activities was judged to be less complicated than the corresponding activities in the test phase 7 missions; so the worker requirements and activity times were reduced accordingly. The data reduction activity (DATARD) required a longer activity time because the Computer Sciences Laboratory must hand process the aircraft position information.

(Ref 2)

Workers. After the activities are grouped into missions, the worker information must be determined. Nine workers were divided into three

levels for this sample run. Different skill factors were entered for the workers. This action reflects that workers were more proficient in performing activities which they normally performed by that worker level than for those activities which they only occasionally perform. For example, it would take a TT twice as long to perform a data analysis activity (ANLOGS) as it would take a TA to perform that same activity. (See Appendix F Section 3)

Program Variables. The program variables were entered as directed in the Data Input Instructions listed in Appendix D.

Activity Test Phase Codes. The two segments of the activity test phase codes were entered to correspond to the missions in each test phase. The first segment corresponds to the test phase number (1 through 8) based on the chronological order in which the missions were scheduled. The second segment corresponds to the order in which activities are to be performed and is the same order in which they were discussed in the respective test phase. All activity codes meet the requirement of having the same number of digits.

Mission Priorities. Priorities between missions of different test phases were explicitly assigned by prefixing the activity test phase codes. In the Pave Tack simulation test phase 5, missions were given a priority over all other missions by prefixing test phase 5 activity codes with a 1 and all other activity codes with a 2.

Options. Most of the options were illustrated in this example. Since only selected output from the final report could be reasonably presented in this thesis, weekly and monthly reports were not requested. Both reduction of mission effectiveness and mission cancellation were

entered. A workweek of 5 days was entered, and priority activities were not permitted to be worked on the weekends. Individual skill factors, as discussed in Workers, were entered for each worker. Special missions were scheduled using external event notices, and one run of the data package was requested.

Discussion of Output

Section 4. Selected outputs are presented in Appendix F Sections 4, 5, and 6 to show some of the results obtained by entering the sample data. The source of this data was discussed in the previous section. Section 4 shows the first and last test phase mission statistics. Since no cancellation factors were entered and no missions were less than 100% effective in test phase 1, 6 missions were scheduled and the effective productivity was also 6. Cancellation factors for test phase 8 were entered as reflected by the cancellation of 2 missions. Of the remaining 11 missions, the effective productivity was slightly more than 10 missions. Similar results were observed in the other test phases.

Section 5. Section 5 presents the statistics collected on the missions which were waiting for worker assignments. The state percentages show that most of the time the missions were being worked as quickly as they arrived. (Almost 60% of the time 2 or less missions were in the queue.) The percentages then peak at 7 - 8 missions indicating a backlog for a substantial period of time. Monthly reports could be consulted to show the phase and time frame of the backlog.

Section 6. Section 6 shows the information collected on the workers by levels and as individuals. The maximum number of idle workers in each level is 3, which corresponds to the total number of workers in the input

data. The total idle time is lengthy for all levels, and level 2 workers were used the least. The individual worker statistics indicate that TE2 and PO were sent TDY to Nellis AFB since their overtime hours are considerably larger than the other level 1 worker statistics. If cost data on overtime hours were to be determined, these figures would be adjusted to reflect the actual number of work hours per day.

Parametric Studies. Multiple runs were made with reduced personnel. Based on the results shown in the tables, the number of workers in levels 2 and 3 were reduced until a total of six workers were in the simulation. Fewer than six workers would have resulted in a programming fatal error since an activity in test phase 4 requires six workers. The completion date remained the same for all runs; and, as expected, the idle time in all worker levels progressively decreased as the number of workers decreased. For this model configuration a substantial savings can be realized without a change in completion date by reducing the personnel requirement by one-third.

IV. Conclusions and Extension of Research

The main thrust of this research project was the development of the event-oriented simulation model which is useful in estimating work completion dates and personnel requirements. The results obtained by entering the data for Pave Tack compared favorably with the ongoing test; however, as with all stochastic simulations, the results produced from a single run are of little significance. The illustrative example merely shows how to apply the simulation to a JTF and exercises many of the model's significant features. The real value of the simulation model will be realized through extensive parametric studies with replications in order to develop confidence intervals for the results.

The statistics accumulated in the simulation model readily lend themselves to cost-benefit analysis for various work rules versus completion dates. The flexibility of data input allows the model to accurately simulate situations which have yet to be considered. A continuation of this study should concentrate on the application of the simulation model to extensively test its intended flexibility. Techniques for establishing confidence intervals for the output data should be developed. In conjunction with the confidence intervals, different seed values for the random number generators should be tested for significance.

Further research should determine which distributions can be used to better describe the activity times. Uniform distributions were used with parameter estimates verified by JTF members in order to insure structural validity of the simulation model. Since the simulation results compare favorably with the actual test, more descriptive distribu-

tions can now be selected to refine the estimates. Guidance on which distributions should be used with various classes of activities would be of value to the Test Wing.

Additional research could be conducted in the areas of mission cancellations and reductions in mission effectiveness. Extensive data is available in this area, but time permitted only a superficial analysis of this data to provide inputs for the sample run. This information, which is accumulated by the Test Wing, should be further studied to provide better estimates under a variety of conditions. Additional analysis should compare this past data with the final statistics of Pave Tack. Some differences can be expected; because, under the JTF concept, some resources are dedicated to the test force. If there is an appreciable difference, future data inputs should favor the results obtained under the JTF concept.

Bibliography

1. Air Force Systems Command. RF-4C/Flir Reconnaissance System. Eglin AFB, Florida, 1976.
2. Air Force Systems Command. F-4E/Night Attack System. Eglin AFB, Florida, 1976.
3. Armament Development and Test Center. Test Engineers' Handbook. Eglin AFB, Florida, 1976.
4. Armament Development and Test Center. Data Reduction and Computational Capabilities. Eglin AFB, Florida, 1975.
5. Emshoff, James R. and Roger L. Sission. Design and Use of Computer Simulation Models. New York: Macmillan Publishing Co., Inc., 1970.
6. Fishman, George S. and Phillip J. Kiviat, Digital Computer Simulation: Statistical Considerations, (AD66 0953; California: The Rand Corporation, November 1967).
7. Fryer, J. S. "Operating Policies in Multiechelon Dual-constraint Job Shops." Management Science, 19:1001-1012 (May 1973).
8. Hillier, F. S. and G. J. Lieberman. Introduction to Operations Research (Second Edition). San Francisco, Calif.: Holden-Day Inc., 1974.
9. Johnson, Glen and Edward C. Russell. SIMSCRIPT II.5 Users Manual, Honeywell 600.6000 Version. Los Angeles, Calif.: CACI, Inc. 1973.
10. Kiviat, Phillip J., Computer Simulation Programming Languages: Perspective and Prognosis, (AD659358; California: The Rand Corporation, September 1967).
11. Kiviat, Phillip J., Digital Computer Simulation: Computer Programming Languages (AD84124; California: The Rand Corporation, January 1969).
12. Kiviat, P. J., R. Villanueva, and H. M. Markowitz, SIMSCRIPT II.5 Programming Language. Los Angeles, Calif.: CACI, Inc. 1973
13. Naylor, T. H. and J. M. Finger. "Verification of Computer Simulation Models." Management Science, 14:B92-B106 (October 1967).
14. Russel, Edward C. Simulation and SIMSCRIPT II.5 (Preliminary Printing). Los Angeles, Calif: CACI, Inc., 1975.
15. Samuelson, Paul A. Economics (Eighth Edition). New York: McGraw-Hill Book Co., 1970.

16. Schribner, Thomas J. Simulation Using GPSS. New York: John Wiley and Sons, 1974.
17. Tocher, K. D. "Review of Simulation Languages." Operations Research, 16:189-217 (June 1965).
18. Van Horn, R. L. "Validation of Simulation Results." Management Science, 17:247-258 (January 1971).

ATTRIBUTES OF TEMPORARY ENTITY "ACTIVITY"
 - CURRENT EFFECTIVENESS LEVEL
 EACH ACTIVITY HAS THE POTENTIAL TO REDUCE THE
 EFFECTIVENESS OF THE MISSION. THE ADEL SHOWS
 THE MISSION EFFECTIVENESS AT THE END OF THE
 ACTIVITY.
 - FIRST NUMBER LEVEL
 FIRST OR THREE NUMBER LEVEL REFERENCES WHICH
 CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
 - DISTRIBUTION COORDINATION TIME
 TIME REQUIRED TO COMPLETE AN ACTIVITY WHICH HAS
 BEEN EITHER PLANNED OR INTERRUPTED
 - LOWER LEVEL FOR A DURING DISTRIBUTION WHICH IS
 USED TO SET THE AMOUNT THAT THE ADEL IS TO BE
 REDUCED.
 - NUMBER OF ACTIVITY
 SAME NUMBER AS THE NUMBER OF KINDS OF ACTIVITY
 NUMBER OF WORKERS DESIGNATED
 EACH ACTIVITY MAY REQUIRE FROM ZERO TO THE TOTAL
 NUMBER OF WORKERS IN THE MISSION TO PERFORM
 THE ACTIVITY.
 - PARTIAL TEAM OF ACTIVITY
 YES/NO ATTRIBUTE WHICH INDICATES WHETHER A
 PARTIAL TEAM CAN BE USED TO PERFORM THE ACTIVITY
 FROM THE NUMBER IN THE NUMBER OF WORKERS DESIGNATED
 - PERCENT MISTAKE APPETITE
 IS A RANDOM NUMBER IS LESS THAN OR EQUAL TO THIS
 VALUE, THE ACTIVITY IS REPEATED BY AN AMOUNT
 BETWEEN THE VALUE OF THE PERCENT MISTAKE APPETITE
 AND THE VALUE OF THE PERCENT MISTAKE APPETITE
 - PRIORITY OF ACTIVITY
 ONE OF THREE PRIORITY LEVELS WHICH INDICATES THE TIME
 OF DAY THAT AN ACTIVITY CAN BE PERFORMED AND
 ITS ABILITY TO TAKE WORKERS AWAY FROM OTHER
 WORKING ACTIVITIES.
 - SECOND NUMBER LEVEL
 SECOND OR THREE NUMBER LEVEL REFERENCES WHICH
 CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
 - TIME FUNCTION CODE
 INDICATES THE TIME OF DAY THAT AN ACTIVITY CAN
 BE PERFORMED.
 - TEST POINT CODE
 SEVERAL CODE WORDS WHICH ARE ASSIGNED TO
 INDICATE THE TYPE OF ACTIVITY.
 - THIRD NUMBER LEVEL
 THIRD OR FOUR NUMBER LEVEL REFERENCES WHICH CAN
 BE DESIGNATED FOR EACH KIND OF ACTIVITY.

APPENDIX A

DEFINITION OF TERMS

ACTIVITY - A TASK WHICH CAN BE PERFORMED BY
 ONE OR MORE WORKERS.
 ACTIVITY LEVEL - THE NUMBER OF WORKERS
 DESIGNATED TO PERFORM AN ACTIVITY.
 ACTIVITY PRIORITY - A VALUE WHICH INDICATES
 THE TIME OF DAY THAT AN ACTIVITY CAN BE
 PERFORMED.
 ACTIVITY TEST POINT CODE - A VALUE WHICH
 INDICATES THE TYPE OF ACTIVITY.
 ACTIVITY TIME FUNCTION CODE - A VALUE WHICH
 INDICATES THE TIME OF DAY THAT AN ACTIVITY
 CAN BE PERFORMED.
 ACTIVITY THIRD NUMBER LEVEL - A VALUE WHICH
 CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
 ACTIVITY SECOND NUMBER LEVEL - A VALUE WHICH
 CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
 ACTIVITY FIRST NUMBER LEVEL - A VALUE WHICH
 CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
 ACTIVITY PERCENT MISTAKE APPETITE - A VALUE
 WHICH INDICATES THE PROBABILITY THAT AN
 ACTIVITY WILL BE REPEATED.
 ACTIVITY PARTIAL TEAM - A VALUE WHICH
 INDICATES WHETHER A PARTIAL TEAM CAN BE
 USED TO PERFORM AN ACTIVITY.
 ACTIVITY DISTRIBUTION COORDINATION TIME -
 THE TIME REQUIRED TO COMPLETE AN ACTIVITY
 WHICH HAS BEEN EITHER PLANNED OR INTERRUPTED.
 ACTIVITY LOWER LEVEL - A VALUE WHICH IS
 USED TO SET THE AMOUNT THAT THE ADEL IS
 TO BE REDUCED.
 ACTIVITY NUMBER OF ACTIVITY - THE NUMBER
 OF KINDS OF ACTIVITY.
 ACTIVITY NUMBER OF WORKERS DESIGNATED -
 THE NUMBER OF WORKERS DESIGNATED TO
 PERFORM AN ACTIVITY.
 ACTIVITY EFFECTIVENESS LEVEL - A VALUE
 WHICH INDICATES THE CURRENT EFFECTIVENESS
 OF THE MISSION.

- ATTRIBUTES OF TEMPORARY ENTITY "ACTIVITY"**
- ACEL - CURRENT EFFECTIVENESS LEVEL**
EACH ACTIVITY HAS THE POTENTIAL TO REDUCE THE EFFECTIVENESS OF THE MISSION. THE ACEL SHOWS THE MISSION EFFECTIVENESS AT THE END OF THE ACTIVITY.
- AFWL - FIRST WORKER LEVEL**
FIRST OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
- AICT - INTERRUPTION CONTINUATION TIME**
TIME REQUIRED TO COMPLETE AN ACTIVITY WHICH HAS BEEN EITHER PREEMPTED OR INTERRUPTED
- ALRL - LOWER REDUCTION LIMIT**
LOWER LIMIT FOR A UNIFORM DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT THAT THE ACEL IS TO BE REDUCED.
- ANOA - NUMBER OF ACTIVITY**
SAME NUMBER AS THE CORRESPONDING KIND OF ACTIVITY
- ANWR - NUMBER OF WORKERS REQUIRED**
EACH ACTIVITY MAY REQUIRE FROM ZERO TO THE TOTAL NUMBER OF WORKERS IN THE SIMULATION TO PERFORM THE ACTIVITY.
- APAR - PARTIAL TEAM CAPABILITY**
YES/NO ATTRIBUTE WHICH INDICATES WHETHER AN ACTIVITY CAN BE PERFORMED WITH FEWER WORKERS THAN THE NUMBER REQUESTED.
- APMA - PERCENT MISSIONS AFFECTED**
IF A RANDOM NUMBER IS LESS THAN OR EQUAL TO THIS VALUE, THE ACEL WILL BE REDUCED BY AN AMOUNT BETWEEN THE ALRL AND THE AURL.
- APOA - PRIORITY OF ACTIVITY**
ONE OF THREE PRIORITIES WHICH INDICATES THE TIME OF DAY THAT AN ACTIVITY CAN BE PERFORMED AND ITS ABILITY TO TAKE WORKERS AWAY FROM LOWER PRIORITY MISSIONS.
- ASWL - SECOND WORKER LEVEL**
SECOND OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY.
- ATFC - TIME FUNCTION CODE**
INDICATES THE TYPE OF DISTRIBUTION FROM WHICH AN ACTIVITY TIME WILL BE SELECTED.
- ATPC - TEST PHASE CODE**
SEQUENTIAL CODE NUMBERS ASSIGNED TO ACTIVITIES WHICH COMPRISE A MISSION IN EACH TEST PHASE. PRIORITIES MAY BE ASSIGNED BETWEEN THE DIFFERENT TEST PHASES BY PREFIXING THE CODE WITH PRIORITY NUMBERS. (LOW NUMBER IS HIGH PRIORITY)
- ATWL - THIRD WORKER LEVEL**
THIRD OF THREE WORKER LEVEL PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF ACTIVITY

- AURL - UPPER REDUCTION LIMIT
UPPER LIMIT FOR A UNIFORM DISTRIBUTION WHICH IS USED
TO SELECT THE AMOUNT THE ACEL IS TO BE REDUCED.
- ATTRIBUTES OF EVENT NOTICES
- CTPN - CONFIRMED MISSION TEST PHASE NUMBER
PROGRAMMING VALUE USED TO DETERMINE THE TEST PHASE
OF THE NEXT MISSION.
- EAMN - END OF ACTIVITY MISSION NUMBER
PROGRAMMING VALUE TO IDENTIFY THE MISSION NUMBER
OF A COMPLETED ACTIVITY.
- ATTRIBUTES OF PERMANENT ENTITY "KIND.OF.ACTIVITY"
- KPAR - PARTIAL TEAM CAPABILITY
PERMANENT COPY OF YES/NO ATTRIBUTE OF EACH
KIND OF ACTIVITY WHICH INDECATES WHETHER AN
ACTIVITY CAN BE PERFORMED WITH FEWER WORKERS
THAN THE NUMBER REQUESTED.
- KFWL - FIRST WORKER LEVEL
PERMANENT COPY OF THE FIRST OF THREE WORKER LEVEL
PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF
ACTIVITY.
- KLRL - LOWER REDUCTION LIMIT
PERMANENT COPY OF LOWER LIMIT FOR A UNIFORM
DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT THAT
THE ACEL IS TO BE REDUCED.
- KPHA - PERCENT MISSIONS AFFECTED
PERMANENT COPY OF THE MAXIMUM VALUE OF A RANDOM
NUMBER WHICH INDICATES THAT THE MISSION EFFECTIVENESS
IS TO BE REDUCED BY AN AMOUNT BETWEEN KLRL AND KURL.
- KPOA - PRIORITY OF ACTIVITY
PERMANENT COPY DESIGNATING ONE OF THE THREE PRIORITIES
WHICH INDICATES THE TIME OF DAY THAT AN ACTIVITY
CAN BE PERFORMED AND ITS ABILITY TO TAKE WORKERS
AWAY FROM LOWER PRIORITY MISSIONS.
- KSWL - SECOND WORKER LEVEL
PERMANENT COPY OF THE SECOND OF THREE WORKER LEVEL
PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF
ACTIVITY.
- KTFC - TIME FUNCTION CODE
PERMANENT COPY OF THE TYPE OF DISTRIBUTION FROM WHICH
AN ACTIVITY TIME WILL BE SELECTED.
- KTWL - THIRD WORKER LEVEL
PERMANENT COPY OF THE THIRD OF THREE WORKER LEVEL
PREFERENCES WHICH CAN BE DESIGNATED FOR EACH KIND OF
ACTIVITY.
- KURL - UPPER REDUCTION LIMIT
PERMANENT COPY OF THE UPPER LIMIT FOR A UNIFORM
DISTRIBUTION WHICH IS USED TO SELECT THE AMOUNT
THAT THE ACEL IS TO BE REDUCED.

- ATTRIBUTES OF PERMANENT ENTITY "LEVEL.OF.WORKER"
- LNIN** - NUMBER OF IDLE WORKERS
CONTAINS THE VALUES INDICATING THE NUMBER OF IDLE WORKERS IN EACH LEVEL. USED TO GATHER STATISTICS AND FOR PROGRAM CONTROL.
- LTIT** - TOTAL IDLE TIME
CONTAINS THE SUM OF THE IDLE TIME IN A LEVEL. ITS VALUES ARE PRINTED IN THE REPORTS.
- ATTRIBUTES OF TEMPORARY ENTITY "MISSION"
- MCEL** - CURRENT EFFECTIVENESS LEVEL
VALUE OF MISSION EFFECTIVENESS AND MAY BE REDUCED BY EACH ACTIVITY IN THE MISSION.
- MNAC** - NEXT ACTIVITY CODE
NUMBER CORRESPONDING TO THE ATPC OF THE NEXT ACTIVITY OF THE MISSION. VALUE IS USED TO ORDER THE MISSIONS WAITING IN THE QUEUE.
- MNAP** - NEXT ACTIVITY PRIORITY
NUMBER CORRESPONDING TO THE APOA OF THE NEXT ACTIVITY OF THE MISSION. VALUE IS USED TO ORDER THE MISSIONS WAITING IN THE QUEUE.
- MTOA** - TIME OF ARRIVAL
TIME AT WHICH NOTIFICATION OF A MISSION WAS RECEIVED. VALUE IS USED TO DETERMINE MISSION DURATION AND THE TIME AT WHICH A PRIORITY 1 MISSION CAN BE CONDUCTED.
- MTPN** - TEST PHASE NUMBER
IDENTIFIES THE TEST PHASE TO WHICH THE MISSION BELONGS
- MXCI** - CANCELL CODE INDICATOR
IF THE VALUE IS 1 WHEN THE MISSION RECORD RETURNS FROM DETERMINE.PREMISSION.CANCELLATION OR CALCULATE. MISSION.EFFECTIVENESS, THE MISSION IS CANCELLED AND 1 IS ADDED TO THE NUMBER OF CANCELLED MISSIONS IN THE TEST PHASE.
- ATTRIBUTES OF PERMANENT ENTITY "TEST.PHASE"
- TCMP** - NUMBER OF COMPLETED MISSIONS
CURRENT COUNT IN EACH TEST PHASE
- TCNF** - NUMBER OF CONFIRMED MISSIONS
CURRENT COUNT IN EACH TEST PHASE. SOME OF THESE MISSIONS MAY BE CANCELLED.
- TCNX** - NUMBER OF CANCELLED MISSIONS
CURRENT COUNT IN EACH TEST PHASE.
- TDBM** - DAYS BEFORE NEXT MISSION
IF THE VALUE IS GREATER THAN OR EQUAL TO 1 IT IS USED WITH THE TTMC RANDOM VARIABLE TO DETERMINE THE TIME OF THE NEXT MISSION. IF THE VALUE IS LESS THAN 1, IT IS ADDED TO THE CURRENT SIMULATION TIME. IF THE SUM IS DURING NORMAL DUTY HOURS, THE MISSION IS SCHEDULED AT THAT TIME; OTHERWISE, THE MISSION IS SCHEDULED FOR

THE BEGINNING OF DUTY HOURS ON THE NEXT DUTY DAY.

TDCD - DATE COMPLETED
DATE THAT THE LAST MISSION WAS COMPLETED IN EACH TEST PHASE.

TNUM - NUMBER OF TEST PHASE
NUMBERS RANGE FROM 1 TO THE TOTAL NUMBER OF TEST PHASES

TSTY - STAY TIME STATISTIC
VALUE COMPUTED FOR EACH MISSION IN A TEST PHASE AS THE DIFFERENCE BETWEEN THE TIME COMPLETED AND THE TIME OF NOTIFICATION.

TTMC - TIME MISSION CONFIRMED RANDOM VARIABLE
USER PROVIDED DISTRIBUTION FOR THE TIME OF DAY THAT THE MISSION IS TO BE SCHEDULED.

TREQ - NUMBER OF REQUIRED MISSIONS
VALUE TO WHICH THE SUM OF ALL FINAL MCEL MUST EQUAL OR EXCEED IN EACH TEST PHASE.

GLOBAL VARIABLES

VBB1T - BEGIN BREAK 1 TIME

VBB2T - BEGIN BREAK 2 TIME

VBDOT - BEGIN DUTY DAY TIME

VBLBT - BEGIN LUNCH BREAK TIME

VEB1T - END BREAK 1 TIME

VEB2T - END BREAK 2 TIME

VEDDT - END DUTY DAY TIME

VELBT - END LUNCH BREAK TIME

VMROP - MONTHLY REPORT OPTION
ENTERING NO IN THE INPUT DATA DELETES THIS REPORT.

VRMPT - REQ MISSION PREPARATION TIME (INCLUDED IN DURATION TIME)
TIME REQUIRED TO PREPARE FOR AN ACTIVITY. (BETWEEN MISSION NOTIFICATION AND THE FIRST PRIORITY 1 ACTIVITY) IF ALL PREMISSION ACTIVITIES HAVE BEEN COMPLETED PRIOR TO THIS TIME, THE MISSION IS FILED IN WAIT.FOR.MISSION.TIME.

VRUNS - NUMBER OF SIMULATION RUNS FOR ENTIRE DATA PACKAGE
NUMBER OF TIMES THE ENTIRE DATA PACKAGE IS TO BE RUN WITH DIFFERENT SEED VALUES IN THE RANDOM NUMBER GENERATORS TO PROVIDE A DISTRIBUTION OF VALUES.

VWEAM - WEEKEND ACTIVE MISSIONS
YES OR NO OPTION TO ALLOW PRIORITY MISSIONS TO BE CONDUCTED ON WEEKENDS.

VWROP - WEEKLY REPORT OPTION
ENTERING NO IN THE INPUT DATA DELETES THIS REPORT

VWDD - WORK WEEK DUTY DAYS
NUMBER OF DAYS IN THE NORMAL DUTY WEEK. TO INTERNALLY SCHEDULE MISSIONS ON THE WEEKEND THIS VALUE MUST BE INCREASED ACCORDINGLY.

