A TRANSIENT STATE MAINTENANCE REQUIREMENTS PLANNING MODEL

by

Kenneth W. Brammer

A Master's Thesis submitted to the faculty of the

Department of Industrial Engineering and Operations Research

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Industrial Engineering and Operations Research

Approved:

Charles J. Malmborg, Chairman

Marvin H. Agee

Wolter J. Fabrycky

February, 1985

Blacksburg, Virginia

A TRANSIENT STATE MAINTENANCE REQUIREMENTS PLANNING MODEL

by

Kenneth W. Brammer

Committee Chairman: Charles J. Malmborg
Industrial Engineering and Operations Research

(ABSTRACT)

A model is developed for use by logistics planners in determining period by period maintenance requirements for repairable item populations. The model generates maintenance requirements with respect to manpower and facility requirements and spare parts requirements. The model is capable of capturing nonsteady-state failure behavior of populations of repairable items. Each item within the population is broken down into one or more families of components and subcomponents that can have different failure/repair characteristics. Probability of component failure may be generally distributed. A specific data requirement for the model is established. The model is structured to allow the user to conduct various "what if gaming" through an iterative procedure on a personal computer.

This research document includes a literature review that establishes a history of logistics modeling. The literature review provides impetus to the proposed research by defining a need for a transient state model for maintenance requirements planning. The model is validated by a case study involving the generation of maintenance requirements for a case population.

ACKNOWLEDGMENTS

The author wishes to express his appreciation to all of those who have had a part in the research effort leading to the completion of this thesis. Five individuals deserve special recognition.

Dr. Marvin Agee and Dr. Wolter Fabrycky, members of the author's graduate committee, provided support, suggestions, and encouragement during the development and completion of this research. Gratitude is expressed for their sincere interest in the author's research project.

Dr. Franz Frisch deserves special recognition for the mortality concepts that he developed. These concepts served as the seed from which this research project grew. His support of the research effort has been sincerely appreciated.

A special thank you is reserved for Dr. Charles Malmborg, the author's advisor, who provided valuable assistance in concept development and then enthusiastically supported the entire research effort. Dr. Malmborg's expert guidance and optimistic schedules interest insured the expedient completion of this sometimes unwieldy research project.

The biggest thank you goes to my loving wife, , whose patience, understanding, and help contributed greatly to the success of this research effort.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

CHAPTER I

INTRODUCTION

The planning and management of logistics requirements for repairable item populations has become a monumental task as the size, complexity, and expense of such populations has increased. A repairable item population consists of units that, up to a point, are more economical to repair than to replace. Logistics for both military and civilian purposes has been referred to as "the study and application of decision making techniques intended to achieve efficient allocation and use of resources in satisfying stated objectives of an organization" [6]. As can be seen from this definition, a logistics planner has the responsibility for making decisions that can have impact on budgets and on the state of readiness of repairable item populations. To illustrate the significance of the logistics responsibility consider that in 1968, 52% of the total investment in spare parts by the Air Force was in repairable items, which at the time amounted to about ten billion dollars [31]. By 1975, the percentage had risen to about 65% [23]. Logistics decisions regarding inventory management, procurement, maintenance, and distribution must be made continuously in order to support any system containing repairable items.

In order to make effective logistics decisions in the face of complex logistics systems, planners need analysis tools that can help determine the maintenance requirements of recoverable item populations. The logistics planner needs a tool that provides him with these maintenance requirements prior to the procurement and deployment of recoverable item populations. Some data, provided by Air Force sources, suggest that "about 80% of the life time logistics costs of an aircraft is determined by decisions made before the aircraft begins to fly. These decisions relate to design tradeoffs of performance against reliability and maintainability, with resulting impact on maintenance manning, initial spare parts procurement, location of repair, and ground support equipment investment" [7].

In order to forecast the true maintenance requirements of a repairable item population, the maintenance requirements of the population during its entire life cycle must be forecast. This life cycle can include several stages that vary from population to population. The failure characteristics demonstrated by each member of the population may vary from stage to stage. If the failure characteristics of a population vary significantly from period to period then the total population maintenance requirements can vary significantly from one time period to

another.

Stage dependent failure characteristics are observed among many populations. An example would be populations that experience a "burn in" stage where the rate of failure for the initial period of usage is greater than for subsequent periods. Later on in the life cycle of the population, the failure rate again accelerates as the population ages and begins to "wear out". Other populations may exhibit a continuously accelerating failure rate over their life cycles. The point is that in many cases it is very difficult to establish an average failure rate that will generate accurate maintenance requirements on a period by period basis.

1.1. STATEMENT OF THE PROBLEM.

There are two basic problems with logistics models intended to serve as planning and analysis tools. First, logistics models do not evaluate the complete maintenance requirements of repairable item populations over the entire life cycle of the population items. Models have tended to focus on one particular maintenance requirement or another, by making simplifying assumptions about the other related maintenance requirements. One of the major simplifications made has been the assumption that systems operate in steady-state all of the time and that logistics requirements may be determined from these steady-state

conditions. Under this assumption, an average rate of failure is applied to the total population life cycle in order to generate maintenance requirements. Some populations never attain a steady-state of operation. This could be due to characteristics inherent in the components that comprise each member of the population, or due to external stimuli such as war conditions, temperature, moisture, or dust.

Logisticians such as Geisler and Murrie [7] and Kline [18] have called for the development of models that are useful during all phases of a population life cycle and are robust in that they address a broad range of maintenance requirements. These logisticians and others also allude to the second major problem with logistics models that exists today, and continues to exist after the completion of this research. This second problem pertains to the data requirements typically imposed by logistics models. In particular, to evaluate the multitude of maintenance requirements generated by a repairable item population, logistics models have generated extensive data requirements for failure and repair information that is not readily available.

1.2 OBJECTIVE.

The intent of this research is to develop the methodology for a logistics model that determines and

evaluates maintenance requirements of repairable item populations. The model is capable of generating maintenance requirements for the total life cycle, whether the repairable items are operating under steady-state or nonsteady-state conditions. The model allows the logistics planner to develop and evaluate the desired logistics system in the face of availability constraints, budget constraints, space constraints, procurement constraints, and personnel constraints. Development of the model has produced a data requirement. This data requirement is expounded upon. The results produced by the model emphasize what type of data base the logistician must have in order to field and operate a population at a reasonable cost and maintain the required degree of readiness.

The objectives of the research can thus be summarized within three major categories. They are as follows:

1. to develop a transient state model for describing the failure behavior of a population of repairable items that has the ability to
    a. capture nonsteady-state behavior
    b. model the failure behavior of a population of items where each item is comprised of one or more families of components and subcomponents that possess different failure/repair characteristics

c. model the flow of the population into repair facilities

d. evaluate system availability over time

e. specify the logistical consequences of alternative item quality (quality is described by the failure behavior of items) and item procurement policies

2. to generate maintenance requirements on a period by period basis with respect to

a. manpower and facility requirements

b. spare parts requirements

3. to establish the practicality of the model by

a. providing the flexibility to conduct various "what if gaming" that is of interest to logisticians

b. clearly establishing the specific data requirements of the model

c. implementation of the model for a realistic case study.

1.3 OVERVIEW OF THE RESEARCH.

The remaining chapters of this research document contain a review of literature pertinent to logistics modeling, a discussion of the computational basis of the model developed, a discussion of the implementation of the model developed, case study documentation, and conclusions

and recommendations. A partial program listing is attached as an appendix. A list of referenced authors is included as a bibliography.

After a brief introduction, the literature review discusses logistics models by functional classification. An attempt has been made to discuss some of the earlier, prototype models in each classification and then to move forward in time, illustrating extensions made of prototype models and also new approaches that have been developed. The first classification described is "The Repair/Replace Decision". This classification includes models whose basic function is to determine when it is advantageous to repair units upon failure and when is is advantageous to replace a failed unit with a new unit. The second classification is "Determining Inventory Requirements". This classification discusses logistics models whose primary function is to evaluate either spares or spare part inventories required to support repairable item populations. The concept of spares versus spare parts is discussed in the literature review. The third classification is "General Logistics Models" that includes a discussion of logistics models that attempt to address a broad range of logistics problems rather than concentrate on specific areas. The fourth classification used is "Queueing Theory and Logistics". These models encompass a family of logistics models that

include queueing theory to determine the effects of finite capacity repair facilities on other logistics requirements. The literature review closes with a general discussion of the state of the art in logistics modeling and a summary.

In Chapter III, the discussion of the computational basis of the model opens with an introduction. This is followed by definitions of key terms used in describing the model. The development of the model is described with special emphasis placed on two major concepts applied during the research. The first of these concepts, the multi-stream concept, is basically the same as the nested Markov Chain concept proposed by Frisch [5]. The second concept involves the use of a typical population member bill of materials to establish a maintenance tree structure that is used to track specific component maintenance requirements. The description of the computational basis of the model continues with discussions of assumptions, parameters, decision variables, calculations, and measures of performance used. The flexibility of the model is exhibited through a discussion of possible uses of the model. Chapter III closes with information on limitations of the modeling approach and a summary.

Chapter IV opens with an introduction and continues with a profile of the data requirement generated by the model. The remainder of the chapter is concerned with

possible applications of the model in lieu of certain model limitations. Model limitations result from assumptions made and from physical limitations of computing facilities utilized.

The model is validated through a case study which is discussed in Chapter V. Due to the scarcity of failure and repair data, much of the data used for the case study is fabricated but is, in most cases, representative of a real population of Side Loadable Warping Tugs employed by the U.S. Navy. Also included in Chapter V are scenario descriptions for two additional case studies, designed to further exhibit the flexibility and "what if" gaming capabilities of the model.

The final chapter of this thesis contains conclusions resulting from the research conducted. The chapter closes with a list of recommended computer enhancements and research extensions relative to transient state modeling of maintenance requirements. An operator's guide and a listing of the computer program are included as appendices.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

In a survey of maintenance models, Pierskalla and Voelker [26], write that, "for two decades, there has been a large and continuing interest in the study of maintenance models for items with stochastic failure." This observation is substantiated not only in Pierskalla and Voelker's survey, but also in surveys of maintenance models conducted by Barlow and Proschan [2], McCall [21], Sherif and Smith [32], and Nahmias [23]. Additional sources such as Department of Defense bibliographies, an assessment of military logistic models by Geisler and Murrie [7], and reviews of related topics such as availability by Lie, Hwang, and Tillman [19] verify that papers written since the late 1950's concerning the maintenance of stochastically failing systems number well into the hundreds and probably over one thousand.

The maintenance of stochastically failing items can involve decisions regarding the repair or replacement of items, the determination of spare or spare part inventory levels, inspection and preventive maintenance policy, and the determination of repair facility characteristics. These decisions can be made under conditions of certainty

where failure and repair rate probability distributions are known and related costs also are known. Uncertain conditions exist where some aspect of the maintenance model is not known. Besides unknown failure and repair distributions and unknown costs, there can be uncertainty associated with the determination of the state of a system through inspection or uncertainty regarding the state to which a unit is returned by repair. Models developed assuming certainty in the knowledge of model parameters are reviewed in this chapter.

Some of the models reviewed in this chapter have concentrated on a particular facet of the maintenance process and others have chosen to encompass many maintenance decisions. This literature review will move from specific maintenance models to the more generalized models that consider a broader scope of the logistics process.

2.2 THE REPAIR/REPLACE DECISION

Some of the basic decisions involved in the maintenance of any item are when to repair the item and when to replace the item. These decisions may be made at discrete time intervals or can be made on a continuous basis, that is, at any time on a continuous time axis.

Repair or replace decisions are important because of the impact they have on the entire logistics system. Kline

[18] writes that the repair or replace decision "impacts support resources (technician numbers, skill levels and training, spares and repair parts, facilities, test and support equipment, and maintenance information), as well as the specific maintenance actions to be taken and the levels at which repairs are made. The decision also affects such system and design attributes as safety, reliability, accessibility, and human factors." Repair or replace decisions are also going to have a specific impact on the life cycle cost of the system items themselves.

2.2.1 Basic Discrete Time Markov Process Model. Pierskalla and Voelker [26] point out that most discrete time maintenance models are Markov decision models, in which the state of the system is described by the level of deterioration or the number of spare units available in inventory. They also go on to describe a model developed by Derman [3] as being the basic model for Markov process repair/replace models.

Derman's original model considers the inspection of a unit at discrete time intervals with the unit being observed to be in a particular state of deterioration. Each time interval represents a step in a Markov chain. The user of the model has two options: take no action or replace the unit. If no action is taken there are corresponding probabilities of transitioning from the

presently occupied state to any state of further deterioration in one period. If the unit is replaced, then the unit immediately returns to state 0 and the transition probabilities for state 0 again apply. The only way to return to state 0 is through replacement, otherwise the unit continues to deteriorate toward complete failure. The transition probabilities that make up the deterioration process Markov chain are steady-state probabilities.

2.2.2 Discrete Time Semi-Markov Processes.

Generalizations of the basic Derman model have accounted for a great number of subsequent repair/replace maintenance models. One reason that Derman's model was generalized was that the model does not illustrate that the probability of moving into a less desirable state changes with time. The steady-state assumption and the use of a single Markov chain prevent this consideration. Kao [17] and Hayre [13] utilized a discrete time semi-Markov process to overcome this problem. The semi-Markov process is described in Barlow and Proschan [2, pages 131-139] and Ross [28, chapter 7].

In a semi-Markov process, transitions from state to state are made in accordance with a Markov chain similar to the chain used by Derman. The time spent in each state before a transition occurs is a random variable dependent on the state presently occupied. By using these random

variables in conjunction with the Markov transition probabilities, Kao and Hayre determine an expected time of occupancy for each state. By associating the expected occupancy time with known costs of occupancy, a cost figure analagous to a repair cost is developed. Comparison of this cost with a known cost to replace a unit allows the user of these semi-Markov models to determine at what stage it becomes more advantageous to replace instead of repair. Both authors assume that the failure stage is inevitable.

2.2.3 Continuous Time Models. Continuous time repair or replace models assume that the actions of the decision maker may potentially take place anywhere on the continuous time axis. Some continuous time models permit the maintenance activity to occur as a continuous stream. In this type of situation, the decision maker must optimize over the function m(t), where m(t) is the maintenance expenditure rate at time t. A review of the literature reveals that these types of models, called control theory models by Pierskalla and Voelker [26], appear to have been developed primarily for commercial use because parameters such as production revenues and sale date are considered. The control theory models could be generalized to apply to commercial, government, and military situations. For example, the sale date parameter is analagous to retirement age. Instead of maximizing production revenue, total life

cycle costs could be minimized.

A model developed by Sethi [30] is an example of a control theory model. Sethi's model allows the consideration of the machine replacement problem in a changing technological environment where replacement machines may cost different amounts and possess different production and maintenance characteristics. Sethi considers an infinite chain of machines, where each link in the chain represents a uniform length replacement cycle with identical preventive maintenance. The objective is to maximize the present value of all future net profits from the chain of machines. In order to obtain the present value, the following continuous functions are employed by the model: P(t), the production rate at time t; E(t), the ordinary maintenance; S(t), the salvage value; m(t), the preventive maintenance rate; C(T), the cost of a machine purchased at time T; and d(t), the obsolescence function at time t in terms of dollars subtracted from S(t). Decision variables are the length of the replacement cycle, T, and the schedule of preventive maintenance, m(t). When the preventive maintenance rate, m(t), increases, Sethi assumes that P(t) increases, E(t) decreases, and S(t) increases. The problem is solved using a nonlinear programming method.

A second category of continuous repair or replace models, as defined by Pierskalla and Voelker [26], are age

replacement models. These types of models introduce the concept of age dependent operating costs to the traditional repair or replace models. An important model of this type was developed by Barlow and Hunter [1]. Through this model, they introduced the concept of minimal repair, which is the idea that the failure of a single component of a system does not call for system replacement, but rather a repair that does not affect the system's failure rate.

Barlow and Hunter assume that the system failure rate corresponds to a failure distribution with a density parameter. It is further assumed that replacement or overhaul occurs at times T, 2T, 3T,.... The objective of the model is to select T so as to minimize

$$C(T) = \lim_{t \to \infty} \frac{c_1 EN_1(t) + c_2 EN_2(t)}{t}$$

where c_1 = cost of minimal repair,
c_2 = cost of replacement,
$EN_1(t)$ = expected number of failures in [0,t],
$EN_2(t)$ = expected number of replacements in [0,t],
$C(T)$ = operating cost.

2.3 DETERMINING INVENTORY REQUIREMENTS

Nahmias [23] states that most classical inventory models assume that units are completely consumable, i.e. not repairable. Logisticians cannot afford to take this point of view. In a paper published in 1967, Schrady [29] estimated that the Navy's investment in repairable units was approximately 58% of their total dollar investment in

inventory. It can only be assumed that this percentage has remained steady or increased as inflation and other costs have boosted the price of new units. Nahmias points out that generally it is the high value items that are more economical to repair than to replace. One result of the large number of repairable items in use is that there are situations, especially in the military, where being able to determine inventory requirements for spares and spare parts to repair these items can be of significant economical advantage.

As an example of the cost effectiveness of managing inventory levels, consider the military's stockage of spare parts for aircraft repair. Since aircraft are very expensive it follows that repair parts would be expensive also. If inventory requirements for the parts are not established through some analytical means, it is conceivable that logisticians may overstock repair parts, resulting in a large and unnecessary investment in parts and storage space. If, on the other hand, parts were understocked and shortages occurred, expensive aircraft and skilled pilots would be grounded and unable to carry out possible crucial missions.

In addition to economic reasons, an appropriate inventory level must be maintained to meet availability constraints often imposed in both commercial and military

settings. When understocking occurs, key equipment could be down during either crucial production periods or during a military operation. Nahmias' review [23] reveals that many maintenance models have been developed exclusively to provide spare and spare parts inventory levels and other models have included inventory level determination as one of several maintenance outputs.

Nahmias evaluates two basic types of inventory models, continuous review and periodic review models. The continuous review models assume that demands for spare parts can occur at any instant in time. As a result of this approach, Nahmias states that the majority of the continuous review models use a one for one ordering policy, i.e., when a unit is shipped to the repair facility an order is simultaneously placed for a spare unit or a particular spare part. Periodic review models assume that inventory levels are reviewed periodically and that total demand in a period is a random variable with some specified distribution function and density.

Before going further, a brief discussion of terminology is called for. The word "spare" will be used to denote the number of additional, complete end items that will be kept in inventory. Generally, a spare is ready for use immediately upon failure of an identical unit. "Spare parts" denote additional components kept in inventory.

These components are used to repair end items that require repair.

Level of "indenture", as described by Nahmias, refers to the relationship between an end item and its components or modules. For example, an end item could consist of several modules each of which has its own spare parts requirements. The end item represents a level of indenture, the modules a second level of indenture, and the spare parts for the modules a third level of indenture.

The term "echelon" will apply to the levels of repair available for use by repairable items. As an example, consider an end item that could be repaired at a base facility if one type of failure occurred, or would require repair at a centrally located depot if another type of failure occurred. In this case there are two echelons or levels of repair.

2.3.1 Repairable Spares Requirements, Single Indenture Analysis, Continuous Review. An important inventory model, that according to Nahmias has served as a basis for some of the small number of multi-echelon inventory models that have actually been implemented, was the METRIC model developed by the RAND Corporation and reported by Sherbrooke [31]. METRIC is a two echelon, single indenture model that allows the user to determine inventory requirements for spare end items. METRIC seeks

to minimize the number of backorders of end items within a given budget constraint. A stationary compound Poisson probability distribution describes the demand for spares. Muckstadt [22] notes that a predictable nonstationary distribution for demand could be used.

Input data required by the model includes average base and depot repair times for each item, unit costs, certain probability distribution parameters, Not Repairable This Station (NRTS) rates, and average order and ship times. Additional assumptions include: there is no lateral supply between bases, there are no condemnations (all failed parts are repaired), repair times are statistically independent, there are infinite queues at repair channels, repair level depends only on repair complexity, items are equally essential, and a failure of one type is statistically independent of another type.

Porteus and Lansdowne [27] consider a single indenture model in which they investigate stochastic failures that can be one of several types. The particular type of failure is governed by a probability distribution. Failures are assumed to occur according to a simple Poisson process. The system is assumed to be operating in steady-state. When an end item fails, it is assumed to be replaced by a spare end item if one is available, otherwise a shortage occurs. Expected repair times are considered as

decision variables. Different expected repair times are associated with different types of failures.

Through an optimization procedure that utilizes a Lagrangian approach, Porteus and Lansdowne minimize expected weighted shortages of all items at all locations given a budget constraint. The weights assigned to shortages are based on item essentiality. The costs constrained by the budget include the cost of implementing an expected repair time and the cost of stocking spares.

2.3.2 Spare Parts Requirements, Multi Indenture Analysis, Continuous Review. As stated in Nahmias, the METRIC model itself has not been implemented. However, extensions of the model were in use by the Air Force in 1981. One primary enhancement to the model was the addition of multi-indenture capabilities.

2.3.2.1 Steady state evaluation. As evidenced by this literature review and as noted by reviewers such as Geisler and Murrie [7], the great majority of existing maintenance models assume that the demand for spares or spare parts is in steady-state. Murrie and Geisler and Kline [18] note that this assumption is also true of extensions of the METRIC model that the Air Force has implemented.

One of the extensions of METRIC, that was in use by the Air Force in 1981, is MODMETRIC reported by Muckstadt

[22]. MODMETRIC extends METRIC to include a multi-indenture analysis. Muckstadt considers two levels of indenture, the end item and its components. The model may be extended to consider additional levels of indenture. Muckstadt considers the fact that backorders of spare parts for components do not necessarily carry the same penalty as backorders for components themselves. Otherwise, MODMETRIC makes the same assumptions and produces similar output as METRIC.

Graves and Keilson [8] developed a single echelon, two indenture inventory model. Their model differs from the METRIC concept in that they consider a finite repair facility and use system availability and system persistence as performance measures. System persistence is measured by the time until the next system failure. The two indentures evaluated are the end items and critical modules.

Graves and Keilson assume that each end item requires exactly one unit of each module to be operable, complete cannibalism of the inventories of repairable modules is allowed, the failure and repair of distinct types of modules are independent, and each random variable representing the state of the modules can be modeled as a stationary birth-death process. Also, it is assumed that the mean time between system failures follows an exponential distribution. It is assumed that the end items

operate as a system (like a squadron of aircraft) and that system failures are rare.

Besides determining appropriate inventory stockage levels for repairable modules, Graves and Keilson illustrate economic benefits that would occur with improved repair capability or improved module reliability. The economic benefits are determined through the use of cost functions specific to the individual repairable modules.

It should be noted that this model, along with the other inventory models discussed thus far, considers a series relationship between failures of lower levels of indenture versus the failure of the end item, i.e. when a component fails, the entire end item is considered to be in failure. When a series relationship between components exists, failures of components that do not result in end item failure do not occur. An alternate arrangement, not mentioned thus far, is a parallel relationship between failures of lower levels of indenture versus the failure of the end item. In this type of situation there may be redundant component parts where the failure of one component does not cause the failure of the end item.

2.3.2.2 Transient state evaluation. Hillestad [14] presents a model, DYNA-METRIC that will evaluate transient state demand for aircraft spare parts. This model also is an extension of the METRIC model. Hillestad's model is a

multi-echelon, multi-indenture model that can evaluate series and parallel systems. The rate of demand for spare parts is Poisson. In place of a constant repair time, the model uses a repair function which is the probability that a component entering repair at time s is still in repair at time t. Hillestad defines the daily demand rate d(t) under transient conditions to be:

d(t) = (failures per flying hour)
x (flying hours/sortie at time t) x (number of sorties per day per aircraft at time t)
x (number of aircraft at time t)
x (quantity of the component on the aircraft)
x (percentage of aircraft with the component)

Under Hillestad's definition of the transient state, the failures per flying hour remains constant. DYNA-METRIC's measure of performance is aircraft availability. An infinite queue at the repair channels is assumed.