REPORT VARIABLES

VWCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST WEEK

VWCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST WEEK
VWCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST WEEK
VMCHP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST MONTH
VMCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST MONTH
VMCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST MONTH

LOCAL VARIABLES (WITH A SINGLE LETTER PREFIX OR SUFFIX)

"AZ" - ACTIVITY.....EXAMPLE AZG OR PAZ
"MZ" - MISSION.....EXAMPLE MZF OR QMZ
"TZ" - TEST PHASE..EXAMPLE TZB OR YTZ
"WZ" - WORKER.....EXAMPLE WZD OR UWZ

ATTRIBUTES OF PERMANENT ENTITY "WORKER"

WLEV - LEVEL
CATAGORY OR TYPE OF WORKER
WNAM - NAME
NAME OF EACH WORKER (6 DIGETS OR LESS).
WSTA - STATE
SHOWS THE CURRENT STATUS OF THE WORKER AS IDLE, BUSY,
BREAK, OFF DUTY, OR OVERTIME.

ATTRIBUTES OF TEMPORARY ENTITY "XFACTOR"

XUPL - UPPER PERCENTAGE LIMIT
IF A RANDOM NUMBER IS LESS THAN OR EQUAL TO THIS
VALUE THE MISSION IS CANCELLED.

PROGRAMMING CODES

ACTIVITY PRIORITY CODES

- 1 - ACTIVE MISSION, PREEMPT WORKERS, CONTINUE UNTIL FINISHED
- 2 - MUST BE COMPLETED IN CURRENT DAY
- 3 - NORMAL DUTY HOURS

WORKER STATUS CODES

- 0 - IDLE
- 1 - BUSY
- 2 - BREAK
- 3 - OFF.DUTY
- 4 - OVERTIME

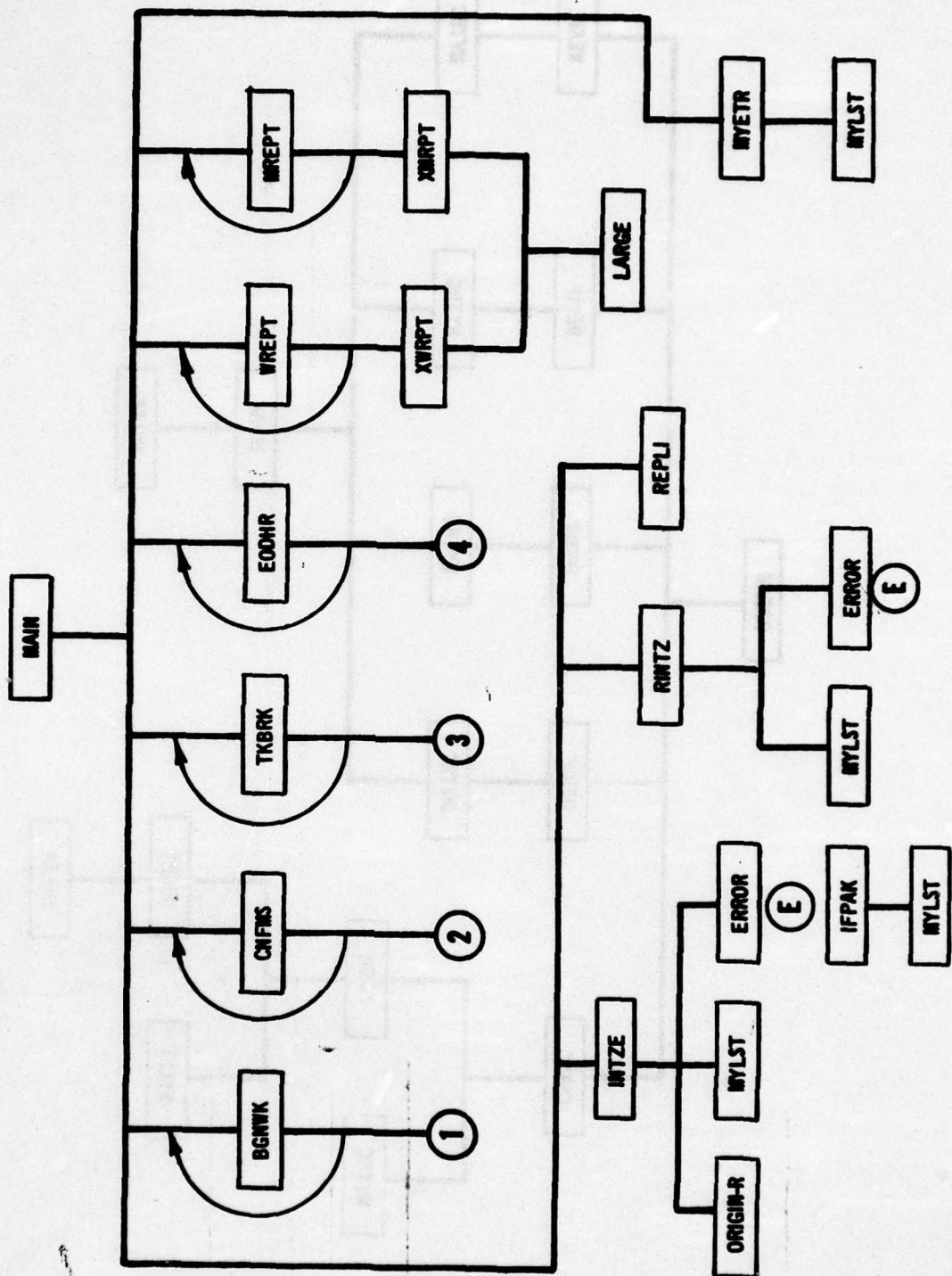
TIME FUNCTION CODES

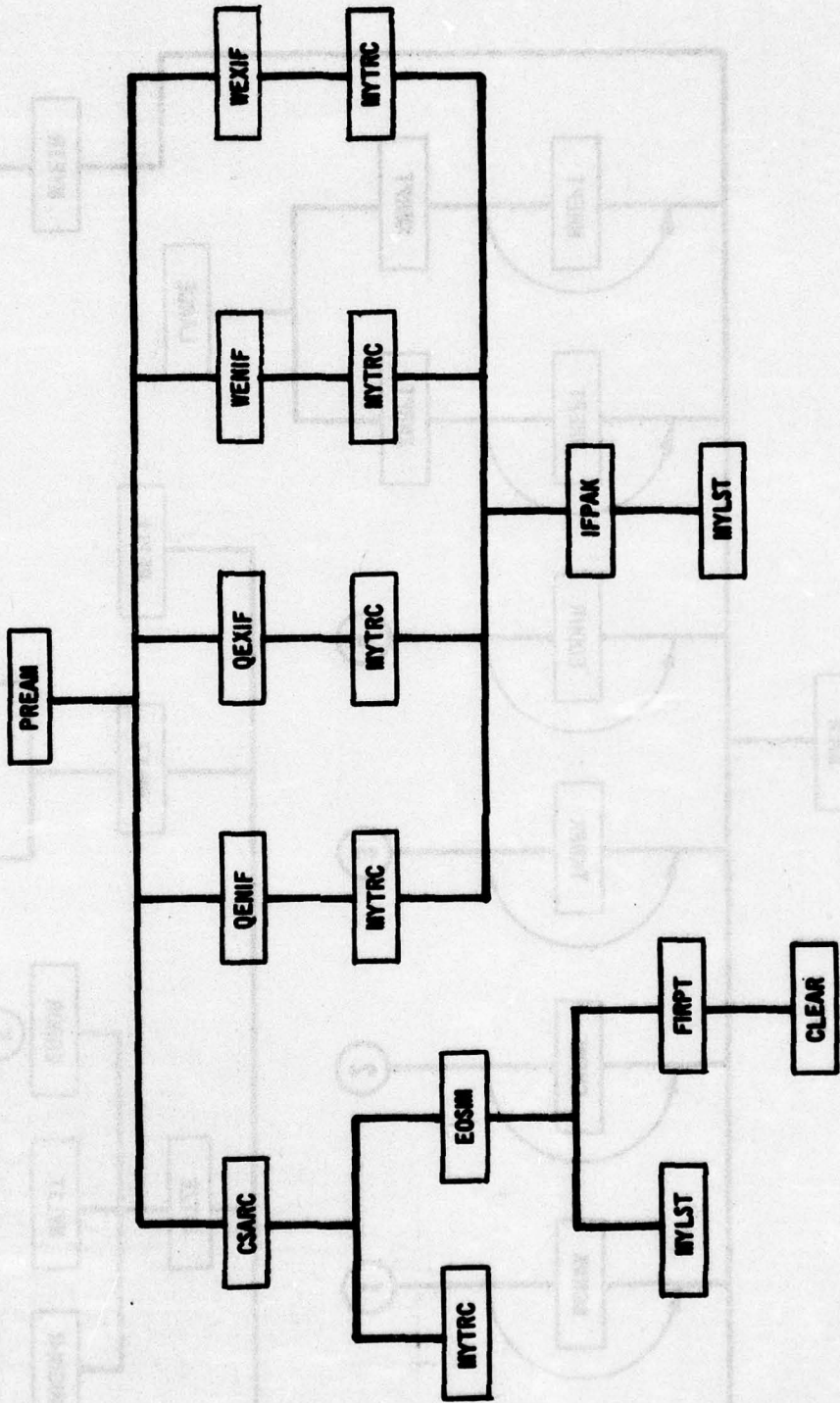
ARGUMENTS

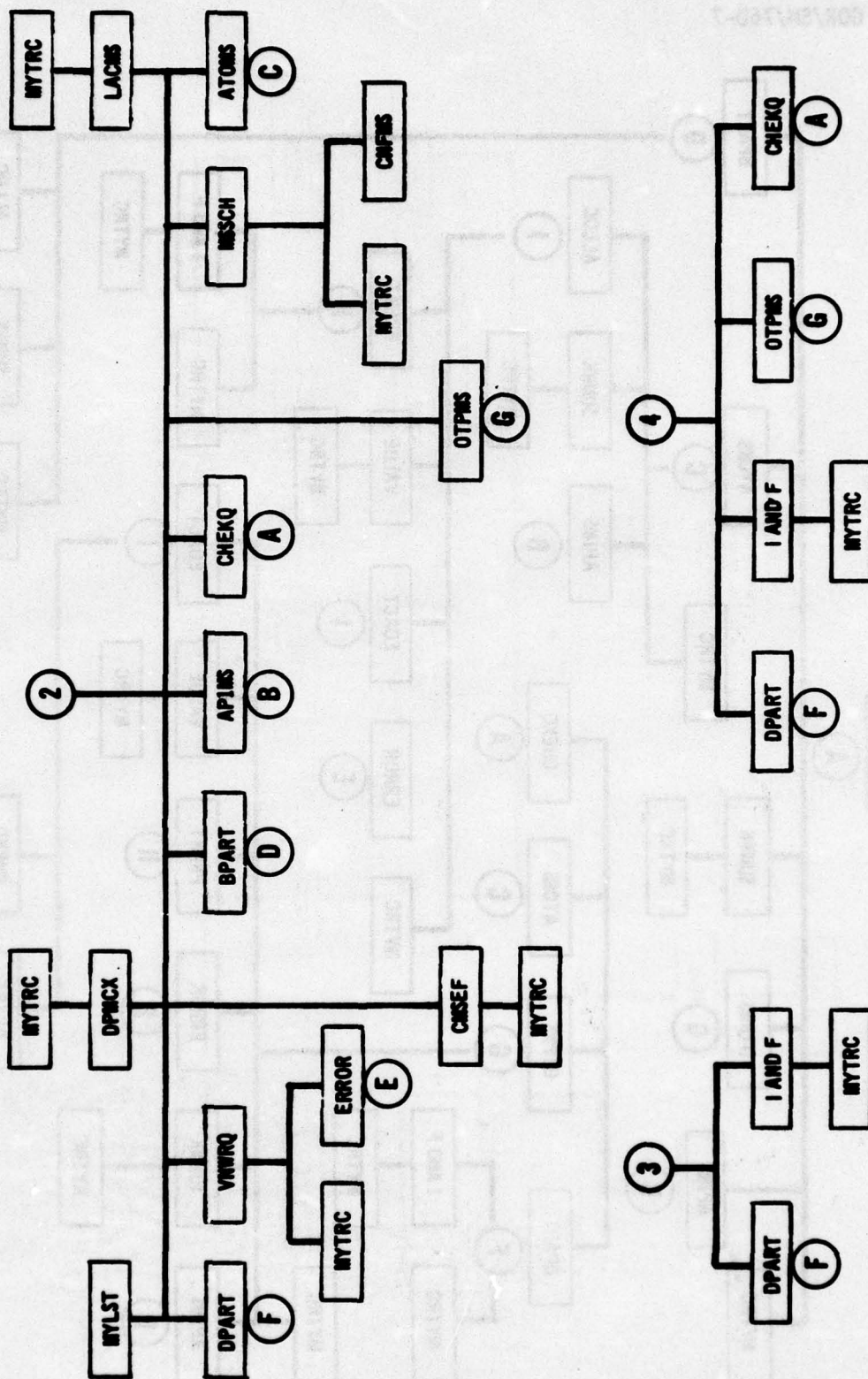
	1ARG	2ARG
1 - BETA	POWER OF X	POWER OF (1-X)
2 - BINOMIAL	NUMBER OF TRIALS	PROB OF SUCCESS
3 - CONSTANT	MEAN	VARIANCE
4 - ERLANG	MEAN	SHAPE PARAMETER
5 - EXPONENTIAL	MEAN	MEAN
6 - GAMMA	MEAN	SHAPE PARAMETER
7 - NORMAL	MEAN	STD DEVIATION
8 - POISSON	MEAN	MEAN
9 - UNIFORM	BEGINNING VALUE	ENDING VALUE
10 - WEIBULL	SCALE PARAMETER	SHAPE PARAMETER
-1 - INTEGER STEP	N/A	N/A
-2 - REAL STEP	N/A	N/A
-3 - CONTINUOUS	N/A	N/A

Flow Chart Abbreviations

ABBREVIATION	EVENT OR ROUTINE
ALLOC	ALLOCATION
AP1MS-----	ACCOMPLISH. PRI. MISSION
ATOMS	ATTEND. TO. MISSION
BGNWK	BEGIN. WORKING
BPART-----	BUILD. PARTIAL. TEAMS
CHEKQ	CHECK. QUEUE
CLEAR	CLEAR
CMSEF-----	CALCULATE. MISSION. EFFECTIVENESS
CNFMS	CONFIRMED. MISSION
CSARC	COMPUTE. STAY. TIME. AND. RECORD. CANCELLATION
DPART-----	DISBAND. PARTIAL. TEAMS
DPMCX	DETERMINE. PREMISSION. CANCELLATION
EOACT	EO. ACTIVITY
EODHR-----	EO. DUTY. HOURS
EOSIM	EO. SIMULATION
ERROR	ERROR. PACKAGE
FIRPT-----	FINAL. REPORT
IANDF	INTERRUPT. AND. FILE
IFPAK	INFO. PACKAGE
INTZE-----	INITIALIZE
LACMS	LOOK. AHEAD. COMPLETE. MISSIONS
LARGE	LARGEST
MAIN-----	MAIN
MREPT	MREPORT
MSSCH	MISSION. SCHEDULE
MYETR-----	MYE. TRACE
MYLST	MY. LIST
MYTRC	MY. TRACE
OTPMS-----	OVERTIME. ACTIVITIES. FOR. PRIORITY. MISSIONS
PREAM	PREAMBLE
PREMT	PREEMPT
QENIF-----	QENTER. INFO
QEXIF	QEXIT. INFO
REPLI	REPLICATION
RINTZ	RUN. INITIALIZATION
SUMWK	SUM. WORKERS
TKBRK	TAKE. BREAK
VALUE-----	VALUE
VNWRQ	VALIDATE. NUMBER. OF. WORKERS. REQUESTED
WENIF	WENTER. INFO
WEXIF-----	WENTER. INFO
WREPT	WREPORT
XMRPT	XMREPORT
XMRPT-----	XWREPORT







00000360
 00000370
 00000380
 00000390
 00000400
 00000410
 00000420
 00000430
 00000440
 00000450
 00000460
 00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700

```

A APAR, A AFWL, A ASWL, A ATWL, A ACEL, A APMA, A ALRL, A AURL
  BELONGS TO A STRUCTURE AND A ROUTING, AND OWNS A TEAM
  EVERY XFACTOR HAS A XUPL, AND BELONG TO A
  TCANCELLATION.PROBABILITY AND A MCANCELLATION.PROBABILITY
..
THE SYSTEM OWNS THE QUEUE
  DEFINE QUEUE AS A FIFO SET RANKED BY LOW MNAP, THEN BY LOW MNAC,
  THEN BY LOW MTOA
THE SYSTEM OWNS THE WAIT.FOR.MISSION.TIME
  DEFINE WAIT.FOR.MISSION.TIME AS A FIFO SET RANKED BY LOW MTOA
..
**DECLARATION OF DYNAMIC SYSTEM STRUCTURE
..
EVENT NOTICES INCLUDE BEGIN.WORKING, TAKE.BREAK, EO.DUTY.HOURS,
ACCOMPLISH.PRI.MISSION, MREPORT, MREPORT,
EO.SIMULATION, AND FINAL.REPORT
EVERY CONFIRMED.MISSION HAS A CTPN
EVERY EO.ACTIVITY HAS A EAMN
EXTERNAL EVENTS ARE CONFIRMED.MISSION AND EO.SIMULATION
EXTERNAL EVENT UNITS ARE 8 AND 9
PRIORITY ORDER IS EO.ACTIVITY, EO.DUTY.HOURS, EO.SIMULATION,
CONFIRMED.MISSION, TAKE.BREAK, ACCOMPLISH.PRI.MISSION,
BEGIN.WORKING, MREPORT, MREPORT, AND FINAL.REPORT
DEFINE CSTARC TO MEAN COMPUTE.STAY.TIME.AND.RECORD.CANCELLATIONS
BEFORE DESTROYING MISSION CALL CSTARC
BEFORE FILING IN QUEUE CALL CENTER.INFO
BEFORE FILING IN WAIT.FOR.MISSION.TIME CALL WENTER.INFO
BEFORE REMOVING FROM QUEUE CALL QEXIT.INFO
BEFORE REMOVING FROM WAIT.FOR.MISSION.TIME CALL MEXIT.INFO
..
**DATA COLLECTION AND ANALYSIS DECLARATIONS
..
NORMALLY MODE IS REAL
TALLY WAVG.STAY AS THE WEEKLY MEAN, WSTD.STAY AS THE WEEKLY STD.DEV,
WSUM.STAY AS THE WEEKLY SUM, WSQSOME.SQUARES.STAY AS THE WEEKLY
    
```

```

SUM.OF.SQUARES, AND MNUM.STAY AS THE WEEKLY NUMBER OF TSTY
TALLY MAVG.STAY AS THE MONTHLY MEAN, MSTD.STAY AS THE MONTHLY STD.DEV,
MSUM.STAY AS THE MONTHLY SUM, MSOME.SQUARES.STAY AS THE MONTHLY
SUM.OF.SQUARES, AND MNUM.STAY AS THE MONTHLY NUMBER OF TSTY
TALLY TAVG.STAY AS THE MEAN, TSTD.STAY AS THE STD.DEV, TSUM.STAY AS
THE SUM, TSOME.SQUARES.STAY AS THE SUM.OF.SQUARES, AND TNUM.STAY AS
THE NUMBER OF TSTY
ACCUMULATE HQSUM.QUEUE AS THE WEEKLY SUM, HQSTD.QUEUE AS THE WEEKLY
STD.DEV, HQAVG.QUEUE AS THE WEEKLY MEAN, HQMIN.QUEUE AS THE WEEKLY
MINIMUM, HQMAX.QUEUE AS THE WEEKLY MAXIMUM AND HREQ(0 TO 500 BY 1)
AS THE WEEKLY HISTOGRAM OF N.QUEUE
ACCUMULATE MQSUM.QUEUE AS THE MONTHLY SUM, MQSTD.QUEUE AS THE MONTHLY
STD.DEV, MQAVG.QUEUE AS THE MONTHLY MEAN, MQMIN.QUEUE AS THE MONTHLY
MINIMUM, MQMAX.QUEUE AS THE MONTHLY MAXIMUM AND MREQ(0 TO 500 BY 1)
AS THE MONTHLY HISTOGRAM OF N.QUEUE
ACCUMULATE TQSUM.QUEUE AS THE SUM, TQSTD.QUEUE AS THE STD.DEV,
TQAVG.QUEUE AS THE MEAN, TQMAX.QUEUE AS
THE MAXIMUM, AND TFREQ(0 TO 500 BY 1) AS THE HISTOGRAM OF N.QUEUE
ACCUMULATE HLMAX.IDLE.WORKERS AS THE WEEKLY MAXIMUM,
HLMAX.IDLE.WORKERS AS THE MONTHLY MAXIMUM,
TLMAX.IDLE.WORKERS AS THE MAXIMUM,
MIDDLE.WORKERS(0 TO 100 BY 1) AS THE WEEKLY HISTOGRAM,
MIDDLE.WORKERS(0 TO 100 BY 1) AS THE MONTHLY HISTOGRAM, AND
TIDLE.WORKERS(0 TO 100 BY 1) AS THE HISTOGRAM OF LNIW
ACCUMULATE HOBSERVED.STATE(0 TO 4 BY 1) AS THE WEEKLY HISTOGRAM,
HOBSERVED.STATE(0 TO 4 BY 1) AS THE MONTHLY HISTOGRAM, AND
TOBSERVED.STATE(0 TO 4 BY 1) AS THE HISTOGRAM OF MSTA
NORMALLY MODE IS INTEGER
..
.. MISC DECLARATIONS
..
DEFINE IDLE TO MEAN 0
DEFINE BUSY TO MEAN 1
DEFINE BREAK TO MEAN 2
DEFINE OFF.DUTY TO MEAN 3

```

```

DEFINE OVERTIME TO MEAN 4
DEFINE WK TO MEAN *HOURS.V+7 HOURS
DEFINE NEXT.MONTH TO
  MEAN AT REAL.F(MONTH.F(TIME.V+31),1, YEAR.F(TIME.V+31))
DEFINE KOA TO MEAN KIND.OF.ACTIVITY
DEFINE LOFW TO MEAN LEVEL.OF.WORKER
DEFINE N.LOFW TO MEAN N.LEVEL.OF.WORKER
DEFINE SAME.SKILL.FACTOR, DELETE.REDUCTION.OF.MISSION.EFFECTIVENESS,
CHECK, EMPIRICAL.FUNCTIONS, APAR, KNAM, KPAR, VMROP, VMEAN, VMROP,
  AND WNAM AS ALPHA VARIABLES
DEFINE ACEL, AICT, ALRL, APMA, AURL, KLRL, KPMA, KURL, LTIT, MCEL,
MTOA, TCMP, TOBM, TDCD, TSTY, V9BIT, V8B2T, VBOOT, VBLBT, VEBIT,
VEB2T, VEDDT, VELBT, VRMPT, XUPL, CK.MIN, CK.MAX AS REAL VARIABLES
DEFINE VRUNS, MONTH, DAY, YEAR, VVDD AS INTEGER VARIABLES
DEFINE SAVESEED, VMCNF, VMCNX, VMCNF, VMCNX AS
  1-DIMENSIONAL INTEGER VARIABLES
DEFINE VMCMP, VMCMP AS 1-DIMENSIONAL REAL VARIABLES
DEFINE SUM.WORKERS AS AN INTEGER FUNCTION WITH 3 ARGUMENTS
DEFINE LACH TO MEAN LOOK.AHEAD.COMPLETE.MISSIONS
DEFINE LACH AS A REAL FUNCTION WITH 1 ARGUMENT
DEFINE LARGEST AS A REAL FUNCTION WITH 1 ARGUMENT
DEFINE VALUE AS A REAL FUNCTION WITH 1 ARGUMENT
DEFINE INITIALIZE, REPLICATION, XWREPORT, XWREPORT AS
  RELEASABLE ROUTINES
END

```

```

00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300

```

00001310
 00001320
 00001330
 00001340
 00001350
 00001360
 00001370
 00001380
 00001390
 00001400
 00001410
 00001420
 00001430
 00001440
 00001450
 00001460
 00001470
 00001480
 00001490
 00001500
 00001510
 00001520
 00001530
 00001540
 00001550
 00001560
 00001570
 00001580
 00001590
 00001600
 00001610
 00001620
 00001630
 00001640
 00001650

MAIN

```

..
*REPLICATE*
CALL INITIALIZE
IF VRUNS = 1
  RELEASE INITIALIZE
  REFUSE REPLICATION
ALWAYS
LET BETWEEN.V = 'WYE.TRACE'
*SIMULATE*
CALL RUN.INITIALIZATION
SCHEDULE A BEGIN.WORKING AT VBDDT/24
SCHEDULE A CONFIRMED.MISSION(1) AT 8/24
SCHEDULE A TAKE.BREAK AT VBB1T/24
SCHEDULE A BEGIN.WORKING AT VEB1T/24
SCHEDULE A TAKE.BREAK AT VBLBT/24
SCHEDULE A BEGIN.WORKING AT VELBT/24
SCHEDULE A TAKE.BREAK AT VBB2T/24
SCHEDULE A BEGIN.WORKING AT VEB2T/24
SCHEDULE A EO.DUTY.HOURS AT VEDDT/24
IF VMROP = "NO"
  RELEASE XWREPORT
  GO TO HRPT
ELSE
  SCHEDULE A WREPORT IN 1 WK
  *HRPT*
  IF VMROP = "NO"
    RELEASE XWREPORT
    GO TO START
  ELSE
    SCHEDULE A MREPORT NEXT.MONTH
  *START*
  START SIMULATION
  USE UNIT 7 FOR INPUT
  IF DATA IS ENDED

```

```

IF VRUNS GT 1
  CALL REPLICATION
  GO REPLICATE
ELSE
  STOP
ELSE
  **RUN SIMULATION WITH DIFFERENT DATA
  START NEW PAGE
  PRINT 1 LINE THUS
  RUN SIMULATION WITH DIFFERENT WORKERS AND TIME BETWEEN MISSIONS
  GO *SIMULATE*
END

```

```

00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770

```

```

..
ROUTINE TO INITIALIZE

USE UNIT 7 FOR INPUT
IF VRUNS NE 0
  SKIP 4 FIELDS
  GO READ
ELSE
  READ VRUNS, MONTH, DAY, YEAR
  CALL ORIGIN.R(MONTH, DAY, YEAR)
  *READ*
  READ VMROP, VMROP,
  N.KIND.OF.ACTIVITY, DELETE.REDUCTION.OF.MISSION.EFFECTIVENESS
  CREATE EVERY KIND.OF.ACTIVITY
  IF DELETE.REDUCTION.OF.MISSION.EFFECTIVENESS = "YES"
    FOR EACH KIND.OF.ACTIVITY DO
      READ KNAM(KOA), KPOA(KOA), KTFC(KOA), KPAR(KOA), KFWL(KOA),
        KSWL(KOA), KTHL(KOA)
      LET KPMA(KOA) = 0
      LET KLRL(KOA) = 0
      LET KURL(KOA) = 0
      LOOP
      JUMP AHEAD
    ELSE
      FOR EACH KIND.OF.ACTIVITY
        READ KNAM(KOA), KPOA(KOA), KTFC(KOA), KPAR(KOA), KFWL(KOA),
          KSWL(KOA), KTHL(KOA), KPMA(KOA), KLRL(KOA), KURL(KOA)
        HERE
        READ N.TEST.PHASE
        CREATE EVERY TEST.PHASE
        FOR EACH TEST.PHASE DO
          READ TNUM(TEST.PHASE), TREQ(TEST.PHASE), TTMC(TEST.PHASE)
          FOR EACH KIND.OF.ACTIVITY
            READ 1ARG(TEST.PHASE, KOA), 2ARG(TEST.PHASE, KOA)
          UNTIL MODE IS ALPHA DO
            CREATE AN ACTIVITY CALLED BAZ

```

```

00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120

```

00002130
 00002140
 00002150
 00002160
 00002170
 00002180
 00002190
 00002200
 00002210
 00002220
 00002230
 00002240
 00002250
 00002260
 00002270
 00002280
 00002290
 00002300
 00002310
 00002320
 00002330
 00002340
 00002350
 00002360
 00002370
 00002380
 00002390
 00002400
 00002410
 00002420
 00002430
 00002440
 00002450
 00002460
 00002470

```

READ ATPC(BAZ), ANOA(BAZ), ANHR(BAZ)
LET APOA(BAZ) = KPOA(ANOA(BAZ))
LET ATFC(BAZ) = KTFC(ANOA(BAZ))
LET AICT(BAZ) = 0
LET APAR(BAZ) = KPAR(ANOA(BAZ))
LET AFWL(BAZ) = KFWL(ANOA(BAZ))
LET ASWL(BAZ) = KSWL(ANOA(BAZ))
LET ATWL(BAZ) = KTWL(ANOA(BAZ))
LET ACEL(BAZ) = 1
LET APMA(BAZ) = KPMA(ANOA(BAZ))
LET ALRL(BAZ) = KLRL(ANOA(BAZ))
LET AURL(BAZ) = KURL(ANOA(BAZ))
FILE BAZ IN STRUCTURE( TEST.PHASE)
LOOP
SKIP 1 FIELD
**TO DELETE THIS FEATURE ENTER A "**
UNTIL MODE IS ALPHA DO
CREATE AN XFACTOR
READ XUPL
FILE XFACTOR IN TCANCELLATION.PROBABILITY
LOOP
SKIP 1 FIELD
LOOP
READ CHECK
IF CHECK E "OK"
JUMP AHEAD
ELSE
READ EMPERICAL.FUNCTIONS
IF EMPERICAL.FUNCTIONS = "NO"
GO TO NONE
ELSE
READ N.INTEGER.STEP.FUNCTION
CREATE EVERY INTEGER.STEP.FUNCTION
FOR EVERY INTEGER.STEP.FUNCTION
READ ONE( INTEGER.STEP.FUNCTION)
    
```


ANMR - NUMBER OF WORKERS REQUIRED	00002630
APAR - PARTIAL TEAM CAPABILITY	00002640
APHA - PERCENT MISSIONS AFFECTED	00002650
APOA - PRIORITY OF ACTIVITY	00002660
ASWL - SECOND WORKER LEVEL	00002670
ATFC - TIME FUNCTION CODE	00002680
ATPC - TEST PHASE CODE	00002690
ATWL - THIRD WORKER LEVEL	00002900
AURL - UPPER REDUCTION LIMIT	00002910
	00002920
	00002930
	00002940
CTPN - CONFIRMED MISSION TEST PHASE NUMBER	00002950
EAMN - END OF ACTIVITY MISSION NUMBER	00002960
	00002970
	00002980
	00002990
	00003000
	00003010
	00003020
	00003030
	00003040
	00003050
	00003060
	00003070
	00003080
	00003090
	00003100
	00003110
	00003120
	00003130
	00003140
	00003150
	00003160
	00003170

ATTRIBUTES OF EVENT NOTICES
CTPN - CONFIRMED MISSION TEST PHASE NUMBER
EAMN - END OF ACTIVITY MISSION NUMBER
ATTRIBUTES OF PERMANENT ENTITY "KIND.OF.ACTIVITY"
KFML - FIRST WORKER LEVEL
KLRL - LOWER REDUCTION LIMIT
KPHA - PERCENT MISSIONS AFFECTED
KPAR - PARTIAL TEAM CAPABILITY
KPOA - PRIORITY OF ACTIVITY
KSWL - SECOND WORKER LEVEL
KTFC - TIME FUNCTION CODE
KTWL - THIRD WORKER LEVEL
KURL - UPPER REDUCTION LIMIT
ATTRIBUTES OF PERMANENT ENTITY "LEVEL.OF.WORKER"
LNIM - NUMBER OF IDLE WORKERS
LYIT - TOTAL IDLE TIME
ATTRIBUTES OF TEMPORARY ENTITY "MISSION"
MCEL - CURRENT EFFECTIVENESS LEVEL
MNAC - NEXT ACTIVITY CODE
MNAP - NEXT ACTIVITY PRIORITY
MTOA - TIME OF ARRIVAL
MTPN - TEST PHASE NUMBER

MXCI - CANCEL CODE INDICATOR	00003160
	00003190
	00003200
ATTRIBUTES OF PERMANENT ENTITY "TEST.PHASE"	00003210
TCMP - NUMBER OF COMPLETED MISSIONS	00003220
TCNF - NUMBER OF CONFIRMED MISSIONS	00003230
TCNX - NUMBER OF CANCELLED MISSIONS	00003240
TDBM - DAYS BEFORE NEXT MISSION	00003250
TDCD - DATE COMPLETED	00003260
TNUM - NUMBER OF TEST PHASE	00003270
TSTY - STAY TIME STATISTIC	00003280
TTMC - TIME MISSION CONFIRMED RANDOM VARIABLE	00003290
TREQ - NUMBER OF REQUIRED MISSIONS	00003300
	00003310
	00003320
GLOBAL VARIABLES	00003330
VBB1T - BEGIN BREAK 1 TIME	00003340
VBB2T - BEGIN BREAK 2 TIME	00003350
VBD0T - BEGIN DUTY DAY TIME	00003360
VBLBT - BEGIN LUNCH BREAK TIME	00003370
VEB1T - END BREAK 1 TIME	00003380
VEB2T - END BREAK 2 TIME	00003390
VED0T - END DUTY DAY TIME	00003400
VELBT - END LUNCH BREAK TIME	00003410
VMROP - MONTHLY REPORT OPTION	00003420
VRMPT - REQ MISSION PREPARATION TIME (INCLUDED IN DURATION TIME)	00003430
VRUNS - NUMBER OF SIMULATION RUNS FOR ENTIRE DATA PACKAGE	00003440
VWEAM - WEEKEND ACTIVE MISSIONS	00003450
VWROP - WEEKLY REPORT OPTION	00003460
VWDD - WORK WEEK DUTY DAYS	00003470
	00003480
REPORT VARIABLES	00003490
VWCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST WEEK	00003500
VMCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST WEEK	00003510
VMCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST WEEK	00003520
VMCMP - CUMULATIVE NUMBER OF COMPLETED MISSIONS AS OF LAST MONTH	
VMCNF - CUMULATIVE NUMBER OF CONFIRMED MISSIONS AS OF LAST MONTH	

VMCNX - CUMULATIVE NUMBER OF CANCELLED MISSIONS AS OF LAST MONTH

00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870

LOCAL VARIABLES(WITH A SINGLE LETTER PREFIX OR SUFFIX)

- "AZ" - ACTIVITY...EXAMPLE AZG OR PAZ
- "MZ" - MISSION...EXAMPLE MZF OR QMZ
- "TZ" - TEST PHASE...EXAMPLE TZB OR YTZ
- "WZ" - WORKER...EXAMPLE WZD OR UWZ

ATTRIBUTES OF PERMANENT ENTITY "WORKER"

- WLEV - LEVEL
- WNAH - NAME
- WSTA - STATE

ATTRIBUTES OF TEMPORARY ENTITY "XFACTOR"

- XUPL - UPPER PERCENTAGE LIMIT

PROGRAMMING CODES

ACTIVITY PRIORITY CODES

- 1 - ACTIVE MISSION, PREEMPT WORKERS, CONTINUE UNTIL FINISHED
- 2 - MUST BE COMPLETED IN CURRENT DAY
- 3 - NORMAL DUTY HOURS

WORKER STATUS CODES

- 0 - IDLE
- 1 - BUSY
- 2 - BREAK
- 3 - OFF-DUTY
- 4 - OVERTIME

TIME FUNCTION CODES

- 1 - BETA POWER OF X
- 2 - BINOMIAL -----NUMBER OF TRIALS-----PROBABILITY OF SUCCESS

ARGUMENTS

1ARG POWER OF X
2ARG POWER OF (1-X)

```

3 - CONSTANT          MEAN          VARIANCE          00003860
4 - ERLANG            MEAN          SHAPE PARAMETER  00003890
5 - EXPONENTIAL      MEAN          MEAN              00003900
6 - GAMMA            MEAN          SHAPE PARAMETER  00003910
7 - NORMAL           MEAN          STANDARD DEVIATION 00003920
8 - POISSON          MEAN          MEAN              00003930
9 - UNIFORM          BEGINNING VALUE  ENDING VALUE      00003940
10 - WEIBULL         SCALE PARAMETER  SHAPE PARAMETER   00003950
-1 - INTEGER STEP    N/A             N/A                00003960
-2 - REAL STEP      N/A             N/A                00003970
-3 - CONTINUOUS     N/A             N/A                00003980
START NEW PAGE      00003990
PRINT 11 LINES THUS 00004000
                     00004010
                     00004020
                     00004030
                     00004040
                     00004050
                     00004060
                     00004070
                     00004080
                     00004090
                     00004100
                     00004110
                     00004120
                     00004130
                     00004140
                     00004150
                     00004160
                     00004170
                     00004180
                     00004190
                     00004200
                     00004210
                     00004220

```

```

#####
INFORMATION COMMON TO ALL SIMULATION RUNS
#####

```

```

-----
INFORMATION COMMON TO ALL TEST PHASES
-----

```

```

CALL MY.LIST(12)*KIND OF ACTIVITY
CALL MY.LIST(9)*1ARG
CALL MY.LIST(10)*2ARG
SKIP 3 LINES
PRINT 10 LINES WITH MONTH, DAY, YEAR, VMDD, VBDDT, VBB1T, VEB1T,
VBLBT, VELBT, VBB2T, VEB2T, VEDDT THUS
SIMULATION BEGINS----- ** ** **
THE NUMBER OF DUTY DAYS PER WEEK IS-- *
THE DUTY DAY BEGINS AT----- ** **
THE MORNING COFFEE BREAK BEGINS AT--- ** **
THE MORNING COFFEE BREAK ENDS AT---- ** **

```

```

THE LUNCH PERIOD BEGINS AT----- **.**
THE LUNCH PERIOD ENDS AT----- **.**
THE AFTERNOON COFFEE BREAK BEGINS AT- **.**
THE AFTERNOON COFFEE BREAK ENDS AT--- **.**
THE DUTY DAY ENDS AT----- **.**
PRINT 7 LINES THUS

```

TEST PHASE INFORMATION

```

FOR EACH TEST.PHASE DO
PRINT 2 LINES THUS
### NUMBER MISSION TIME PROB TIME ACTIVITY SEQUENCE AND RQMTS
LET I = F.TTMC(TEST.PHASE) CODE ACTIVITY WORKERS REQ
LET J = F.STRUCTURE(TEST.PHASE)
PRINT 1 LINE WITH TNUM(TEST.PHASE), PROB.A(I), RVALUE.A(I), ATPC(J),
ANOAA(J), ANMR(J) THUS
** * ** ** ** **
UNTIL I=J=0 DO
IF I NE 0
LET I = S.TTMC(I)
THEN IF I NE 0
WRITE PROB.A(I) AND RVALUE.A(I) AS B 16,2 D(9,2)
ALWAYS
IF J NE 0
LET J = S.STRUCTURE(J)
THEN IF J NE 0
WRITE ATPC(J), ANOA(J), ANMR(J) AS B 35, 3 I 11
ALWAYS
START NEW LINE
LOOP

```

```

00004230
00004240
00004250
00004260
00004270
00004280
00004290
00004300
00004310
00004320
00004330
00004340
00004350
00004360
00004370
00004380
00004390
00004400
00004410
00004420
00004430
00004440
00004450
00004460
00004470
00004480
00004490
00004500
00004510
00004520
00004530
00004540
00004550
00004560
00004570

```

CALL MY.LIST(3,TEST.PHASE)..TEST PHASE ATTRIBUTES
CALL MY.LIST(4,TEST.PHASE)..ACTIVITIES IN STRUCTURE(TEST PHASE)
CALL MY.LIST(6,TEST.PHASE)..XUPL IN T.CX.PROB(TEST PHASE)

SKIP 5 LINES

LOOP

RETURN

HERE

CALL ERROR.PACKAGE(1)

END

00004580
00004590
00004600
00004610
00004620
00004630
00004640
00004650
00004660

00004670
00004680
00004690
00004700
00004710
00004720
00004730
00004740
00004750
00004760
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010

ROUTINE FOR RUN. INITIALIZATION

```

..
USE UNIT 7 FOR INPUT
UNTIL MODE IS ALPHA
  READ CK. MIN, CK. MAX
  SKIP 1 FIELD
  READ CHECK
  IF CHECK NE "OK"
    JUMP AHEAD
  ELSE
    READ SAME. SKILL. FACTOR
    READ N. LEVEL. OF. WORKER
    CREATE EVERY LEVEL. OF. WORKER
    READ N. WORKER
    CREATE EVERY WORKER
    FOR EACH WORKER DO
      READ WNAME(WORKER), WLEV(WORKER)
      LET WSTA(WORKER) = OFF. DUTY
      IF SAME. SKILL. FACTOR = "YES"
        FOR EACH KIND. OF. ACTIVITY
          LET SKILL. FACTOR(WORKER, KOA) = 1
        GO LOOP
      ELSE
        FOR EACH KIND. OF. ACTIVITY
          READ SKILL. FACTOR(WORKER, KOA)
        LOOP
    LOOP
  START NEW PAGE
  PRINT 7 LINES THUS

```


INFORMATION ENTERED FOR CURRENT SIMULATION RUN
#####

00005020
 00005030
 00005040
 00005050
 00005060
 00005070
 00005080
 00005090
 00005100
 00005110
 00005120
 00005130
 00005140
 00005150
 00005160
 00005170
 00005180
 00005190
 00005200
 00005210
 00005220
 00005230
 00005240
 00005250
 00005260
 00005270
 00005280
 00005290
 00005300
 00005310
 00005320
 00005330
 00005340
 00005350
 00005360

PRINT 5 LINE THUS

 INFORMATION COMMON TO ALL TEST PHASES

CALL MY.LIST(1) WORKERS
 CALL MY.LIST(11) SKILL FACTOR
 PRINT 8 LINES THUS

 TEST PHASE INFORMATION

TEST PHASE DAYS BETWEEN MISSIONS
 FOR EACH TEST.PHASE
 READ TOBM(TEST.PHASE)
 FOR EVERY TEST.PHASE PRINT 1 LINE WITH TEST.PHASE AND
 TOBM(TEST.PHASE) THUS

 PRINT 8 LINES THUS

 #####
 #####
 END OF INPUT SUMMARY
 #####
 #####

SKIP 10 LINES
 READ CHECK
 HERE

00005580
 00005590
 00005600
 00005610
 00005620
 00005630
 00005640
 00005650
 00005660
 00005670
 00005680
 00005690
 00005700
 00005710
 00005720
 00005730
 00005740
 00005750
 00005760
 00005770
 00005780
 00005790
 00005800
 00005810
 00005820
 00005830
 00005840
 00005850
 00005860
 00005870
 00005880
 00005890
 00005900
 00005910
 00005920

```

..
ROUTINE TO CHECK.QUEUE

CALL MY.TRACE(1)
DEFINE T AS A REAL VARIABLE
FOR EVERY MZJ IN QUEUE WITH MMAP(MZJ) = 1 AND
    MTOA(MZJ) + VRMPT GT TIME.V DO
    REMOVE MZJ FROM QUEUE
    FILE MZJ IN WAIT.FOR.MISSION.TIME
    SCHEDULE AN ACCOMPLISH.PRI.MISSION AT MTOA(MZJ) + VRMPT
LOOP
IF FRAC.F(TIME.V) GE VEDDT/24 OR FRAC.F(TIME.V) LT VBDDT/24
    GO TO OVERTIME
ELSE
IF WEEKDAY.F(TIME.V) GT VNMDD
    *OVERTIME*
    CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS
    RETURN
ELSE
** IF CHECKED DURING BREAK, PRI MISSIONS SHOULD BE INITIATED
LET T = FRAC.F(TIME.V) * 24
IF VBB1T LT T LE VEB1T OR
    VBLB1T LT T LE VELB1T OR
    VBB2T LT T LE VEB2T
    FOR EVERY MZJ IN QUEUE WITH MMAP(MZJ) = 1 AND
        ANWR(F.ROUTING(MZJ)) LE SUM.WORKERS(2,MZJ,0) DO
        REMOVE MZJ FROM QUEUE
        CALL MY.TRACE(2,MZJ)
        CALL ATTEND.TO.MISSION GIVING MZJ
LOOP
ALWAYS
FOR EVERY DMZ IN QUEUE WITH ANWR(F.ROUTING(DMZ)) LE
    SUM.WORKERS(1,DMZ,0) DO
    REMOVE DMZ FROM QUEUE
    CALL ATTEND.TO.MISSION GIVING DMZ
LOOP
    
```

00005420
 00005430
 00005440
 00005450
 00005460
 00005470
 00005480
 00005490
 00005500
 00005510
 00005520
 00005530
 00005540
 00005550
 00005560
 00005570

00000000
 00000000
 00000000
 00000000

```

..
EVENT BEGIN.WORKING SAVING THE EVENT NOTICE
SCHEDULE THIS BEGIN.WORKING IN 24 HOURS
IF WEEKDAY.F(TIME.V) GT VNMDD
  RETURN **OFFICE IS CLOSED FOR THE WEEKEND
ELSE
FOR EVERY WORKER CALLED WZU WITH WSTA(WZU) = BREAK OR
  WSTA(WZU) = OFF.DUTY DO
  LET WSTA(WZU) = IDLE
  ADD 1 TO LNIH(WLEV(WZU))
LOOP
FOR EVERY WORKER CALLED WZV WITH WSTA(WZV) = OVERTIME
  LET WSTA(WZV) = BUSY
  CALL CHECK.QUEUE
RETURN
END

```

END
 RE 1724
 EVER 0010 0011 0012 0013

00005960
 00005970
 00005980
 00005990
 00006000
 00006010
 00006020
 00006030
 00006040
 00006050
 00006060
 00006070
 00006080
 00006090
 00006100
 00006110
 00006120
 00006130
 00006140
 00006150
 00006160
 00006170
 00006180
 00006190
 00006200
 00006210
 00006220
 00006230
 00006240
 00006250
 00006260
 00006270
 00006280
 00006290
 00006300

```

.. EVENT CONFIRMED.MISSION GIVEN GTZ SAVING THE EVENT NOTICE
   DEFINE T AS A REAL VARIABLE
   CALL DISBAND.PARTIAL.TEAMS
   IF EVENT IS EXTERNAL
     READ GTZ
     GO TO EXTERNAL
   ELSE
     IF WEEKDAY.F(TIME.V) LE VWMDD
       GO TO WEEKDAY
     ELSE
       SCHEDULE THIS CONFIRMED.MISSION(GTZ) AT TIME.V + 1
       ..NOTE INTERNAL MISSIONS MAY BE SCHEDULED FOR NON DUTY HOURS BUT
       .. ONLY EXTERNAL MISSIONS CAN BE SCHEDULED ON NON DUTY DAYS.
   RETURN
   *WEEKDAY*
   *EXTERNAL*
   ADD 1 TO TCNF(GTZ)
   CREATE A MISSION CALLED GMZ
   LET MTPN(GMZ) = GTZ
   LET MCEL(GMZ) = 1
   LET MTOA(GMZ) = TIME.V
   LET MXCI(GMZ) = 0
   IF EVENT IS INTERNAL
     FOR EACH AZK OF STRUCTURE(GTZ) DO
       CREATE AN ACTIVITY CALLED GAZ
       LET ATPC(GAZ) = ATPC(AZK)
       LET ANOA(GAZ) = ANOA(AZK)
       LET APOA(GAZ) = APOA(AZK)
       LET ATFC(GAZ) = ATFC(AZK)
       LET AICT(GAZ) = AICT(AZK)
       LET ANHR(GAZ) = ANHR(AZK)
       LET APAR(GAZ) = APAR(AZK)
       LET AFHL(GAZ) = AFHL(AZK)
       LET ASWL(GAZ) = ASWL(AZK)