2.3.3 Periodic Review. The prototype periodic review model for a repairable item inventory is due to Phelps [25]. In that model, each stage of the review has two state variables: the amount of serviceable stock on hand and the amount of repairable stock on hand. Serviceable stock includes items that have been repaired or procured from outside sources and repairable stock includes items that are yet to be repaired and returned to service. An expected demand for spares per period must be provided by

the user of the model.

The user of the model utilizes two decision variables; the number of serviceables and the number of repairables on hand after all decisions have been made. The decisions that must be made at each stage require the specification of: the number of items to procure from outside, the number of repairables to repair, and the number of repairables and/or serviceables to scrap for salvage. Utilizing the results of these decisions, the model maximizes the total expected return in a single period.

2.4 GENERAL LOGISTICS MODELS

There are a few models in existence that provide the logistician with information on a broad scope of logistics problems, from repair or replace decisions to spare parts requirements. Following are descriptions of two of these models.

2.4.1 Level of Repair Model, Constant Failure Rate.

A group of extended repair or replace models that appear to be more general in nature are the level of repair models used by the military. In 1981, Kline [18] wrote that each of the three major branches of the military had their own level of repair models in operation. According to Kline, the level of repair models base their decisions at least partly on costs just as the other repair or replace models do. The level of repair models are more general than other

repair or replace models because they also determine where a repair should be made, where spare parts should be located, how many spare parts should be at each location, and the costs associated with these policies.

A paper by Jensen [16] illustrates some of the facets involved with optimum repair level analysis. Jensen evaluates equipment that can be broken down into repairable modules. His analysis is a multi-indenture, two echelon analysis that combines a family of more specific models into one large level of repair model to determine the optimum maintenance and logistics policies for the parts, modules, and higher modular assemblies in the face of an availability constraint. Optimum maintenance policy is defined with respect to when to repair and when to replace. Optimum logistics policy is defined with respect to where to repair and where and in what quantities to keep spare parts. Jensen's model also determines which parts are line replaceable. Line replaceable means that the end item or higher modular assembly does not have to be sent to a repair facility if a line replaceable part fails.

Jensen develops a method for classifying maintenance and logistics policies. There are three possible logistics policies: spare at the organization level, spare at the field level, or spare at the depot level. There are four possible maintenance policies: repair at either the

organization, field, or depot level or discard. Life cycle costs associated with these policies are determined by assigning a particular cost to each possible alternative for each element that makes up the life cycle costs. The optimal maintenance and logistics policy, taking into account the availability constraint, is determined through an enumerative scheme where each possible policy alternative is evaluated.

Some of the assumptions utilized in this level of repair model include: components of the equipment fail independently and randomly with a constant failure rate over time, the number of failures in a cycle time has a Poisson distribution, and the modules and their parts are in a series arrangement. Spares levels are determined using a Lagrangian approach.

2.4.2 A Stochastic Model. Frisch [5] provides the framework for the development of a periodic review model that not only determines inventory requirements but would allow the logistician to make repair and replace decisions. The construction of Frisch's conceptual model is unique in that it models the units of a repairable item population as a series of nested Markov Chains. If a population is made up of cohorts or units that are of different ages because of procurement practices or for other reasons, then each age group has its own failure characteristics at a given

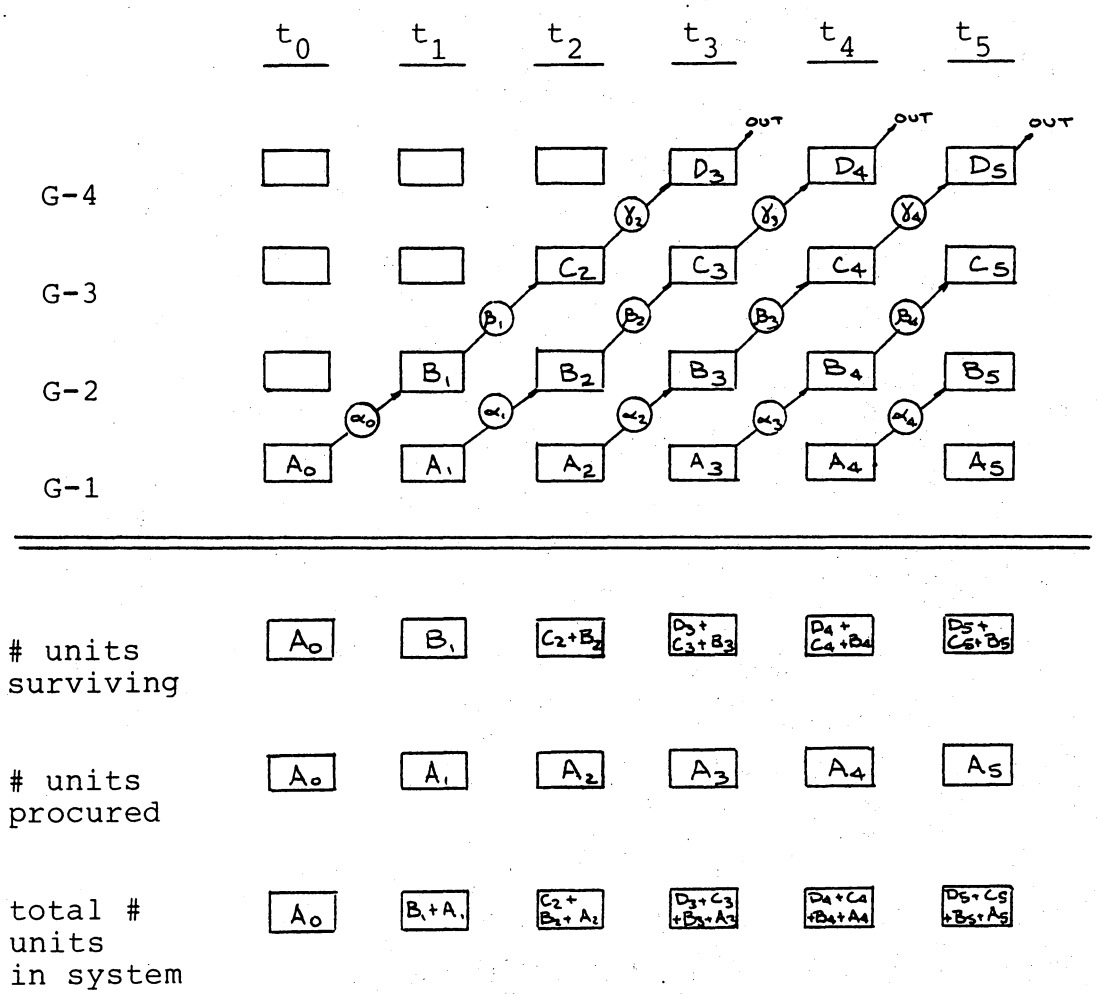
point in time. Frisch accounts for the different failure characteristics by modeling each group of age cohorts as a separate stream or Markov Chain, thus resulting in the nested Markov Chains. See Figure 2.1 for an illustration of the nested Markov Chain concept.

The Markov Chains are driven by the mortality rate or the failure rate of the units. The Markov process used by Frisch is stationary in time. Barlow and Proschan [2, page 122] describe a stationary Markov process as follows:

> In a Markov Chain, a conditional probability (called a transition probability) of a transition at time n for each pair i,j = 0,1,2,...,m from state i to state j must be given. This probability is denoted by
>
> $$p_{ij}^{n,n+1} = P[X(n+1) = j \mid X(n)=i].$$
>
> The Markov process is stationary if the transition probability functions depend only on the time difference, that is,
>
> $$p_{ij}^{n,n+1} = p_{ij}^{0,1} = p_{ij}$$

The Markov process utilized by Frisch is also a discrete time process in that the chain is indexed by time.

A discrete time, stationary Markov process, such as the one employed by Frisch, transitions until eventually a steady-state is reached. The Frisch model can evaluate the transition phase leading to steady-state conditions. This transition phase is analagous to the build up phase experienced between the time a population is first fielded and the time that full population size is attained. Frisch does assume, however, that his population operates in



$A_nB_nC_nD_n$ = stream of units t_n = time period

$\alpha_n, \beta_n, \gamma_n$ = survival rates G = age group

A_n, B_n, C_n, D_n = # units in each age group

A_n units are procured at the end of each time period t.

FIGURE 2.1. NESTED MARKOV CHAIN CONCEPT
MULTI-STREAM MODEL

steady-state the majority of the time. Frisch also notes that within the Markov chains, number of operating hours or some similar parameter may be used in lieu of age when describing the state of the units.

By applying the failure probabilities to the respective number of cohorts in a particular age group, the expected requirement for spares may be determined for the period of time that each step of the Markov process represents. The model would have to either be extended to multiple indentures or applied to spare parts individually in order to determine "spare parts" requirements as Frisch alludes to in the title of the paper. By computing spares requirements, the model is providing the logistician with a forecast that can be used for maintaining spares or spare parts inventories and a forecast that can be used to determine when it will be economically advantageous to replace rather than repair.

The Frisch model is not an optimization process but rather a "what if" model that allows the determination of the effects of two decision variables on the logistics system. The two decision variables are quality and mode of fielding.

Frisch notes that quality is hard to quantify. As a means of measuring quality, he suggests that performance (P) and quality (Q) can be measured with the same

dimension. Performance is defined as the output of a machine at its design point. Quality is defined by the delta (Δ) or loss of output of a machine during its operation time, so $P - \Delta_t = Q_t$. The critical quality (Q_c) is defined as the limit to which the performance can deteriorate without rendering the machine useless for its mission. Frisch implies with this definition that only if two machines have identical design parameters can they be qualitatively compared. Thus, a qualitative comparison between functionally similar but not identical units can be accomplished by comparing the failure rates and the resulting repair costs of the respective units.

The mode of fielding has to do with the method with which a population is built up. Different modes of fielding could be as follows: procure all units in the first year; procure an equal amount of units each year (plus replacements for failed units) until the population reaches full size; procure a decreasing amount of units each year (plus replacements for failed units) until the population reaches full size; and so forth.

Frisch evaluated the effects of the two decision variables on the logistics system. His findings are summarized in Table 2.1. Findings 2, 3, and 4 are illustrated by the following example. Assume that the task of the logistician in this example is to make an initial

TABLE 2.1. EFFECTS OF DECISION VARIABLES, FRISCH MODEL

DECISION	RESULT
1) no decision allowed	Logistics requirements have been determined, no total life cycle cost minimization is possible.
2) mode of fielding = block procurement	Support requirements will fluctuate.
3) block procurement and high quality	The span between support requirement fluctuations will increase with quality and the rate at which support requirements reach steady-state will decrease as quality increases.
4) block procurement and low quality	The span between support requirement fluctuations will decrease as quality decreases, however, support requirements will reach steady-state more rapidly as quality decreases.
5) quality and mode of fielding other than block procurement	System support requirements decrease as quality increases. Also, it is possible to build up systems of any desired quality with steady-state support requirements from the beginning.
6) mode of fielding and quality	Both decision variables affect total systems life cycle cost.

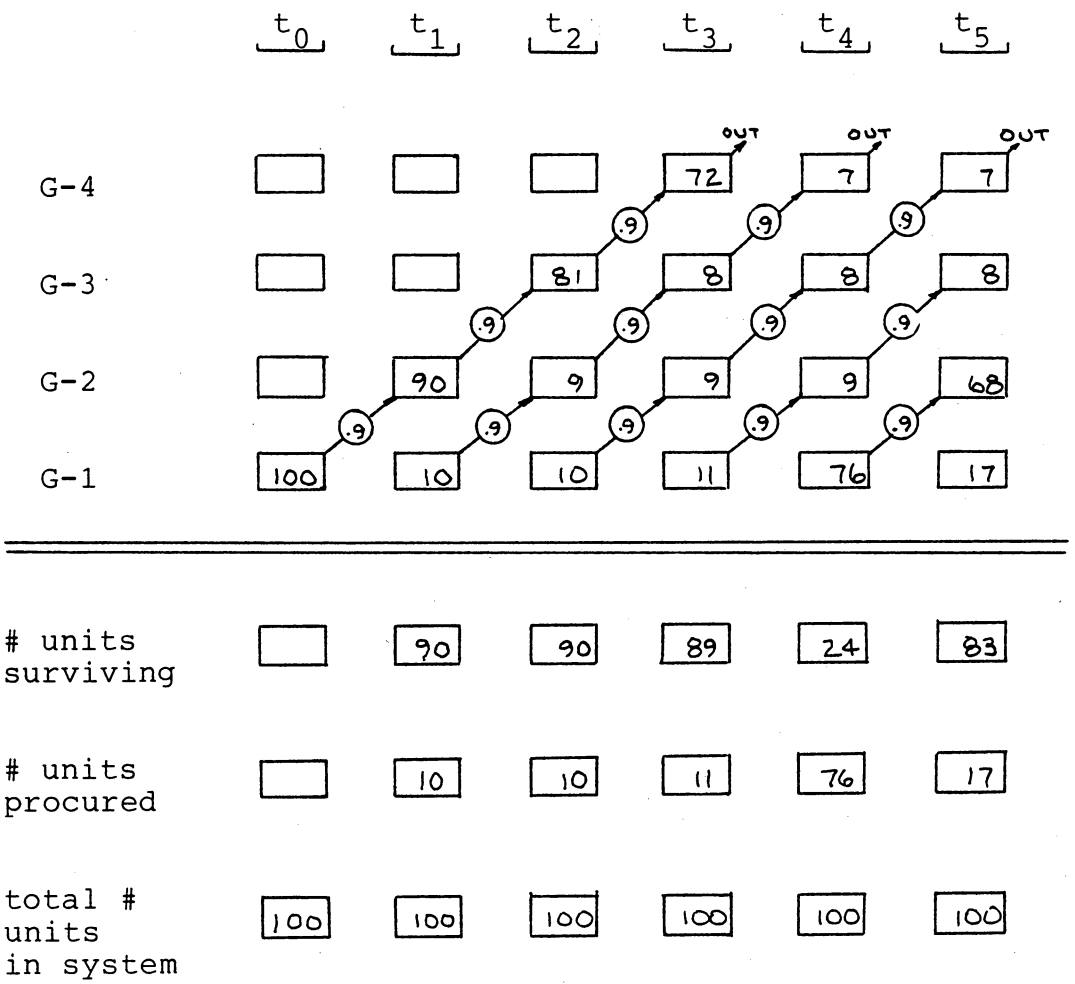
block procurement of 100 units and then maintain a population size of 100 from that time forward. Procurements may be made at the start of each year. The population has an annual survival rate of .9 and if a unit lasts for four years, it is retired from service at that time. See Figure 2.2 for a diagram of the nested Markov chains that represent this population.

The logistician in the example looks ahead 20 years and calculates the amount of units that will have to be procured each year in order to replace failed units and maintain the population size of 100. The yearly procurements are as follows:

Year	1	2	3	4	5	6	7	8	9	10
Units	100	10	10	11	76	17	17	18	60	21

Year	11	12	13	14	15	16	17	18	19	20	21
Units	22	22	50	23	25	26	42	26	26	28	37

Notice the large procurement amounts occurring at years 1, 5, 9, 13, 17, and 21. These peaks will generate corresponding peaks of support requirements, meaning that the demand for warehousing, maintenance work, and distribution will be uneven having its peak times and its slow times. See graph a. in Figure 2.3. This graph clearly shows the peaks and valleys that occur due to block procurement practices. When extended over a period of many years, a steady-state is reached where annual procurements and the corresponding support requirements become the same

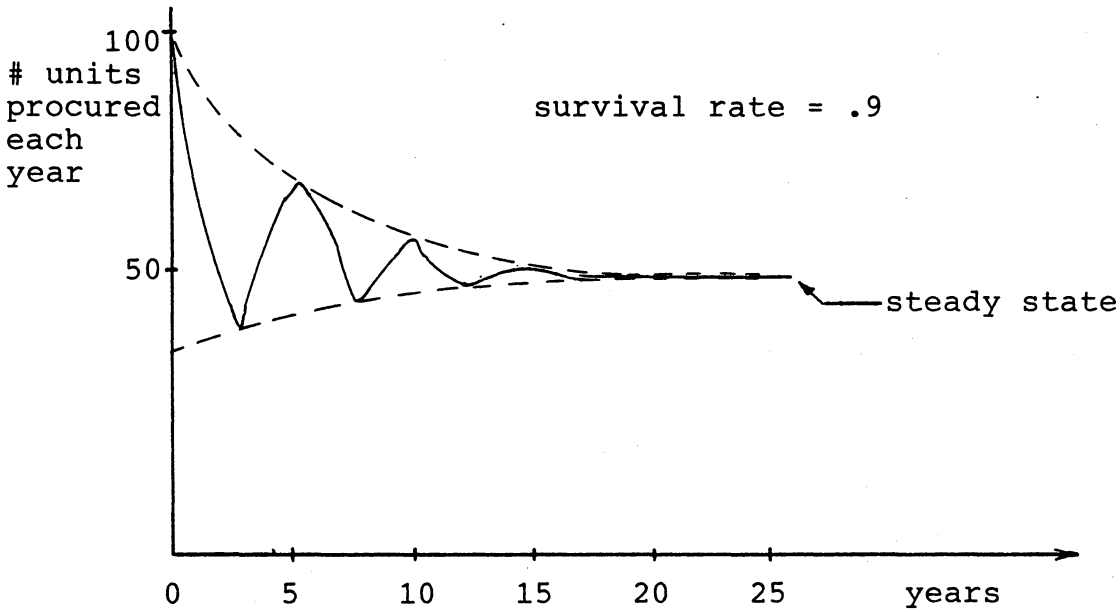


t_n = year n survival rate = .9 throughout each stream

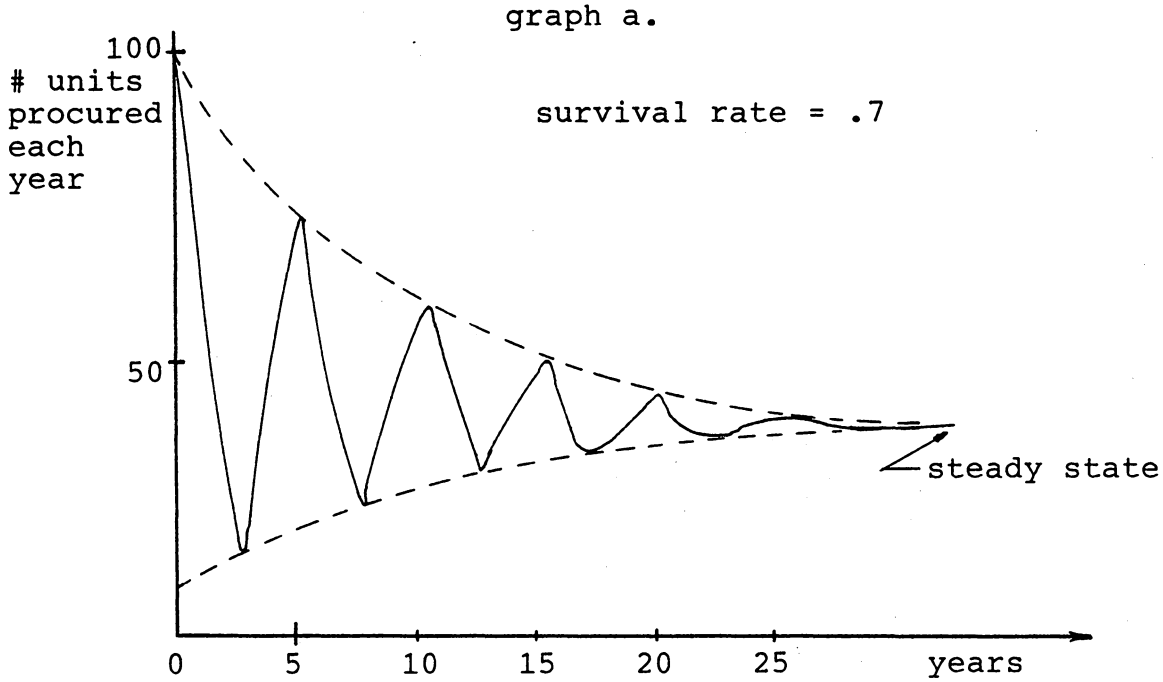
Surviving units are retired after four years of operation

G-1 = 1st year of operation G-2 = 2nd year of operation
G-3 = 3rd year of operation G-4 = 4th year of operation

FIGURE 2.2. NESTED MARKOV CHAIN EXAMPLE



graph a.



graph b.

FIGURE 2.3. BLOCK PROCUREMENT FLUCTUATIONS

from year to year.

Graph b. in Figure 2.3 represents the effects of quality on block procurement. When the quality, as represented by the annual survival rate of the population, is lowered to .7, steady-state is reached sooner than in the case where the survival rate was .9. Another result of the decrease in quality is that the peaks during the transient phase are lower, but the steady-state value eventually attained is higher than the steady-state value for annual procurements for the population with the .9 survival rate. This higher steady-state value translates to a greater annual support requirement. Note also that the lower quality units reach steady-state sooner than the higher quality units.

Frisch notes that an example such as this illustrates that

1) Low quality tends to promote constant replacement rates in the early stage of a system and hence may need little warehousing.

2) High quality is detrimental to a stabilization of the system and hence detrimental to economic replacement-production; it also requires high warehousing efforts.

Frisch also demonstrates the effects of uncertainty on support requirements and on repair/replace decisions. His

model allows for the evaluation of the trade-off of costs for testing to obtain more certainty versus the costs of operating a logistics system under uncertainty. Frisch states that a result of this trade-off evaluation is that it becomes apparent that under conditions of complete certainty a system should be repaired until it ceases to perform to standards. If uncertain conditions are present, then a point is reached where the total system's life cycle cost can be reduced by replacing instead of continuing to repair units. Other models contradict Frisch in that they can be used to illustrate that there are situations where certainty exists, but a point prior to the occurrence of unsatisfactory performance is reached where it is more economical to replace rather than repair.

A model by Fabrycky and Hart [4] illustrates this contradiction. Fabrycky uses his finite population queueing model to work an example that optimizes the replacement age. His example illustrates that a point can be reached where the costs of maintaining and operating an older unit exceeds the costs involved with purchasing a newer unit. This phenomenon occurs because Fabrycky assumes that the failure rate follows a bathtub function where the rate of failure begins to increase with age after a certain time. It is also assumed that mean times to repair increase with age. The combined effect of these

assumptions is that at some point, repair costs begin to increase with age until eventually these costs exceed the cost of purchasing and operating a new unit in place of the aging unit. Even at this point where a new unit becomes more cost effective, the old unit is still performing satisfactorily.

Frisch also illustrates that assuming there is in-house production capability, his model may be used for the evaluation of the production rate of spares or spare parts. The demand for spares or spare parts is of course determined by the nested Markov Chain model. By associating a penalty cost with shortages, the trade off between shortage costs incurred and costs saved by evening out production of spare parts may be evaluated. The same evaluation could be made by substituting costs saved by evening out procurement of inventory items for saved production costs.

Frisch closes his paper by pointing out that his model is flexible and extendable to cover large populations composed of many different cohort age groups. The model also may be extended to handle composite streams, where each stream within the nested Markov Chain can have a totally different probability distribution of failure and/or a different life expectancy.

2.5 QUEUEING THEORY AND LOGISTICS.

Of the models discussed thus far, that have considered repair times, a constant mean repair time has been assumed, independent of the state of the system, for items entering service. The models have not considered the queue that occurs when items are waiting for service. In most real life cases, where repair channel capacity is not infinite, the true time to repair is dependent upon the number of units queued up at the repair facility. Also, the failure rate of a population of units is dependent on the number of operable units. Unless an infinite number of spares is assumed, the number of units queued up at the repair facility must be known in order to calculate the number of operable units. In order to determine more exactly the time to repair and the failure rate, a class of maintenance models utilizing queueing theory have been developed.

These queueing models have the potential to consider perhaps the broadest scope of logistics problems because not only can repair or replace decisions or spare parts inventory level decisions be made, but now the actual operation of the repair channels can be studied. The application of queueing theory allows the logistician to determine the effects of repair and replace decisions on the operation of the repair channel. Queueing theory also provides the logistician with a means of more accurately

evaluating the effects of repair facility design on the total logistics system.

2.5.1 Queueing and the Repair/Replace Decision. Nahmias [23] writes that one approach for analyzing the queues that form at repair channels is through the classic machine repair model. Also, Nahmias points out that the original application of the machine repair model to the problem of determining suitable levels of spares inventory appears to be due to Taylor and Jackson [33]. The classic machine repair scenario as utilized by Taylor and Jackson supposes that M machines operate continuously and are supported by S spares. When a failure occurs, the item enters a repair facility and queues up for service. If available, a spare immediately replaces the failed item. The repair facility is assumed to consist of $r \geq 1$ parallel servers that repair units on a first come first serve basis.

Taylor and Jackson assume that the items become unserviceable at random, but they utilize a constant average rate of failure. They use a birth/death process in conjunction with the failure rate and they assume that when fewer than M machines are operational, all machines cease to operate. From this assumption a stationary distribution of the number of machines in the repair queue under steady-state conditions is derived. The model then

considers the probability that a total breakdown in the system will occur as a function of the number of spare machines.

Gross and Harris [9, page 123] extended the classic machine scenario further and developed expressions for the expected rate of failure and the expected rate of repair dependent on the number of units in the repair system (queue plus repair channels). Their expressions apply only when the rate of failure and the time to repair are distributed exponentially with the respective parameters λ and u.

The expressions developed are as follows. The state of the system, n, corresponds to the number of machines in the repair system so that, if λ_n is the failure rate when the state of the system is n, then

$$\lambda_n = \begin{cases} M\lambda & \text{if } 0 \leq n < S, \\ (M - n + S)\lambda & \text{if } S \leq n < M+S. \end{cases}$$

Let u_n be the rate at which units leave the repair facility when the system state is n, then

$$u_n = \begin{cases} n \cdot u & \text{if } 0 \leq n < r, \\ r \cdot u & \text{if } r \leq n. \end{cases}$$

<u>2.5.2 Queueing and Repair Facilities.</u> Hart [12] and Fabrycky and Hart [4] extend the machine repair model concept. Utilizing finite source queueing theory, they developed an acquisition/maintenance model that allows the

user to design an optimal cost system by specifying the values for certain decision variables. These decision variables include the population size, the replacement age of units, and the number of repair channels. The model also evaluates a second scenario where the mean time between failure and the mean time to repair are considered to be design variables. The model is a single indenture, single echelon model. The performance of the model is measured by comparing the total annual equivalent costs generated by different design scenarios embracing three general costs:

1. population annual equivalent cost which represents the cost of ownership of the population by evaluating acquisition cost of the unit, salvage value of the unit, and the interest rate.

2. repair facility annual equivalent cost which represents the cost of owning and operating a repair facility by evaluating the first cost of the facility by channel, the salvage value of the facility per channel, and the yearly administrative, labor, and overhead cost per channel.

3. out of operation or down cost which represents the cost of not having enough units operable to meet the demand.

Fabrycky and Hart employ a birth-death process that evaluates only the steady-state phase of operation of the finite population of units. They assume that the inter-arrival times of units for repair are exponentially distributed as are the times for repair. The arrival times are statistically independent of the repair times. The

exponential distributions are utilized for two reasons. First, a distribution that will produce a constant mean is necessary in order to allow the evaluation of steady-state conditions. Secondly, a distribution that "forgets" is required in order to be used with a birth-death process which has the Markovian property of "forgetfulness".

The meaning of "forgets" may be illustrated by the following, as taken from Hillier and Lieberman [15, page 325]:

> "If the random variable T is the length of time a process continues before an event occurs, and it has an exponential distribution, then
>
> $P[T>t+\Delta t \mid T>\Delta t] = P[T>t]$
>
> for any non-negative quantity Δt, thus, the process "forgets", in effect how long it has been going."

Just as in the machine repair problem, Fabrycky and Hart assume that repair channels operate in parallel and serve units on a first come first serve basis. (See Figure 2.4 for a diagram representing the relationship of the repair channels to the rest of the system.) Each repair channel has the same capability and only one unit at a time is repaired per channel. Fabrycky and Hart recognize that a population can consist of different aged units, thus at any given point in time the mean time between failure and the mean time to repair varies for each age group. It is

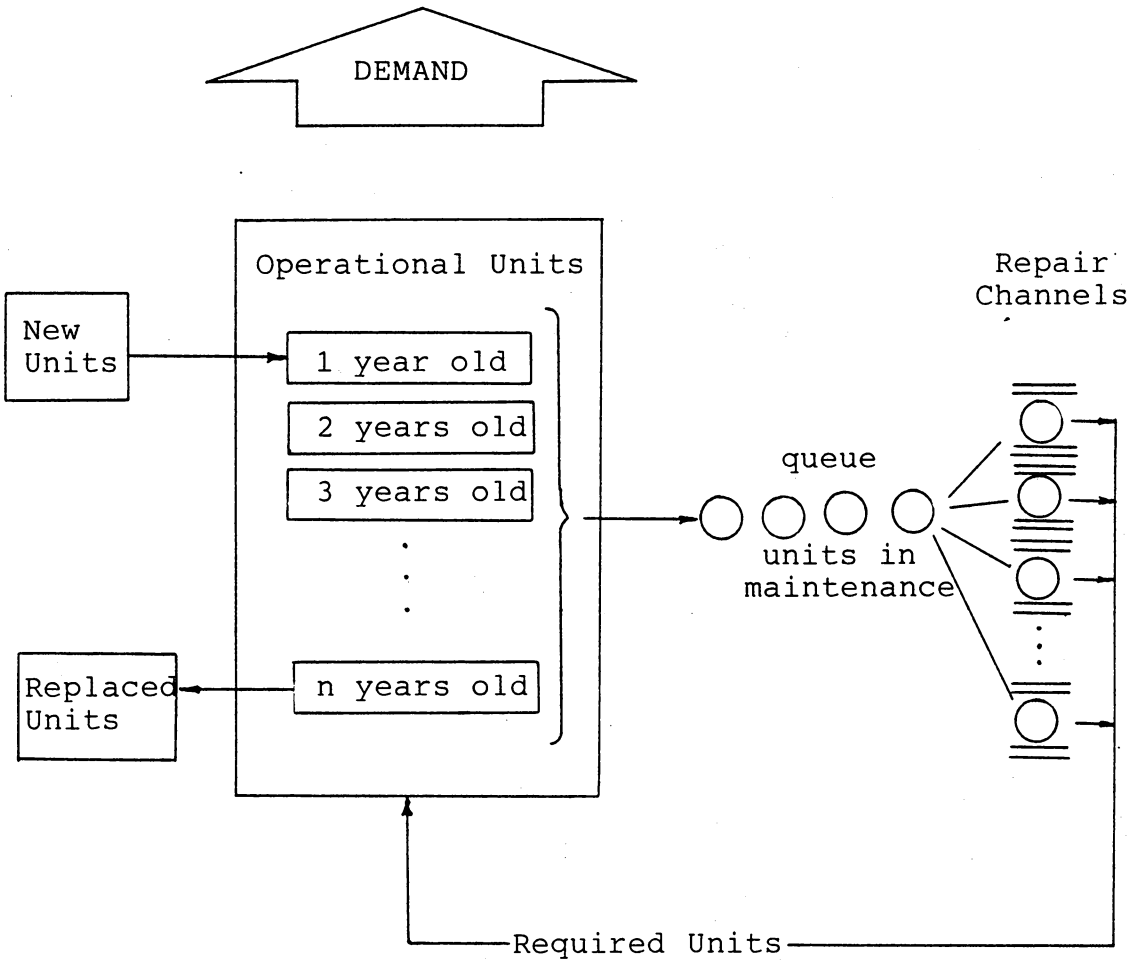


FIGURE 2.4. FINITE POPULATION SYSTEM

assumed that the population consists of an equal number of each age group. As shown in Figure 2.4, when a unit reaches the designated age of replacement, that unit leaves the population and a new unit is procured in its place and added to the population.

In the model, units that have been repaired return to the population with the same operational characteristics of their age group. The time spent in repair is the time elapsed from the instant the unit fails until the instant that the unit becomes operational again.

In order to calculate the amount of time a unit spends down, Fabrycky and Hart obtain the expected values for the MTBF and MTTR of each age group and utilize finite queueing theory to determine the expected time that a unit waits in the queue for repair, W_q, and the expected number of units in the repair system (queue plus repair

channels), R. The following queueing equations from White, Schmidt, and Bennett [35] were used:

$$W_q = \frac{\sum_{n=C}^{n-1} \frac{n-C+1}{Cu} (N-n) \frac{N!}{(N-n)!} \frac{C^C (\lambda/uC)^n}{C!}}{\sum_{n=0}^{C-1} (N-n) \frac{N!}{n!(N-n)!} \left(\frac{\lambda}{u}\right)^n + \sum_{n=C}^{N} (N-n) \frac{N!}{(N-n)!} \frac{C^C (\lambda/uC)^n}{C!}}$$

$$R = N\left[1 - \frac{1/u + 1/\lambda}{1/u + 1/\lambda + W_q}\left(1 - \frac{1/u}{1/u + 1/\lambda}\right)\right]$$

where:
N = total number of items in population
C = total number of repair channels
λ = the arrival rate = 1/MTBF
u = the service rate = 1/MTTR

Knowing the amount of time that units are out of service allows the calculation of the number of units down, which in turn makes the calculation of the out of operation or down costs possible.

A limited source queueing model by Hillier and Lieberman [15, page 311-312], provides an alternate method of calculating the expected number of units down. They assume a finite source population and exponentially distributed failure rates ($1/\lambda$) and service times ($1/u$). The queueing model has multiple, parallel servers, each with identical service capability. The resulting model is

a special case of the birth-death process such that

$$\lambda_n = \begin{cases} (M-n)\lambda, & \text{if } n \leq M \\ 0, & \text{if } n \geq M \end{cases}$$

and

$$u_n = \begin{cases} nu, & \text{if } 0 \leq n \leq s \\ su, & \text{if } n \geq s. \end{cases}$$

Hence,

$$P_n = \begin{cases} P_0 \dfrac{M!}{(M-n)!n!} \left(\dfrac{\lambda}{u}\right)^n, & \text{if } 0 \leq n \leq s \\ P_0 \dfrac{M!}{(M-n)!s!s^{n-s}} \left(\dfrac{\lambda}{u}\right)^n, & \text{if } s \leq n \leq M \\ 0, & \text{if } n > M, \end{cases}$$

where

$$P_0 = \left[\sum_{n=0}^{s-1} \frac{M!}{(M-n)!n!} \left(\frac{\lambda}{u}\right)^n + \sum_{n=s}^{M} \frac{M!}{(M-n)!s!s^{n-s}} \left(\frac{\lambda}{u}\right)^n \right]^{-1} .$$

Finally,

$$L_q = \sum_{n=s}^{M} (n-s) P_n .$$

where:
M = total number of units in population
s = number of servers or repair channels
λ = failure rate
u = service rate
P_0 = probability of no units down
P_n = steady-state probability of n units down
L_q = expected number of items in the queueing system

Gross, Kahn, and Marsh [11] present a queueing model where infinite instead of finite queueing theory is utilized. The model is an optimization model built around the basic machine repair model. They determine an adequate number of spares and repair channels for replacing and repairing components which randomly fail. Upon failure,

failed components are replaced by spares. A service level constraint is imposed in that there must be adequate spares available at least 90% of the time. It is assumed that there is an infinite input of units into the repair channels. The objective of the model is to minimize expenditures for spares and repair channels subject to the 90% availability constraint. Failure rates and times to repair are considered to be exponentially distributed and an expected MTBF and MTTR are utilized in conjunction with a series queue to determine the number of units in repair.

The cost performance measure utilized in the Gross, Kahn, and Marsh model consists of the following four costs: purchase cost of spares, purchase cost of service channels, repair costs, and investment costs in component reliability improvement programs. No shortage cost is considered since the model does not allow trade-offs between cost and availability. At least 90% availability must be maintained at all times. It is assumed that no shortage costs are associated with the 10% of the time that spares are not available and shortages do occur.

The optimization of the model is complicated, as an integer-nonlinear-programming problem is required. The authors develop a heuristic method instead, that actually considers only the purchase cost of spares and the purchase cost of service channels in the optimization procedure. It

is assumed that the repair costs and the investment costs in component improvement programs are negligible in comparison to the first two costs. Also, the heuristic method considers only one year at a time, allowing the determination of the best maintenance policy on a year to year basis. The model is a single indenture, single echelon model.

Gross and Ince [10] extend the classical machine repair model to allow for the consideration of more than a single stage in the repair phase. A cyclic queueing system is utilized to study the movement of a unit through several phases of repair. An example of a 4 phase system could be removal of a failed machine, transportation to the repair facility, the repair itself, and transportation from the repair facility. Gross and Ince actually assume that stage 1 of the cyclic queue is the machine operating phase. They treat M operating machines as being equal to c_1 parallel servers at stage one.

Since a cyclic queue is utilized no assumption of infinite input of units (infinite queue assumption) to the repair cycle is necessary. Also, there is no requirement for a high availability of parts as the Gross, Kahn, and Marsh model called for. Gross and Ince derive exact and heuristic expressions for availability.

The exact expression is:

$$AVAIL_E = A_3 c_1 u_1 \sum_{n_1=c_1+1}^{N} P_E(n_1)$$

where:

$$A_3 = \left[\sum_{n_1=0}^{c_1} n_1 u_1 P_E(n_1) + \sum_{n_1=c_1+1}^{N} c_1 u_1 P_E(n_1) \right]^{-1}.$$

and where:

c_1 = the number of servers (operators) at stage 1
u_1 = the steady-state service rate at stage 1
$P_E(n_1)$ = the steady-state marginal probability that there are n_1 customers at stage 1

The heuristic expression is:

$$AVAIL_A = \sum_{n_1=c_1+1}^{N} P_A(n_1).$$

where:

$P_A(n_1)$ = the approximate model's steady-state probability that there are n_1 at stage 1

In order to obtain the approximate model, Gross and Ince assumed that the first stage, the operating machine stage, is almost always operating at full capacity or, in other words, all c_1 machines are up. This is an implication of a high availability constraint such as Gross, Kahn, and Marsh [11] used. The result of this assumption is that the output from the first stage is a pure Poisson process with parameter $c_1 u_1$ and the first stage acts like an infinite source input to the rest of the system.

The decision variables in the Gross and Ince model are: the number of repair channels at each phase of repair

and the total number of units in the population. The objective of the model is to adjust the decision variables to achieve a desired availability of spares.

2.6 LOGISTICS MODELING, STATE OF THE ART.

Maintenance models such as those discussed in this chapter are used in civilian and military settings. Geisler and Murrie [7] write that the military planner needs maintenance or logistic models to help estimate the cost and effectiveness involved with the use of a particular item or family of items. The logistics model also helps the military planner adopt development and production strategies that will improve the cost-effectiveness of the item being evaluated. The civilian planner is most interested in modeling the characteristics of the market in which an item must operate. The information gained from the model will be used to help reduce operating costs which will in effect increase profits. Although Geisler and Murrie do not mention it, civilian planners involved with the maintenance of production machinery can use maintenance models to gain information on the cost effectiveness of the way in which machinery is procured and utilized and in the way in which the machinery is maintained.

A key point in the above definitions is that logistics models are of great value to personnel involved in planning

functions. Geisler and Murrie go on to write that "Some data suggest that about 80% of the life time logistics costs of an aircraft is determined by decisions made before the aircraft begins to fly. These decisions relate to design tradeoffs of performance against reliability and maintainability, with resulting impact on maintenance manning, initial spare parts procurement, location of repair, and ground support equipment investment." This statement, although specific in referring to aircraft, points out that logistics models can be invaluable as planning tools, thus logistics models must be able to perform trade off and sensitivity analysis to take account of the flexibility and uncertainties existing during planning phases. Frisch [5] reinforces the value of logistics models as planning tools and Fabrycky and Hart [4] emphasize the importance of the trade-off capabilities of a logistics model through their analysis of initial cost and logistics costs tradeoffs. This trade-off is analagous to the design trade-offs listed by Geisler and Murrie.

Geisler and Murrie point out that logistics models should be able to reflect a weapon system's logistics during peacetime, surge, and sustained combat scenarios. Surge conditions occur when a mobilization begins in response to an act of war. In their article, Geisler and Murrie evaluated logistics models in use in 1981 by the

three major branches of the military. They only evaluated models that could be used to model the logistics requirements of aircraft because the measures of performance applied to aircraft are well defined. The logistics models evaluated were placed in three groups: design models, base-aircraft operations models, and spares allocation models. Following are some of the author's findings regarding logistics models.

Design models are similar to the repair/replace models described in section 2.2 of this chapter in that they evaluate the effects of an item's design characteristics on mission reliability. Base-aircraft operations models are generally large-scale Monte Carlo simulations, in which aircraft are prepared and launched and sorties are flown. These simulated events yield information on the logistics needs of the aircraft and on the performance of the logistics support included in the simulation. The queueing models described in section 2.5 of this chapter provide this type of information, among other things. The spares allocation models evaluated by Geisler and Murrie were generally derivations of the METRIC model described in section 2.3 of this chapter. Geisler and Murrie found that "individually and collectively, their (the models) ability to satisfy all the planning requirements for new weapon systems is limited." Generally, the authors found that

most of the models were designed to evalute a limited number of maintenance requirements in which it was assumed that necessary data was readily obtainable. Then the models were found to relate their specific maintenance requirement output to logistic performance without considering the interaction of the overall logistics needs of a population. Because of this limitation, surrogate measures of logistics performance have had to be used. For example, the base-aircraft operations models use probabilities of spare parts availability as a surrogate for explicit modeling of inventory structure.

The authors found that most models had been designed to evaluate the logistics requirements of existing weapons systems rather than being designed as a planning tool for new systems. The design models were found to be the most limited in scope. The spares allocation models were not adaptable to nonsteady-state conditions as they assume a steady-state pattern of spare parts demand. The base-aircraft operations models were difficult to adapt to surge and extended warfare conditions and especially to the transient phases leading into and out of those conditions. Furthermore, owing to their Monte Carlo processes, numerical output from simulations is subject to chance variations making the interpretation of their output difficult.

Geisler and Murrie call for the introduction of comprehensive logistics models that can incorporate the required planning capabilities of the three types of models evaluated in their paper. This comprehensive model would have to be analytic to permit optimization and rapid computation to alternative performance measures. The model would have to be dynamic in order to treat nonsteady-state as well as steady-state conditions. Finally, the model should also be able to contend with different levels of detail for a new weapon system so that useful planning could be done with it over the entire range of the life cycle of the system.

A survey of logistics models by Kline [18] echoes some of Geisler and Murrie's conclusions. He states that "It would be desirable to have logistic support models which are useful during all phases of the system life cycle. Unfortunately, most models are useful only beginning with the Full-Scale Development Phase." Kline's paper was a review of logistics models in general that were in use in the military in this country in 1981.

Kline notes that one of the most common assumptions of the logistics models in use is that of steady-state conditions. The models further limit themselves by requiring input data that is not readily available. Another limitation is that the models assume all items to

be in series. Work has been done to eliminate this latter assumption, as is evidenced in an article by Lohmar and Mandel [20] that illustrates the uses of the TIGER computer program developed by the Navy Sea Systems Command. TIGER generates reliability block diagrams which can be used to model a system operating in series, parallel, or series and parallel combined. The model can then be used to provide simulated mean time between failures for the system. Kline notes that reliability block diagram models, such as TIGER, can be used with spares allocation models to overcome the current series limitation.

2.7 SUMMARY.

The literature review has revealed that research applicable to logistics and logistics modeling has been extensive in the last 25 years. Models have been developed to analyze, optimize, or simulate one or more phases of the logistics process utilizing mathematical programming, dynamic programming, and/or computer simulations. Even though logistics tends to be associated with the military, models have been developed with commercial as well as military applications. Many models have gained acceptance in military circles and are in use today, yet research in logistics modeling continues as researchers grope for modeling techniques that provide more accurate answers to real life logistics problems.

The logistics process involves several dependent considerations. The design characteristics of a system being procured must be evaluated with respect to the spares or spare parts inventories and the repair facilities required to meet a specified demand. A complete logistics evaluation includes a determination of the life cycle costs of owning and operating a system of units. Logistics modeling has been further complicated by the lack of sufficient data regarding failure and repair characteristics of systems of units.

Logistics models have tended to focus on one consideration or another, by making simplifying assumptions about the other dependent facets of logistics. One of the major simplifications made has been the assumption that systems operate in steady-state and that logistics requirements may be determined from these steady-state conditions. Logisticians, primarily those involved with military logistics, have begun to call for logistics models that evaluate the complete logistics requirements of a system that does not necessarily operate under steady-state conditions at all times. As stated by Geisler and Murrie [7] "rather than devoting energy to minor improvements in existing models, the model developer should focus on an effort that will produce a big payoff." The "big payoff" will be a new model that logisticians can depend on for

viable solutions to the planning and design of the total logistics system.

This "new" model will have to achieve a balance between simplicity and detailed diagnostic analysis. Just as previously developed logistics models have done, the "new" model will generate a requirement for empirical logistics data, but that empirical requirement must be tempered by certain behavioral assumptions about logistics systems. A model so balanced will possess a generality that will allow application to a broad range of logistics problems, yet retain enough specialization to realistically assist the logistics decision maker with the problems at hand.

CHAPTER III

THE MODELING APPROACH

3.1 INTRODUCTION.

The information presented in this chapter will provide insight into the approach taken in developing the logistics model. The concepts utilized are the key to the model because they allow the evaluation of maintenance requirements for repairable populations during transient as well as steady state phases of operation. Information included in the chapter will describe why these concepts were chosen and how they are applied.

Assumptions made as a part of this model development are detailed as are the parameters and decision variables included in the model. Both the parameters and the decision variables represent data that must be input by the user of the model. Calculations necessary to compute maintenance requirements are detailed using variables that appear in the computerized version of the model. Performance measures used by the model to judge the effectiveness of logistics policy versus maintenance requirements are also described.

The parameters mentioned above indicate variables that the logistics planner generally has no control over and the decision variables represent variables that in many cases

are left to the discretion of the logistics planner. By using the model to play "what if" games with the decision variables, the logistics planner can use the model as an aid to logistics decision making. Changing parameters and evaluating their effect on maintenance requirements may provide information for high level decision makers for their use in establishing parameter values in the future. Besides parameter and decision variable manipulation, other uses of the model are described later in this chapter.

An outline of the limitations created by the modeling approach follows the section on Model Uses. The chapter closes with a summary of the modeling approach.