```

00006310
 00006320
 00006330
 00006340
 00006350
 00006360
 00006370
 00006380
 00006390
 00006400
 00006410
 00006420
 00006430
 00006440
 00006450
 00006460
 00006470
 00006480
 00006490
 00006500
 00006510
 00006520
 00006530
 00006540
 00006550
 00006560
 00006570
 00006580
 00006590
 00006600
 00006610
 00006620
 00006630
 00006640
 00006650

```

LET ATWL(GAZ) = ATWL(AZK)
LET ACEL(GAZ) = ACEL(AZK)
LET APMA(GAZ) = APMA(AZK)
LET ALRL(GAZ) = ALRL(AZK)
LET AURL(GAZ) = AURL(AZK)
FILE GAZ IN ROUTING(GMZ)
LOOP
GO TO CNX.PROB
ELSE
**PERFORM A SPECIAL MISSION
UNTIL MODE IS ALPHA DD
READ N
FOR EACH AZK IN STRUCTURE(GTZ) WITH ATPC(AZK) = N
FIND THE FIRST CASE
IF NONE
PRINT 3 LINES WITH N AND GTZ THUS
WARNING ACTIVITY CODE *** IS NOT PART OF THE SEQUENCE OF
TEST PHASE ***, VERIFY THE INPUT
CREATE AN ACTIVITY CALLED GAZ
READ ANOA(GAZ), APOA(GAZ), ANMR(GAZ),
AFWL(GAZ), ASHL(GAZ), ATWL(GAZ)
LET ATPC(GAZ) = N
LET ATFC(GAZ) = KTFC(ANOA(GAZ))
LET AICT(GAZ) = 0
LET APAR(GAZ) = KPAR(ANOA(GAZ))
LET ACEL(GAZ) = 1
LET APMA(GAZ) = KPMA(ANOA(GAZ))
LET ALRL(GAZ) = KLRL(ANOA(GAZ))
LET AURL(GAZ) = KURL(ANOA(GAZ))
FILE GAZ IN ROUTING(GMZ)
CALL MY.LIST(5,GMZ)**ACTIVITIES IN ROUTING(GMZ)
GO TO LOOP
ELSE
CREATE AN ACTIVITY CALLED GAZ
    
```

AD-A035 753

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/6 5/1
JOINT TEST FORCE SIMULATION.(U)

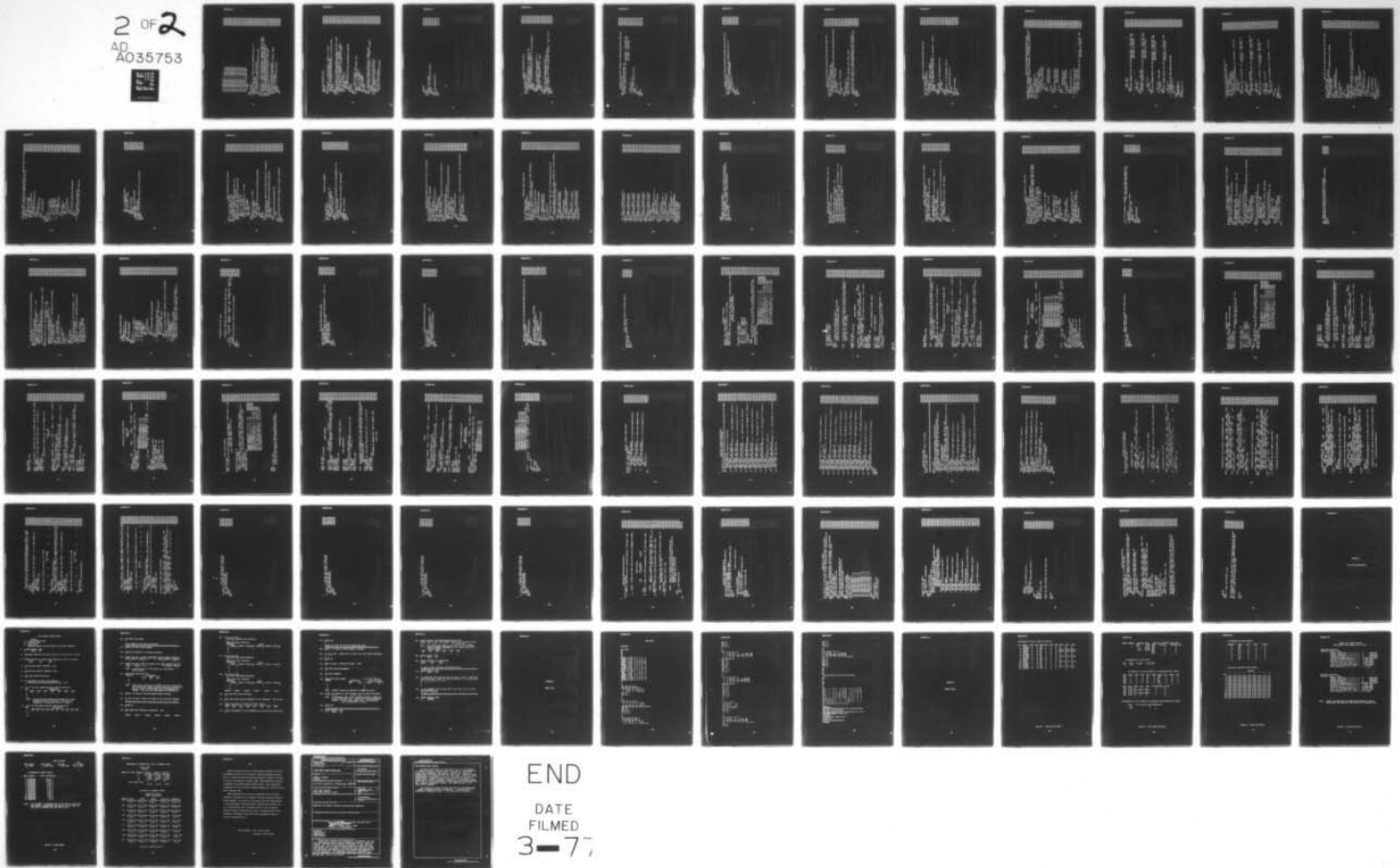
DEC 76 J P EGGERS

60R/SM/76D-7

UNCLASSIFIED

NL

2 OF 2
AD
A035753



00006660
 00006670
 00006680
 00006690
 00006700
 00006710
 00006720
 00006730
 00006740
 00006750
 00006760
 00006770
 00006780
 00006790
 00006800
 00006810
 00006820
 00006830
 00006840
 00006850
 00006860
 00006870
 00006880
 00006890
 00006900
 00006910
 00006920
 00006930
 00006940
 00006950
 00006960
 00006970
 00006980
 00006990
 00007000

```

LET ATPC(GAZ) = N
LET ANOA(GAZ) = ANOA(AZK)
LET APOA(GAZ) = APOA(AZK)
LET ATFC(GAZ) = ATFC(AZK)
LET AICT(GAZ) = AICT(AZK)
LET ANMR(GAZ) = ANMR(AZK)
LET APAR(GAZ) = APAR(AZK)
LET AFWL(GAZ) = AFWL(AZK)
LET ASWL(GAZ) = ASWL(AZK)
LET ATWL(GAZ) = ATWL(AZK)
LET ACEL(GAZ) = ACEL(AZK)
LET APHA(GAZ) = APHA(AZK)
LET ALRL(GAZ) = ALRL(AZK)
LET AUPL(GAZ) = AUPL(AZK)
FILE GAZ IN ROUTING(GMZ)
    *LOOP*
LOOP
    *CNX.PROB*
FOR EACH FACTOR OF TCANCELLATION.PROBABILITY(GTZ) DO
    CREATE AN XFACTOR
    LET XUPL(XFACTOR) = XUPL(FACTOR)
    FILE XFACTOR IN MCANCELLATION.PROBABILITY(GMZ)
LOOP
LET MMAP(GMZ) = APOA(F.ROUTING(GMZ))*RANKING CODE IN QUEUE
LET MNAC(GMZ) = ATPC(F.ROUTING(GMZ))*RANKING CODE IN QUEUE
CALL VALIDATE.NUMBER.OF.WORKERS.REQUESTED GIVING GMZ
CALL DETERMINE.PREMISSION.CANCELLATION GIVING GMZ
CALL CALCULATE.MISSION.EFFECTIVENESS GIVING GMZ
IF MXCI(GMZ) = 1
    DESTROY THIS MISSION CALLED GMZ
    CALL BUILD.PARTIAL.TEAMS**REBUILD PARTIAL TEAMS
    IF EVENT IS EXTERNAL
        DESTROY THIS CONFIRMED.MISSION
    RETURN
ELSE

```

00007010
 00007020
 00007030
 00007040
 00007050
 00007060
 00007070
 00007080
 00007090
 00007100
 00007110
 00007120
 00007130
 00007140
 00007150
 00007160
 00007170
 00007180
 00007190
 00007200
 00007210
 00007220
 00007230
 00007240
 00007250
 00007260
 00007270
 00007280
 00007290
 00007300
 00007310
 00007320
 00007330
 00007340
 00007350

```

CALL MSN.SCHEDULE(GTZ)
RETURN
ELSE
IF MNAPI(GMZ) = 1 AND MTOA(GMZ) + VRMPT GT TIME.V
FILE GMZ IN WAIT.FOR.MISSION.TIME
SCHEDULE AN ACCOMPLISH.PRI.MISSION AT MTOA(GMZ)+VRMPT
GO COUNT
ELSE
**PERFORM PRIORITY MISSIONS SCHEDULED DURING NON DUTY HOURS
IF WEEKDAY.F(TIME.V) GT VHMDD OR **WEEKEND
FRAC.F(TIME.V) GE VEDDT/24 OR **AFTER DUTY HOURS
FRAC.F(TIME.V) LT VBDDT/24 **BEFORE DUTY HOURS
FILE GMZ IN QUEUE
IF MNAPI(GMZ) = 1 OR
MNAPI(GMZ) = 2
CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS
ALWAYS
GO COUNT
ELSE
LET T = FRAC.F(TIME.V) * 24
IF VBB1T LT T LE VEB1T OR **BREAK TIME
VBLBT LT T LE VELBT OR
VBB2T LT T LE VEB2T
FILE GMZ IN QUEUE
CALL CHECK.QUEUE
GO COUNT
ELSE
CALL ATTEND.TO.MISSION GIVING GMZ **NORMAL DUTY
CALL CHECK.QUEUE**REASSIGN WORKERS IF PREEMPT OCCURRED
*COUNT*
IF TCHP(GTZ)+LACH(GTZ) GE TREQ(GTZ)
**CURRENT MISSION IS COUNTED IN LACH
ADD 1 TO GTZ
IF GTZ GT N.TEST.PHASE
DESTROY THIS CONFIRMED.MISSION
    
```


00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
00007580
00007590
00007600
00007610
00007620
00007630
00007640

```

..
    ROUTINE MSN.SCHEDULE(GTZ)
    CALL MY.TRACE(19,GTZ)
    IF TOBM(GTZ) LT 1.0
    IF FRAC.F(TIME.V+TOBM(GTZ)) GE VBDDT/24 AND
    FRAC.F(TIME.V+TOBM(GTZ)) LT VEDDT/24
    SCHEDULE THE CONFIRMED.MISSION(GTZ) AT TIME.V+TOBM(GTZ)
    RETURN
    ELSE
    SCHEDULE THE CONFIRMED.MISSION(GTZ) AT
    TRUNC.F(TIME.V+1) + VBDDT/24
    .. TOMMORROW + BEGINNING OF DUTY HOURS
    RETURN
    ELSE
    SCHEDULE THE CONFIRMED.MISSION(GTZ) AT
    TRUNC.F(TIME.V) + TOBM(GTZ) + TTMC(GTZ)/24
    .. BEGIN OF CURRENT DAY + DAYS BEFORE NEXT MISSION + TIME OF DAY
    RETURN
    END

```

00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
00007580
00007590
00007600
00007610
00007620
00007630
00007640

MSN.SCHEDULE(GTZ)
MY.TRACE(19,GTZ)
TOBM(GTZ)
FRAC.F(TIME.V+TOBM(GTZ))
VBDDT/24
VEDDT/24
TIME.V+TOBM(GTZ)
CONFIRMED.MISSION(GTZ)
TRUNC.F(TIME.V+1)
TTMC(GTZ)
CURRENT DAY
DAYS BEFORE NEXT MISSION
TIME OF DAY

00007650
 00007660
 00007670
 00007680
 00007690
 00007700
 00007710
 00007720
 00007730
 00007740
 00007750
 00007760
 00007770
 00007780
 00007790

00007650
 00007660
 00007670
 00007680
 00007690
 00007700
 00007710
 00007720
 00007730
 00007740
 00007750
 00007760
 00007770
 00007780
 00007790

```

ROUTINE TO VALIDATE NUMBER OF WORKERS REQUESTED GIVEN RMZ
CALL MYTRACE(8,RMZ)
FOR EVERY RAZ IN ROUTING(RMZ) DO
  FOR EVERY WORKER CALLED RMZ WITH
    WLEV(RMZ) = AFHL(RAZ) OR
    WLEV(RMZ) = ASHL(RAZ) OR
    WLEV(RMZ) = ATML(RAZ)
    ADD 1 TO I
  IF ANWR(RAZ) GT I
    CALL ERROR.PACKAGE(4,RMZ)
  ALWAYS
  LET I = 0
LOOP
RETURN
END

```

ENG
 MICH
 TUDS
 VERBAZ
 PUA MODILAS1 = J
 DE PONDUM EAT DE XAPITAL YOLUM
 FOR EVERY WORKER CALLED RMZ WITH
 WLEV(RMZ) = AFHL(RAZ) OR
 WLEV(RMZ) = ASHL(RAZ) OR
 WLEV(RMZ) = ATML(RAZ)
 ROUTINE TO VALIDATE NUMBER OF WORKERS REQUESTED GIVEN RMZ

00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090

```
..
ROUTINE TO DETERMINE.PREMISSION.CANCELLATION GIVEN FMZ
CALL MY.TRACE(3,FMZ)
FOR EVERY XFACTOR IN MCANCELLATION.PROBABILITY(FMZ) DO
IF RANDOM.F(7) LE XUPL(XFACTOR)
LET MXCI(FMZ) = 1
ALWAYS
LOOP
RETURN
END
```

00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090

```
END  
BEGIN  
F000  
LET I = 0  
FMZ = 1  
CALL TRACE(3,FMZ)  
DO EVERY XFACTOR IN MCANCELLATION.PROBABILITY(FMZ)  
IF RANDOM.F(7) LE XUPL(XFACTOR)  
LET MXCI(FMZ) = 1  
ALWAYS  
LOOP  
RETURN  
END
```

00007900
00007910
00007920
00007930
00007940
00007950
00007960
00007970
00007980
00007990
00008000
00008010
00008020
00008030
00008040
00008050
00008060
00008070
00008080
00008090
00008100

```

..
ROUTINE TO CALCULATE MISSION EFFECTIVENESS GIVEN HMZ
CALL MYTRACE(4, HMZ)
FOR EACH HAZ IN ROUTING(HMZ) DO
  IF RANDOM.F(7) LE ALPHA(HAZ)/100
    LET ACEL(HAZ) = MCEL(HMZ) - UNIFORM.F(ALRL(HAZ)/100, AURL(HAZ)/100, 8)
    LET MCEL(HMZ) = ACEL(HAZ)
    JUMP AHEAD
  ELSE
    LET ACEL(HAZ) = MCEL(HMZ)
  HERE
LOOP
IF MCEL(HMZ) LE 0
  LET MXCI(HMZ) = 1
  PRINT 3 LINES WITH MTPN(HMZ) AND TIME.V THUS
  A MISSION IN TEST PHASE *** WAS ELIMINATED AT ****.*** DAYS
  BECAUSE THE TOTAL MISSION EFFECTIVENESS REDUCTIONS EXCEEDED 100%
  ALWAYS
  RETURN
END

```

00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```

..
ROUTINE TO ATTEND. TO MISSION GIVEN IMZ
CALL MY.TRACE(5,IMZ)
LET IAZ = F.ROUTING(IMZ)
IF MMAP(IMZ) = 1
  IF MTOA(IMZ) + VRMPT GT TIME.V
    FILE IMZ IN WAIT.FOR.MISSION.TIME
    SCHEDULE AN ACCOMPLISH.PRI.MISSION AT MTOA(IMZ) + VRMPT
    RETURN
  ELSE
    **BREAK TIME
    IF ANWR(IAZ) LE SUM.WORKERS(2,IMZ,0)
      JUMP AHEAD
    ELSE
      ALWAYS
    IF ANWR(IAZ) LE SUM.WORKERS(1,IMZ,0)
      HERE
      CALL ALLOCATION GIVING IMZ
      RETURN
    ELSE
      FILE IMZ IN QUEUE
      RETURN
    END

```

00008340
 00008350
 00008360
 00008370
 00008380
 00008390
 00008400
 00008410
 00008420
 00008430
 00008440
 00008450
 00008460
 00008470
 00008480
 00008490
 00008500
 00008510
 00008520
 00008530
 00008540
 00008550
 00008560
 00008570
 00008580
 00008590
 00008600
 00008610
 00008620
 00008630
 00008640
 00008650
 00008660
 00008670
 00008680

```

ROUTINE SUM.WORKERS (NUMB, JMZ, PART, TEAM)
**CONSIDER ALL COMBINATIONS OF WORKER LEVEL PREFERENCES IN DETERMINING
**THE NUMBER OF WORKERS CURRENTLY AVAILABLE TO PERFORM AN ACTIVITY.
CALL MY.TRACE(6, JMZ)
LET STATUS = 1
LET JAZ = F.ROUTING(JMZ)
GO TO 1,2,3 PER NUMB
*1. **WORKERS WHO ARE IDLE
  IF AFWL(JAZ) = ASWL(JAZ)
  IF ASWL(JAZ) = ATWL(JAZ)
  LET X = LNIH(AFWL(JAZ))
  GO PR1
  ELSE
  LET X = LNIH(AFWL(JAZ)) +
  LNIH(ATWL(JAZ))
  GO PR1
  ELSE
  IF ASWL(JAZ) = ATWL(JAZ)
  LET X = LNIH(AFWL(JAZ)) +
  LNIH(ASWL(JAZ))
  GO PR1
  ELSE
  LET X = LNIH(ASWL(JAZ)) +
  LNIH(ATWL(JAZ))
  IF AFWL(JAZ) = ATWL(JAZ)
  GO PR1
  ELSE
  ADD LNIH(AFWL(JAZ)) TO X
  GO PR1
*2. **WORKERS WHO ARE NOT BUSY
  IF AFWL(JAZ) = ASWL(JAZ)
  IF ASWL(JAZ) = ATWL(JAZ)
  FOR EVERY WORKER CALLED JMZ WITH MLEV(JMZ) = AFWL(JAZ) AND
  HSTA(JMZ) NE STATUS
    
```

00008690
 00008700
 00008710
 00008720
 00008730
 00008740
 00008750
 00008760
 00008770
 00008780
 00008790
 00008800
 00008810
 00008820
 00008830
 00008840
 00008850
 00008860
 00008870
 00008880
 00008890
 00008900
 00008910
 00008920
 00008930
 00008940
 00008950
 00008960
 00008970
 00008980
 00008990
 00009000
 00009010
 00009020
 00009030

```

ADD 1 TO X
GO TO PR1
ELSE
FOR EVERY WORKER CALLED JWZ WITH WLEV(JMZ) = AFHL(JAZ) OR
WLEV(JMZ) = ATML(JAZ) AND
WSTA(JMZ) NE STATUS
    ADD 1 TO X
GO TO PR1
ELSE
IF ASHL(JAZ) = ATML(JAZ)
FOR EVERY WORKER CALLED JWZ WITH WLEV(JMZ) = AFHL(JAZ) OR
WLEV(JMZ) = ASHL(JAZ) AND
WSTA(JMZ) NE STATUS
    ADD 1 TO X
GO TO PR1
ELSE
FOR EVERY WORKER CALLED JWZ WITH WLEV(JMZ) = ASHL(JAZ) OR
WSTA(JMZ) NE STATUS
    ADD 1 TO X
GO TO PR1
ELSE
FOR EVERY WORKER CALLED JWZ WITH WLEV(JMZ) = AFHL(JAZ) AND
WSTA(JMZ) NE STATUS
    ADD 1 TO X
GO TO PR1
ELSE
FOR EVERY WORKER CALLED JWZ WITH WLEV(JMZ) = AFHL(JAZ) AND
WSTA(JMZ) NE STATUS
    ADD 1 TO X
GO TO PR1

```

3 WORKERS WHO ARE NOT WORKING OVERTIME
 LET STATUS = 4
 GO TO 2
 *PR1.
 IF PART.TEAM = 1
 RETURN WITH X
 ELSE
 IF APOA(JAZ) = 1


```

00009380
00009390
00009400
00009410
00009420
00009430
00009440
00009450
00009460
00009470
00009480
00009490
00009500
00009510
00009520
00009530
00009540
00009550
00009560
00009570
00009580
00009590
00009600
00009610
00009620
00009630
00009640
00009650
00009660
00009670
00009680
00009690
00009700
00009710
00009720

..
ROUTINE FOR ALLOCATION GIVEN PMZ AND PART.TEAM
CALL MY.TRACE(7,PMZ)
DEFINE VAL AS A REAL VARIABLE
IF ANWR(F.ROUTING(PMZ)) = 0 ..MANPOWER REQ IS EXTERNAL
  GO TO WORKER.FOUND
ELSE
  LET PAZ = F.ROUTING(PMZ)
  *PREEMPT.COMPLETED.LOOK.AGAIN*
  LET J=0
  LET I = AFWL(PAZ)
  *LOOK.AGAIN*
  IF I=0
    GO TO U
  ELSE
    *MORE.REQUIRED*
    LET PMZ = 0
    FOR EVERY WORKER CALLED WZI WITH WLEV(WZI) = I AND WSTA(WZI) = IDLE
      COMPUTE PMZ = MIN(WZI) OF TOBSERVED.STATE(WZI,2)
      IF PMZ NE 0
        FILE PMZ IN TEAM(PAZ)
        LET WSTA(PMZ) = BUSY
        SUBTRACT 1 FROM LNIH(WLEV(PMZ))
        SUBTRACT 1 FROM ANWR(PAZ)
        IF ANWR(PAZ) LT 0
          CALL ERROR.PACKAGE(2,PMZ)
          ALWAYS
        IF ANWR(PAZ) = 0
          GO TO WORKER.FOUND
        ELSE
          GO TO MORE.REQUIRED
        ALWAYS
    ..BREAK TIME ACTIVITIES
    IF APOA(PMZ) = 1
      LET PMZ = 0

```

```

FOR EVERY WORKER CALLED WZI WITH WLEV(WZI) = I AND WSTA(WZI) NE BUSY000009730
  COMPUTE PWZ = MIN(WZI) OF TOBSERVED.STATE(WZI,2) 00009740
  IF PWZ NE 0 00009750
    FILE PWZ IN TEAM(PAZ) 00009760
    LET WSTA(PWZ) = BUSY 00009770
    SUBTRACT 1 FROM ANWR(PAZ) 00009780
    IF ANWR(PAZ) LT 0 00009790
      CALL ERROR.PACKAGE(2,PWZ) 00009800
      ALWAYS 00009810
      IF ANWR(PAZ) = 0 00009820
        GO TO WORKER.FOUND 00009830
      ELSE 00009840
        GO TO MORE.REQUIRED 00009850
      ELSE 00009860
        ALWAYS 00009870
        ADD 1 TO J 00009880
        GO TO H, X, Y, V PER J 00009890
        *W LET I = ASWL(PAZ) 00009900
          GO TO LOOK.AGAIN 00009910
        *X LET I = ATWL(PAZ) 00009920
          GO TO LOOK.AGAIN 00009930
        *Y IF PART.TEAM = 1 00009940
          GO TO WORKER.FOUND 00009950
        ELSE 00009960
          IF HMAP(PWZ) = 1 00009970
            CALL PREEMPT 00009980
            ADD 1 TO L 00009990
            IF L GT N.WORKER 00010000
              GO TO V 00010010
            ELSE 00010020
              GO TO PREEMPT.COMPLETED.LOOK.AGAIN 00010030
            ELSE 00010040
              GO TO V 00010050
          *V UNTIL TEAM(PAZ) IS EMPTY DO 00010060
            REMOVE THE FIRST PWZ FROM TEAM(PAZ) 00010070

```

```

00010060
00010090
00010100
00010110
00010120
00010130
00010140
00010150
00010160
00010170
00010180
00010190

```

```

ADD 1 TO LNIH(WLEV(PHZ))
ADD 1 TO ANHR(PAZ)
LET NSTA(PHZ) = IOLE
LOOP
FILE PHZ IN QUEUE
RETURN
*U*CALL ERROR.PACKAGE(3,PHZ)
*WORKER.FOUND*
LET VAL = VALUE(PHZ)
SCHEDULE AN EO.ACTIVITY GIVING PHZ IN VAL MINUTES
RETURN
END

```

00010200
 00010210
 00010220
 00010230
 00010240
 00010250
 00010260
 00010270
 00010280
 00010290
 00010300
 00010310
 00010320
 00010330
 00010340
 00010350
 00010360
 00010370
 00010380
 00010390
 00010400
 00010410
 00010420
 00010430
 00010440
 00010450
 00010460
 00010470
 00010480
 00010490
 00010500
 00010510
 00010520
 00010530
 00010540