3.2 TERMINOLOGY.

Certain terms are used frequently in describing the research model. Following are definitions of key terms:

Component - An end item can be broken down into components. The word component refers to subassemblies, parts, or modules of the end item. Each component defined must be assigned a node on the maintenance tree. Note that on the maintenance tree, components at lower levels of indenture are components of components.

End Item - This model is designed to evaluate a population of repairable items. Each item in the repairable population is an end item. The end item occupies $Node_1$ at the top of the maintenance tree.

Failure Environment - Many times, identically aged end items do not fail at the same rate because of how they are used, where they are used, or because of slight differences in component quality. Therefore, failure environments are created. Identically aged end items within the same failure environment experience the same rate of failure because all of the components that comprise the end item have the same rate of failure. Identically aged end items operating in different failure environments do not experience identical rates of failure for all components.

In Operation - This term applies to end items and is meant to imply that the end items are not in the repair phase, i.e. they are ready for operation or are in use.

Level of Indenture - There are five levels of indenture on the sample maintenance trees in Figures 3.3 and 3.4. The first level of indenture is the end item. The second level of indenture includes major components that make up the end item. (Note that $Node_6$ represents the probability of fatal failure. It is not a component.) The third level of indenture includes components that are components of the second level of indenture and so forth. The last level of indenture includes spare parts or stockable items required to support the higher levels of indenture. The number of levels of indenture is an indication of the extent to which an end item can be broken

down into components.

Level of Repair - If there is more than one level of repair then there is more than one type of repair facility available to handle end item failures. The type of failure will determine which repair facility should handle the repair. A level of repair then is a specific type of repair facility.

Node - Each "box" on the maintenance tree is a node. $Node_1$ represents the end item. $Node_6$ represents a probability of fatal failure of the end item. Each of the other nodes represents a component of the end item. Data input by the user of the model is used to establish a probability of failure of components represented by bottom of the tree nodes. By moving upward along branches of the tree, probabilities of failure may be determined for each node of the maintenance tree.

Period - The discrete time periods specified by the user of the model, over which all model outputs are calculated. The user directs the model to evaluate maintenance requirements for a certain number of periods. The user can use inventory cycles or any other convenient means for determining the lengths of the periods to be used.

Period Unit - If a period is five days long, the period unit is days. If a period is one month long, the

period unit is months. Period units of days, weeks, months, and years may be used in the present version of the model.

Stockable Item - Any spare part that is required to support the components listed as "bottom of the tree nodes" is a stockable item. By support, it is meant that if the bottom of the tree component experiences a failure, the stockable items represent the spare parts that will be needed to repair that component.

Stream - A stream is a homogeneous group of end items. By homogeneous, it is meant that the end items are identically aged and have identical failure rates for all components.

3.3 MODEL DEVELOPMENT.

The model under development will draw heavily upon some of the concepts utilized by Frisch [5] in his development of a conceptual Markov model. The model will attempt to gain a breadth of coverage found in very few logistics models. By breadth of coverage it is meant that the model will allow the investigation of alternative maintenance policies regarding repair or replace, spare parts inventories, and repair facility characteristics.

3.3.1 Concepts. The model is built upon two major concepts. The first concept is an expansion of the multi-stream idea utilized by Frisch [5] in the development

of a conceptual Markov Model that evaluates the mortality characteristics of populations of spare parts needed to support military weapon systems. The second concept is that of using a maintenance tree to give an organized breakdown of component failure characteristics that govern the maintenance requirements of an end item.

3.3.1.1 Multi-Stream Concept. The model treats the failure of end items as a discrete, nonstationary, Markov process. Generally distributed probabilities of failure are used to determine maintenance requirements at each step of the Markov process. The general distributions used are described in Section 3.3.5. Each step of the Markov Process represents one period. See Section 3.2 for the definition of a period.

A Markov process was chosen because the life cycle of items evaluated by logistics models generally is finite. This finite period may be broken into stages or states in which the item deteriorates toward a final state where one of two things could happen: 1) It is no longer economically or operationally feasible to continue to repair the end item so the end item is retired from operation; or 2) The end item experiences a fatal failure. At each state leading to the final state, there is a probability of a failure and there is the probability that no failure will occur. The no failure/failure possibility

in each state is analagous to a birth-death process. A Markov chain represents a good way to model a birth-death process as long as end items fail randomly. It has already been stated that the failure of end items is governed by generally distributed probabilities of failure, so the random failure requirement is met. Further, the state by state evaluation of the unit life cycle possesses the Markov property, in that it is only necessary to know the present state of an end item in order to forecast future states.

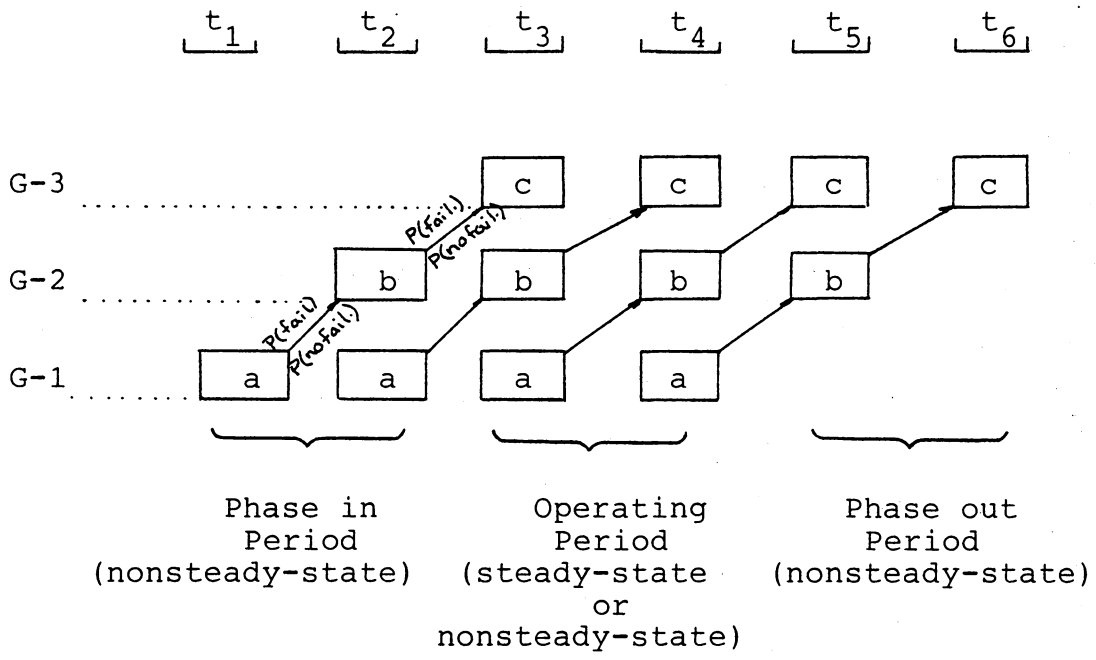
A nonstationary Markov process will be utilized because the model will be used to evaluate nonsteady-state conditions where there is no uniform transition probability from state to state. There may be circumstances under which a population of units never attains steady-state.

In the multi-stream concept employed by this research, a family of nested Markov chains is used to model a population of repairable end items. The population is divided into streams depending upon such parameters as the age of the end item, the number of hours that the end item has operated, or the environment in which the end item operates. Because of the parameter used to divide the end items into streams, each stream represents a homogeneous group of end items. The homogeneity within each group or stream is maintained by applying the same failure

distribution and repair characteristics to each item within a particular stream. In other words, items within a stream are similarly aged, are at the same stage of deterioration, and experience a similar failure environment. Grouping end items into streams avoids the need for item by item tracking.

A stream begins when the first item in that grouping is procured and the stream ends when the last item in that grouping is removed from operation through retirement or fatal failure. Each stream is in effect a Markov chain and all of the streams grouped together or nested together form a model representing the failure characteristics of a population of repairable end items. Each step of the individual streams (Markov chains) represents a period specified by the user of the model. See Figure 3.1 for an illustration of this stream concept.

Even though a discrete Markov process is used to model failure characteristics of end items, a continuous approximation is derived by the model. The accuracy of that approximation is dependent, to a large extent, on the lengths of the periods chosen for evaluation purposes. Each step (period) of the Markov chain representing a particular stream is a discrete approximation of the expected maintenance requirements generated by the end items in that stream. As illustrated by Figure 3.2, when



t_n = time period over which logistician is evaluating inventory requirements (could be on a weekly basis, on an annual basis, etc.)

G-n = age of units. The largest G-n represents the retirement age of the unit.

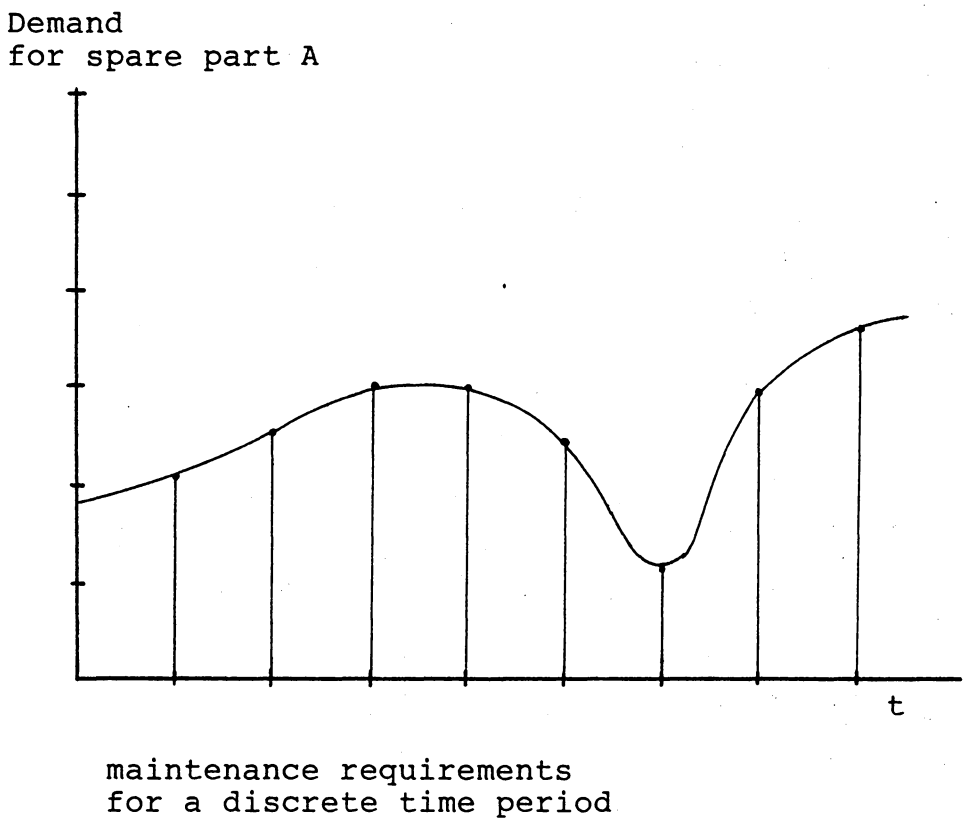
abc = a stream or a Markov chain (All of the streams together represent the nested Markov chain concept.)

a = number of units procured during time period, t_n

$a,b,c \geq 0$

Each node has a maintenance tree attached to it (see Figure 3.3)

FIGURE 3.1. NESTED MARKOV CHAIN (MULTI-STREAM) CONCEPT



Each vertical spike represents the total demand for spare part A during time period t. Connecting the tops of the spikes produces a continuous approximation of the demand for A over all time periods. The continuous approximation becomes more accurate as the length of the individual discrete time periods is shortened.

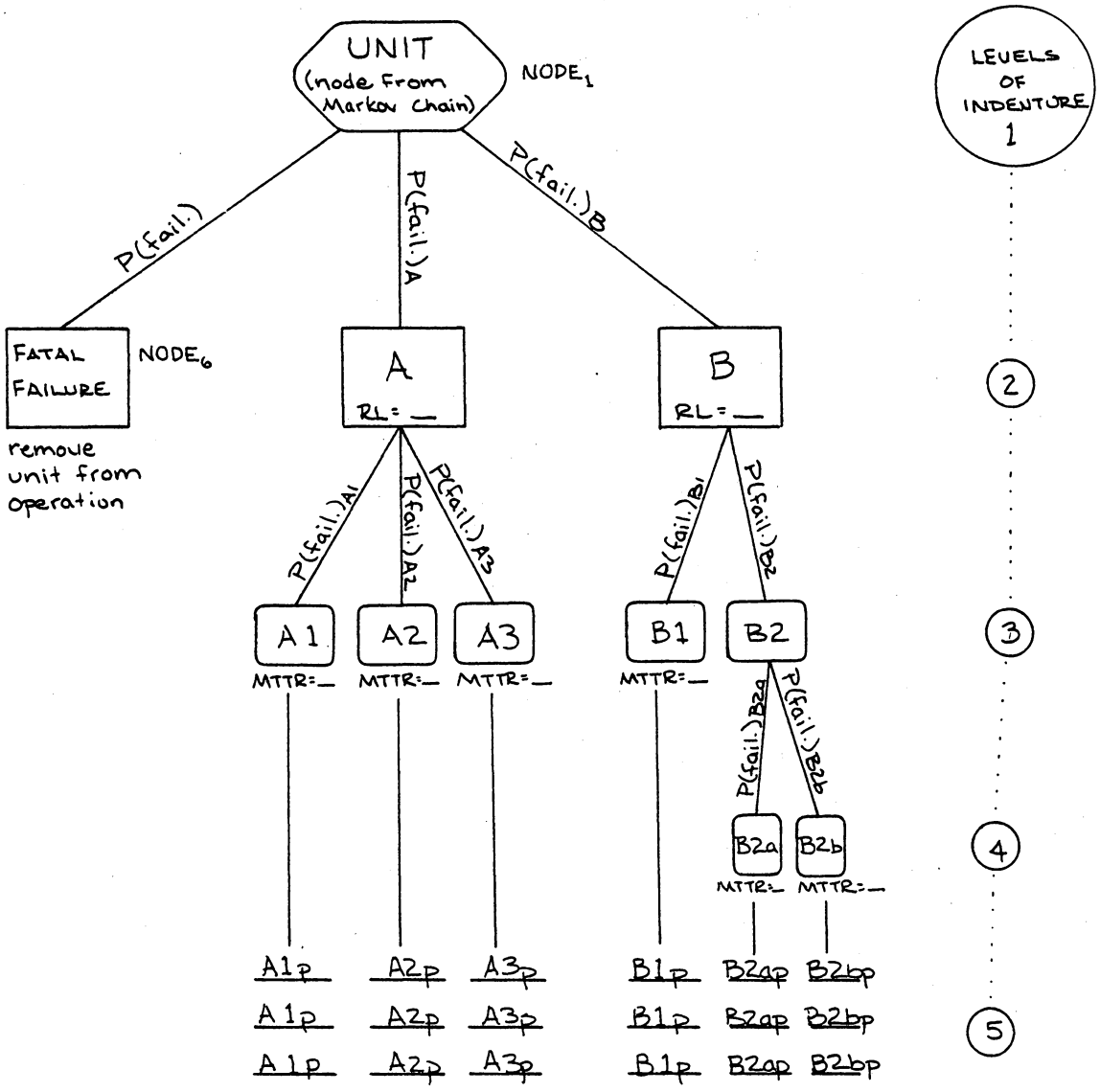
FIGURE 3.2.

CONTINUOUS APPROXIMATION OF MAINTENANCE REQUIREMENTS

these discrete approximations are grouped together, a continuous approximation of the expected maintenance requirements for the end item may be derived. Figure 3.2 uses a period by period spare part requirement as an example.

3.3.1.2 Maintenance Tree Concept. As previously described, for each step (period) of the individual Markov chains (streams), there will be a probability of failure or a probability of no failure for the end items included in that stream. When no failure occurs, no action need be taken by maintenance personnel and the unit continues in operation. When a failure occurs, the affected unit is removed from service and is directed to the appropriate repair facility. A maintenance tree is used to derive the probabilities of these end item failures occurring. The maintenance tree also contains information that governs the repair process (see Figures 3.3 and 3.4).

The maintenance tree allows the end item to be systematically divided into various major components. Below each major component in the maintenance tree will be the family of subcomponents and spare parts that make up the major component. Each level of the maintenance tree represents a level of indenture. At some level of indenture a "bottom of the tree node" will be encountered where it is not practical to break the component



RL = repair level

MTTR = mean time to repair

FIGURE 3.3 MAINTENANCE TREE

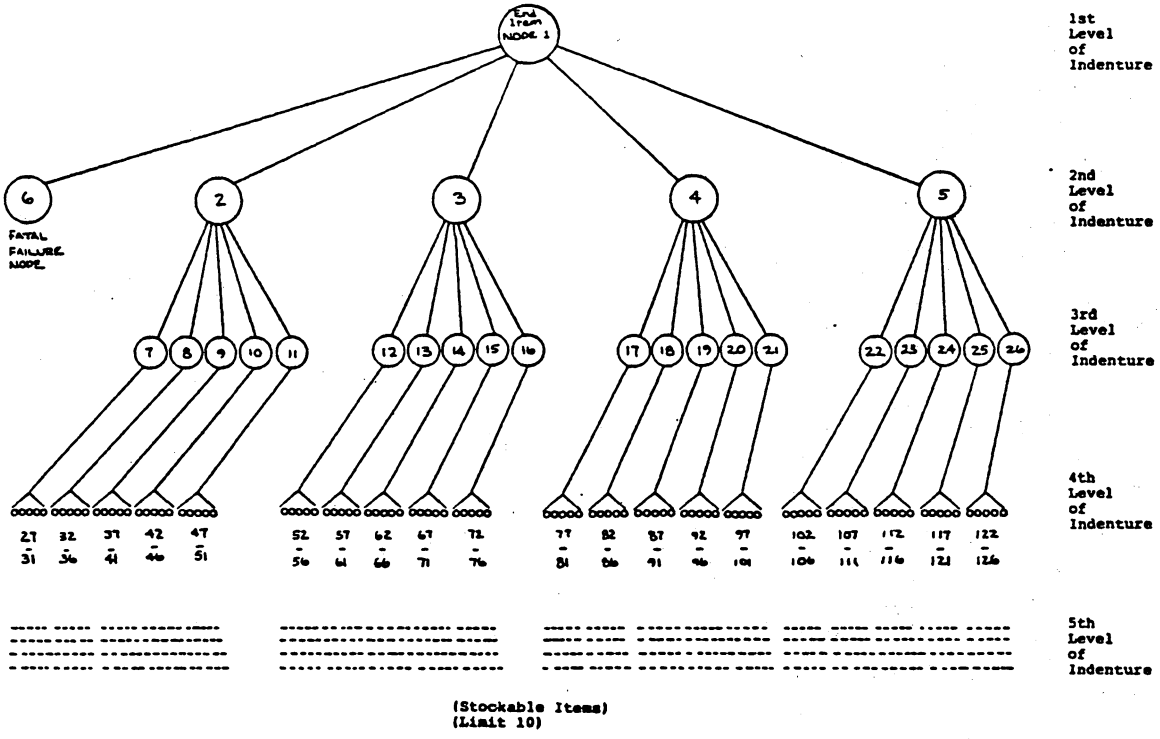


FIGURE 3.4. MAXIMUM MAINTENANCE TREE CONSIDERED BY MODEL

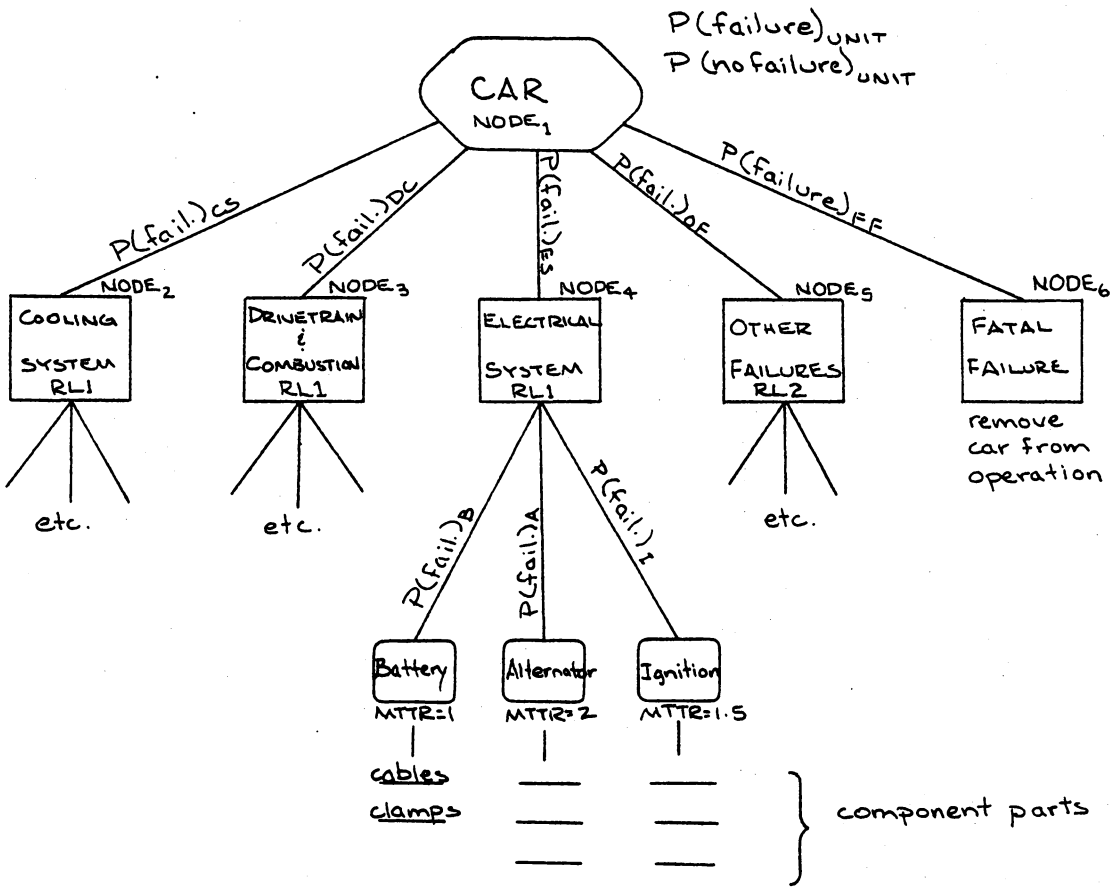
down further. At this node, either failure data is available or can be estimated. Once this level of indenture is reached, a list of stockable parts must be attached to the "bottom of the tree node". This list of stockable parts should include all parts that are intended to be stocked to support the spare parts requirements of that particular bottom of the tree node. The sum of all of the lists of stockable parts should represent a complete list of all the parts that are intended to be stocked to support the spare parts requirements of the end item.

The failure information available for the "bottom of the tree node" is used to establish a failure rate probability distribution for each "bottom of the tree node". Failure information may be derived from data supplied by manufacturers of components, from in house maintenance records, or by extrapolating data from other sources. Once these failure probabilities have been determined and input into the model, they are in turn used to derive the failure probabilities of components at higher levels of the maintenance tree. There is a vertical dependency of failure rates in the tree and a horizontal independence of failure rate at each level of indenture. (See the section on Assumptions, Section 3.3.2, for a discussion of failure rate dependency.) It follows then that conditional probability may be used to compute failure

probabilities at each node of the tree beginning with the given information at "bottom of the tree nodes" and working up to the top of the tree node which represents the end item itself.