```

..
      EVENT ACCOMPLISH.PRI.MISSION
      DEFINE T AS A REAL VARIABLE
      LET T = FRAC.F(TIME.V) * 24
      CALL DISBAND.PARTIAL.TEAMS
      FOR EVERY SMZ IN WAIT.FOR.MISSION.TIME WITH
      MTOA(SMZ) + VRMPT LE TIME.V DO
      REMOVE THIS SMZ FROM WAIT.FOR.MISSION.TIME
      IF VBB1T LT T LE VEB1T OR
      VBLBT LT T LE VELBT OR
      VBB2T LT T LE VEB2T
      THEN IF WEEKDAY.F(TIME.V) LE VMWDD
      FILE SMZ IN QUEUE
      CALL CHECK.QUEUE
      JUMP AHEAD
    ELSE
      IF WEEKDAY.F(TIME.V) GT VMWDD
      FILE SMZ IN QUEUE
      IF VWEAM = "YES"
      CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS
      JUMP AHEAD
    ELSE
      RETURN
    ELSE
      IF FRAC.F(TIME.V) GE VEDDT/24 OR FRAC.F(TIME.V) LT VBDDT/24
      FILE SMZ IN QUEUE
      CALL OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS
      JUMP AHEAD
    ELSE
      CALL ATTEND.TO.MISSION GIVING SMZ
      HERE
    LOOP
    CALL CHECK.QUEUE**REASSIGN WORKERS IF PREEMPT OCCURRED
    RETURN
  END

```

00010550
00010560
00010570
00010580
00010590
00010600
00010610
00010620
00010630
00010640
00010650
00010660
00010670
00010680
00010690
00010700

ROUTINE TO PREEMPT

```

..
CALL MY.TRACE(9)
LET REASON = IDLE
IF FRAC.F(TIME.V) GE VEEDT/24 OR
FRAC.F(TIME.V) LT VBODT/24
LET REASON = OFF.DUTY
ALWAYS
FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH MNAP(EAMN(K)) NE 1
FIND THE FIRST CASE
IF FOUND
CALL INTERRUPT.AND.FILE GIVING K AND REASON
ALWAYS
..PREEMPT ONLY 1 MISSION
RETURN
END

```

00010710
 00010720
 00010730
 00010740
 00010750
 00010760
 00010770
 00010780
 00010790
 00010800
 00010810
 00010820
 00010830
 00010840
 00010850
 00010860
 00010870
 00010880
 00010890
 00010900
 00010910
 00010920
 00010930
 00010940
 00010950
 00010960

```

ROUTINE TO INTERRUPT.AND.FILE GIVEN K AND REASON
CALL MY.TRACE(10,EAMN(K))
DEFINE AV.FACTOR AS A REAL VARIABLE
LET AV.FACTOR = 1
LET PART.TEAM.FACTOR = 1
REMOVE K FROM EV.S(I.EO.ACTIVITY)
LET UMZ = EAMN(K)
LET UAZ = F.ROUTING(UMZ)
IF TEAM(UAZ) IS NOT EMPTY
LET PART.TEAM.FACTOR = (ANMR(UAZ)+N.TEAM(UAZ))/N.TEAM(UAZ)
ALWAYS
UNTIL TEAM(UAZ) IS EMPTY DO
REMOVE THE FIRST UMZ FROM TEAM(UAZ)
COMPUTE AV.FACTOR = AVG OF SKILL.FACTOR(UMZ,ANO(A(UAZ))
ADD 1 TO ANMR(UAZ)
LET WSTA(UMZ) = REASON
IF WSTA(UMZ) = IDLE
ADD 1 TO LNIW(WLEV(UMZ))
ALWAYS
LOOP
LET AICT(UAZ) = (TIME.A(K) - TIME.V)/(AV.FACTOR + PART.TEAM.FACTOR)
FILE UMZ IN QUEUE
DESTROY EO.ACTIVITY CALLED K
RETURN
END
    
```

00010970
 00010980
 00010990
 00011000
 00011010
 00011020
 00011030
 00011040
 00011050
 00011060
 00011070
 00011080
 00011090
 00011100
 00011110
 00011120
 00011130
 00011140
 00011150
 00011160
 00011170
 00011180
 00011190
 00011200
 00011210
 00011220
 00011230
 00011240
 00011250
 00011260
 00011270
 00011280
 00011290
 00011300
 00011310

```

..
ROUTINE VALUE GIVEN YMZ
CALL MY.TRACE(11, YMZ)
LET YAZ = F.ROUTING(YMZ)
LET YTZ = MTPN(YMZ)
DEFINE A, B, SAMPLE, AV.FACTOR, AND VAL AS REAL VARIABLES
DEFINE C, D, AND E AS INTEGER VARIABLES
IF AICT(YAZ) GT 0
    LET SAMPLE = AICT(YAZ)*24*60
    LET AICT(YAZ) = 0
    GO TO PROFICIENCY.OF.WORKER
ELSE
    LET A = 1ARG(YTZ, ANOA(YAZ))
    LET B = 2ARG(YTZ, ANOA(YAZ))
    LET C = 10
    LET E = ATFC(YAZ)
    IF E LT 0
        GO TO INTEGER-STEP, REAL-STEP OR CONTINUOUS PER ABS.F(E)
    ELSE
        GO TO BETA, BINOMIAL, CONSTANT, ERLANG, EXPONENTIAL, GAMMA, NORMAL,
        POISSON, UNIFORM OR WEIBULL PER E
    *BETA*
    LET SAMPLE = BETA.F(A, B, C)
    GO TO PROFICIENCY.OF.WORKER
    *BINOMIAL*
    LET D = TRUNC.F(A)
    LET SAMPLE = BINOMIAL.F(D, B, C)
    GO TO PROFICIENCY.OF.WORKER
    *CONSTANT*
    LET SAMPLE = A
    GO TO PROFICIENCY.OF.WORKER
    *ERLANG*
    LET SAMPLE = ERLANG.F(A, B, C)
    GO TO PROFICIENCY.OF.WORKER
    *EXPONENTIAL*

```

00011320
 00011330
 00011340
 00011350
 00011360
 00011370
 00011380
 00011390
 00011400
 00011410
 00011420
 00011430
 00011440
 00011450
 00011460
 00011470
 00011480
 00011490
 00011500
 00011510
 00011520
 00011530
 00011540
 00011550
 00011560
 00011570
 00011580
 00011590
 00011600
 00011610
 00011620
 00011630
 00011640
 00011650
 00011660

LET SAMPLE = EXPONENTIAL.F(A,C)
 GO TO PROFICIENCY.OF.WORKER
 •GAMMA.
 LET SAMPLE = GAMMA.F(A,B,C)
 GO TO PROFICIENCY.OF.WORKER
 •NORMAL.
 LET SAMPLE = NORMAL.F(A,R,C)
 GO TO PROFICIENCY.OF.WORKER
 •POISSON.
 LET SAMPLE = POISSON.F(A,C)
 GO TO PROFICIENCY.OF.WORKER
 •UNIFORM.
 LET SAMPLE = UNIFORM.F(A,B,C)
 GO TO PROFICIENCY.OF.WORKER
 •WEIBULL.
 LET SAMPLE = WEIBULL.F(A,B,C)
 GO TO PROFICIENCY.OF.WORKER
 •INTEGER.STEP.
 LET D = TRUNC.F(B)
 ••FOR INTEGER.STEP.FUNCTION = D
 LET SAMPLE = ONE(D)
 GO TO PROFICIENCY.OF.WORKER
 •REAL.STEP.
 LET D = TRUNC.F(B)
 ••FOR REEL.STEP.FUNCTION = D
 LET SAMPLE = TWO(D)
 GO TO PROFICIENCY.OF.WORKER
 •CONTINUOUS.
 LET D = TRUNC.F(B)
 ••FOR CONTINUOUS.FUNCTION = D
 LET SAMPLE = THREE(D)
 GO TO PROFICIENCY.OF.WORKER
 •PROFICIENCY.OF.WORKER.
 IF TEAM(YAZ) IS EMPTY
 RETURN WITH SAMPLE

```

ELSE
FOR EVERY YMZ IN TEAM(YAZ)
  COMPUTE AV.FACTOR = AVG OF SKILL.FACTOR(YHZ,ANOAC(YAZ))
  LET PART.TEAM.FACTOR = (ANMR(YAZ)+N.TEAM(YAZ))/N.TEAM(YAZ)
  LET VAL = SAMPLE * AV.FACTOR + PART.TEAM.FACTOR
  RETURN WITH VAL
END

```

```

00011670
00011680
00011690
00011700
00011710
00011720
00011730

```

```

00011740
00011750
00011760
00011770
00011780
00011790
00011800
00011810
00011820
00011830
00011840
00011850
00011860
00011870
00011880
00011890
00011900
00011910
00011920
00011930
00011940
00011950
00011960
00011970
00011980
00011990

```


00011670
00011680
00011690
00011900
00011910
00011920
00011930
00011940
00011950
00011960
00011970
00011980
00011990
00012000
00012010
00012020
00012030

```
..
EVENT TAKE.BREAK SAVING THE EVENT NOTICE
SCHEDULE THIS TAKE.BREAK IN 24 HOURS
IF WEEKDAY.F(TIME.V) GT VNMDD
  RETURN ..OFFICE IS CLOSED FOR THE WEEKEND
ELSE
  CALL DISBAND.PARTIAL.TEAMS
  FOR EVERY WORKER CALLED WZL WITH WSTA(WZL) = IDLE DO
    LET WSTA(WZL) = BREAK
    SUBTRACT 1 FROM LNIN(WLEV(WZL))
  LOOP
  FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH MMAP(EAMN(K)) NE 1 DO
    LET REASON = BREAK
    CALL INTERRUPT.AND.FILE GIVING K AND REASON
  LOOP
RETURN
END
```

00011670
00011680
00011690
00011900
00011910
00011920
00011930
00011940
00011950
00011960
00011970
00011980
00011990
00012000
00012010
00012020
00012030

..
EVENT TAKE.BREAK SAVING THE EVENT NOTICE
SCHEDULE THIS TAKE.BREAK IN 24 HOURS
IF WEEKDAY.F(TIME.V) GT VNMDD
 RETURN ..OFFICE IS CLOSED FOR THE WEEKEND
ELSE
 CALL DISBAND.PARTIAL.TEAMS
 FOR EVERY WORKER CALLED WZL WITH WSTA(WZL) = IDLE DO
 LET WSTA(WZL) = BREAK
 SUBTRACT 1 FROM LNIN(WLEV(WZL))
 LOOP
 FOR EVERY K IN EV.S(I.EO.ACTIVITY) WITH MMAP(EAMN(K)) NE 1 DO
 LET REASON = BREAK
 CALL INTERRUPT.AND.FILE GIVING K AND REASON
 LOOP
RETURN
END

00012040
 00012050
 00012060
 00012070
 00012080
 00012090
 00012100
 00012110
 00012120
 00012130
 00012140
 00012150
 00012160
 00012170
 00012180
 00012190
 00012200
 00012210
 00012220
 00012230
 00012240
 00012250
 00012260
 00012270
 00012280
 00012290
 00012300
 00012310
 00012320
 00012330
 00012340
 00012350
 00012360
 00012370
 00012380

```

EVENT EO.ACTIVITY GIVEN MZB
DEFINE T AS A REAL VARIABLE
CALL DISBAND.PARTIAL.TEAMS
REMOVE THE FIRST AZB FROM ROUTING(MZB)
LET MNAF(MZB) = APOA(F.ROUTING(MZB))--RANKING CODE IN QUEUE
LET MNAC(MZB) = ATPC(F.ROUTING(MZB))--RANKING CODE IN QUEUE
UNTIL TEAM(AZB) IS EMPTY DO
    REMOVE THE FIRST HZB FROM TEAM(AZB)
    IF FRAC.F(TIME.V) * 24 GE VEDDT OR
        FRAC.F(TIME.V) * 24 LT VBDDT
        GO TO OFF.DUTY
    ELSE
        IF WEEKDAY.F(TIME.V) GT VHWDD
            *OFF.DUTY*
            LET WSTA(MZB) = OFF.DUTY
            GO LOOP
        ELSE
            LET T = FRAC.F(TIME.V) * 24
            IF VB81T LT T LE VE81T OR
                VBL8T LT T LE VEL8T OR
                VB82T LT T LE VE82T
                LET WSTA(MZB) = BREAK
                GO LOOP
            ELSE
                LET WSTA(MZB) = IDLE
                ADD 1 TO LNIW(WLEV(MZB))
                *LOOP*
LOOP
DESTROY THIS ACTIVITY CALLED AZB
IF ROUTING(MZB) IS EMPTY
    LET TDCD(MTPN(MZB)) = TIME.V
DESTROY THIS MISSION CALLED MZB
GO CHECK.QUEUE
ELSE
    
```

```

.. FILE MZB IN QUEUE ..INSURES THAT LOW CODED ACTIVITIES OF
   ALL MISSIONS ARE PERFORMED FIRST.
   *CHECK.QUEUE*
   IF WEEKDAY.F(TIME.V) GT VNMDD
   THEN IF VWEAM = "NO"
   RETURN
   ELSE
   CALL CHECK.QUEUE
   RETURN
   END

```

```

00012390
00012400
00012410
00012420
00012430
00012440
00012450
00012460
00012470
00012480

```

00012490
 00012500
 00012510
 00012520
 00012530
 00012540
 00012550
 00012560
 00012570
 00012580
 00012590
 00012600
 00012610
 00012620
 00012630
 00012640
 00012650
 00012660
 00012670
 00012680
 00012690
 00012700
 00012710
 00012720
 00012730
 00012740
 00012750
 00012760
 00012770
 00012780
 00012790
 00012800
 00012810
 00012820
 00012830

```

..          EVENT EO.DUTY.HOURS SAVING THE EVENT NOTICE
SCHEDULE THIS EO.DUTY.HOURS IN 24 HOURS
IF WEEKDAY.F(TIME.V) GT VHMDD
RETURN ..OFFICE IS CLOSED FOR THE WEEKEND
ELSE
CALL DISBAND.PARTIAL.TEAMS
FOR EVERY WORKER CALLED WZX WITH WSTA(WZX) = IOLE DO
  LET WSTA(WZX) = OFF.DUTY
  SUBTRACT 1 FROM LNIW(MLEV(WZX))
LOOP
FOR EVERY K IN EV.S(I.EO.ACTIVITY)
  WITH TIME.A(K)-TIME.V LT 30/(24*60) DO
..  ACTIVITIES WITH LESS THAN 30 MINUTES REMAINING WILL BE COMPLETED
  LET MZD = EAMN(K)
  LET AZD = F.ROUTING(MZD)
  FOR EVERY WZD IN TEAM(AZD)
    LET WSTA(WZD) = OVERTIME
LOOP
FOR EVERY K IN EV.S(I.EO.ACTIVITY)
  WITH TIME.A(K)-TIME.V GE 30/(24*60) AND
  MNAPEAMN(K) NE 1 AND
  MNAPEAMN(K) NE 2 DO
  LET REASON = OFF.DUTY
  CALL INTERRUPT.AND.FILE GIVING K AND REASON
LOOP
FOR EVERY K IN EV.S(I.EO.ACTIVITY)
  WITH TIME.A(K)-TIME.V GE 30/(24*60) AND
  MNAPEAMN(K) = 1 OR
  MNAPEAMN(K) = 2 DO
  LET MZD = EAMN(K)
  LET AZD = F.ROUTING(MZD)
  FOR EVERY WZD IN TEAM(AZD)
    LET WSTA(WZD) = OVERTIME
LOOP

```

CALL OVERTIME ACTIVITIES FOR PRIORITY MISSIONS
CALL CHECK QUEUE REASSIGN WORKERS IF A PREEMPT OCCURRED
RETURN
END

00012640
00012650
00012660
00012670

00015000
00015010
00015020
00015030
00015040
00015050
00015060
00015070
00015080
00015090
00015100
00015110
00015120
00015130
00015140
00015150
00015160
00015170
00015180
00015190
00015200
00015210
00015220
00015230
00015240
00015250
00015260
00015270
00015280
00015290
00015300

00015310
00015320
00015330
00015340
00015350
00015360
00015370
00015380
00015390
00015400
00015410
00015420
00015430
00015440
00015450
00015460
00015470
00015480
00015490
00015500
00015510
00015520
00015530
00015540
00015550
00015560
00015570
00015580
00015590
00015600
00015610
00015620
00015630
00015640
00015650
00015660
00015670
00015680
00015690
00015700
00015710
00015720
00015730
00015740
00015750
00015760
00015770
00015780
00015790
00015800
00015810
00015820
00015830
00015840
00015850
00015860
00015870
00015880
00015890
00015900
00015910
00015920
00015930
00015940
00015950
00015960
00015970
00015980
00015990
00016000

00012880
 00012890
 00012900
 00012910
 00012920
 00012930
 00012940
 00012950
 00012960
 00012970
 00012980
 00012990
 00013000
 00013010
 00013020
 00013030
 00013040
 00013050
 00013060
 00013070
 00013080
 00013090
 00013100
 00013110
 00013120
 00013130
 00013140
 00013150
 00013160
 00013170
 00013180
 00013190
 00013200
 00013210
 00013220

```

..
ROUTINE OVERTIME.ACTIVITIES.FOR.PRIORITY.MISSIONS
CALL MY.TRACE(13)
DEFINE VAL AS REAL VARIABLE
FOR EVERY MZF IN QUEUE WITH MNA(MZF) = 1 AND
  MTOA(MZF) + VRMPT GT TIME.V DO
  REMOVE MZF FROM QUEUE
  FILE MZF IN WAIT.FOR.MISSION.TIME
  SCHEDULE AN ACCOMPLISH.PR1.MISSION AT MTOA(MZF) + VRMPT
  LOOP
FOR EVERY MZF IN QUEUE WITH MNA(MZF) = 1 OR
  MNA(MZF) = 2 AND
  ANMR(MZF) LE SUM.WORKERS(3,MZF,0) DO
  REMOVE THIS MZF FROM QUEUE
  IF ANMR(F.ROUTING(MZF)) = 0 **MANPOWER REQ IS EXTERNAL
  GO TO WORKER.FOUND
ELSE
LET AZF = F.ROUTING(MZF)
*PREEMPT.COMPLETED.LOOK.AGAIN*
LET J=0
LET I = AFWL(AZF)
*LOOK.AGAIN*
IF I=0
  GO TO V
ELSE
*MORE.REQUIRED*
LET MZF = 0
FOR EVERY WORKER CALLED WZP WITH WLEV(WZP) = I AND
  WSTA(WZP) NE OVERTIME
  COMPUTE MZF = MIN(WZP) OF TOBSERVED.STATE(WZP,5)
  IF MZF NE 0
    FILE MZF IN TEAM(AZF)
    LET WSTA(WZF) = OVERTIME
    SUBTRACT 1 FROM ANMR(AZF)
    IF ANMR(AZF) LT 0

```

```

CALL ERROR.PACKAGE(2,MZF)
ALWAYS
IF ANMR(AZF) = 0
GO TO WORKER.FOUND
ELSE
GO TO MORE.REQUIRED
ELSE
ADD 1 TO J
GO TO M, X, Y, V, U PER J
.W. LET I = ASHL(AZF)
GO TO LOOK.AGAIN
.X. LET I = ATWL(AZF)
GO TO LOOK.AGAIN
.Y. IF HMAP(MZF) = 1
CALL PREEMPT
ADD 1 TO L
IF L GT N.WORKER
GO TO V
ELSE
GO TO PREEMPT.COMPLETED.LOOK.AGAIN
ELSE
GO TO V
.V. UNTIL TEAM(AZF) IS EMPTY DO
REMOVE THE FIRST MZF FROM TEAM(AZF)
ADD 1 TO ANMR(AZF)
LET WSTA(MZF) = OFF.DUTY
LOOP
FILE MZF IN QUEUE..ANOTHER PRIORITY ACTIVITY IN PROGRESS
RETURN
.U. CALL ERROR.PACKAGE(3,MZF)
.WORKER.FOUND
LET VAL = VALUE(MZF)
IF WEEKDAY.F(TIME.V) GT VHMDD
PRINT 6 LINES WITH MONTH.F(TIME.V), DAY.F(TIME.V),
YEAR.F(TIME.V), HTPN(MZF), MZF, ANOA(AZF), APOA(AZF),

```

00013230
00013240
00013250
00013260
00013270
00013280
00013290
00013300
00013310
00013320
00013330
00013340
00013350
00013360
00013370
00013380
00013390
00013400
00013410
00013420
00013430
00013440
00013450
00013460
00013470
00013480
00013490
00013500
00013510
00013520
00013530
00013540
00013550
00013560
00013570

N. TEAM(AZF) AND VAL THUS

A PRIORITY MISSION HAS INITIATED ON A NON DUTY DAY

DATE	TEST PHASE	MISSION	ACTIVITY	PR	WORKERS	REQ	DURATION
** ** *	** *	*****	** *	** *	** *	** *	*****

ALWAYS

SCHEDULE AN EO.ACTIVITY GIVING MZF IN VAL MINUTES

LOOP
RETURN
END

00013580
 00013590
 00013600
 00013610
 00013620
 00013630
 0000013640
 00013650
 00013660
 00013670
 00013680
 00013690

00013800
00013810
00013820
00013830
00013840
00013850
00013860
00013870
00013880

ROUTINE DISBAND.PARTIAL.TEAMS

..
CALL MY.TRACE(21)
LET REASON = IDLE
FOR EVERY K IN EV.SII.ED.ACTIVITY) WITH
ANMR(F.ROUTING(EAMN(K))) NE 0
CALL INTERRUPT.AND.FILE(K,REASON)
RETURN
END

00013890
00013900
00013910
00013920
00013930
00013940
00013950
00013960
00013970
00013980
00013990

..
CALL MY.TRACE(21)
LET REASON = IDLE
FOR EVERY K IN EV.SII.ED.ACTIVITY) WITH
ANMR(F.ROUTING(EAMN(K))) NE 0
CALL INTERRUPT.AND.FILE(K,REASON)
RETURN
END

00013890
00013900
00013910
00013920
00013930
00013940
00013950
00013960
00013970
00013980
00013990
00014000
00014010
00014020
00014030

```

..      ROUTINE TO COMPUTE STAY-TIME AND RECORD CANCELLATION GIVEN MZG
      CALL MYTRACE(14,MZG)
      IF MXCI(MZG) = 1
        ADD 1 TO TCNX(MTPN(MZG))
        RETURN
      ELSE
        LET TSTY(MTPN(MZG)) = TIME-V - MTOA(MZG)
        ADD MCEL(MZG) TO TCHP(MTPN(MZG))
        FOR I = 1 TO N.TEST.PHASE
          IF TCHP(I) GE TREQ(I)
            SCHEDULE AN EO.SIMULATION NON
        ALWAYS
        RETURN
      END

```

00013980
00013990
00014000
00014010
00014020
00014030
00014040
00014050
00014060
00014070

END
OF
PROGRAM
MAY 1964
BY
J. W. B. /
MAY 1964
BY
J. W. B. /
MAY 1964
BY
J. W. B. /

00014040
00014050
00014060
00014070
00014080
00014090

EVENT WREPORT SAVING THE EVENT NOTICE
SCHEDULE THIS WREPORT IN 1 WK
CALL XWREPORT
RETURN
END

00014040
00014050
00014060
00014070
00014080
00014090
00014100
00014110
00014120
00014130
00014140
00014150
00014160
00014170
00014180
00014190
00014200
00014210
00014220
00014230
00014240
00014250
00014260
00014270
00014280
00014290
00014300
00014310
00014320
00014330
00014340
00014350
00014360
00014370
00014380
00014390
00014400
00014410
00014420
00014430
00014440
00014450
00014460
00014470
00014480
00014490
00014500
00014510
00014520
00014530
00014540
00014550
00014560
00014570
00014580
00014590
00014600
00014610
00014620
00014630
00014640
00014650
00014660
00014670
00014680
00014690
00014700
00014710
00014720
00014730
00014740
00014750
00014760
00014770
00014780
00014790
00014800
00014810
00014820
00014830
00014840
00014850
00014860
00014870
00014880
00014890
00014900
00014910
00014920
00014930
00014940
00014950
00014960
00014970
00014980
00014990
00015000

00014100
00014110
00014120
00014130
00014140
00014150
00014160
00014170
00014180
00014190
00014200
00014210
00014220
00014230
00014240
00014250
00014260
00014270
00014280
00014290
00014300
00014310
00014320
00014330
00014340
00014350
00014360
00014370
00014380
00014390
00014400
00014410
00014420
00014430
00014440

ROUTINE XMREPORT

DEFINE P AS A REAL VARIABLE
RESERVE VMCNF, VMCNX, VMCHP AS N.TEST.PHASE
START NEW PAGE
PRINT 4 LINES WITH MONTH.F(TIME.V), DAY.F(TIME.V), YEAR.F(TIME.V) THUS
WEEKLY TEST FORCE REPORT
(FOR WEEK ENDING **/**/****

FOR EVERY TEST.PHASE DO

LET I = TEST.PHASE
LET J = TCNF(I) - VMCNF(I)
LET K = TCNX(I) - VMCNX(I)
LET P = TCMP(I) - VMCHP(I)
IF REAL.F(J) = P = 0.0
GO LOOP
ELSE

PRINT 15 LINES WITH I, J, K, TREQ(I), P, TCMP(I), HAVG.STAY(I),
WSTD.STAY(I), MSUM.STAY(I), AND WSOME.SQUARES.STAY(I) THUS
MISSION STATISTICS

FOR TEST PHASE ** ARE

NUMBER SCHEDULED----- = *** MISSIONS
NUMBER CANCELLED----- = *** MISSIONS
NUMBER REQUIRED----- = *** MISSIONS
EFFECTIVE PRODUCTIVITY----- = ***,**** MISSIONS
CUMULATIVE NUMBER COMPLETED = ***,**** MISSIONS
AVERAGE MISSION DURATION---- = ***,**** DAYS
STD DEV----- = ***,**** DAYS
TOTAL OF DURATION TIMES----- = ***,**** DAYS
SUM OF SQ OF DJURATION TIMES = *****,** DAYS