As an example of how the probabilities of failure are computed, consider the maintenance tree in Figure 3.5. The lowest level at which a failure distribution is available (in the detailed branch of the tree) is the battery. Component parts listed below the battery are stockable items and do not have failure distributions associated with them. The probabilities of failures of components listed in the tree above the battery are derived by using conditional probability to combine the probabilities of lower levels of indenture, beginning with the level of the battery. As mentioned previously, the failure distribution for the battery node may follow any of several general distributions or specific period by period failure probabilities could be input if they are known. See Figure 3.6 for examples of the general cumulative failure probability distributions that may be used.

Note in Figure 3.4 that Node 6 represents the probability of fatal failure. This is fatal failure of the end item and represents situations in which acts of war, serious accidents, or serious failures cause an end item to be removed from operation permanently. Since this node is



At the component part level, it may be desirable to supply certain information, such as, new cables and clamps are only needed for 10% of the battery replacement. For this example, we are assuming that the last level of indenture possessing a known failure probability distribution is at the battery level. The probabilities of failure of the battery, alternator, and ignition will be combined, using applicable laws of probability, to obtain the $P(\text{fail.})_{ES}$.

$$P(\text{failure})_{unit} = P(\text{failure})_{CS} \cup P(\text{failure})_{DC} \cup P(\text{failure})_{ES} \cup P(\text{failure})_{OF} \cup P(\text{failure})_{FF}.$$

$$P(\text{no failure})_{unit} = 1 - P(\text{failure})_{unit}$$

RL = repair level
MTTR = mean time to repair

FIGURE 3.5. MAINTENANCE TREE FOR BATTERY EXAMPLE

connected directly only to the end item node ($node_1$), only the probability of failure of the end item is dependent on this fatal failure probability. Fatal failure of components is inherently resolved if components that can fail fatally are listed as stockable items at the bottom of the maintenance tree.

The maintenance tree will also be used as a frame to carry other information such as a mean time to repair and a level of repair for each "bottom of the tree node". The mean time to repair (MTTR) and the level of repair (LOR) are input at the same time failure information for each bottom of the tree node is input. See Section 3.3.2.2 for a description of what is included in the MTTR. Several levels of repair will be considered to be available for use upon failure of an item. These levels of repair are analagous to the echelons of repair used by logistics models such as METRIC [31]. LOR is defined in Section 3.2.

3.3.2 Assumptions. This model is intended to introduce concepts that may be employed in evaluating transient state maintenance requirements. With the introduction of these concepts being the main objective of the model several simplifying assumptions have been made. While some of these assumptions may detract from the realistic application aspect of the model, most of these assumptions may be removed by building onto the model.

Sections 4.5 and 6.3 discuss some paths that future researchers may take in expanding this model and making it more suitable for practical application.

3.3.2.1 List of Assumptions. The assumptions utilized may be summarized as follows:

1) Each stream in the nested Markov chains contains a homogeneous group of end items.

2) Minimal repair of end items and certain components is possible, i.e., not all failures are fatal.

3) Minimal repair returns the end item to the state of readiness it was in just prior to failure.

4) The failure of any component represented by a node on the maintenance tree results in an immediate failure of the end item.

5) Failure probabilities are not dynamic unless manipulated manually by the user of the model. For example, if a repair is made on a component part in step n-1 of the Markov chain, the probability of failure of that component part in step n will not be affected by the fact that a repair has occurred.

6) Components at any level of indenture fail independently of the failure of other components at that level of indenture. Only one component can fail at a time.

7) The failure of a component at one level of indenture is dependent on the failure of components at lower levels of indenture that are on a connected branch of the maintenance tree.

8) There is always a chance of fatal failure of an end item. Upon fatal failure, an end item must be removed from service permanently.

9) The user's choice of failure rate distribution will in all likelihood be an assumption on the user's part.

10) Several levels of repair facilities are available.

11) All repair facilities have an infinite capacity.

12) Mean Time to Repair includes the entire amount of time that elapses between failure of the end item/component and the point in time at which the end item is considered in operation.

13) Replenishment of end items through procurement is instantaneous.

14) Purchase costs, procurement costs, repair costs are constant from period to period and are treated as end of period costs in the cost analysis.

15) Repair costs per unit time are the same at all levels of repair. This repair cost includes labor and overhead.

16) The same shortage cost, per unit of time short, is incurred whether the system is one end item short or many end items short. The system is considered inoperable once it becomes one end item short.

17) When a particular "bottom of the tree" component fails, all stockable items listed under that component are required for repair.

18) Demand on end items is constant from period to period.

3.3.2.2 Discussion of Assumptions. Following is a discussion of the assumptions in the order in which they appear above. If the assumption limits realistic application in any way, some comment will be made as to the means for removing the assumption.

Assumption 1) results from the application of the multi-stream concept described in Section 3.2.1.1. This is a timesaving, structural assumption created by the model, not a limiting assumption. End items are grouped into homogeneous streams so that maintenance requirements may be calculated for groups of end items rather than for individual end items.

Assumption 2) simply points out that a repairable population of end items is being dealt with. The end items and many of the components cannot be regarded as throw away

items. This would not qualify as a limiting assumption, but as a descriptive assumption.

Assumption 3) points out that a repair does not return an end item to a "good as new" condition. The respective probabilities of failure of the end items and all components is time dependent, and is not affected by the number or type of repairs that have been performed.

Assumption 4) is a limiting assumption in that it states that the failure of any component listed as a node of the maintenance tree will result in a failure of the end item. This precludes the direct consideration of systems containing noncritical components or redundant (parallel) components. If this model were applied to an end item containing noncritical or redundant components, the availability output given with Assumption 4 would, in effect, be a conservative estimate of the availability for that end item. Creative manipulation of the input data could produce more realistic "out of operation time" data and a more accurate measure of availability.

Assumption 5) can be construed as a limiting assumption, because in many cases a repair can change the failure characteristics of a component or an end item. If the user of this model takes this fact into account when choosing a failure rate distribution and a mean time between failures, then the assumption is no longer

limiting. In other words, it is possible to construct failure data that reflects what the probability of failure becomes after a repair has been made.

Assumption 6) concerns the horizontal independence of failure existing in the maintenance tree. By constructing the tree so that the failure of components at the same level of indenture is independent, the calculation of conditional probabilities of failure in the tree are greatly simplified.

If the failure of one component is dependent on the failure of another component, then Assumption 7) states that if these components could be arranged vertically in the same branch of the tree, then this dependence would be taken into account. The result of Assumption 7) is that the probability of failure of the end item is dependent on the probabilities of failure of all of the "bottom of the tree" nodes plus the fatal failure node 6.

Assumption 8) could be considered a realistic assumption in that it states that the probability of a fatal failure of the end item must be taken into account. This fatal failure aspect can be very useful in any situation where, for some reason, end items must be permanently removed from operation on a frequent basis.

Assumption 9) emphasizes that the user of the model will in all likelihood make assumptions in determining the

failure rate input data. Assumptions about mean time between failures and the distribution of failure rates about that mean must be made in order to obtain the expected failure rates which, in turn, are used by the model to perform maintenance requirements planning. The section on Data Requirements (Section 4.2) will comment further on the number of assumptions that must be made when organizing failure data of any kind.

Assumption 10) points out that there may be cases where there is more than one level of repair available to perform repairs. Since this model is designed to function in a military logistics setting, the U.S. Navy LOR system would serve as a good example of levels of repair available. The first level of repair in this system is the organizational level which represents repairs that are made, for instance, aboard a ship. For various reasons, certain repairs may not be made aboard ship, but are sent to a second level of repair, the intermediate LOR, which may be located at the ship's home port. The third level of repair handles repairs that cannot be made at the first two levels and is called depot repair. This type of facility represents an even more centralized repair facility that may serve a large geographical area.

Assumption 11) states that there is infinite capacity in the repair facilities. This assumption of infinite

capacity was made in an effort to allow the use of expected MTTR's and to avoid simulating repair channel operations. A major extension of this model would be to model repair channel operations more exactly. This would probably entail the use of an event driven simulation model.

Assumption 12) states that the entire amount of time that a unit is out of operation due to a failure is included in the expected MTTR's. This means that time spent diagnosing the failure, travel time to the repair facility, queue time at the repair facility, time spent waiting on parts, the actual repair time, and transportation time back to the theater of operation is all included in the one MTTR term that is input by the user of the model. This assumption allows the user of the model to take into account the queueing time assumed away by assumption 11. The assumption does require that quite a bit of data be collected regarding the repair process before the user can state an appropriate MTTR.

Assumption 13) is strictly a simplifying assumption that does depart from realism in many cases. Certain end items can require a significant amount of lead time before they can be obtained and placed into operation. This assumption can be circumvented through the proper interpretation of the output results. Since this model is used for planning purposes, the output concerning end items

needed should be interpreted strictly as end items needed and at what time period they are needed. If the user of the output data is familiar with lead times involved in procuring end items, it would be an easy matter to calculate the points in time at which end item procurements should be initiated. The only model output that would change as a result of this adjustment is that procurement costs for end items would be incurred in earlier periods.

Assumption 14) also is a simplifying assumption that can depart significantly from realism in certain cases. Additional scenario generators that operate much the same as the model subroutines that compute procurement amounts, could be appended to the model to simulate cost increases or inflation. The length of period chosen by the user of the model will determine whether or not the treatment of cost as occurring at the end of the period will have a significant effect on the evaluation of logistics policy costs.

Assumption 15) is a limiting assumption in cases where there are different levels of repair that typically have significantly different overhead rates or labor rates from level to level. Additional model subroutines could also overcome this assumption.

Assumption 16) is a limiting assumption because it does not allow the model to evaluate systems of end items

that can continue to operate even when one or more end items short of the number of end items required to meet the demand. Information regarding the number of end items short could be easily derived from output presently calculated by the model. A shortage cost that is sensitive to the number of end items short could be incorporated into the model so that the effects on cost of being a specific number of end items short can be evaluated.

Assumption 17) limits the model's ability to give accurate spare parts inventory requirements for components that do not require the same spare parts at every failure. Again, subroutines to accommodate these situations could be built into the framework of the existing model.

The last assumption, Assumption 18) is a simplifying assumption that again could be overcome by adding a scenario generator that would operate in a manner similar to the model subroutines that compute procurement amounts. The demand being referred to here is the number of end items that must be operating in each failure environment in order to complete their respective missions. Demand is further described in Section 3.3.3 on Parameters.

3.3.3 Parameters. The model requires the input of certain information that generally will not be under the control of the logistics decision maker. It is the task of the user of this model to input these fixed parameters and

then develop the best logistics policy possible given these specific circumstances. Following is a description of the input parameters considered by the model:

Interest Rate - An interest rate must be input so that the time value of money may be considered when calculating logistics policy costs.

End Item Costs - The first cost of end items, the book values of used end items, and the salvage cost of a retired end item all are controlled by market conditions beyond the control of the logistics decision maker. If the decision maker has a selection of end items from which to procure, then the decision maker can have some effect on the end item dollar values that are actually input into the model. For this reason, end item costs are also included under the decision variable of quality which is described in the next section. Another end item cost is the administrative cost of procurement which is a cost probably not under the control of the logistics decision maker.

Spare Parts Costs and Spare Parts Lead Time - The market, not the logistics decision maker will control the price of spare parts and also the lead time involved with procuring spare parts. The decision maker can have some impact on spare parts costs through the brand name part that is selected. This impact is described further under the decision variable of quality in the next section. The

procurement and warehousing costs associated with spare parts is also assumed not to be under the control of the logistics decision maker.

Failure Characteristics of End Items and Components - A brand-dependent mean time between failure and mean time to repair is typical to the brand name end item or spare part procured. The logistics decision maker has no direct control over these characteristics. It is conceivable that preventive maintenance policies could affect the MTBF or even the brand-dependent MTTR. The model has no direct ability to calculate this effect, so if preventive maintenance programs are in place, the effect on MTBF and MTTR must be allowed for before this information is input into the model. Brand-dependent MTBF and MTTR characteristics are the characteristics that are affected by the quality of the end item or spare part being procured.

Demand - Within each failure environment there will be a specified number of end items required in order for a particular mission or service to be performed. Demand establishes a framework within which the decision maker must form his logistics policy. The logistician's success in meeting the demand is measured by the availability of the system(s) of end items.

Availability - A level of availability desired or required is generally established by someone other than a logistics decision maker. The availability and the demand parameters are interrelated.

Repair Costs - Costs associated with the repair of an end item must include labor and overhead associated with diagnosis of the cause of failure, transportation to the repair facility, the actual repair, and transportation back to the site of operation. More specifically, the overhead cost should include cost of operating transportation equipment and the cost of operating the repair facility. Capital costs involved with establishing repair facilities should not be included. If the logistician's purpose in using the model is to obtain accurate cost data, a great deal of data must be collected in order to arrive at an accurate repair cost per period.

Shortage Costs - When the demand is not met, there is a certain penalty incurred. In order to gauge the effects of shortages, a dollar value is assigned to this penalty. Shortage penalties may include lost revenue, lost time, damaged machinery, personnel injury, or loss of life. The decision maker must control the decision variables available to him in order to meet specified levels of availability and keep shortage penalties to a minimum.

3.3.4 Decision Variables. Since logistics models are important planning tools, a great deal of flexibility must be built into the models so that many variables and many alternatives may be considered. Of the variables that must be input into the model, many will be under the control of the decision makers using the model. By playing "what if" games with the model, the effects of various combinations of decision variables on the cost of logistics policy and availability of end items may be evaluated.

Following is a description of the decision variables considered by the model:

Length of Period - In the development of this model it was assumed that in most cases it would be desirable to know the maintenance requirements of end items over consecutive time periods of a particular length. For example, if inventory policy requires reporting spare parts quantities once a month then it would be logical to choose a month time period and let the model not only compute spare parts requirements on a month by month basis, but give repair manpower requirements, failure probabilities, end item availability, and logistics policy costs on the same monthly basis.

Quality - A family of variables including first costs of end items and spare parts, failure characteristics of end items and components (MTBF), repair characteristics of

end items and components (MTTR), and depreciated values of end items all combine to help determine the relative quality of end items and components that are chosen for use. Assuming that there is more than one manufacturer of the end items and components that must be procured, there should be variations in quality and price, and therefore choices to be made by those responsible for procurement. Frisch [5] utilizes quality as a decision variable in his mortality model and discusses the implications of quality decisions.

Procurement Policy - This is another decision variable discussed in depth by Frisch [5]. The method by which end items are procured and therefore the way in which a population of end items is built can have a significant impact on the availability of end items, the cost of logistics policy, spare parts requirements, and repair facility requirements. This model allows the user to be very flexible in selecting procurement policy. The following alternatives are available:

1) Build the population all at one time through one block procurement.
2) Build the population through uniform procurement amounts occurring at equal time intervals.
3) Build the population through decreasing procurement amounts occurring at equal time intervals.
4) Build the population through increasing procurement amounts occurring at equal time intervals.
5) Build the population strictly at the discretion of the user by procuring designated amounts at specified points in time.

Further variations possible include: building the population of end items infinitely, building to an upper population limit, or replacing end items only upon fatal failure in order to maintain a specified population size.

Size of the End Item Population - This decision variable is intertwined with several of the other decision variables. For example, if the size of the end item population is varied, some other variable will probably have to change if there is an availability requirement to be met. The decision maker can choose to have a smaller population, but this will probably mean that better quality end items that fail less frequently and are easier to repair must be utilized. Or it may be possible to survive with a smaller population if the MTTR per end item can be reduced significantly by implementing changes at the repair facilities.

Retirement Age - A very important decision generally at the discretion of the logistician is, when should end items be retired from service? Setting the retirement age of end items must be based on several different factors. A point may be reached where it is no longer economical to continue to repair an end item. The economics are affected by the required availability of the population of end items, the cost of parts, the cost involved with repair, and the first cost of end items. A second factor that must

be considered is the fact that populations of end items become obsolete. If there is the option to gracefully retire end items over a period of time, varying the retirement age variable could offer some insight on the cost involved with some type of phase out policy.

Mean Time to Repair - Since the MTTR in this model includes such a wide range of events, there would be many opportunities for the decision maker to decide to make certain changes in repair policies that would affect availability of the end items, spare parts requirements, repair facility manpower requirements, and logistics policy costs. Any changes made in repair policies would have to be documented separately by the user of this model as the model only tracks the final MTTR value for components in the maintenance tree.

Inventory Policies for Spare Parts - The logistician will probably have some control over lot sizing techniques used for spare parts. For large populations of end items, the amount of spare parts stocked can carry a significant cost as far as inventory costs are concerned or a significant penalty if down time of end items is increased because of part shortages.

It is recognized that in most situations certain of the above variables are fixed and may not be changed by the decision maker. It is, however, the job of the decision

maker to formulate the best logistics policy under the given circumstances. Varying the decision variables available to the logistician and weighing the effects of changes in these variables on availability, logistics policy cost, repair manpower requirements, and spare parts requirements will help the logistician to approach the most effective and efficient logistics policy possible.

3.3.5 Calculations. As would be expected, many calculations and approximations are used by the model to produce maintenance requirements. This section will detail the calculations used. The variable names and symbols used here will not necessarily match the variable names used in the computerized version of the model.

3.3.5.1 Probability of Failure. A probability of failure for each period must be calculated for the component represented by each "bottom of the tree node". A probability of failure for each period also must be calculated. These probabilities of failure either may be input directly by the user of the model or may be generated by the model. If the model is to generate these probabilities, one of the general failure distributions, identified by the user as appropriate to a specific component, will be used. The user programs the model by selecting a failure distribution and inputting an MTBF for each component. There are five choices of failure

distributions that may be used to compute failure probabilities.

Following is a description of the equations used by each of the five failure distributions to compute the probability of failure for the component represented by each "bottom of the tree node". The user of the model also assigns a failure distribution to the fatal failure node, $NODE_6$. See Figure 3.6 for a graphical representation of the failure distributions employed.

List of Variables:

$1/\lambda$ = Mean Time Between Failures for the component represented by the "bottom of the tree" node. This value is input by the user of model. is the failure rate of the component.

e = irrational number = 2.718282.

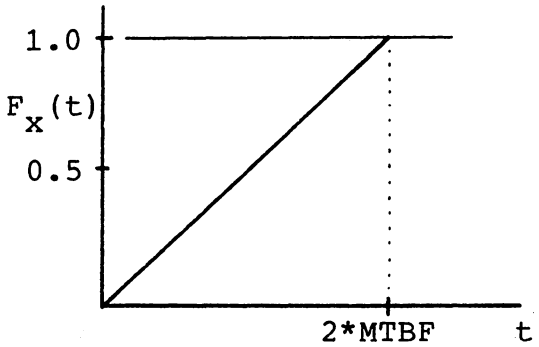
t = # of periods that the end item, of which the component is a part, has been in operation.

$F_x(t)$ = cumulative probability of a failure of the component at some point in time t.

$NODE_i$ = probability that a failure of the component will occur during the given time period.

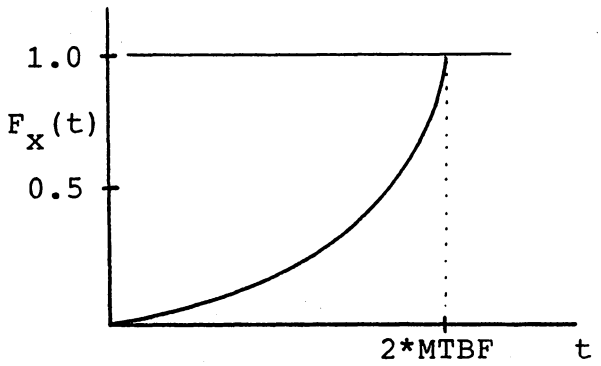
i = "bottom of the tree" node number as identified by position in maintenance tree.

PI = irrational number = 3.141593



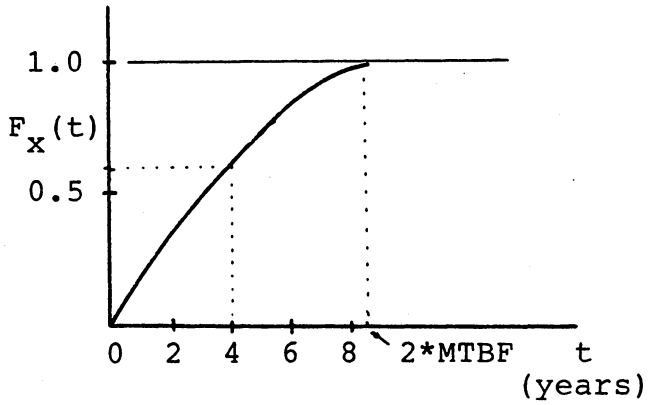
graph a.

Uniform



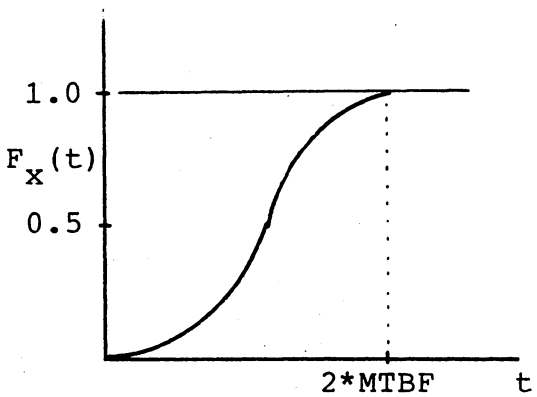
graph b.

Increasing Exponential



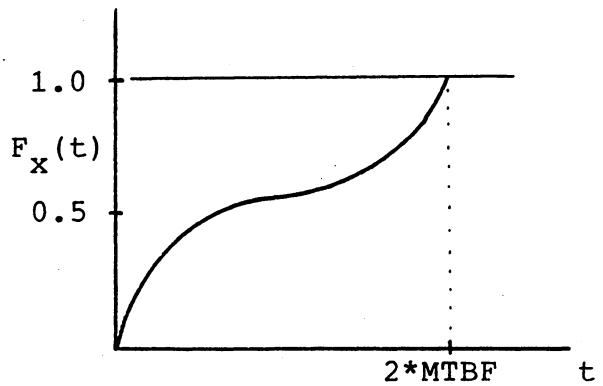
graph c.

Exponential



graph d.

Approximate Normal



graph e.

Bathtub

FIGURE 3.6. CUMULATIVE FAILURE PROBABILITY DISTRIBUTIONS

List of Equations:

Uniform Distribution:

$$F_x(t) = \frac{\lambda t}{2}$$

Exponential Distribution:

$$F_x(t) = 1 - \frac{1}{e^{\lambda t}}$$

Increasing Exponential Distribution:

$$F_x(t) = \frac{1}{e^{1/\lambda}} * e^{t/2}$$

Approximate Normal Distribution:
(Weibull with shape parameter = 3)

$$F_x(t) = 1 - \frac{1}{e^{(\lambda t)^3}}$$

Bathtub Distribution:

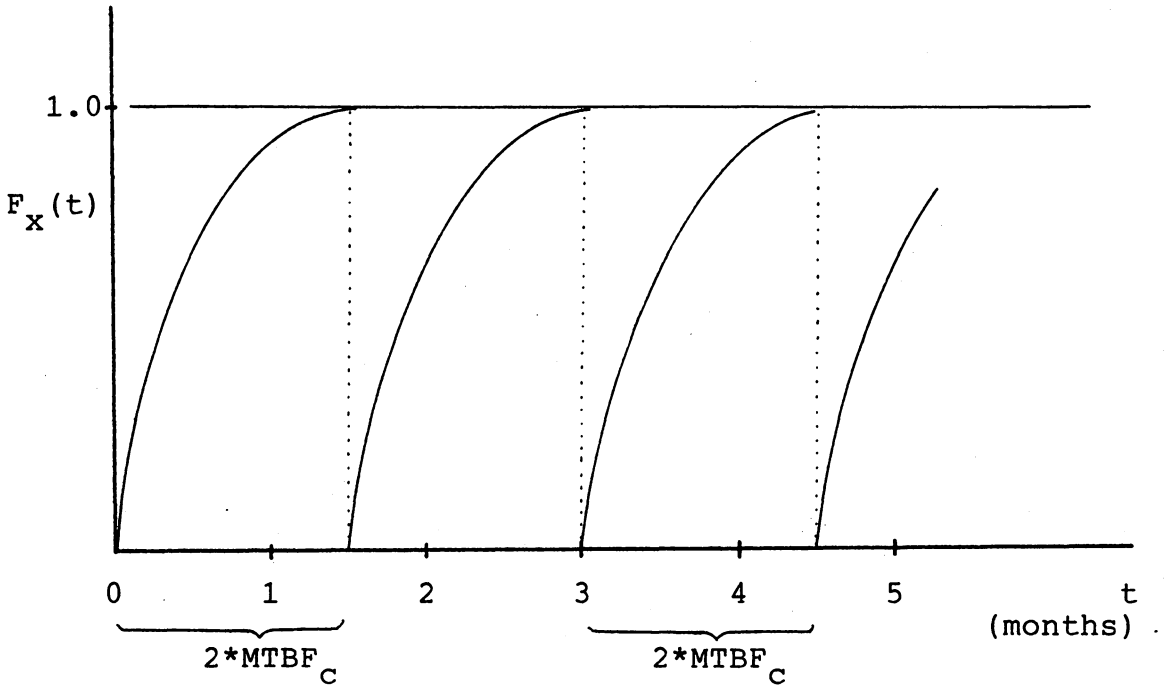
$$F_x(t) = \frac{1}{e^2} * e^{\lambda t}$$

In the Uniform, Exponential Increase, and Bathtub distributions, the coefficients have been chosen such that when an end item has been in operation for a period of time equalling twice the MTBF of the respective function, the probability of a failure of the component represented

becomes 1.0. The Exponential distribution is a tangential function that approaches a failure probability of 1.0 as time approaches infinity. The approximate normal shape is produced by using a Weibull distribution with a shape parameter of 3. This produces a probability density function that is shaped similarly to the Normal but skewed slightly to the right. The cumulative distribution function is tangential and approaches 1.0 as time approaches infinity.

Even though the Exponential and Weibull cumulative probability functions theoretically never attain the probability of 1.0, it is assumed in the model that the probability of failure = 1.0 for a given component at the end of that component's expected life span. A component's expected life span is assumed to be equal to twice the MTBF for the component. After the end item has been in operation a period equal to twice the component's MTBF, the probability distribution for this component regenerates. Figure 3.7 illustrates this concept.

The Uniform distribution would be suitable to use in situations where the component being evaluated has a constant probability of failure from period to period. The Exponential distribution is suitable in



$MTBF_c$ = .75t = .75 months

$MTBF_c$ is the mean time between failures for component c.

t = 0 is the point in time at which the end item, of which component c is a part, first went into operation.

The cumulative failure probability distribution for component c regenerates after reaching an upper limit of $2*MTBF_c$.

FIGURE 3.7.

UPPER LIMIT OF CUMULATIVE FAILURE PROBABILITY DISTRIBUTIONS

situations where the probability of failure decreases exponentially with time. The increasing exponential distribution represents the opposite scenario where the probability of failure increases exponentially with time.

The last two curves also represent opposite situations. The approximate normal distribution would apply where failure rates are low in the early and late stages of the life of a component. The bathtub distribution applies in situations where there are higher failure rates during the burn-in period of a new component, lower rates of failure during the majority of the life of the component and then higher rates again during the later stages of component life.

The user of the model will be able to obtain inventory requirements based on a time period of his choice. This may require that given failure rate information be adjusted to meet the chosen time scale. As an example, consider a car battery that has the cumulative failure probability distribution shown in graph c of Figure 3.6. The battery is a part of the maintenance tree for an automobile. The battery's position in the automobile maintenance tree is shown in Figure 3.5. Note that information regarding mean time to repair and level of repair for battery replacement are noted in that figure. A large fleet of automobiles is being operated and the logistician must know how many

batteries to stock on a weekly basis. Graph c is an Exponential probability distribution, so the following calculations may be used to find the mean time between failure for each battery assuming that $F_x(4) = 0.6$.

$$F_x(t) = 1 - e^{-\lambda t}$$

$$0.6 = 1 - e^{-4\lambda}$$

$$1/\lambda = \text{MTBF} = 4.365 \text{ years}$$

To find the probability of failure of a battery during week one, the following calculation would be used:

$$p(f)_1 = 1 - e^{-(1/52)/4.365} = .004$$

To find the probability of failure of a battery during the first two weeks, the following calculation would be used:

$$p(f)_{1,2} = 1 - e^{-(2/52)/4.365} = .009$$

The probability of failure of a battery during week two alone is:

$$p(f)_{1,2} - p(f)_1 = .009 - .004 = .005$$

The probability of failure of a battery during subsequent weeks would be calculated using this logic.

<u>3.3.5.2 Probability of Failure of End Items.</u> The probability of failure of the end item is conditional on the probability of failure of all of the "bottom of the tree" nodes plus the fatal failure node, $NODE_6$. The probability of failure for nodes located at intermediate levels of the tree (between the "bottom of the tree nodes" and the end item node) are conditional on the probability

of failure of nodes connected directly beneath them.

Steps required to compute $NODE_1$ (probability of failure of end item) are:

Step 1: Compute $NODE_i$ of the bottom of the tree nodes as described in previous section.

Step 2: Use conditional probabilities, assuming horizontal independence of failure in the maintenance tree, and work towards the top of the tree. The formula listed below would be used to calculate $NODE_1$ once the conditional probability calculations have proceeded to that level of indenture in the tree. See the Venn Diagrams in Figure 3.8 for a graphical explanation of the formula.

$$NODE_1 = NODE_2 + NODE_3 + NODE_4 + NODE_5 + NODE_6$$

$$- \quad [NODE_i * NODE_j] \quad (i,j = 2,..,6)$$

(all possible combinations i and j)

$$+ \quad [NODE_i * NODE_j * NODE_k] \quad (i,j,k = 2,..,6)$$

(all possible combinations i, j, and k)

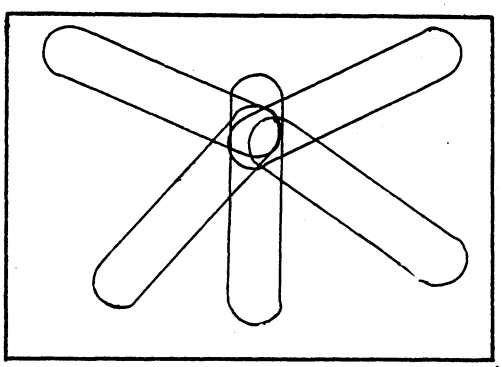
$$- \quad [NODE_i * NODE_j * NODE_k * NODE_m]$$

$$(i,j,k,m = 2,..,6)$$

(all possible combinations i,j,k, and m)

$$+ \quad [NODE_2 * NODE_3 * NODE_4 * NODE_5 * NODE_6]$$

VENN DIAGRAM



Each ellipse represents the probability of failure of a separate component. When the components are assembled together they form an end item. Since components fail independently, the following combinations of component probabilities must be evaluated and combined in order to determine the probability of an end item failure.

$$f_1(t) \cup f_2(t) \cup f_3(t) \cup f_4(t) \cup f_5(t) =$$

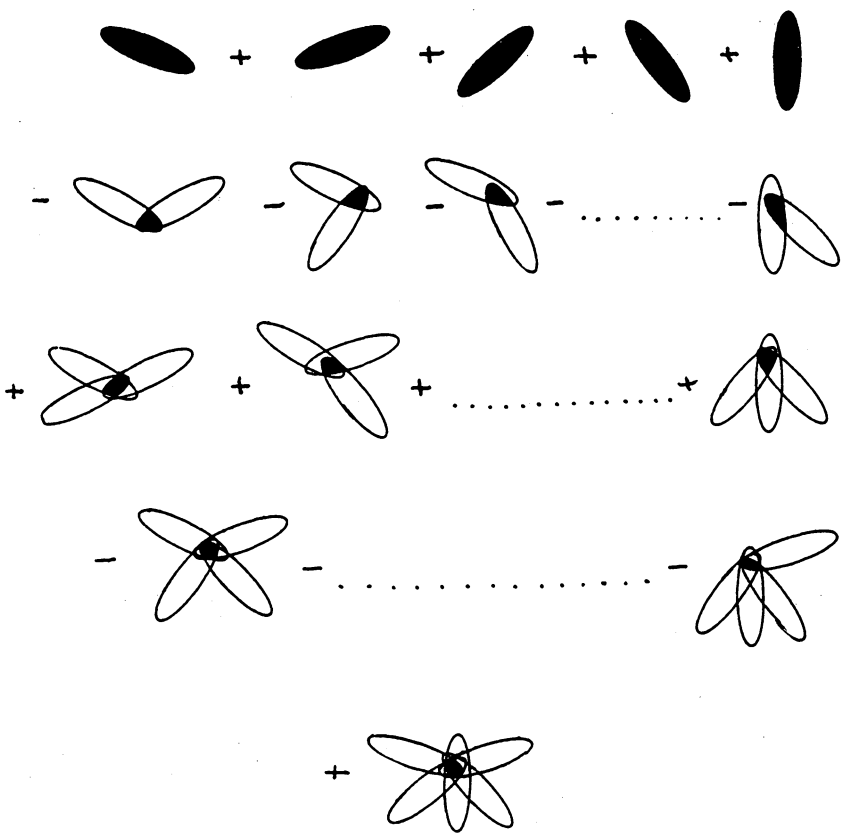


FIGURE 3.8. COMPUTATION OF CONDITIONAL PROBABILITIES

3.3.5.3 End Item MTTR. The mean time to repair end items must be known in order to compute availability and repair facility manpower requirements. This value is computed using the MTTR values entered for the individual components represented by "bottom of the tree" nodes. Following are the variables and equations used to compute end item MTTR:

List of Variables:

$MTTR_i$ = MTTR if the component represented by node i fails. MTTR = $1/\mu$ where μ = the rate of repair. (MTTR is input by the user of model.)

$NODE_i$ = probability of failure in a given time period of the component represented by the bottom of the tree node i.

$MTTRSUM_s$ = weighted sum of MTTR's for one end item in a given stream s.

$MTTRAVG_s$ = average amount of time spent in repair during a given period by an end item in stream s. End items in different streams may have different $MTTRAVG_s$.

B = total number of "bottom of the tree" nodes

Equations:

$$MTTRSUM_s = \sum_i [MTTR_i * NODE_i]$$

$$MTTRAVG_s = MTTRSUM_s / B$$

3.3.5.4 Availability. Now that the probability of a failure of the end item and the amount of time that an end item spends in repair have been calculated, all of the data is in place for the availability calculations. Following are the variables and equations used to compute end item availability.

List of Variables:

N_s' = # end items in operation in stream s at the beginning of time period, s = 1,..., total # streams.

N_s'' = # end items in operation in stream s at end of time period.

N_s = average # end items in operation in stream s during time period.

N_e = total # end items in operation in environment e during time period, e = 1,...,total # environments.

R_e = total amount of time spent in repair by end items in environment e.

$MTTR_e$ = MTTR per end item in environment e.

$NODE_{1s}$ = probability of failure of an end item in stream s. The probability of fatal failure is not considered. The effect of fatal failure on availability is taken into account when determining the average number of end items in operation in a stream during a given time period.

$SUM_{(f)_e}$ = weighted sum of the chance of failure of end items in environment e.

$P(f)_e$ = average chance of failure of an end item in environment e.

$P(f)_e'$ = average chance of failure of an end item in environment e during a period of time equal to $MTTR_e$.

LTP = length of a time period (input by user).

The equations are as follows:

$$N_s = [N_s' + N_s''] / 2$$

$$N_e = \sum_{s \text{ in } e} N_s$$

$$R_e = \sum_{s \text{ in } e} [N_s * MTTRAVG_s]$$

$$MTTR_e = R_e / N_e$$

$$SUM(f)_e = \sum_{s \text{ in } e} [N_s * NODE_{1s}]$$

$$P(f)_e = SUM(f)_e / N_e$$

$$P(f)_e' = [MTTR_e / LTP] * P(f)_e$$

In order to calculate the availability of end items, some means must be derived for calculating the probability of being short one or more units. The availability will be:

$$A_e = 1 - \sum_{x=1}^{D_e} M_x$$

D_e = the demand in environment e.

M_x = the probability of x end items out of operation. Another way to write this would be:

$$A_e = \sum_{x=0}^{k} M_x \qquad \text{where } k = N_e - D_e$$

It is assumed that when the number of end items in operation in a given environment becomes less than the demand for end items in that environment, then at that point in time, the availability of that system of end items is 0. The availability calculations determine the probability that, for a given time period, the number of units in operation is greater than or equal to the demand. This probability represents the availability of end items versus the demand. If the number of end items out of operation at any given time remains between 0 and k for an entire time period, then the availability for that time period would be 100%.

The chance of failure of any one end item is independent from the chance of failure of any other end item within or outside of the same stream. Further, because of the homogeneity of the streams, all end items within a stream have the same probability of failure. In the above calculations the value $P(f)_e$ establishes an average probability of failure for all end items operating in the same failure environment, so all end items operating

in the same failure environment in effect have an equal probability of failure. There are two things that can happen to an end item during a given period: either the end item experiences a failure, or the end item experiences no failure.

With the characteristics of the foregoing paragraph in mind, it can be assumed that the probabilities of $0,1,2,\ldots,N_e$ simultaneous failures of end items is binomially distributed. The binomial distribution as applied to a population of end items would be defined as follows:

$$\sum_{x=0}^{n} b(x;\ n,p) = 1$$

n = total number of end items in the population.

p = probability of failure of an end item.

To calculate availability, we only need to know the probability of 0 to k end items being out of operation simultaneously. Thus the binomial calculation for availability becomes:

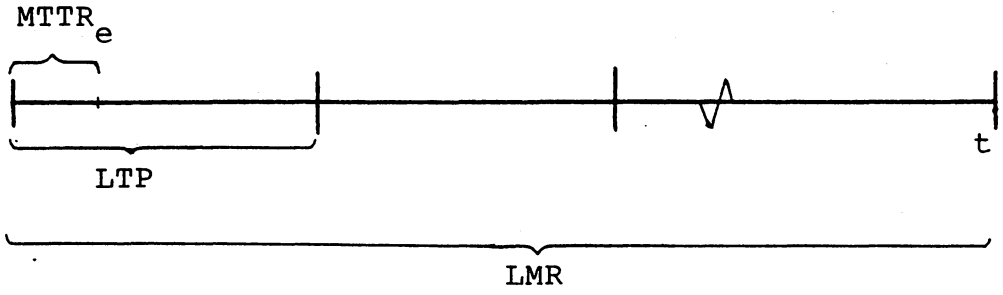
$$A_e = \sum_{x=0}^{k} b(x;n,p)$$

$$A = \sum_{e} [(A_e * N_e) / N]$$

A_e	= the availability of end items operating in environment e.
k	= $N_e - D_e$
D_e	= the demand for end items operating in environment e.
p	= $P(f)_e'$
A	= availability of the entire population of end items.
N	= total # of end items in operation during a given period.

The $P(f)_e'$ term needs some explanation. Referring to Figure 3.9 it can be seen that the average MTTR ($MTTR_e$) is not necessarily equal to the length of a time period (LTP). In order to find availability, it is necessary to evaluate the probability of simultaneously having more than k units out of operation. By evaluating a time period of length $MTTR_e$, the probability of multiple failures occurring at the same time can be approximated. In order to calculate $P(f)_e'$, it must be assumed that the probability of failure is linear across any given time period (LTP).

Graph a. of Figure 3.10 illustrates the method used to calculate availability when at some point during a time period, the total number of end items operating in an environment becomes less than the demand of end items in

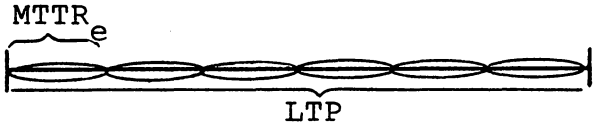


MTTR$_e$ = the average MTTR for an end item operating in a particular failure environment.

LTP = length of one period

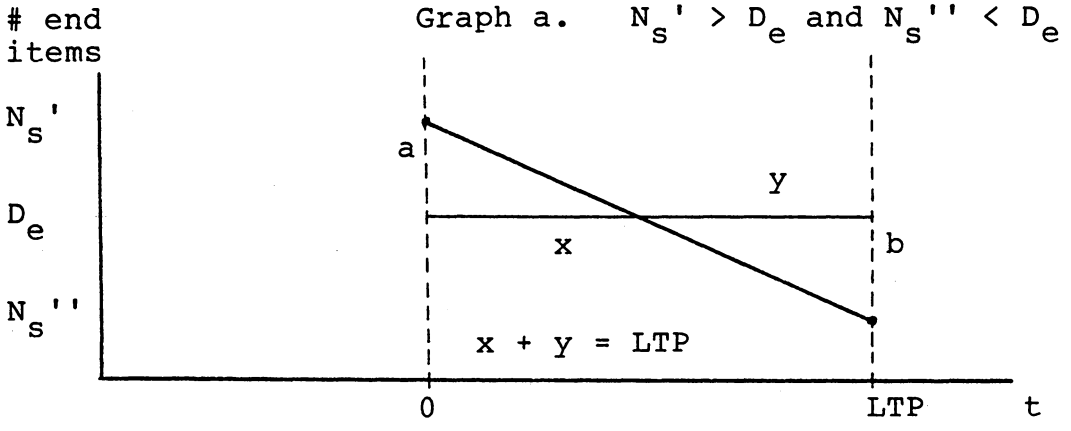
LMR = length of model run

If the probability of failure of end items was evaluated over one entire period, LTP, the evaluation of simultaneous failures of end items would be difficult. Difficulty occurs because there is a probability that events such as the following could occur. If MTTR = 1/6 * LTP then it is possible that 6 end item failures could occur during the time period, LTP, without an overlap of "out of operation time".



If the evaluation period is shortened to MTTR$_e$, the model assumes that if more than one end item failure occurs during the period, MTTR$_e$, then simultaneous failure has occurred. The availability subroutine evaluates the probability of 0 to k failures occurring within a time period of length MTTR$_e$.

FIGURE 3.9. EXPLANATION OF P(f)$_e$

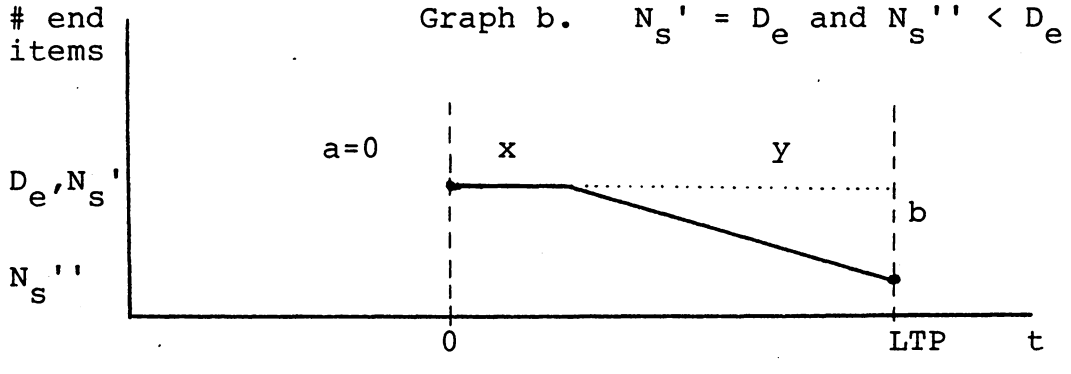


$a = N_s' - D_e$ $\quad b = D_e - N_s''$ $\quad x = \dfrac{(a/b)}{(a/b + 1)} * LTP$

LTP = one time period $\quad A$ = Availability for LTP
A_x = Availability for x $\quad A_y$ = Availability for y

A_x is calculated as outlined in Section 3.3.5.4.

$A_y = 0$ $\quad$ then $\quad \boxed{A = \dfrac{x}{x+y} * A_x}$



$b = D_e - N_s''$ $\quad x = \dfrac{1}{b + 1} * LTP$

$A_x = \sum_{x=0}^{0} b(x;\ n,p)] * \dfrac{x}{x+y}$: $\quad n = N_e, \quad p = P(f)_e'$

$A_y = 0$ $\quad$ then $\quad \boxed{A = A_x}$

FIGURE 3.10. AVAILABILITY WHEN $N_s'' < D_e$

that environment. Total number of end items operating includes the end items actually in operation plus end items in repair. During the course of a time period, fatal failure is the only cause of the reduction in total number of end items.

If at the start of a period, the number of end items in an environment is less than the demand for that environment, the availability for that period is 0. This must be the case since it was assumed that the number of end items in operation must meet the demand for that particular system of end items in order for that system to be operable. If at the start of a period, the number of end items in an environment is equal to the demand for end items in that environment, it is assumed that the fatal failure(s) of end items occurring within the period occur at equal intervals. Then, the time to first failure is calculated as shown on graph b. in Figure 3.10.

3.3.5.5 Procurement Amounts. As described previously, the user of the model may select one of five different procurement policies to follow when procuring end items. Besides the option of procuring based on the period by period inputs of the user, the model could be programmed to procure in uniform amounts, block amounts, increasing amounts, or decreasing amounts. The model must perform calculations of the amount to procure if an increasing or

decreasing amount policy is selected. The user inputs a percentage increase or decrease and the model computes the procurement amount simply as:

increase:

$$P_p = P_{p-1} + [P_{p-1} * (g/100)]$$

decrease:

$$P_p = P_{p-1} - [P_{p-1} * (g/100)]$$

P_p = amount to be procured on the pth procurement.

P_{p-1} = amount procured on the p-1 procurement.

g = percentage increase or decrease in procurement amount input by user.

Under the uniform procurement options, P_p is defined by the user at the beginning of the program and remains constant except as noted below. Under the block procurement policy there is of course only an initial block procurement amount defined by the user. In the four programmable procurement policies, the user may specify that an upper limit on the number of end items in the population be maintained. In this case, the model checks the diferrence between the upper limit and the number of

units in operation at the end of the period and compares this to P_p before determining the actual amount to procure:

$$\text{difference} = \text{UPLIMIT} - [N - (N * \text{NODE}_6)]$$

UPLIMIT = upper population limit input by user.

N = total number of end items in operation.

NODE_6 = probability of failure occurring within the given period.

If the difference is greater than P_p, then the amount procured is P_p. If P_p is greater than the difference, then the difference is the amount actually procured and P_p is set equal to the difference. The amount procured is rounded to the nearest integer value.

3.3.5.6 Spare Parts Requirements. Totals of each type of spare part required to support the population of end items are calculated for each bottom of the tree component. This information is provided for each stream and for the population in total at the end of each period. From these values, the period warehouse space requirements at each level of repair are derived.

$$Q_j = \text{NODE}_i * n * N_s$$

$$W_j = Q_j * w_j$$

$NODE_i$ = the probability of failure for the given time period of the component represented by "bottom of the tree" node i.

n = the total number of spare part j required to support the "bottom of the tree" node component to which it is attached.

N_s = the average number of end items in operation in stream s during the given time period.