```

LET VMCNF(I) = TCNF(I)
LET VMCNX(I) = TCNX(I)
LET VMCMP(I) = TCMP(I)
*LOOP.

```

PRINT 9 LINES WITH N.TEST.PHASE THUS
 STATISTICS ARE COMPUTED ONLY FOR MISSIONS
 COMPLETED IN THIS REPORTING PERIOD.

NOTE IF DURING THIS REPORTING PERIOD NO MISSIONS WERE SCHEDULED OR
 COMPLETED IN A TEST PHASE, MISSION STATISTICS WERE OMITTED.
 THERE ARE *** TEST PHASES.

START NEW PAGE
 PRINT 3 LINES AS FOLLOWS

```

MAX LENGTH MIN LENGTH AVG LENGTH STD DEV OF Q TIMES SUM
PRINT 1 LINE WITH MQMAX.QUEUE, MQMIN.QUEUE, MQAVG.QUEUE,
MQSTD.QUEUE, MQSUM.QUEUE THUS
*** MSN *** MSN *** MSN *** MSN *** DAYS
SKIP 3 LINES
BEGIN REPORT
BEGIN HEADING
PRINT 3 LINES THUS

```

QUEUE REPORT

WEEKLY HISTOGRAM OF QUEUE LENGTH

```

QUEUE LENGTH STATE PERCENTAGE
SKIP 1 LINE
END**HEADING
FOR I = INT.(MQMIN.QUEUE + 1) TO INT.(MQMAX.QUEUE + 1)
PRINT 1 LINE WITH I-1 AND MFREQ(I)/.07 THUS
*** MISSIONS
*** ** %

```

00014450
 00014460
 00014470
 00014480
 00014490
 00014500
 00014510
 00014520
 00014530
 00014540
 00014550
 00014560
 00014570
 00014580
 00014590
 00014600
 00014610
 00014620
 00014630
 00014640
 00014650
 00014660
 00014670
 00014680
 00014690
 00014700
 00014710
 00014720
 00014730
 00014740
 00014750
 00014760
 00014770
 00014780
 00014790

```

00014800
00014810
00014820
00014830
00014840
00014850
00014860
00014870
00014880
00014890
00014900
00014910
00014920
00014930
00014940
00014950
00014960
00014970
00014980
00014990
00015000
00015010
00015020
00015030
00015040
00015050
00015060
00015070
00015080
00015090
00015100
00015110
00015120
00015130
00015140

END*REPORT
PRINT 5 LINES THUS

NOTE ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS.
PERCENTAGES ARE BASED ON A 24 HOUR DAY AND DUE TO ROUNDING
MAY NOT SUM TO 100.

SKIP 3 LINES
BEGIN REPORT ON A NEW PAGE PRINTING FOR LOFM = 1 TO N.LOFM IN
GROUPS OF 5
BEGIN HEAD ING
PRINT 6 LINES THUS

HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL

STATE TIME
(DAYS)

NUMBER OF IDLE WORKERS LEVEL 1 LEVEL 2 LEVEL 3 LEVEL 4 ...LEVEL
END*HEADING
FOR I = 1 TO INT.F(LARGEST(I)) + 2 DO
PRINT 1 LINE WITH I-1 AND A GROUP OF
MIDLE.WORKERS(LEVEL.OF.WORKER,I) FIELDS THUS
*** **.**** **.**** **.**** **.**** **.****
LOOP
FOR L = 1 TO N.LOFM DO
FOR EVERY WORKER WITH WLEV(WORKER) = L
COMPUTE LTIT(L) = SUM OF WOBERVED.STATE(WORKER,1)
LOOP
PRINT 1 LINE THUS
TOTAL IDLE TIME
PRINT 1 LINE WITH A GROUP OF LTIT(LEVEL.OF.WORKER) FIELDS THUS
****.* **.**** **.**** **.**** **.**** **.****
END*REPORT
SKIP 3 LINES
BEGIN REPORT ON A NEW PAGE

```

BEGIN HEADING
PRINT 6 LINES THUS

ANALYSIS OF WORKER STATUS

HOURS IN STATE
STATE PERCENTAGES

WORKER IDLE BUSY BREAK OFF DUTY OVERTIME

END**HEADING

**STATE NOTATION IS FOR THE FIRST STATE, SECOND STATE, ETC.
FOR EACH WORKER DO

PRINT 2 LINES WITH

WNAME(WORKER),

WOBERVED.STATE(WORKER,1) * 24,

WOBERVED.STATE(WORKER,2) * 24,

WOBERVED.STATE(WORKER,3) * 24,

WOBERVED.STATE(WORKER,4) * 24,

WOBERVED.STATE(WORKER,5) * 24,

WOBERVED.STATE(WORKER,1) / .07,

WOBERVED.STATE(WORKER,2) / .07,

WOBERVED.STATE(WORKER,3) / .07,

WOBERVED.STATE(WORKER,4) / .07,

WOBERVED.STATE(WORKER,5) / .07 THUS

***** **.* HRS **.* HRS **.* HRS **.* HRS

***** **.* % **.* % **.* % **.* %

SKIP 1 LINE

LOOP

END**REPORT

RESET WEEKLY TOTALS OF N.QUEUE

FOR EVERY LEVEL.OF.WORKER

RESET WEEKLY TOTALS OF LNIW

FOR EVERY TEST.PHASE

RESET WEEKLY TOTALS OF TSTY

FOR EVERY WORKER

RESET WEEKLY TOTALS OF WSTA

RETURN

END

00015150
00015160
00015170
00015180
00015190
00015200
00015210
00015220
00015230
00015240
00015250
00015260
00015270
00015280
00015290
00015300
00015310
00015320
00015330
00015340
00015350
00015360
00015370
00015380
00015390
00015400
00015410
00015420
00015430
00015440
00015450
00015460
00015470
00015480
00015490
00015500

00015510
00015520
00015530
00015540
00015550
00015560

EVENT MREPORT SAVING THE EVENT NOTICE
SCHEDULE THIS MREPORT NEXT MONTH
CALL XMREPORT

RETURN
END

END REPORT

00015510
 00015520
 00015530
 00015540
 00015550
 00015560
 00015570
 00015580
 00015590
 00015600
 00015610
 00015620
 00015630
 00015640
 00015650
 00015660
 00015670
 00015680
 00015690
 00015700
 00015710
 00015720
 00015730
 00015740
 00015750
 00015760
 00015770
 00015780
 00015790
 00015800
 00015810
 00015820
 00015830
 00015840
 00015850
 00015860
 00015870
 00015880
 00015890
 00015900
 00015910
 00015920
 00015930
 00015940
 00015950
 00015960
 00015970
 00015980
 00015990
 00016000

00015570
 00015580
 00015590
 00015600
 00015610
 00015620
 00015630
 00015640
 00015650
 00015660
 00015670
 00015680
 00015690
 00015700
 00015710
 00015720
 00015730
 00015740
 00015750
 00015760
 00015770
 00015780
 00015790
 00015800
 00015810
 00015820
 00015830
 00015840
 00015850
 00015860
 00015870
 00015880
 00015890
 00015900
 00015910

```

ROUTINE XMREPORT
DEFINE P AS A REAL VARIABLE
RESERVE VMCNF, VMCNX, VMCMP AS N.TEST.PHASE
START NEW PAGE
PRINT 3 LINES WITH
MONTH.F(TIME.V-8.) AND MOD.F(YEAR.F(TIME.V-8.),100) THUS
REPORT FOR MONTH OF **/**
    
```

```

FOR EVERY TEST.PHASE DO
LET I = TEST.PHASE
LET J = TCNF(I) - VMCNF(I)
LET K = TCNX(I) - VMCNX(I)
LET P = TCOMP(I) - VMCMP(I)
IF REAL.F(J) = P = 0.0
GO LOOP
    
```

```

ELSE
PRINT 15 LINES WITH I, J, K, TREQ(I), P, TCOMP(I), MAVG.STAY(I),
MSTD.STAY(I), MSUM.STAY(I), AND MSOME.SQUARES.STAY(I) THUS
MISSION STATISTICS
FOR TEST PHASE ** ARE
    
```

```

NUMBER SCHEDULED----- = *** MISSIONS
NUMBER CANCELLED----- = *** MISSIONS
NUMBER REQUIRED----- = *** MISSIONS
EFFECTIVE PRODUCTIVITY----- = ***.**** MISSIONS
CUMULATIVE NUMBER COMPLETED- = ***.**** MISSIONS
AVERAGE MISSION DURATION----- = ***.**** DAYS
STD DEV----- = ***.**** DAYS
TOTAL OF DURATION TIMES----- = ***.**** DAYS
SUM OF SQ OF DURATION TIMES- = *****.** DAYS
    
```

```

LET VMCNF(I) = TCNF(I)
LET VMCNX(I) = TCNX(I)
LET VMCMP(I) = TCMP(I)
  *LOOP*

```

LOOP

PRINT 9 LINES WITH N.TEST.PHASE THUS

NOTE STATISTICS ARE COMPUTED ONLY FOR MISSIONS
 COMPLETED IN THIS REPORTING PERIOD.

NOTE IF DURING THIS REPORTING PERIOD NO MISSIONS WERE SCHEDULED OR
 COMPLETED IN A TEST PHASE, MISSION STATISTICS WERE OMITTED.
 THERE ARE *** TEST PHASES.

START NEW PAGE
 PRINT 3 LINES AS FOLLOWS

QUEUE REPORT

```

MAX LENGTH MIN LENGTH AVG LENGTH STD DEV OF Q TIMES SUM
PRINT 1 LINE WITH MQMAX.QUEUE, MQMIN.QUEUE, MQAVG.QUEUE,
MQSTD.QUEUE, MQSUM.QUEUE THUS
*** * MSN *** * MSN *** * MSN *** * MSN *** * DAYS
SKIP 3 LINES
BEGIN REPORT
BEGIN HEADING
PRINT 3 LINES THUS

```

MONTHLY HISTOGRAM OF QUEUE LENGTH

```

QUEUE LENGTH STATE TIME
SKIP 1 LINE
END * HEADING
FOR I = INT.(MQMIN.QUEUE + 1) TO INT.(MQMAX.QUEUE + 1)
  PRINT 1 LINE WITH I-1 AND MREQ(I) THUS
  *** MISSIONS
  ** * DAYS

```

00015920
 00015930
 00015940
 00015950
 00015960
 00015970
 00015980
 00015990
 00016000
 00016010
 00016020
 00016030
 00016040
 00016050
 00016060
 00016070
 00016080
 00016090
 00016100
 00016110
 00016120
 00016130
 00016140
 00016150
 00016160
 00016170
 00016180
 00016190
 00016200
 00016210
 00016220
 00016230
 00016240
 00016250
 00016260

00016620
00016630
00016640
00016650
00016660
00016670
00016680
00016690
00016700
00016710
00016720
00016730
00016740
00016750
00016760
00016770
00016780
00016790
00016800
00016810
00016820
00016830
00016840
00016850
00016860
00016870
00016880

WORKER IDLE BUSY BREAK OFF DUTY OVERTIME
END**HEADING
**STATE NOTATION IS FOR THE FIRST STATE, SECOND STATE, ETC.
FOR EACH WORKER 00
PRINT 1 LINES WITH WNAME(WORKER),
OBSERVED.STATE(WORKER,1) * 24,
OBSERVED.STATE(WORKER,2) * 24,
OBSERVED.STATE(WORKER,3) * 24,
OBSERVED.STATE(WORKER,4) * 24,
OBSERVED.STATE(WORKER,5) * 24, THUS
** ** HRS ** ** HRS ** ** HRS ** ** HRS ** ** HRS
LOOP ** ** HRS ** ** HRS ** ** HRS ** ** HRS

END**REPORT
RESET MONTHLY TOTALS OF N.QUEUE
FOR EVERY LEVEL.OF.WORKER
RESET MONTHLY TOTALS OF LNIM
FOR EVERY TEST.PHASE
RESET MONTHLY TOTALS OF TSTY
FOR EVERY WORKER
RESET MONTHLY TOTALS OF WSTA
RETURN
END

LOOP
END**REPORT
RESET MONTHLY TOTALS OF N.QUEUE
FOR EVERY LEVEL.OF.WORKER
RESET MONTHLY TOTALS OF LNIM
FOR EVERY TEST.PHASE
RESET MONTHLY TOTALS OF TSTY
FOR EVERY WORKER
RESET MONTHLY TOTALS OF WSTA
RETURN
END

LOOP
END**REPORT
RESET MONTHLY TOTALS OF N.QUEUE
FOR EVERY LEVEL.OF.WORKER
RESET MONTHLY TOTALS OF LNIM
FOR EVERY TEST.PHASE
RESET MONTHLY TOTALS OF TSTY
FOR EVERY WORKER
RESET MONTHLY TOTALS OF WSTA
RETURN
END

LOOP
END**REPORT
RESET MONTHLY TOTALS OF N.QUEUE
FOR EVERY LEVEL.OF.WORKER
RESET MONTHLY TOTALS OF LNIM
FOR EVERY TEST.PHASE
RESET MONTHLY TOTALS OF TSTY
FOR EVERY WORKER
RESET MONTHLY TOTALS OF WSTA
RETURN
END

LOOP
END**REPORT
RESET MONTHLY TOTALS OF N.QUEUE
FOR EVERY LEVEL.OF.WORKER
RESET MONTHLY TOTALS OF LNIM
FOR EVERY TEST.PHASE
RESET MONTHLY TOTALS OF TSTY
FOR EVERY WORKER
RESET MONTHLY TOTALS OF WSTA
RETURN
END

00016890
00016900
00016910
00016920
00016930
00016940
00016950
00016960
00016970
00016980
00016990
00017000
00017010
00017020
00017030
00017040
00017050
00017060
00017070
00017080
00017090
00017100
00017110
00017120
00017130
00017140
00017150
00017160
00017170
00017180
00017190
00017200
00017210
00017220
00017230

EVENT FINAL REPORT

START NEW PAGE

PRINT 4 LINES WITH MONTH.F(TIME.V), DAY.F(TIME.V), YEAR.F(TIME.V) THUS
FINAL TEST FORCE REPORT
(FOR SIMULATION ENDING ** ** ** **)

FOR EVERY TEST.PHASE DO

LET I = TEST.PHASE

PRINT 16 LINES WITH I, TCNF(I), TCNX(I), TREQ(I), TCMP(I),
TAVG.STAY(I), TSTD.STAY(I), TSUM.STAY(I), TISOME.SQUARES.STAY(I),
MONTH.F(TDCD(I)), DAY.F(TDCD(I)), AND YEAR.F(TDCD(I)) THUS
MISSION STATISTICS

FOR TEST PHASE ** ARE

NUMBER SCHEDULED----- = **
NUMBER CANCELLED----- = **
NUMBER REQUIRED----- = **
EFFECTIVE PRODUCTION----- = **, ****
AVERAGE MISSION DURATION--- = **, ****
STD DEV----- = **, ****
TOTAL OF DURATION TIMES----- = **, ****
SUM OF SQ OF DURATION TIMES- = **, ****
DATE COMPLETED----- = ** ** **

LOOP

PRINT 5 LINES THUS

NOTE SINCE THE DURATION OF CANCELLED MISSIONS IS ZERO,
STATISTICS ARE COMPUTED FOR COMPLETED MISSIONS ONLY.

START NEW PAGE
PRINT 3 LINES AS FOLLOWS

MAX LENGTH AVG LENGTH STD DEV
PRINT 1 LINE WITH TMAX.QUEUE, TQAVG.QUEUE, TQSTD.QUEUE, TQSUM.QUEUE THUS
*** MISSIONS *** MISSIONS *** MISSIONS
SKIP 3 LINES

BEGIN REPORT
BEGIN HEADING
PRINT 3 LINES THUS

HISTOGRAM OF QUEUE LENGTH

QUEUE LENGTH STATE PERCENTAGE
SKIP 1 LINE
END-HEADING
FOR I = 1 TO INT.(TQMAX.QUEUE + 1)
PRINT 1 LINE WITH I-1 AND (TFREQ(I)/TIME.V)*100 THUS
*** MISSIONS *** %
END-REPORT
PRINT 5 LINES THUS

NOTE ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS.
PERCENTAGES ARE BASED ON A 24 HOUR DAY AND DUE TO ROUNDING
MAY NOT SUM TO 100.

SKIP 3 LINES
BEGIN REPORT ON A NEW PAGE PRINTING FOR LOFM = 1 TO N.LOFM IN
GROUPS OF 5
BEGIN HEADING
PRINT 6 LINES THUS

HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL

STATE TIME

00017240
00017250
00017260
00017270
00017280
00017290
00017300
00017310
00017320
00017330
00017340
00017350
00017360
00017370
00017380
00017390
00017400
00017410
00017420
00017430
00017440
00017450
00017460
00017470
00017480
00017490
00017500
00017510
00017520
00017530
00017540
00017550
00017560
00017570
00017580

```

00017590
00017600
00017610
00017620
00017630
00017640
00017650
00017660
00017670
00017680
00017690
00017700
00017710
00017720
00017730
00017740
00017750
00017760
00017770
00017780
00017790
00017800
00017810
00017820
00017830
00017840
00017850
00017860
00017870
00017880
00017890
00017900
00017910
00017920
00017930

(DAYS)
NUMBER OF IDLE WORKERS LEVEL 1 LEVEL 2 LEVEL 3 LEVEL 4 ...LEVEL
END**HEADING
FOR I = 1 TO INT.F(LARGEST(3)) + 2 DO
PRINT 1 LINE WITH I-1 AND A GROUP OF
TIDLE.WORKERS(LEVEL.OF.WORKER,I) FIELDS THUS
*** **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.*
LOOP
FOR L = 1 TO N.L.OFW DO
FOR EVERY WORKER WITH MLEV(WORKER) = L
COMPUTE LTIIT(L) = SUM OF TOBSERVED.STATE(WORKER,I)
LOOP
PRINT 1 LINE THUS
TOTAL IDLE TIME
PRINT 1 LINE WITH A GROUP OF LTIIT(LEVEL.OF.WORKER) FIELDS THUS
*** **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.* **.*
END**REPORT
SKIP 3 LINES
BEGIN REPORT ON A NEW PAGE
BEGIN HEADING
PRINT 6 LINES THUS

ANALYSIS OF WORKER STATUS

HOURS IN STATE
STATE PERCENTAGES

WORKER IDLE BUSY BREAK OFF DUTY OVERTIME
END**HEADING
**STATE NOTATION IS FOR THE FIRST STATE, SECOND STATE, ETC.
FOR EACH WORKER DO
PRINT 2 LINES WITH WNAM(WORKER),
TOBSERVED.STATE(WORKER,1) * 24,
TOBSERVED.STATE(WORKER,2) * 24,
TOBSERVED.STATE(WORKER,3) * 24,

```

00017940
00017950
00017960
00017970
00017980
00017990
00018000
00018010
00018020
00018030
00018040
00018050
00018060
00018070
00018080

TOBSERVED.STATE(WORKER,4) * 24.
TOBSERVED.STATE(WORKER,5) * 24.
(TOBSERVED.STATE(WORKER,1)/TIME.V)*100.
(TOBSERVED.STATE(WORKER,2)/TIME.V)*100.
(TOBSERVED.STATE(WORKER,3)/TIME.V)*100.
(TOBSERVED.STATE(WORKER,4)/TIME.V)*100.
(TOBSERVED.STATE(WORKER,5)/TIME.V)*100 THUS
*** HRS **.* Z **.* Z
*** HRS **.* Z **.* Z
*** HRS **.* Z **.* Z
*** HRS **.* Z **.* Z

*** HRS **.* Z
*** HRS **.* Z
*** HRS **.* Z
*** HRS **.* Z
SKIP 1 LINE
LOOP
END**REPORT
CALL CLEAR
RETURN
END

00018090
00018100
00018110
00018120
00018130
00018140
00018150
00018160
00018170
00018180
00018190
00018200
00018210
00018220
00018230

ROUTINE LARGEST GIVEN FUNCT

DEFINE X AS A REAL VARIABLE
GO TO M, M, T PER FUNCT

M FOR EACH LEVEL OF WORKER

 COMPUTE X AS THE MAXIMUM OF WL MAX. IDLE. WORKERS (LOFW)
 JUMP AHEAD

M FOR EACH LEVEL OF WORKER

 COMPUTE X AS THE MAXIMUM OF ML MAX. IDLE. WORKERS (LOFW)

JUMP AHEAD

T FOR EACH LEVEL OF WORKER

 COMPUTE X AS THE MAXIMUM OF TL MAX. IDLE. WORKERS (LOFW)

 HERE

 RETURN WITH X

END

```

..
ROUTINE NY. TRACE(NUMB,IDENT)
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX
GO TO 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 PER NUMB
*1* WRITE TIME.V AS
  "CHECK.QUEUE WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*2* WRITE IDENT AND TIME.V AS
  "A BRK TIME ACTIVITY IN MSN", I 6, " WAS STARTED AT", D(10,4),/
  JUMP AHEAD
*3* WRITE IDENT AND TIME.V AS
  "DET PREMSN CMX FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*4* WRITE IDENT AND TIME.V AS
  "CAL MSN EFF FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*5* WRITE IDENT AND TIME.V AS
  "ATTEND.TO.MISSION FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*6* WRITE IDENT AND TIME.V AS
  "SUM.WORKERS FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*7* WRITE IDENT AND TIME.V AS
  "ALLOCATION FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*8* WRITE IDENT AND TIME.V AS
  "VAL NUM WRKS REQ FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*9* WRITE TIME.V AS
  "PREEMPT WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*10* WRITE IDENT AND TIME.V AS
  "INTERRUPT.AND.FILE FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
  JUMP AHEAD
*11* WRITE IDENT AND TIME.V AS

```

```

00010240
00010250
00010260
00010270
00010280
00010290
00010300
00010310
00010320
00010330
00010340
00010350
00010360
00010370
00010380
00010390
00010400
00010410
00010420
00010430
00010440
00010450
00010460
00010470
00010480
00010490
00010500
00010510
00010520
00010530
00010540
00010550
00010560
00010570
00010580

```

```

"VALUE FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•12• WRITE IDENT AND TIME.V AS
"LAUNCH FOR TEST PHASE", I 3, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•13• WRITE TIME.V AS
"OVERTIME.ACTIVITIES.FOR.PRIORITY.MSN WAS CALLED AT", D(10,4),/
JUMP AHEAD
•14• WRITE IDENT AND TIME.V AS
"CSTARC FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•15• WRITE IDENT AND TIME.V AS
"CENTER.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•16• WRITE IDENT AND TIME.V AS
"QEXIT.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•17• WRITE IDENT AND TIME.V AS
"WENTER.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•18• WRITE IDENT AND TIME.V AS
"WEXIT.INFO FOR MISSION", I 6, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•19• WRITE IDENT AND TIME.V AS
"MSN.SCHEDULE FOR TEST PHASE", I 3, " WAS CALLED AT", D(10,4),/
JUMP AHEAD
•20• WRITE TIME.V AS
"BUILD.PARTIAL.TEAMS WAS CALLED AT", D(10,4),/
JUMP AHEAD
•21• WRITE TIME.V AS
"DISBAND.PARTIAL.TEAMS WAS CALLED AT", D(10,4),/
HERE
ALWAYS
RETURN
END

```