Q_j = the total number of spare part j needed to support stream s during the given period.

w_j = the total amount of warehouse space in cubic feet taken up by one of the spare parts j.

W_j = the total warehouse space in cubic feet required to store Q_j of part j.

The sum of the Q_j required for each stream gives the number of spare part j needed to support the total population for one time period. Summing the W_j of all spare parts by repair level gives an indication of the total warehouse space that must be reserved for parts at each LOR.

If the user of the model instructs the model to calculate inventory levels and generate order policies and costs based on EOQ, the following calculations are performed:

$$Q_j' = \sqrt{\frac{2 * PC_j * Q_j}{HC_j}}$$

$$L_j' = Q_j * LT_j$$

$$DAYS_j = Q_j / Q_j'$$

$$TC_j' = (C_j * Q_j) + \sqrt{(2 * PC_j * HC_j * Q_j)}$$

Q_j = total number of spare part j needed to support a given stream of end items during a given period.

PC_j = procurement cost incurred every time a procurement of spare part j is initiated.

HC_j = the cost to warehouse one unit of spare part j for one period. This includes the cost of space occupied, warehouse facility labor and overhead costs, and cost of equipment supporting warehouse.

Q_j' = the economic order quantity for spare part j.

LT_j = the lead time in days for spare part j.

L_j' = the economic order level for part j.

$DAYS_j$ = the number of days between orders of spare part j.

C_j = the price of one unit of spare part j.

TC_j' = the cost of the EOQ policy for the total number of spare parts j required to support the given stream of end items for the given period.

3.3.5.7 Cost of Logistics Policy. The cost of logistics policy is calculated per stream per period and then totalled for the period. At the end of the model run a final total cost of logistics policy for the time periods covered by that model run is given. The period costs are converted to present worth values, utilizing equations detailed below. After the model has evaluated the last time period, all present values are summed and converted to a period equivalent cost. This period equivalent cost (average logistics policy cost per period) should provide a figure that can be easily compared against different logistics policy period equivalent costs computed on other runs of the model.

The factors for converting to present value and to annual equivalent values, as taken from Thuesen, Fabrycky, and Thuesen [34] are as follows:

$$P = F(P/F\ i,n) = F * \frac{1}{(1+i)^n}$$

$$A = P(A/P\ i,n) = P * \frac{i * (1+i)^n}{(i+1)^n - 1}$$

P = present worth value

A = annual equivalent cost

F = future value

i = time value of money (for example, the current interest rate)

n = time periods

The costs considered in developing the logistics policy cost are:

Capital Cost - The first cost of an end item, the book value received for a failed end item, and the salvage value received for a retired end item are factors considered in calculating the capital cost. Capital cost is computed on a per period basis using the following equation:

$$CC_e = (P_{pe} * CC') - (NF_e * SV') - (NR_e * BV_e')$$

CC_e = capital cost for end items in environment e.

P_{pe} = number of end items procured and placed in environment e.

CC' = first cost of an end item.

NF_e = number of end items failing fatally during period.

SV' = salvage value for a failed end item.

NR_e = number of end items retired at end of period.

BV_e = book value for retired end item from environment e.

Repair Cost - The repair cost per period unit (see Section 3.2 for a definition of period unit) times the total amount of time that units spend out of service gives the repair cost. Repair cost per period is calculated using the following equation:

$$RC_e = RC' * R_e$$

RC_e = repair cost per period for end items in environment e.

RC' = repair cost per period unit. See section 3.2 for definition of period unit.

R_e = total amount of time spent in repair by end items in environment e.

Shortage Cost - The shortage cost per period unit times the amount of time the demand is not met gives shortage cost. Shortage cost per period is calculated using the following equation:

$$SC_e = (1 - A_e) * SC' * LTP$$

SC_e = shortage cost per period for in environment e.

A_e = availability of end items in environment e.

SC' = shortage cost per period unit.

LTP = number of period units in one period, i.e. length of time period.

Inventory Cost - The sum of the TC_j' described in the previous section constitute the inventory cost. The

following equation is used to calculate inventory cost:

$$IC_e = \sum_j TC_{je}'$$

IC_e = total inventory cost per period for end items in environment e.

TC_{je}' = inventory costs for spare part j which is a stockable item for an end item in environment e.

Procurement Cost - The administrative cost incurred when end items are procured constitutes procurement cost. This cost is a total period cost and is not allocated as an environment cost. The following equation is utilized:

$$\text{If } P_t > 0$$

$$PC_t = 1 * PC'$$

$$PPC_t = PC_t \ (P/F\ i,n)$$

P_t = number of end items procured at beginning of period t.

PC_t = procurement cost for period t.

PC' = procurement cost per procurement.

PPC_t = present worth of procurement costs in period t.

(P/F i,n) = factor to convert from end of period cost to present worth value.

i = interest rate adjusted to period length.

n = t (number of time periods)

Total Cost - The model produces a total period cost for each environment. These figures are converted to

present worth values and summed. The sum of period present worth values is then converted to a period equivalent cost and total model run costs for each environment and for the entire population of end items are calculated. The following equations are used:

$$TC_{te} = CC_e + RC_e + SC_e + IC_e$$

$$PTC_{te} = TC_{te}\ (P/F\ i,n)$$

$$ATC_e = [\sum_{t=1}^{LTP} PTC_{te}]\ (A/P\ i,n)$$

$$TC = \sum_e ATC_e + [\sum_{t=1}^{LTP} PPC_t\ (A/P\ i,n)]$$

TC_{te} = total cost of logistics policy for environment e in period t. Procurement costs of end items are not included.

TC = population total cost of logistics policy.

PTC_{te} = present worth value of period total costs.

ATC_e = period equivalent cost of logistics policy for environment e.

$(A/P\ i,n)$ = factor to convert from present worth to period equivalent cost.

i = interest rate adjusted to period length.

n = LTP

3.3.6 Measures of Performance. Two primary measures of performance are utilized. First, when given the appropriate data, the model returns the costs associated with the input logistics strategy. The previous subsection

illustrates the computation of period by period costs for each environment in which end items operate. Also, the computation of period equivalent costs per environment and a total logistics policy period equivalent cost are detailed. This type of data can provide the logistics decision maker with a valuable measure of performance for evaluating end items operating in different environments and under different logistics policies. The second performance measure provided by the model is availability. The model provides the availability of end items in each environment and a total population availability. Again, the logistics decision maker has valuable data for measuring performance.

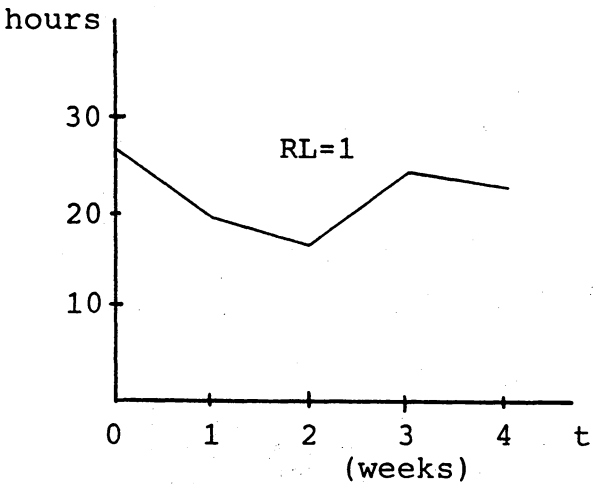
Secondary measures of performance include inventory demand profiles, maintenance facility projections, and manpower projections as determined by the model. Knowledge of inventory demand profiles becomes important if there are warehousing constraints or procurement lot sizing constraints. Maintenance facility projections and manpower projections become important in the face of limited space for maintenance facilities and limited manpower for maintenance functions.

The primary measures of performance can also function as constraints. If a budget constraint is imposed, the logistics strategy must be varied to meet the constraint.

Unless excellent data is available for computing the cost of logistics policy, the cost performance measure is better used as a comparison tool when evaluating one logistics policy versus another. If a demand must be met a certain percentage of the time, then an availability constraint exists and logistics strategy must be varied to meet the constraint. Knowledge of inventory profiles, maintenance facility projections, and manpower projections again become important in the face of constraints such as budget or availability constraints.

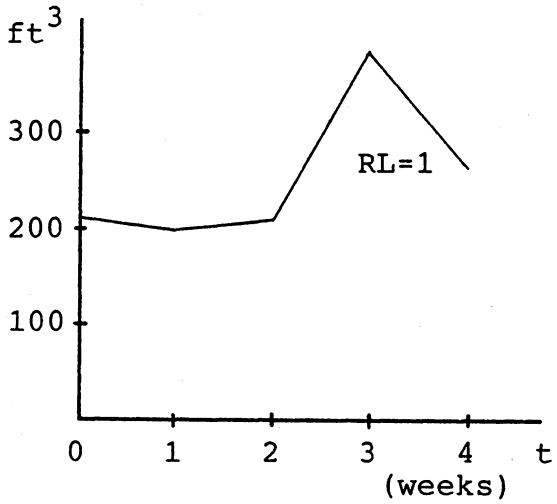
3.4 USES OF THE MODEL.

The model should be able to satisfy many of the logistician's planning needs. In general, the model will generate the maintenance requirement demand of a finite population of repairable items. This will include spare parts requirements in the form of EOQ inventory level policies, manpower hours required to staff maintenance facilities, warehouse space required for spare parts inventories, the costs associated with the operation and maintenance of the designed logistics system, and the availability of units under the proposed logistics policies. The computerized version of the model produces the diagrams shown in Figure 3.11. The performance measures described in the previous section serve as output to be used by the logistician in judging the effectiveness



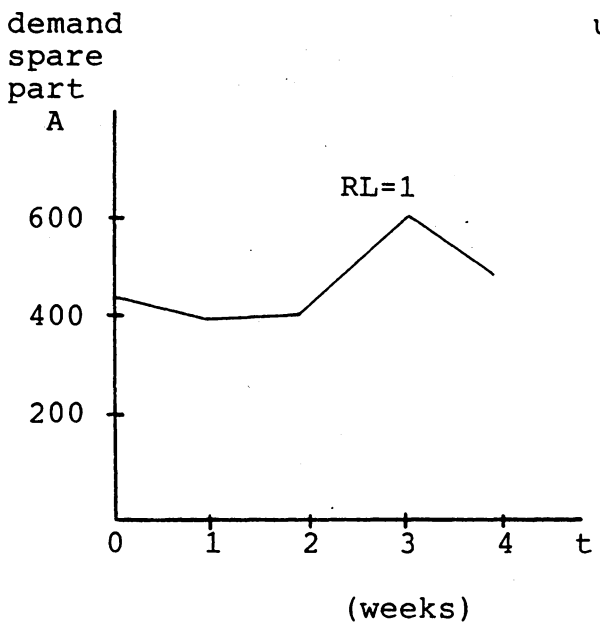
graph a.

Manpower Requirements



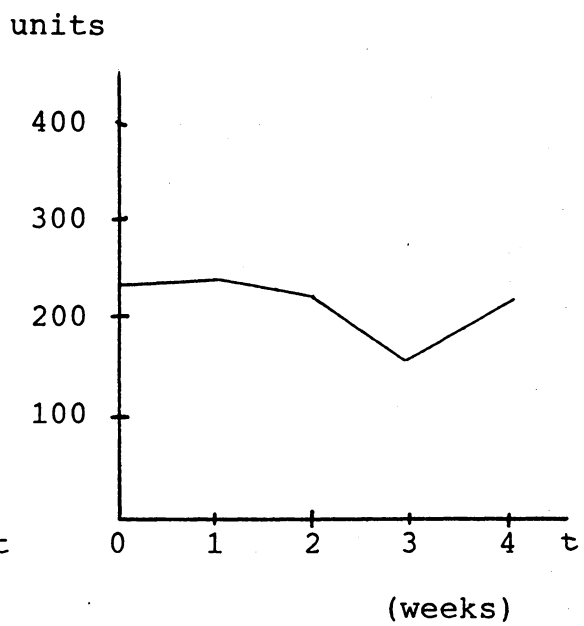
graph b.

Warehouse Requirements



graph c.

Inventory Requirements



graph d.

Availability

FIGURE 3.11. MODEL OUTPUT

of proposed logistics policy. These performance measures allow the logistician to evaluate the effects of cost and availability constraints. Also warehousing, personnel, and procurement policy constraints can be evaluated.

The use of various methods of spare parts inventory lot sizing may be evaluated by adding subroutines to the model that produce data similar to the EOQ routine already incorporated into the model. Also, warehouse and budget constraints could be considered versus varying methods of lot sizing. In situations where spare parts are manufactured in house, lot sizing of manufacturing batches could be planned using the spare parts requirements generated by the model.

The model is structured to handle more than one level of repair simply by specifying in the maintenance tree what type of repair facility is required for a specific repair. The model produces spare parts requirements based on the repair facility at which the repair must be made so that spare parts inventory requirements for separate levels of repair are totaled separately. By comparing the spare parts requirements at the various levels of repair, common spare parts requirements can be identified. This information can assist in the establishment of centralized procurement policies which could save on spare parts prices and procurement costs.

The model considers the MTTR associated with the repair being made at the specified repair facility. From this information, the expected demand on various repair facilities is calculated. Knowing these repair facility requirements can assist in the design and construction of, or alteration of, repair facilities. Knowing the requirements ahead of time would also help with staffing and training. Since nonsteady-state conditions are being evaluated, the characteristics of the repair channels can be adjusted to keep pace with the needs of the population. The same adjustments may be made to the spare parts inventory. Only enough inventory to meet the requirements will need to be procured.

The logistics decision maker can use the model as an aid in making traditional repair or replace decisions. By varying the retirement age of end items the user of the model can view the cause and effect relationships between retirement age and cost, availability, spare parts inventory requirements and maintenance facility space and manpower requirements. Decision variables besides the retirement age, such as the total number of units in the population may also be varied. The user of the model will observe that adjusting one decision variable in response to changes in another decision variable can lessen negative impact on the performance measures of the model.

Quality and its effects on the performance measures can be modeled by using the model to evaluate the different failure rates common to different quality equipment. Also the effects of various procurement policies on performance measures may be evaluated. The model could be used as a tool to forecast how many units would have to be procured each year in order to maintain a certain population size or a population growth rate. These capabilities, in conjunction with the model's ability to forecast maintenance requirements, make the model an excellent capital and operating budgeting tool.

In order to make the model easier for the logistician to use, procurement policy scenario generators are included. The generation of these scenarios means that in order to achieve more realistic scenarios, the user of the model does not have to input data at each time period. The computer is generating the period by period "realism". This scenario concept could be extended to make the model less cumbersome for the user and possibly more realistic in its outputs. See Sections 4.5 and 6.3 for discussions of model extensions.

Since the concept of nested Markov chains is applied, nonhomogeneous populations could be evaluated as long as all units within a particular chain possess similar failure characteristics, and as long as the same maintenance tree

can be attached to each end item. The populations could be called nonhomogeneous because of several factors: different brand names may be present, improved models may be procured as replacements for original units, or units may be employed in different environments. As a hypothetical example of the "different environments" idea, consider a scenario in which a logistician for a widespread military research division purchases 200 batteries in 1980. Half of the batteries are immediately placed in use at a research facility in Antarctica and half of the batteries are immediately placed in use at a research facility in Brazil, near the equator. In the proposed logistics model, the batteries sent to Antarctica would form a stream with a Markov chain of failure probabilities and the batteries sent to Brazil would form a separate stream with its own Markov chain of failure probabilities. The separate chains are required because the two sets of units would obviously have different failure rates. Batteries can last considerably longer in a tropical climate as compared with a frigid climate. The batteries would thus be operating in two different failure environments.

The concept of preventive maintenance may be handled by establishing a life cycle for the major component parts considered in the nested Markov chains. This "preventive maintenance" would take place in the form of replacing

major component parts at prescribed intervals. The model will not presently evaluate the concept of preventive maintenance. Extensions would be required.

With the division of the population into streams, each with its own Markov chain, a division of cost calculation is also accomplished. The costs for operating each stream of equipment are calculated, thus giving a comparison of costs between streams and failure environments within the population.

Finally, the model may be used to illustrate just how effective a logistician can be when the correct data base exists. The model will in effect be used as a demonstration of data requirements necessary for more efficient logistics modeling.

3.5 LIMITATIONS OF THE MODELING APPROACH.

In approaching the problem of maintenance requirements planning by investigating both transient and steady state modes of operation, two major limitations have been encountered. First, trying to model transient state conditions creates the need to consider an overwhelming number of possible combinations of events. The need for simplifying assumptions manifests itself. The second major limitation encountered in this transient state modeling approach is a problem probably encountered by all logistics model developers. All of the data necessary to make

accurate model maintenance requirements of repairable item populations is not readily available. A great deal of assuming, estimating, and extrapolating must take place.

The concept of the maintenance tree can serve as a limitation if complex populations are being evaluated. How big does one make the maintenance tree? How small should the components at the "bottom of the tree nodes" be? If a complex piece of machinery with hundreds or more repairable parts is being evaluated, the construction of a maintenance tree that will identify every stockable item may prove too large in scope for this model.

The multi-stream concept presents limitations when populations are evaluated that operate in many different failure environments or if a population is particularly long lived. In this type of situation a tremendously large number of streams of end items may be generated. Keeping track of data and outputting concise information for each stream would be difficult.

Choosing to avoid simulation of what happens when an end item fails and is removed from operation was a large simplifying decision. The repair phase can be a significantly expensive part of operating and maintaining a population of repairable items. This model would not be of much assistance in fine tuning repair processes.

Choosing to place the computerized version of this model on a personal computer created a limitation in the amount of data that could be considered. Thus, the number of end items, the number of maintenance tree nodes, the number of stockable items, and the number of failure environments that can be considered are limited. Implementing the present model on an IBM PC with 256 K memory proved to be a programming challenge. Also, in using a computer to calculate maintenance requirements, it must be remembered that there will be round off errors. Some of the round offs are forced by the method of programming, other round offs are inherent in the use of a computer. In large populations with extensive maintenance requirements, these round offs should have a negligible effect on output. In situations where small streams of end items exist with corresponding small maintenance requirements, the user of the model must take care in interpreting output provided by the model.

As long as the idea of a shortage cost is included in a model of this sort, there is going to be a problem with quantifying that cost. Using an arbitrary figure, such as shortage cost, in conjunction with other types of logistics costs, makes the overall logistics policy costs better suited for comparative purposes. Only in situations where the shortage cost may either be

omitted or quantified can such things as budget constraints be accurately assessed with a model of this type.

The limitations imposed by simplifying assumptions are discussed in the section on Assumptions, Section 3.3.2. Certain extensions of the model that may bypass these limiting assumptions and alleviate several of the other limitations imposed by the modeling approach taken are discussed in more detail in Sections 4.5 and 6.3.

3.6 SUMMARY OF THE MODELING APPROACH.

The modeling approach used in the development of this maintenance requirements planning model started with the establishment of the basic concepts upon which the model has subsequently been built. Two major concepts were determined as being conducive to the development of a model that could evaluate nonsteady-state phases of end item operation, those being the multi-stream concept and the maintenance tree concept.

The objectives of the research established the type of data that the model was to produce. In order to produce the data required with a model, assumptions had to be made. Some assumptions were necessary to keep the research within manageable proportions and some assumptions were necessary because of the inability of a mathematical model to emulate real life occurrences. The next step of the modeling procedure involved deciding what decision variables and

parameters must be evaluated in order to produce the data called for in the research objectives.

After defining the decision variables and parameters, the modeling approach called for the development of a set of mathematical equations. The equations represent the mechanical link between the decision variables, parameters, and other input data and the maintenance requirements generated as model output. The output data produced by the equations provide measures of performance which the logistics decision maker can use to compare the effects of operating environments and logistics policy decisions on the maintenance requirements of repairable end items. Based on the type of output produced by the final version of the model, a variety of uses for the model were established.

Once it was determined what the model could do, the final step of the modeling approach was taken. This final step involved taking a realistic look at the assumptions made and the methods used in constructing the model. This "look" helped to determine the model's limitations. The next chapter will take a look at the model in its final form and discuss the implementation of the model in the face of these limitations.

CHAPTER IV

MODEL IMPLEMENTATION

4.1 INTRODUCTION.

This chapter will discuss the data requirement generated by the model and try to illustrate just how worthwhile it is to collect good failure and repair data. The chapter also will show how that data is organized, by the model, for use with the assumptions and calculations detailed in Chapter 3.

Since the model is programmed on a microcomputer, there are certain hardware limitations. Possible ways to alleviate these limitations, as well as some of the limiting assumptions described in Section 3.3.2, are discussed. This discussion produces a host of ideas for model extensions. The description of these model extensions should give the reader a feel for the potential uses of the model.

A section on the level of application of the model, as it exists today, provides some insight as to who should use the model and how it should be applied. Ideas on the level of application of the model, if certain extensions are made, are included also. A summary of the limitations and potential of the model closes the chapter.

4.2 PROFILE OF DATA REQUIREMENTS.

The data requirement generated by this logistics model can become extensive. Since the transient state operation of repairable item populations is being evaluated, more detailed failure and repair data must be provided in order to produce maintenance requirements. Since different failure environments can produce different sets of failure and repair rates, failure and repair data requirements are multiplied as the number of failure environments increases. To support the maintenance tree concept and produce spare parts requirements, a great deal of bill of material type data is required. The model serves as an illustration of the trade off that exists between the realism of model output versus the amount of data required to produce results with the model. The amount of flexibility and the level of realism inherent in this model has resulted in a large data requirement. As can be seen in the discussions on simplifying assumptions in Section 3.3.2 and on model extensions in Section 4.5, a great deal more flexibility and realism could be added to the model. Many of these extensions would serve to increase the data requirement even further.

Besides requiring a large quantity of data, the data that is required is often difficult to obtain, or in many cases nonexistent. In order to obtain information on end

item availability, end item maintenance requirements, and logistics policy cost, the user of this model must devise methods of collecting or estimating this data. The data that must be input by the user is summarized in Table 4.1.

Data such as the period unit, the length of a period, and the length of a model run can be determined if the user of the model is familiar with inventory, cost, and availability reporting schedules for the organization within which the model is being applied. Some idea of the frequency of failures and the average time spent in repair for end items and components will also assist in choosing period length. Generally, MTTR's should be less than the period length and MTBF's should be longer than a period length. Data such as the demand for end items or the required availability of end items will probably be specified by someone other than the logistician and can be treated as given data.

Some of the cost data required, such as the first cost of end items and spare parts is readily available to the logistician. The interest rate also should be relatively easy to obtain. The rate used should represent the time value of money based on a rate of return established by existing market conditions. Other cost data such as book values and salvage values of end items, procurement costs for end items and spare parts, repair costs, and cost of

TABLE 4.1. INPUT DATA REQUIRED BY THE MODEL

Period Unit
Length of Period
Length of Model Run

End Item Cost Data:
- First Cost
- Salvage Value upon Fatal Failure
- Book Value at Retirement
- Procurement Cost
- Average Repair Cost
- Average Shortage Cost
- Inventory Holding Cost for Stockable Items
- Interest Rate

Bill of Materials for End Item

Stockable Item Data:
- Part Number
- First Cost
- Procurement Cost
- Warehouse Space Requirement
- Inventory Holding Cost
- Lead Time

Number of Failure Environments

"Bottom of Tree" Component Failure Data (Per Environment):
- Cumulative Distribution Failure Probability
- MTBF
- MTTR
- Repair Level

Demand for End Items per Environment

Retirement Age for End Items per Environment

Age of Existing End Items per Environment

warehousing should be obtainable through calculation or in-house sources. Determining the cost of repair, as it is defined in this model, presents more of a challenge. Since the cost of repair represents the costs incurred during the time that elapses between failure of the end item and the time the end item returns to service, quite a bit of cost data may have to be collected before a representative repair cost per unit time can be determined.