```

00018940
00018950
00018960
00018970
00018980
00018990
00019000
00019010
00019020
00019030
00019040
00019050
00019060
00019070
00019080
00019090
00019100
00019110
00019120
00019130
00019140
00019150
00019160
00019170
00019180
00019190
00019200
00019210
00019220
00019230
00019240
00019250
00019260
00019270
00019280

ROUTINE MYE.TRACE
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX
PRINT 3 LINES THUS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
SKIP 1 LINE
GO TO EO.ACTIVITY, EO.DUTY.HOURS, EO.SIMULATION,
CONFIRMED.MISSION, TAKE.BREAK, ACCOMPLISH.PR1.MISSION,
BEGIN.WORKING, WREPORT, WREPORT OR FINAL.REPORT PER EVENT.V
*EO.ACTIVITY*
LET MZH = EAMN(EO.ACTIVITY)
LET AZH = F.ROUTING(MZH)
PRINT 1 LINE WITH ANDA(AZH), MZH, AND TIME.V THUS
ACTIVITY NUMBER ** OF MSN NUMBER ***** WAS COMPLETED AT ***.****
CALL MY.LIST(5, MZH)**ACTIVITIES IN ROUTING(MZH)
JUMP AHEAD
*EO.DUTY.HOURS*
WRITE TIME.V AS "DUTY HOURS ENDED AT", D(10,4),/
JUMP AHEAD
*EO.SIMULATION*
WRITE TIME.V AS "SIMULATION ENDED AT", O(10,4),/
JUMP AHEAD
*CONFIRMED.MISSION*
PRINT 1 LINE WITH CTPN(CONFIRMED.MISSION) AND TIME.V THUS
A CNF MSN IN TEST PHASE ** WAS RECEIVED AT *****.****
JUMP AHEAD
*TAKE.BREAK*
WRITE TIME.V AS "TAKE BREAK AT", D(10,4),/
JUMP AHEAD
*ACCOMPLISH.PR1.MISSION*
WRITE TIME.V AS "A PRIORITY 1 ACTIVITY WAS INITIATED AT", D(10,4),/
RETURN
*BEGIN.WORKING*

```

00019290
 00019300
 00019310
 00019320
 00019330
 00019340
 00019350
 00019360
 00019370
 00019380
 00019390
 00019400
 00019410
 00019420
 00019430
 00019440
 00019450
 00019460
 00019470
 00019480
 00019490

```

WRITE TIME.V AS "BEGIN WORKING AT", D(10,4),/
  JUMP AHEAD
•MREPORT.
WRITE TIME.V AS "A WEEKLY REPORT WAS GENERATED AT", D(10,4),/
  GO WAITING.INFO
•MREPORT.
WRITE TIME.V AS "A MONTHLY REPORT WAS GENERATED AT", D(10,4),/
  GO WAITING.INFO
•FINAL.REPORT.
WRITE TIME.V AS "THE FINAL REPORT WAS GENERATED AT", D(10,4),/
  •WAITING.INFO.
CALL MY.LIST(6)••MISSIONS IN QUEUE
CALL MY.LIST(7)••MISSIONS IN WAIT FOR MISSION TIME
  HERE
  IF WEEKDAY.F(TIME.V) LE VHHDD
  CALL MY.LIST(1)••WORKERS
  ALWAYS
  SKIP 3 LINES
  ALWAYS
  RETURN
  END

```

00019500
 00019510
 00019520
 00019530
 00019540
 00019550
 00019560
 00019570
 00019580
 00019590
 00019600
 00019610
 00019620
 00019630
 00019640
 00019650
 00019660
 00019670
 00019680
 00019690
 00019700
 00019710
 00019720
 00019730
 00019740
 00019750
 00019760
 00019770
 00019780
 00019790
 00019800
 00019810
 00019820
 00019830
 00019840

```

ROUTINE MY.LIST(NUMB, ID)
GO TO 1,2,3,4,5,6,7,8,9,10,11,12 PER NUMB
*1. PRINT 5 LINES THUS

    ATTRIBUTES OF EACH WORKER

    WORKER      WNAM      WLEV      WSTA      M.TEAM
    FOR EACH WORKER PRINT 1 LINE WITH WORKER, WNAM, WLEV, WSTA,
    M.TEAM THUS
    ***          ***          *          *
    JUMP AHEAD
*2. PRINT 5 LINES THUS

    ATTRIBUTES OF EACH LEVEL OF WORKER

    LEVEL      LNMW
    FOR EACH LEVEL OF WORKER PRINT 1 LINE WITH LOFW, LNIW(LOFW) THUS
    ***
    JUMP AHEAD
*3. PRINT 5 LINES THUS

    ATTRIBUTES OF TEST PHASE

    TNUM      TREQ      TCNF      TCMP      TCNX      TOCD      TOBM      N.STRUC      N.CX.PROB
    PRINT 1 LINE WITH TNUM(ID), TREQ(ID), TCNF(ID), TCMP(ID),
    TCNX(ID), TOCD(ID), TOBM(ID), N.STRUCTURE(ID),
    N.TCANCELLATION.PROBABILITY(ID) THUS
    ***          ***          ***          ***          ***          ***          ***          ***
    JUMP AHEAD
*4. PRINT 5 LINES THUS
    
```

00019850
 00019860
 00019870
 00019880
 00019890
 00019900
 00019910
 00019920
 00019930
 00019940
 00019950
 00019960
 00019970
 00019980
 00019990
 00020000
 00020010
 00020020
 00020030
 00020040
 00020050
 00020060
 00020070
 00020080
 00020090
 00020100
 00020110
 00020120
 00020130
 00020140
 00020150
 00020160
 00020170
 00020180
 00020190

ATTRIBUTES OF EACH ACTIVITY IN STRUCTURE(TEST PHASE)

ATPC ANOA APOA ATFC AICT ANWR APAR AFWL ASWL ATWL
 FOR EVERY AZG IN STRUCTURE(ID) PRINT 1 LINE WITH ATPC(AZG),
 ANOA(AZG), APOA(AZG), ATFC(AZG), AICT(AZG), ANWR(AZG),
 APAR(AZG), AFWL(AZG), ASWL(AZG), ATWL(AZG) THUS
 *** ** ** ** ** ** ** **
 PRINT 2 LINES THUS

ATPC ACEL APMA ALRL AURL M.STRUCTURE N.TEAM
 FOR EVERY AZG IN STRUCTURE(ID) PRINT 1 LINE WITH ATPC(AZG),
 ACEL(AZG), APMA(AZG), ALRL(AZG), AURL(AZG), M.STRUCTURE(AZG)
 AND N.TEAM(AZG) THUS
 *** ** ** ** * **

JUMP AHEAD
 *5. PRINT 4 LINES THUS

ATTRIBUTES OF EACH ACTIVITY IN ROUTING(MISSION)

PRINT 1 DOUBLE LINE THUS
 ATPC ANOA APOA ATFC AICT ANWR APAR AFWL ASWL ATWL
 ACEL APMA ALRL AURL M.ROUTING N.TEAM
 FOR EACH AZG IN ROUTING(ID) PRINT 1 DOUBLE LINE WITH ATPC(AZG),
 ANOA(AZG), APOA(AZG), ATFC(AZG), AICT(AZG), ANWR(AZG), APAR(AZG),
 AFWL(AZG), ASWL(AZG), AURL(AZG), M.ROUTING(AZG), APMA(AZG),
 ALRL(AZG), AURL(AZG), M.ROUTING(AZG) AND N.TEAM(AZG) THUS
 *** ** ** ** * **

JUMP AHEAD
 *6. IF QUEUE IS NOT EMPTY PRINT 4 LINES THUS

ATTRIBUTES OF EACH MISSION IN QUEUE

```

00020200
00020210
00020220
00020230
00020240
00020250
00020260
00020270
00020280
00020290
00020300
00020310
00020320
00020330
00020340
00020350
00020360
00020370
00020380
00020390
00020400
00020410
00020420
00020430
00020440
00020450
00020460
00020470
00020480
00020490
00020500
00020510
00020520
00020530
00020540

PRINT 1 DOUBLE LINE THUS
MISSION MTPN MCEL MTOA MNAC MMAP MXCI M-QUEUE
M.WFHT N.CX.PROB N.ROUTING
FOR EVERY MZP IN QUEUE PRINT 1 DOUBLE LINE WITH MZP, MTPN(MZP),
MCEL(MZP), MTOA(MZP), MNAC(MZP), MMAP(MZP), MXCI(MZP),
M.QUEUE(MZP), M.WAIT.FOR.MISSION.TIME(MZP),
M.MCANCELLATION.PROBABILITY(MZP) AND N.ROUTING(MZP) THUS
*****
*
ALWAYS
JUMP AHEAD
*7. IF WAIT.FOR.MISSION.TIME IS NOT EMPTY PRINT 4 LINES THUS

ATTRIBUTES OF EACH MISSION IN WAIT FOR MISSION TIME

PRINT 1 DOUBLE LINE THUS
MISSION MTPN MCEL MTOA MNAC MMAP MXCI M-QUEUE
M.WFHT N.CX.PROB N.ROUTING
DEFINE WFHT TO MEAN WAIT.FOR.MISSION.TIME
FOR EVERY MZP IN WFHT PRINT 1 DOUBLE LINE WITH MZP, MTPN(MZP),
MCEL(MZP), MTOA(MZP), MNAC(MZP), MMAP(MZP), MXCI(MZP),
M.QUEUE(MZP), M.WAIT.FOR.MISSION.TIME(MZP),
M.MCANCELLATION.PROBABILITY(MZP) AND N.ROUTING(MZP) THUS
*****
*
ALWAYS
JUMP AHEAD
*8. IF TCANCELLATION.PROBABILITY IS NOT EMPTY PRINT 5 LINES THUS

ATTRIBUTES OF EACH XFACTOR IN TCANCELLATION.PROB(TEST PHASE)

XUPL M.TCANCELLATION.PROBABILITY

```

00020550
00020560
00020570
00020580
00020590
00020600
00020610
00020620
00020630
00020640
00020650
00020660
00020670
00020680
00020690
00020700
00020710
00020720
00020730
00020740
00020750
00020760
00020770
00020780
00020790
00020800
00020810
00020820
00020830
00020840
00020850
00020860
00020870
00020880
00020890

FOR EVERY FCT IN TCANCELLATION.PROBABILITY(ID) PRINT 1 LINE
WITH XUPL(FCT) AND M.TCANCELLATION.PROBABILITY(FCT) THUS

ALWAYS
JUMP AHEAD
9 BEGIN REPORT PRINTING FOR TEST.PHASE = 1 TO N.TEST.PHASE IN
GROUPS OF 8
BEGIN HEADING
PRINT 7 LINES THUS

1ARG FOR EACH ACTIVITY FOR EACH TEST PHASE

ACT 1 2 3 4 5 6 7 N
TEST PHASE

END**HEADING
FOR I = 1 TO N.KIND.OF.ACTIVITY
PRINT 1 LINE WITH I AND A GROUP OF 1ARG(TEST.PHASE,I) FIELDS THUS
*** **

END**REPORT
JUMP AHEAD
10 BEGIN REPORT PRINTING FOR TEST.PHASE = 1 TO N.TEST.PHASE IN
GROUPS OF 8
BEGIN HEADING
PRINT 7 LINES THUS

2ARG FOR EACH ACTIVITY FOR EACH TEST PHASE

ACT 1 2 3 4 5 6 7 N
TEST PHASE

END**HEADING
FOR I = 1 TO N.KIND.OF.ACTIVITY

PRINT 1 LINE WITH I AND A GROUP OF ZARG(TEST.PHASE,I) FIELDS THUS
 *** *****
 END**REPORT
 JUMP AHEAD
 *11*BEGIN REPORT PRINTING FOR WORKER = 1 TO N.WORKER IN GROUPS OF 13
 BEGIN HEADING
 PRINT 7 LINES THUS

THE SKILL FACTOR OF EACH WORKER IN PERFORMING AN ACTIVITY IS

ACT	1	2	3	4	5	6	7	8	9	10	11	12	N

END**HEADING
 FOR I = 1 TO N.KIND.OF.ACTIVITY
 PRINT 1 LINE WITH I AND A GROUP OF SKILL.FACTOR(WORKER,I)
 FIELDS THUS
 *** *****

END**REPORT
 JUMP AHEAD
 *12*PRINT 5 LINES THUS

ATTRIBUTES OF EACH KIND OF ACTIVITY

ACT KNAM KPOA KTFC KPAR KFWL KSML KTWL KPMA KLRL KURL
 FOR EVERY KIND.OF.ACTIVITY PRINT 1 LINE WITH KOA, KNAM(KOA),
 KPOA(KOA), KTFC(KOA), KPAR(KOA), KFWL(KOA), KSML(KOA), KTWL(KOA),
 KPMA(KOA), KLRL(KOA), KURL(KOA) THUS
 *** *****

HERE
 SKIP 1 LINE
 RETURN
 END

00020900
 00020910
 00020920
 00020930
 00020940
 00020950
 00020960
 00020970
 00020980
 00020990
 00021000
 00021010
 00021020
 00021030
 00021040
 00021050
 00021060
 00021070
 00021080
 00021090
 00021100
 00021110
 00021120
 00021130
 00021140
 00021150
 00021160
 00021170
 00021180
 00021190
 00021200
 00021210
 00021220
 00021230
 00021240

00021250
00021260
00021270
00021280
00021290
00021300
00021310
00021320

ROUTINE CENTER.INFO GIVEN MZN
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX
CALL MY.TRACE(15,MZN)
CALL INFO.PACKAGE(MZN)
ALWAYS
RETURN
END

..

00051250
00051260
00051270
00051280
00051290
00051300
00051310
00051320

END
RETURN
CALL MY.TRACE(15,MZN)
CALL INFO.PACKAGE(MZN)
ALWAYS
RETURN
END

00021330
00021340
00021350
00021360
00021370
00021380
00021390
00021400

00051750
00051760
00051770
00051780
00051790
00051800
00051810
00051820

..
ROUTINE QEXIT.INFO GIVEN MZN
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX
CALL MY.TRACE(16,MZY)
CALL INFO.PACKAGE(MZN)
ALWAYS
RETURN
END

END
MZN
SERV
CVTT INFO*ACQUIRE(MZN)
CVTT AL*INFLATE* MZN
IF TIME.V GT CK*MIN AND TIME.V LT CK*MAX
NO*ALL* GEN*ER*ATE*ING* C*E*A*R* M*Z*N

00021410
00021420
00021430
00021440
00021450
00021460
00021470
00021480

00051200
00051210
00051220
00051230
00051240
00051250
00051260
00051270

ROUTINE WENTER.INFO GIVEN MZN
IF TIME.V GT CK.MIN AND TIME.V LT CK.MAX
CALL MY.TRACE(17,MZN)
CALL INFO.PACKAGE(MZN)
ALWAYS
RETURN
END

END
BEADW
WYBYAZ
CVT IAL 07 0000/0000,0000
CVT AL 10000000000000
IA TIME^A EL CK^MZH WHO TIME^A EL CK^MZH
KOLIME AKALL^IMAO CLAEW MSH

00021570
00021580
00021590
00021600
00021610
00021620
00021630
00021640
00021650
00021660
00021670
00021680
00021690
00021700
00021710
00021720
00021730
00021740
00021750
00021760
00021770
00021780
00021790
00021800
00021810
00021820
00021830
00021840
00021850
00021860
00021870
00021880
00021890
00021900
00021910
00021920
00021930

ROUTINE ERROR.PACKAGE (NUMB, ID)

GO TO 1,2,3,4 PER NUMB

•1. PRINT 9 LINES WITH READ.V, RECORD.V(READ.V), RCOLUMN.V THUS

ERROR - EITHER TOO MUCH DATA OR DATA WAS INCORRECTLY READ.

INPUT STATUS

UNIT	CARD NUMBER	COLUMN
**	****	**

STOP

•2. PRINT 4 LINES WITH ATPC(F.ROUTING(ID)), ID, AND TIME.V THUS

ERROR - NUMBER OF WORKERS REQ FOR ACTIVITY CODE *** IN MISSION ***** IS NEGATIVE. TRACE ABOUT DAY ****.****

JUMP AHEAD

•3. PRINT 4 LINES WITH ATPC(F.ROUTING(ID)) AND TIME.V THUS

ERROR - NO WORKER LEVEL WAS SPECIFIED FOR ACTIVITY CODE *** TRACE ABOUT DAY ****.****

JUMP AHEAD

•4. PRINT 6 LINES WITH TIME.V THUS

ERROR - NUMBER OF WORKERS REQUESTED TO PERFORM THE ACTIVITY EXCEEDS THE TOTAL NUMBER OF QUALIFIED WORKERS IN THE SIMULATION. TRACE ABOUT DAY ****.****

HERE

CALL INFO.PACKAGE(ID)

STOP

END

00021940
 00021950
 00021960
 00021970
 00021980
 00021990
 00022000
 00022010
 00022020
 00022030
 00022040
 00022050
 00022060
 00022070
 00022080
 00022090
 00022100

ROUTINE INFO.PACKAGE(MZI)
 CALL MY.LIST(5,MZI) ..ACTIVITIES IN ROUTING(MZI)
 CALL MY.LIST(1) ..WORKERS
 CALL MY.LIST(6) ..MISSIONS IN QUEUE
 CALL MY.LIST(7) ..MISSIONS IN WAIT FOR MISSION TIME
 RETURN
 END

EVENT EO.SIMULATION

IF EVENT IS EXTERNAL
 CALL MY.LIST(1) ..WORKERS
 CALL MY.LIST(6) ..QUEUE
 CALL MY.LIST(7) ..WAIT FOR MISSION TIME

ALWAYS
 SCHEDULE A FINAL.REPORT NOW
 RETURN
 END

00022110
 00022120
 00022130
 00022140
 00022150
 00022160
 00022170
 00022180
 00022190
 00022200
 00022210
 00022220
 00022230
 00022240
 00022250
 00022260
 00022270
 00022280
 00022290
 00022300
 00022310
 00022320
 00022330
 00022340
 00022350
 00022360
 00022370
 00022380
 00022390
 00022400
 00022410
 00022420
 00022430
 00022440
 00022450

```

..
    ROUTINE TO CLEAR
    LET TIME.V = 0 ..MUST ZERO BEFORE RESETS FOR VALID STATISTICS
    RELEASE MNAM, MLEV, WSTA, M.TEAM, MOBSERVED.STATE, MOBSERVED.STATE,
    TOBSERVED.STATE
    RELEASE LNM, LTIM, MIDLE.WORKERS, MIDLE.WORKERS, TIDLE.WORKERS
    ..MUST EMPTY SET WHEN ED.SIMULATION WAS AN EXTERNAL EVENT
    FOR EVERY QM2 IN QUEUE DO
        REMOVE QM2 FROM QUEUE
        DESTROY MISSION CALLED QM2
    LOOP
    FOR EVERY XMZ IN WAIT.FOR.MISSION.TIME DO
        REMOVE XMZ FROM WAIT.FOR.MISSION.TIME
        DESTROY MISSION CALLED XMZ
    LOOP
    RESET TOTALS OF N.QUEUE
    FOR EVERY LEVEL.OF.WORKER
        RESET TOTALS OF LNM
    FOR EVERY TEST.PHASE DO
        LET TCNF( TEST.PHASE) = 0
        LET TCMF( TEST.PHASE) = 0
        LET TCMX( TEST.PHASE) = 0
        LET TCD( TEST.PHASE) = 0
        LET TSTY( TEST.PHASE) = 0
        LET VMCHP( TEST.PHASE) = 0
        LET VMCNF( TEST.PHASE) = 0
        LET VMCNX( TEST.PHASE) = 0
        LET VMCHP( TEST.PHASE) = 0
        LET VMCNF( TEST.PHASE) = 0
        LET VMCNX( TEST.PHASE) = 0
    RESET TOTALS OF TSTY
    LOOP
    FOR EVERY WORKER
        RESET TOTALS OF WSTA
        USE UNIT 7 FOR INPUT
    
```

00022460
 00022470
 00022480
 00022490
 00022500
 00022510
 00022520
 00022530
 00022540
 00022550
 00022560
 00022570
 00022580
 00022590
 00022600
 00022610
 00022620
 00022630
 00022640
 00022650
 00022660
 00022670
 00022680
 00022690
 00022700
 00022710
 00022720
 00022730
 00022740
 00022750
 00022760
 00022770
 00022780
 00022790
 00022800

```

IF DATA IS NOT ENDED
FOR I = 1 TO 10
  LET SEED.V(I) = SAVESEED(I)
ALWAYS
FOR J = 1 TO EVT.V DO
  UNTIL EV.S(J) IS EMPTY DO
  REMOVE THE FIRST K FROM EV.S(J)
  GO TO EO.ACTIVITY, EO.DUTY.HOURS, EO.SIMULATION,
  CONFIRMED.MISSION, TAKE.BREAK,
  ACCOMPLISH.PRI.MISSION, BEGIN.WORKING,
  WREPORT, WREPORT OR FINAL.REPORT PER J
  *EO.ACTIVITY*
  DESTROY THE EO.ACTIVITY CALLED K
  GO LOOP
  *EO.DUTY.HOURS*
  DESTROY THE EO.DUTY.HOURS CALLED K
  GO LOOP
  *EO.SIMULATION*
  DESTROY THE EO.SIMULATION CALLED K
  GO LOOP
  *CONFIRMED.MISSION*
  DESTROY THE CONFIRMED.MISSION CALLED K
  GO LOOP
  *TAKE.BREAK*
  DESTROY THE TAKE.BREAK CALLED K
  GO LOOP
  *ACCOMPLISH.PRI.MISSION*
  DESTROY THE ACCOMPLISH.PRI.MISSION CALLED K
  GO LOOP
  *BEGIN.WORKING*
  DESTROY THE BEGIN.WORKING CALLED K
  GO LOOP
  *WREPORT*
  DESTROY THE WREPORT CALLED K
  GO LOOP
    
```

00022810
 00022820
 00022830
 00022840
 00022850
 00022860
 00022870
 00022880
 00022890
 00022900
 00022910
 00022920
 00022930
 00022940
 00022950
 00022960
 00022970
 00022980

```

* MREPORT*
  DESTROY THE MREPORT CALLED K
  GO LOOP
* FINAL.REPORT*
  DESTROY THE FINAL.REPORT CALLED K
  * LOOP*
  LOOP
  LOOP
  REWIND 8
  REWIND 9
  PRINT 5 LINES THUS

```

MODEL HAS BEEN CLEARED FOR NEW INPUT DATA

RETURN
END

00022990
 00023000
 00023010
 00023020
 00023030
 00023040
 00023050
 00023060
 00023070
 00023080
 00023090
 00023100
 00023110
 00023120
 00023130
 00023140
 00023150
 00023160
 00023170
 00023180
 00023190
 00023200
 00023210
 00023220
 00023230
 00023240
 00023250
 00023260
 00023270
 00023280
 00023290
 00023300
 00023310
 00023320
 00023330

```

ROUTINE REPLICATION
DEFINE OPTIONAL.SEED AS AN ALPHA VARIABLE
FOR EVERY TEST.PHASE DO
  FOR EVERY XFACTOR IN TCANCELLATION.PROBABILITY(TEST.PHASE) DO
    REMOVE XFACTOR FROM TCANCELLATION.PROBABILITY(TEST.PHASE)
  DESTROY XFACTOR
  LOOP
FOR EVERY ACTIVITY IN STRUCTURE(TEST.PHASE) DO
  REMOVE ACTIVITY FROM STRUCTURE(TEST.PHASE)
  DESTROY ACTIVITY
  LOOP
RELEASE TTMC, TNUM, TREQ, TCMF, TCMX, TOCD, TDBM, TSTY,
M.STRUCTURE, M.TCANCELLATION.PROBABILITY,
M.STRUCTURE, M.TCANCELLATION.PROBABILITY
RELEASE KPOA, KTFC, KFHL, KSWL, KTWL, KPMA, KLRL, KURL
RELEASE IARG, ZARG
RELEASE SKILL.FACTOR
RELEASE ONE, TWO, THREE
REWIND 7
SUBTRACT 1 FROM VRUNS
USE UNIT 10 FOR INPUT
READ OPTIONAL.SEED
IF OPTIONAL.SEED = "YES"
  FOR I = 1 TO 10
    READ SEED.V(I)
  PRINT 7 LINES WITH VRUNS-1 THUS
    
```

THE SIMULATION MODEL HAS BEEN RESET FOR A REPLICATION OF THE ENTIRE DATA PACKAGE. NEW SEED VALUES FOR THE RANDOM NUMBER GENERATORS WERE ENTERED. THERE ARE ** ADDITIONAL RUNS.

00023340
00023350
00023360
00023370
00023380
00023390
00023400
00023410
00023420
00023430
00023440
00023450

RETURN
ELSE
PRINT 7 LINES WITH VRUNS-1 THUS

THE SIMULATION MODEL HAS BEEN RESET FOR A REPLICATION OF THE ENTIRE
DATA PACKAGE. SEED VALUES FOR THE RANDOM NUMBER GENERATORS WERE
NOT RESET. THERE ARE ** ADDITIONAL RUNS.

RETURN
END

DATA INPUT INSTRUCTIONS

CODES

A - ALPHA NUMERIC DATA

I - INTEGER DATA

R - REAL DATA

(DECIMAL POINTS MAY BE OMITTED FOR WHOLE NUMBERS)

1. ENTER CONTROL CARD
S DATA 07
2. HOW MANY TIMES DO YOU WANT TO RUN THE ENTIRE DATA PACKAGE
I
3. ENTER THE DATE TO BEGIN THE SIMULATION (MUST BE MONDAY)
MONTH DAY YEAR
I I I
4. DO YOU WANT WEEKLY REPORTS (NO)
A
5. DO YOU WANT MONTHLY REPORTS (NO)
A
6. HOW MANY KIND.OF.ACTIVITY
I
7. DO YOU WANT TO DELETE THE FEATURE
REDUCTION.OF.MISSION.EFFECTIVENESS (YES)
A
8. IF "7" IS YES, ENTER FOR EACH KIND.OF.ACTIVITY.
KOA ATTRIBUTES
KNAH KPOA KTFC KPAR KFNL KSNL KTNL
1 A I I A I I I
2
-
H
NOTE IF AN ACTIVITY CANNOT BE PERFORMED BY THREE
DIFFERENT LEVELS OF WORKER, MAKE REPEATED
ENTRIES OF A QUALIFIED LEVEL OF WORKER.
9. IF "7" IS NO ENTER FOR EACH KIND.OF.ACTIVITY.
KOA ATTRIBUTES
KNAH KPOA KTFC KPAR KFNL KSNL KTNL KPHA KLRL KURL
1 A I I A I I I R R R
2
-
H

10. HOW MANY TEST.PHASE
I

DO 11 THRU 17 FOR EACH TEST.PHASE

11. ENTER THE TEST.PHASE NUMBER.
I

12. ENTER THE NUMBER OF MISSIONS REQUIRED.
I

13. ENTER THE TIME MISSION CONFIRMED (TTMC) RANDOM VARIABLE.
CUMLA PROB(1) VALUE(1) --- --- CUMLA PROB(N) VALUE(N)
R R R R *

14. ENTER THE 1ARG, 2ARG IN MINUTES FOR EACH KIND.OF.ACTIVITY
1ARG(1) 2ARG(1) ----- ----1ARG(N) 2ARG(N)
R R R R
NOTE IF AN ACTIVITY IS NOT USED IN A TEST PHASE,
ENTER ZEROES

15. ENTER FOR EACH ACTIVITY.
ACTIVITY ATTRIBUTES
ATPC ANOA ANWR
1 I I I
2
-
M

NOTE SINCE ATPC INDICATES THE ORDER IN WHICH ACTIVITIES
ARE PERFORMED, HIGHER NUMBERS SHOULD BE ASSIGNED AS
ATPC'S WITHIN A TEST PHASE AND IN SUCCEEDING TEST
PHASES. SEE TEXT FOR ASSIGNMENT OF PRIORITIES.

16. ENTER * TO DELETE MISSION CANCELLATION FEATURE.
*

17. IF "16" IS NOT * ENTER THE PROB OF CNX FOR EACH FACTOR.
R R *
#####

18. ENTER OK
A

19. ARE THERE ANY EMPERICAL FUNCTIONS (NO)
A
#####

20. IF "19" IS YES,
 HOW MANY INTEGER.STEP.FUNCTION
 I
 ENTER FOR EACH FUNCTION
 FUNCTION PAIRS
 CUMULA PROB(1) VALUE(1)---CUMULA PROB(N) VALUE(N)
 1 R I R I
 2
 -
 M
21. IF "19" IS YES,
 HOW MANY REEL.STEP.FUNCTION
 I
 ENTER FOR EACH FUNCTION
 FUNCTION PAIRS
 CUMULA PROB(1) VALUE(1)---CUMULA PROB(N) VALUE(N)
 1 R I R I
 2
 -
 M
22. IF "19" IS YES,
 HOW MANY CONTINUOUS.FUNCTION
 I
 ENTER FOR EACH FUNCTION
 FUNCTION PAIRS
 CUMULA PROB(1) VAULE(1)---CUMULA PROB(N) VALUE(N)
 1 R R R R
 2
 -
 M
 #####
23. HOW MANY DUTY DAYS PER WEEK
 I
24. WILL YOU ALLOW ACTIVE MISSIONS ON THE WEEKENDS (YES OR NO)
 A
25. ENTER THE TIMES TO START AND STOP WORKING.
 VBDDT VBB1T VEB1T VBLBT VELBT VBB2T VEB2T VEDDT
 R R R R R R R R
26. ENTER THE NUMBER OF DAYS PREPARATION REQ FOR AN ACTIVE MSN.
 R

27. ENTER OK
A

REPEAT 28 THRU 36 FOR EACH SIMULATION RUN.

28. ENTER * TO DELETE TRACE MESSAGE FEATURE.
*

29. IF 28 IS NOT * ENTER DAY TO START AND STOP TRACE MESSAGES.
R R *

30. ENTER OK
A

31. ARE ALL SKILL FACTORS THE SAME (YES)
A

32. HOW MANY LEVEL OF WORKER
I

33. HOW MANY WORKERS
I

34. ENTER FOR EACH WORKER
WORKER

ATTRIBUTES

IF "31" WAS NO,
SKILL FACTORS

WNAH WLEV
A I

KOA(1)-----KOA(N)
R R

1
2
-
N

NOTE WORKERS SHOULD BE ENTERED IN ORDER BY LEVEL

35. ENTER THE NUMBER OF DAYS BETWEEN MSN FOR EACH TEST.PHASE
R ---- R

NOTE TO SCHEDULE MORE THAN 1 MISSION PER DAY, ENTER THE
VALUE ((END DUTY DAY - BEGIN DUTY DAY)/24)/NUMBER
EXAMPLE OFFICE HOURS 0800-1700, 3 MISSIONS/DAY
((17.0-8.0)/24)/3 = 0.125

36. ENTER OK
A

37. ENTER CONTROL CARD
S DATA 08

38. ENTER EXTERNAL CONFIRMED,SCHEDULED,MISSION
 EVENT DATE TIME TEST.PHASE SEQUENTIAL ACTIVITY CODES
 A I I I I I -- SEE NOTE -- I*
 NOTE IF AN ACTIVITY CODE IS NOT PART OF THE INTERNAL
 EVENT SEQUENCE DESCRIBED IN 15, ENTER AFTER THAT
 ACTIVITY CODE.

ANOA APOA ANWR AFWL ASHL ATHL
 I I I I I I

39. ENTER CONTROL CARD
 \$ DATA 09

40. ENTER EXTERNAL EO.SIMULATION
 EVENT DATE TIME
 A I I *

41 THRU 43 ARE REQUIRED FOR REPLICATIONS.

41. ENTER CONTROL CARD.
 \$ DATA 10

42. IF SPECIFIED SEED VALUES FOR THE RANDOM NUMBER GENERATORS
 ARE TO BE USED, ENTER YES AND THE VALUES. (10 VALUES EACH)
 1 A I I I I I I I I I
 2
 -
 M

43. IF THE CURRENT SEED VALUES ARE TO BE USED, ENTER NO FOR
 EACH REPLICATION.
 A ---- ---- A

44. ENTER CONTROL CARD
 \$ ENDJOB

DATA INPUT

2-DATA-01
 1
 E 24/1978
 NO
 NO
 14
 NO
 PAYEE 2 2 YES 1 2 3 0 0
 PAYEE 2 2 YES 1 2 3 0 0
 GROSS 1 2 NO 1 2 3 10 10 20
 CASH 1 2 NO 1 2 3 30 30 100
 PENDING 2 2 NO 1 2 3 0 0
 PENDING 2 2 NO 1 2 3 0 0
 DATED 1 2 NO 3 3 2 10 20
 DATED 2 2 YES 3 1 2 0 0
 PENDING 2 2 YES 1 1 0 0 0
 PENDING 1 2 NO 1 1 10 10 12
 PENDING 2 2 NO 1 2 3 0 0
 PENDING 1 2 NO 1 1 0 0 0
 DATED 2 2 YES 3 1 2 10 20
 DATED 1 2 YES 3 1 2 0 0

APPENDIX E

SAMPLE DATA

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100

DATA INPUT

\$.DATA:07

1
5 24 1976

NO

NO

14

NO

PMSNPG	2	9	YES	1	2	3	0	0	0
PMSNPA	2	9	YES	1	2	3	0	0	0
CGDMSN	1	9	NO	1	2	3	10	10	20
CAMSN	1	9	NO	1	2	3	30	30	100
PSTMNG	3	9	NO	1	2	3	0	0	0
PSTMNA	2	9	NO	1	2	3	0	0	0
DATARD	1	9	NO	3	3	3	5	10	20
DATAAN	3	9	YES	3	1	2	0	0	0
PMSNI	2	9	YES	1	1	1	0	0	0
CMSNI	1	9	NO	1	1	1	10	10	15
PSTMNI	3	9	NO	1	2	2	0	0	0
MSNNEL	1	3	NO	1	1	1	0	0	0
ANTAPE	3	9	YES	3	1	2	10	10	20
ANLOGS	3	9	YES	3	1	2	0	0	0

8

1

6

0	8	.75	10	1.0	12	*			
360	480	0	0	120	180	0	0		
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

211 1 3

212 3 3 *

*

2

12

0	10	.75	13	1.0	14	*			
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
480	600	120	180	120	180	0	0	0	0

221 9 2

222 10 2

223 11 2 *

.1 .2 .1 *

3

4

0	10	.75	13	1.0	14	*			
0	0	90	120	0	0	105	135		
0	0	30	90	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

GOR/SM/76D-7

231 2 3
232 4 2
233 6 3 *
.1 .2 .1 *

4
22
0 10 .75 18 1.0 21 *
0 0 30 120 0 0 240 300
0 0 120 180 11520 14400 1440 2160
0 0 0 0 0 0 0 0

360 480 120 150

241 2 3

242 4 6

243 6 4

244 13 2

246 14 1

247 7 0

248 8 1 *

.2 .2 *

5

7

0 10 .75 18 1.0 21 *
0 0 30 120 0 0 240 300
0 0 120 180 1440 2160 480 600
0 0 0 0 0 0 0 0

360 480 120 150

151 2 3

152 4 3

153 6 3

154 13 2

155 14 1

156 7 0

157 8 2 *

.2 .1 *

6

1

0 7.5 1.0 7.5 *
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 8640 0
0 0 0 0

261 12 2 *

.1 .1 *

7

4

0 10 .75 13 1.0 14 *
0 0 30 120 0 0 180 240
0 0 120 180 1440 4320 720 2880
0 0 0 0 0 0 0 0 0 0 0 0

271 2 4

272 4 4

273 6 4
 274 / 0
 275 8 3 *
 .1 .1 *
 8
 10
 0 10 .75 13 1.0 14 *
 0 0 30 120 0 0 180 240
 0 0 120 180 2880 5768 360 480
 0 0 0 0 0 C 0 0 0 0 0 0

281 2 3
 282 4 3
 283 6 3
 284 7 0
 285 8 1 *
 .1 .1 *

OK
 NO
 5
 NO
 7.5 9.75 10.0 11.5 12.5 14.75 15.0 16.0
 1.0

OK
 *
 OK
 NO
 3
 9

TE1	1	1	1	1	1	1.5	2	1	1	1	1.5	1.5
TE2	1	1	1	1	1	1.5	2	1	1	1	1.5	1.5
PO	1	1	1	1	1	1.5	2	1	1	1	1.5	1.5
TT1	2	1	1	1	1.5	1	1.5	2	1	1	1	2
TT2	2	1	1	1	1.5	1	1.5	2	1	1	1	2
TT	2	1	1	1	1.5	1	1.5	2	1	1	1	2
TA1	3	1.5	1.5	1	1	1	1.5	1	1	1	1	1
TA2	3	1.5	1.5	1	1	1	1.5	1	1	1	1	1
OA	3	1.5	1.5	1	1	1	1.5	1	1	1	1	1
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

OK
 \$:DATA:08
 CONFIRMED.MISSION 8/16/76 9 30 4 241 242 243 244
 245 13 3 3 3 1 2 247 248 *
 CONFIRMED.MISSION 8/30/76 10 00 6 262 8 3 3 3 3 3 *
 CONFIRMED.MISSION 1/1/84 9 30 *

\$:DATA:09
 EO.SIMULATION 1/1/84 10 30 *
 \$:DATA:10
 NO NO NO NO NO NO NO NO NO
 \$:ENDJOB

ATTRIBUTES OF EACH KIND OF ACTIVITY

ACT	NAME	FROM	TO	DATE	TIME	TYPE	STATUS	OTHER
1	PAYROLL	1	1	1	1	1	1	1
2	EXPENSE	1	1	1	1	1	1	1
3	REVENUE	1	1	1	1	1	1	1
4	DEBIT	1	1	1	1	1	1	1
5	CREDIT	1	1	1	1	1	1	1
6	TRANSFER	1	1	1	1	1	1	1
7	ADJUST	1	1	1	1	1	1	1
8	REVERSAL	1	1	1	1	1	1	1
9	REVERSAL	1	1	1	1	1	1	1
10	REVERSAL	1	1	1	1	1	1	1
11	REVERSAL	1	1	1	1	1	1	1
12	REVERSAL	1	1	1	1	1	1	1
13	REVERSAL	1	1	1	1	1	1	1
14	REVERSAL	1	1	1	1	1	1	1

APPENDIX F

SAMPLE OUTPUT

ATTRIBUTES OF EACH KIND OF ACTIVITY

ACT	KNAM	KPOA	KTFC	KPAR	KFWL	KSWL	KTWL	KPHA	KLRL	KURL
1	PMSNPG	2	9	YES	1	2	3	0.	0.	0.
2	PMSNPA	2	9	YES	1	2	3	0.	0.	0.
3	CGDMSN	1	9	NO	1	2	3	10.00	10.00	20.00
4	CAMSN	1	9	NO	1	2	3	30.00	30.00	100.0
5	PSTMNG	3	9	NO	1	2	3	0.	0.	0.
6	PSTMNA	2	9	NO	1	2	3	0.	0.	0.
7	DATARD	1	9	NO	3	3	3	5.00	10.00	20.00
8	DATAAN	3	9	YES	3	1	2	0.	0.	0.
9	PMSNI	2	9	YES	1	1	1	0.	0.	0.
10	CHSNI	1	9	NO	1	1	1	10.00	10.00	15.00
11	PSTMNI	3	9	NO	1	2	2	0.	0.	0.
12	MSNNEL	1	3	NO	1	1	1	0.	0.	0.
13	ANTAPE	3	9	YES	3	1	2	10.00	10.00	20.00
14	ANLOGS	3	9	YES	3	1	2	0.	0.	0.

Section 1. Activity Attributes

#####	NUMBER	MISSION TIME	ACTIVITY SEQUENCE	AND RQMTS		
		PROB	TIME	CODE	ACTIVITY	WORKERS REQ
	8	0.	10.00	281	2	3
		.75	13.00	282	4	3
		1.00	14.00	283	6	3
				284	7	0
				285	8	1

ATTRIBUTES OF TEST PHASE

TNUM	TREQ	N.STRUC	N.CX.PROB
8	10	5	2

ATTRIBUTES OF EACH ACTIVITY IN STRUCTURE (TEST PHASE)

ATPC	ANOA	APOA	ATFC	AICT	ANWR	APAR	AFWL	ASWL	ATHL
281	2	2	9	0.	3	YES	1	2	3
282	4	1	9	0.	3	NO	1	2	3
283	6	2	9	0.	3	NO	1	2	3
284	7	1	9	0.	0	NO	3	3	3
285	8	3	9	0.	1	YES	3	1	2

ATPC	ACEL	APMA	ALRL	AURL	M.STRUCTURE	N.TEAM
281	1.00	0.	0.	0.	1	0
282	1.00	30.00	30.00	100.0	1	0
283	1.00	0.	0.	0.	1	0
284	1.00	5.00	10.00	20.00	1	0
285	1.00	0.	0.	0.	1	0

ATTRIBUTES OF EACH XFACTOR IN TCANCELLATION.PROB (TEST PHASE)

XUPL	M.TCANCELLATION.PROBABILITY
.10	1
.10	1

Section 2. Test Phase Attributes

ATTRIBUTES OF EACH WORKER

WORKER	WNAH	WLEV	WSTA	M.TEAM
1	TE1	1	3	0
2	TE2	1	3	0
3	PO	1	3	0
4	TT1	2	3	0
5	TT2	2	3	0
6	TT	2	3	0
7	TA1	3	3	0
8	TA2	3	3	0
9	OA	3	3	0

THE SKILL FACTOR OF EACH WORKER

ACT	WORKER								
	1	2	3	4	5	6	7	8	9
1	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.50
2	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.50
3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.50	1.50	1.50	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7	1.50	1.50	1.50	1.50	1.50	1.50	1.00	1.00	1.00
8	2.00	2.00	2.00	2.00	2.00	2.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.50
10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	1.50	1.50	1.50	2.00	2.00	2.00	1.00	1.00	1.00
14	1.50	1.50	1.50	2.00	2.00	2.00	1.00	1.00	1.00

Section 3. Worker Attributes

FINAL TEST FORCE REPORT
(FOR SIMULATION ENDING 10 18 1976)

MISSION STATISTICS
FOR TEST PHASE 1 ARE

NUMBER SCHEDULED-----	=	6	MISSIONS
NUMBER CANCELLED-----	=	0	MISSIONS
NUMBER REQUIRED-----	=	6	MISSIONS
EFFECTIVE PRODUCTIVITY-----	=	6.0000	MISSIONS
AVERAGE MISSION DURATION----	=	1.4120	DAYS
STD DEV-----	=	.6779	DAYS
TOTAL OF DURATION TIMES-----	=	8.4718	DAYS
SUM OF SQ OF DURATION TIMES-	=	14.72	DAYS
DATE COMPLETED-----	=	6 1 1976	

MISSION STATISTICS
FOR TEST PHASE 8 ARE

NUMBER SCHEDULED-----	=	13	MISSIONS
NUMBER CANCELLED-----	=	2	MISSIONS
NUMBER REQUIRED-----	=	10	MISSIONS
EFFECTIVE PRODUCTIVITY-----	=	10.3721	MISSIONS
AVERAGE MISSION DURATION----	=	7.7844	DAYS
STD DEV-----	=	2.1568	DAYS
TOTAL OF DURATION TIMES-----	=	85.6289	DAYS
SUM OF SQ OF DURATION TIMES-	=	717.75	DAYS
DATE COMPLETED-----	=	10 18 1976	

NOTE SINCE THE DURATION OF CANCELLED MISSIONS IS ZERO,
 STATISTICS ARE COMPUTED FOR COMPLETED MISSIONS ONLY.

Section 4. Mission Statistics

QUEUE REPORT

MAX LENGTH	AVG LENGTH	STD DEV	SUM
12.0 MSN	3.1854 MSN	3.3276 MSN	OF Q TIMES
			470.4 DAYS

HISTOGRAM OF QUEUE LENGTH

QUEUE LENGTH	STATE PERCENTAGE
0 MISSIONS	32.22 %
1 MISSIONS	15.92 %
2 MISSIONS	8.94 %
3 MISSIONS	3.98 %
4 MISSIONS	4.45 %
5 MISSIONS	4.77 %
6 MISSIONS	4.71 %
7 MISSIONS	8.89 %
8 MISSIONS	8.02 %
9 MISSIONS	4.81 %
10 MISSIONS	2.94 %
11 MISSIONS	.34 %
12 MISSIONS	.01 %

NOTE ALL PRIORITY 3 MISSIONS ARE IN THE QUEUE DURING OFF DUTY HOURS. PERCENTAGES ARE BASED ON A 24 HOUR DAY AND DUE TO ROUNDING MAY NOT SUM TO 100.

Section 5. Queue Report

HISTOGRAM OF WORKER IDLE TIME BY WORKER LEVEL

STATE TIME
(DAYS)

NUMBER OF IDLE WORKERS	LEVEL 1	LEVEL 2	LEVEL 3
0	135.9906	127.8727	132.5936
1	3.4398	2.6188	2.7880
2	.6038	1.9215	1.1470
3	7.6358	15.2571	11.1415
4	0.	0.	0.
TOTAL IDLE TIME	27.55	52.23	38.51

ANALYSIS OF WORKER STATUS

HOURS IN STATE
STATE PERCENTAGES

WORKER	IDLE	BUSY	BREAK	OFF DUTY	OVERTIME
TE1	224.30 HRS 6.33 %	536.42 HRS 15.14 %	140.28 HRS 3.96 %	2399.41 HRS 67.70 %	243.67 HRS 6.88 %
TE2	218.95 HRS 6.18 %	546.45 HRS 15.42 %	135.60 HRS 3.83 %	2289.53 HRS 64.60 %	353.55 HRS 9.98 %
PO	218.06 HRS 6.15 %	548.40 HRS 15.47 %	134.53 HRS 3.80 %	2272.20 HRS 64.11 %	370.88 HRS 10.46 %
TT1	418.86 HRS 11.82 %	336.08 HRS 9.48 %	146.06 HRS 4.12 %	2555.10 HRS 72.09 %	87.98 HRS 2.48 %
TT2	418.28 HRS 11.80 %	334.17 HRS 9.43 %	148.56 HRS 4.19 %	2557.67 HRS 72.17 %	85.41 HRS 2.41 %
TT	416.45 HRS 11.75 %	335.99 HRS 9.48 %	148.56 HRS 4.19 %	2556.09 HRS 72.12 %	86.99 HRS 2.45 %
TA1	307.34 HRS 8.67 %	441.65 HRS 12.46 %	152.01 HRS 4.29 %	2616.78 HRS 73.84 %	26.30 HRS .74 %
TA2	308.95 HRS 8.72 %	440.04 HRS 12.42 %	152.01 HRS 4.29 %	2613.27 HRS 73.74 %	29.81 HRS .84 %
OA	307.86 HRS 8.69 %	440.88 HRS 12.44 %	152.26 HRS 4.30 %	2619.81 HRS 73.92 %	23.27 HRS .66 %

Section 6. Worker Statistics

VITA

James Paul Eggers was born on 12 March 1946 in Centralia, Illinois. He graduated from Centralia High School in 1964 and attended the University of Illinois from which he received the degree of Bachelor of Science in Electrical Engineering in February 1969. Upon graduation he received a commission in the USAF through the ROTC program. He was employed as an engineer for the Illinois Bell Telephone Company until called to active duty in February 1969.

After completing pilot training at Laredo AFB, Texas, he served 18 months in Southeast Asia as a member of the 361st Tactical Electronic Warfare Squadron. He served as an instructor pilot and flight examiner in the 4410th Special Operations Group at England AFB, Louisiana, and as a training monitor and air operation officer in the 1st Special Operations Wing at Hurlburt Field, Florida. He entered the Air Force Institute of Technology in May 1975 to earn the degree of Master of Science in Systems Analysis.

Mailing address: 219 S. Cherry Street
Centralia, Illinois 62801

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GOR/SM/76D-7	2. GOVT ACCESSION NO:	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) JOINT TEST FORCE SIMULATION		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) James P. Eggers Captain		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN)		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS 3246 Test Wing/TG Eglin AFB, Florida 32542		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December, 1976
		13. NUMBER OF PAGES 180
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 <i>Jerald F. Guess</i> JERALD F. GUESS, Captain USAF Director of Information		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Manpower Requirements Simulation Test Methods		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This study presents the background, development, and use of a generalized event-oriented simulation model of work activities. The particular question addressed in this research effort was: Can an event-oriented simulation model be developed which will be useful in estimating work completion dates and personnel requirements using the information presented in a work directive? The program is written in SIMSCRIPT II.5 and is adaptable to a particular joint test force by using		

UNCLASSIFIED

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

free-format data inputs.

Any test force which is responsible for the performance of activities may be studied using this model. Any number of activities, missions, test phases, workers, and levels of workers may be defined through the input data. Detailed input instructions are provided to tailor the model to the analyst's requirements. Numerous options are available to include: worker skill factors, mission reduction or cancellation probabilities, activity performance capability with reduced personnel, reports, parametric studies, and replications of the entire data package.

The function of each routine and event is discussed and a complete program listing is appended. A sample study is provided to illustrate many of the significant features.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)