The one piece of cost data that could be especially difficult to quantify is the shortage cost. If the population of repairable end items being evaluated is operating in a situation where shortages of end items simply result in lost revenues, then the shortage cost is easily quantifiable. On the other hand, if shortages of end items require emergency actions, noncompletion of missions, injury, or death, the establishment of a monetary value for shortage cost can be difficult. The user of the model must establish this cost such that if shortages do occur, the cost of the logistics policy in force is penalized sufficiently. In these cases, the shortage cost is serving as a reminder of the relative severity of a shortage of end items. Here, the magnitude of the shortage penalty is more important than the actual value of the shortage penalty.

In order to construct a maintenance tree for the end item and evaluate spare parts requirements, knowledge of the bill of materials comprising the end item is a must. This information should be obtainable from manufacturer sources. The arrangement of the components in the maintenance tree is also a data input decision that must be made by the user of the model. The tree must be constructed so as to accurately reflect the dependent and independent components of the end item and also provide information pertinent to required inventory levels of spare parts, warehouse space required at various repair levels, and repair facility capacities.

The data that is perhaps the key to using this model is failure data and repair data. If it cannot be determined how often an end item will fail, and how long it will take to repair it, then it becomes nearly impossible to predict how large a population of end items should be maintained. If the same failure and repair data is not available for the end item components that fail, then spare parts requirements and repair facility requirements cannot be readily estimated.

Failure rates can be estimated if a mean time between failures (MTBF) and an accompanying probability distribution can be determined for end items and end item components that fail. A mean time to repair (MTTR) for the

end item can be determined only after component MTTR's have been estimated. These component MTTR's must take into account everything that might happen to an end item between the time that the component causes an end item failure and the time that the end item returns to operation.

The user of the model must also know how failure environments affect the failure rate and repair rates of end items and components. A whole new set of failure and repair data may be required for each failure environment.

Where does the logistician obtain failure and repair data? This is a major issue any time maintenance requirements planning is performed. Possible sources of information include: in-house maintenance records; outside source maintenance records; manufacturer information or test data; or independent source test data. When data on one component is lacking, the known failure and repair data from similar components may be applied. Worst case failure and repair data could be constructed by applying very conservative failure and repair rate estimates to components.

Even though there are several sources of failure and repair information, the quality and the quantity of information available remains low. Manufacturers either do not make enough test data on component failure rates

available, or manufacturers do not test component failure rates. Whatever the case may be, the logistician does not generally receive a great deal of manufacturer assistance in the determination of reliability and maintainability characteristics of end items and components.

Repair facility records hold forth the promise of being a good source of data that can be used to establish MTBF's and MTTR's. In the past, repair facilities have generally been notorious for poor record keeping. With the advent of bar codes and other electronic data collection systems, it is becoming possible for repair facilities to maintain large, thorough data banks. The maintenance history of end items and their components can be recorded with electronic sensing devices, and then stored in computer related memory devices. Depending on the type of data collection system used, the storage of the collected data may be accomplished automatically or manually.

It is hoped that models, such as the one described in this paper, will establish the importance of having good failure and repair records available. It is also hoped that models, such as this one, will specify what kind of repair data and what kind of failure data should be maintained. The data requirements established by this and other logistics models should aid in the construction of electronic data collection systems that are built around

the specific needs of the maintenance requirements planners, and are conducive to proper use in repair facilities.

4.3 COMPUTATIONAL BASIS OF MODEL.

The model can be looked upon as a system of generators and aggregation routines. After being provided with input data by the user, the model generates the maintenance requirements for streams of end items, sums the results and then generates maintenance requirements for the entire population of repairable end items. In the process, the model computes the cost of the associated logistics policy and the availability of the population of end items. The operation of the model is structured as follows:

1) user inputs data for model generators
2) model generates stream by stream failure characteristics and maintenance requirements
3) model sums stream by stream results
4) model generates period maintenance requirements availability and logistics policy costs
5) model sums period by period results and returns to step 2 until last period is reached

6) model generates charts and graphs depicting period by period maintenance requirements per failure environment, logistics policy costs and availability

See the flow chart in Figure 4.1. Figure 4.2 gives a more detailed picture of the operation of the family of generators that make up the model.

As the user inputs data, several scenario generators are being "seeded". By specifying a retirement age, a generator that removes streams of end items from operation as they reach the appropriate age is seeded. Specifying the chance of fatal failure for the end items seeds a generator that reduces the size of the population of end items in response to the estimated number of fatal failures that occur within a given period.

End items are procured and new streams formed at specified intervals and in specified amounts after the user inputs data for the procurement policy scenario generator. By supplying an MTBF and failure curve shape for each component, the user is seeding the generator that will produce the probabilities of failure of various components. Another subroutine then combines the failure probabilities of the components to produce the probability of failure of the end item.

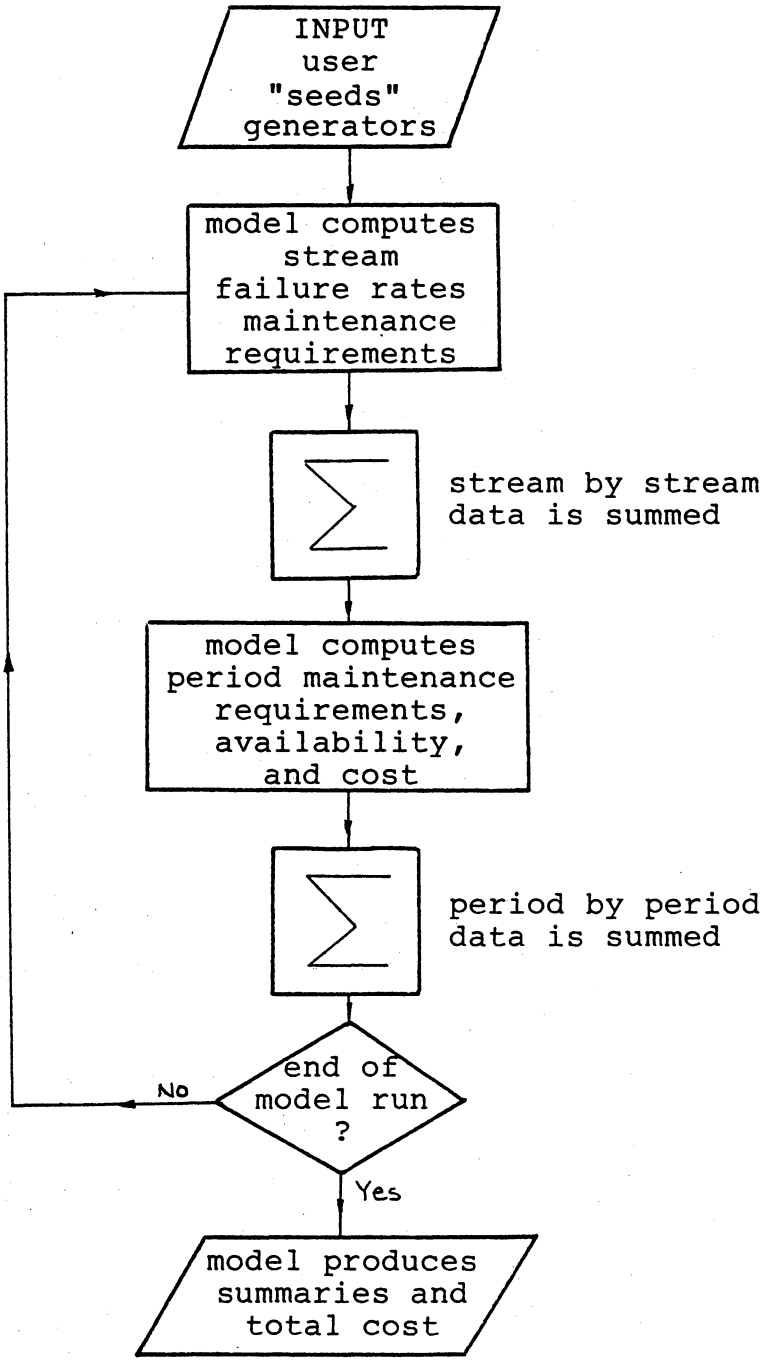


FIGURE 4.1. MODEL OPERATION MACRO FLOW CHART

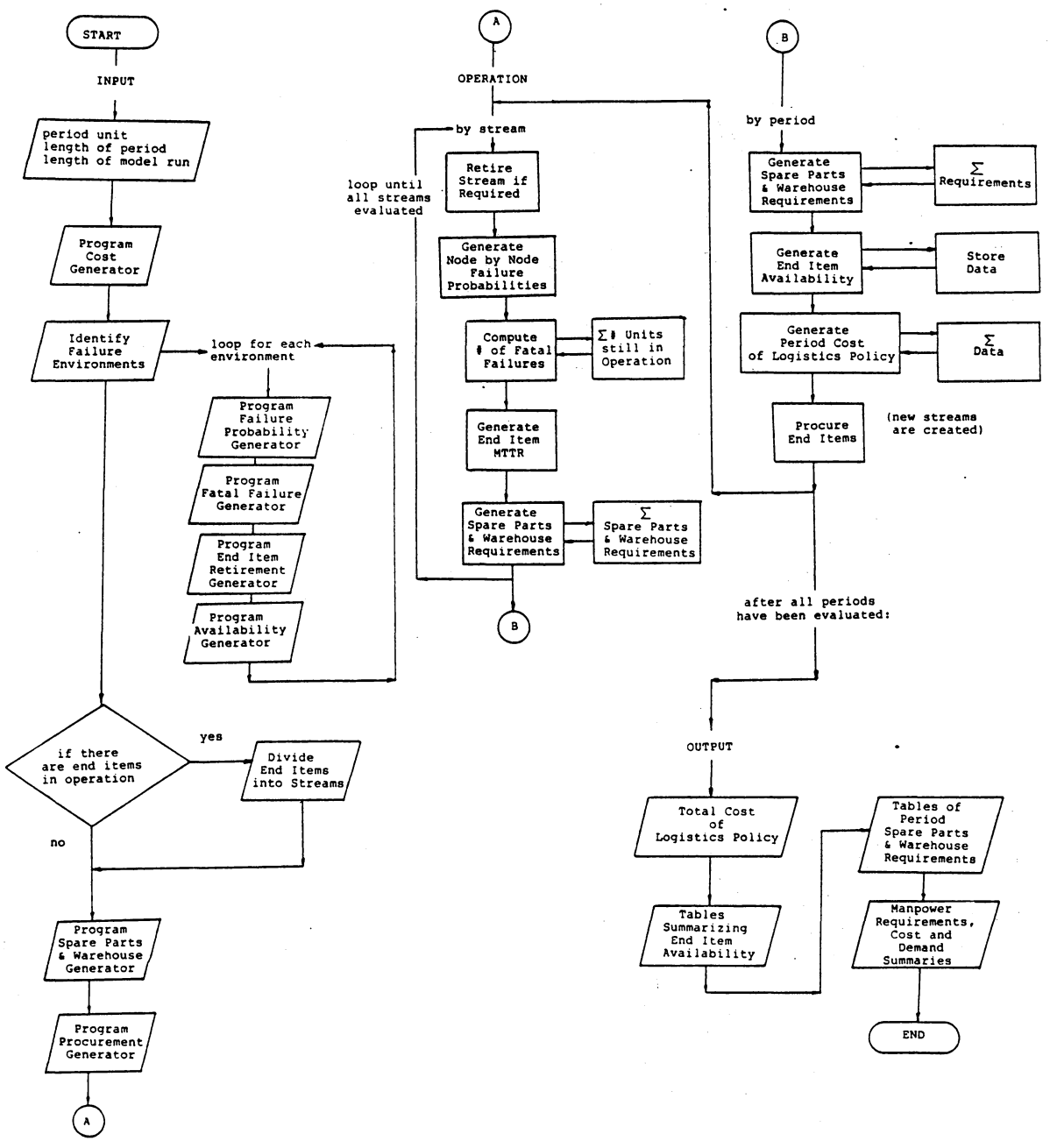


FIGURE 4.2. MODEL OPERATION MICRO FLOW CHART

Cost of logistics policy is generated after the user inputs the necessary cost data. Once period logistics policy has been computed, the cost generator computes present values of the period logistics policy costs and then sums these costs. After the period costs have been summed, the cost generator then computes a period equivalent cost of the logistics policy. This period equivalent cost represents the average per period cost for the length of the model run. The user of the model seeds the cost generator by inputting cost data and an interest rate representing the time value of money.

Additional generators compute spare parts requirements by repair level and specify inventory levels and ordering policies using economic order quantity concepts. Through the process shown by the flow chart in Figure 4.2, these generators perform various computations that result in the outputs noted in the figure. The computations are based on the concepts, assumptions, and calculations detailed in Section 3.3.

4.4 COMPUTER RELATED LIMITATIONS.

By implementing this model on a personal computer (PC), three major PC related limitations were encountered: memory, speed, and display. Central Processing Unit (CPU), Basic Interpreter, diskette, and screen display

characteristics of the PC played a part in the limitations encountered. An IBM PC with 256K of CPU memory and dual floppy disks was utilized.

Techniques used to program the maintenance requirements planning model resulted in an extremely long computer program (coded in Basic computer language). Because of Basic Interpreter capacity limitations and because of CPU memory limitation, the program has been split into several subprograms. When running the model, the subprograms are "chained" together. This means that only one part of the model is actually loaded into the CPU at a time.

Also, because of the Basic Interpreter and the CPU memory limitation, much of the data generated by the model must be stored in files on the floppy disk, rather than in arrays within the CPU. The potential of the model to create a large number of files raises the possibility of a diskette capacity limitation. Only a limited number of files can be stored on any memory device. There are many devices with more storage capacity than floppy disks. The result of the memory limitations is that only a small number of failure environments can be evaluated, the number of stockable items per "bottom of the tree" component is restricted, the size of the maintenance tree (number of nodes) is limited, and the number of streams of end items

that can be created is limited. The memory limitations could be eased by adding capacity to CPU memory and utilizing hard disks instead of floppy disks.

The processing speed of personal computers combined with the fact that programs are chained together and data files are used, cause the model to run slowly. Data files must be used because the model tracks such a great number of variables. Reading and writing these variables to and from disk files can take a considerable amount of time. The only computation within the model that can take longer than five seconds is the availability computation. If populations are evaluated where there is a significant difference between the total number of end items in a particular environment and the demand for end items in that environment, computation times can run as long as thirty seconds for the availability routine alone. These actual computation times could be improved by compiling the Basic programs that make up the model and running the model using Compiled Basic. The use of Compiled Basic would also avoid some of the memory limitations imposed by Interpreter Basic. (The version of Interpreter Basic used for this research only utilized 64k of CPU memory.) Adding CPU memory capacity could eliminate the need for data files and chained programs and significantly decrease the time required for the computer to complete a run.

The display limitation occurs because of the display screen size and the resolution of the average monitor. It is difficult to accurately depict the maintenance tree on one screen or the output data, for such things as spare parts requirements or node by node failure probabilities, on a single screen. If the user of the model wishes to study in detail, the maintenance requirements and failure probabilities of each stream of end items, he or she will be forced to print out the information desired. Printing information from the PC will also serve to slow the execution of the model, especially if printing must be done as each stream is evaluated within a period.

4.5 MAINFRAME IMPLEMENTATION AND OTHER EXTENSIONS

The limitations of implementing the model on a personal computer all point toward the use of larger, faster computing equipment. A mainframe implementation of the model would provide enough memory capacity so that the use of chained programs and data files could be avoided. A mainframe implementation would provide rapid access to any disk files that did have to be used. These improvements would result in a faster running model that could handle larger end item populations, more failure environments, and an expanded maintenance tree. As the length of the model run (number of periods evaluated) increases, it is likely

that the number of streams of end items will continue to increase (as new end items are procured). The added capacity of a main frame implementation would increase the number of periods that could be evaluated.

With the additional memory inherent in a main frame computer, not only can the model run faster, but hardcopy printouts can generally be produced faster since main frame printers are faster than PC printers. The display limitation will not disappear if a main frame implementation occurs. The video monitors used for personal computers and main frames are usually similar in size.

If the model were used on a main frame it is likely that data for more than one end item population could be input at a time, thus making data input a more efficient process. Also, more flexibility could be built into the model, since with additional memory capacity, more scenario generators could be added. Generators that would allow the evaluation of populations of end items, where the end items are rotated from environment to environment, would represent a model extension. This extension could be especially valuable in situations where end items are rotated into and out of storage. Other scenario generators that would represent extensions of the present model could provide inventory lot sizing utilizing techniques besides

EOQ. Also, a scenario generator could be added that when programmed would change the demand or costs associated with end items as the model advances from period to period.

Generators that would automatically change the input data, according to preprogramming by the user of the model, and send the model through another run, would save additional time for the user. The addition of preventive maintenance scenario generators would allow the user to evaluate the effects of performing maintenance on some components before they failed. This extension would, of course, include the capability to change failure probabilities when preventive maintenance was performed.

A main frame implementation would facilitate several model extensions that would relieve some of the restrictive assumptions discussed in Section 3.3.2. For example, additional memory would allow the evaluation of situations where all stockable items are not required every time a failure occurs. This type of situation exists with a car battery, where new clamps and cables are not required each time the battery fails. In most end items the failure of some components does not always signal the failure of the end item. An extension of the model could consider these noncritical components and their contribution to the cost of maintaining a population of end items. The evaluation of parallel or redundant systems of components would

represent a major extension of the model.

The extensions listed above, made possible by main frame implementation, would add to the flexibility and the realism of the model output. Additional flexibility provided by main frame implementation would allow the model to pull data from a central data base. The concept of placing maintenance records on computer through the use of electronic tracking devices is discussed in Section 4.2. It would be easier for the model to become a part of a Material Requirements Planning system if main frame implementation occurred. See the next section for further discussion on the MRP characteristics of this model.

4.6 LEVEL OF APPLICATION.

The questions of, how should this model be used, and who should use this model, must be answered. The model is seen as having application in commercial, public, or military settings. In a commercial environment, the model could be used for fleet sizing and maintenance requirement planning for a fleet of forklift trucks. The model would have applications in local, state, or federal government settings where fleets of vehicles or machinery are maintained. The evaluation of the maintenance requirements for equipment, such as two way radios, could be performed by the model. In military settings, any group of

repairable items could be evaluated. The case studies in Chapter 5 all illustrate military applications of the model.

The model produces much of the information that is required by Materials Requirement Planning systems. Such information includes: the number of end items required to meet a demand, the number of spare parts required to support the end item, the location of end items (by failure environment), and the location of repair parts (by repair facility). Since the model has the potential to be implemented on a main frame computer, it could be tied in with an existing MRP system. This would allow the MRP system to track the location and availability of machinery and vehicles, as well as production items. Actually, the model could serve as a stand alone MRP system for the population of end items that it is evaluating, once a logistics policy has been determined. In this case a more appropriate name for the system would be Maintenance Requirement Planning, which is a descriptor that has been used throughout this document.

The above examples are intended to illustrate that use of the model is limited only by the imagination of the user. The answer to the question of who should use the model has a more limited scope. Since the model provides information on the availability of, the maintenance

requirements of, and the operation cost of a population of repairable items, the user of the model should be a person that has some input into determining the size of end item populations, inventory levels of spare parts, and procurement policies. That person should be in a position to use the model as a planning tool.

The model would probably receive its fullest usage if utilized by the staff in charge of the highest level of repair evaluated on the model. In Case Study 1, in Chapter 5, the depot level of repair represents the highest level of repair, and the lower levels of repair represent a group of repair facilities that report to that depot. The model is not presently conducive to use at levels higher than the manager of the depot in the case study. Including too many groups of repair facilities in the model will make the determination of repair costs and MTTR's very difficult, because different depots may have significant differentials in repair costs and MTTR's due to geographic location. If one depot and its family of repair facilities are evaluated, the degree of accuracy of average repair costs and MTTR's, as applied at each level of repair, will be greater.

With some modification, the model could receive some usage at much higher levels of planning. The bottom of the tree nodes of the maintenance tree could represent

populations of end items, with higher nodes representing systems or groups of populations. By assigning appropriate failure and repair data to the populations of end items, the model would produce information on the rates of failure of, not only the populations of end items, but of systems of end items. The spare parts output would represent the number of spare end items to be kept on hand and the availability output would represent the availability of the entire system being evaluated. This macro-application of the model would take some development, but it is an example of how very high level decision makers could use such a model.

4.7 SUMMARY.

This chapter has expounded on the applications of the model and reemphasized some of the limitations. The intent was to give a clearer picture of the capabilities and potential of the model. The section on data requirements points out that the price paid for the flexibility and realism inherent in the model is a tremendous data requirement. The potential for the use of the model as a logistics planning tool should provide some impetus to data collection efforts. The case studies described in Chapter 5 should further illustrate what kind of logistics forecasting can be done with good data.

This chapter outlined how the model uses the data supplied by the user to perform logistics forecasting. The present model implementation is limited by the hardware on which it is being used, but the potential for model expansions through a main frame implementation is great. The expansions would include the ability to handle more data and produce results at a faster speed. Expansions in computing ability would open the door for model extensions that would alleviate many of the limiting assumptions applied.

The many extensions of the model discussed would identify the model as a potential source of artificial intelligence, if the extensions were implemented successfully. When programmed with the right data, and armed with the appropriate optimization techniques, the model could detail logistics policy for logistics planners. Keep in mind, however, that the model at this time is a conceptual model that depends on "what if" gaming interface with the user of the model, in order to evaluate logistics policy.

CHAPTER V

EMPIRICAL ANALYSIS

5.1 INTRODUCTION.

In order to illustrate the output capabilities of the model, Case Study 1 was developed, and the corresponding data was input into the model. A description of the case population and a description of the data base typical to that population are contained in this chapter. Following that description is an analysis of the maintenance requirements generated by the model for that case population. The format and validity of results generated here represent the type of results that the model can produce for a case population when appropriate data is available.

In order to more fully illustrate the "what if" gaming capabilities of the model, two additional case study scenarios (Case Studies 2 and 3) are developed in Sections 5.5 and 5.6. Case Study 2 illustrates a scenario in which a "what if" gaming procedure can be used to obtain least cost logistics policies and meet availability requirements by varying procurement policies. Case Study 3 illustrates a scenario in which the model can be used to compare the effects of end item quality on maintenance requirements and availability for the case population.

5.2 DESCRIPTION OF CASE POPULATION.

The case population being evaluated in Case Study 1 is a group of 220 Side Loadable Warping Tugs (SLWT's). The SLWT is a component of the Container Offloading and Transfer System (COTS) utilized by the U.S. Navy. The COTS is a deployable system designed to provide seaborne military forces of the U.S. Navy with logistics support during and subsequent to entry on land masses where suitable port facilities are not available. The logistics support provided is basically the unloading (from ships) and transfer to shore of the required quantities of bulk dry cargo and vehicles. The COTS is designed to operate beginning on the day after land entry by military forces (D-day + 1) to D-day + 180 or longer.

The SLWT is a powered causeway section that can serve several functions as a part of COTS. When equipped with a winch and A-frame, one of the major functions of the SLWT is to assist in the construction of the causeway on piers that connects ship to shore. The SLWT can also assist in the mooring of vessels, craft salvage operations, or the transport of shipping containers from ship to shore. Figure 5.1 depicts the SLWT. The case study developed from this point on is fabricated and does not necessarily represent real events, nor does the data used necessarily

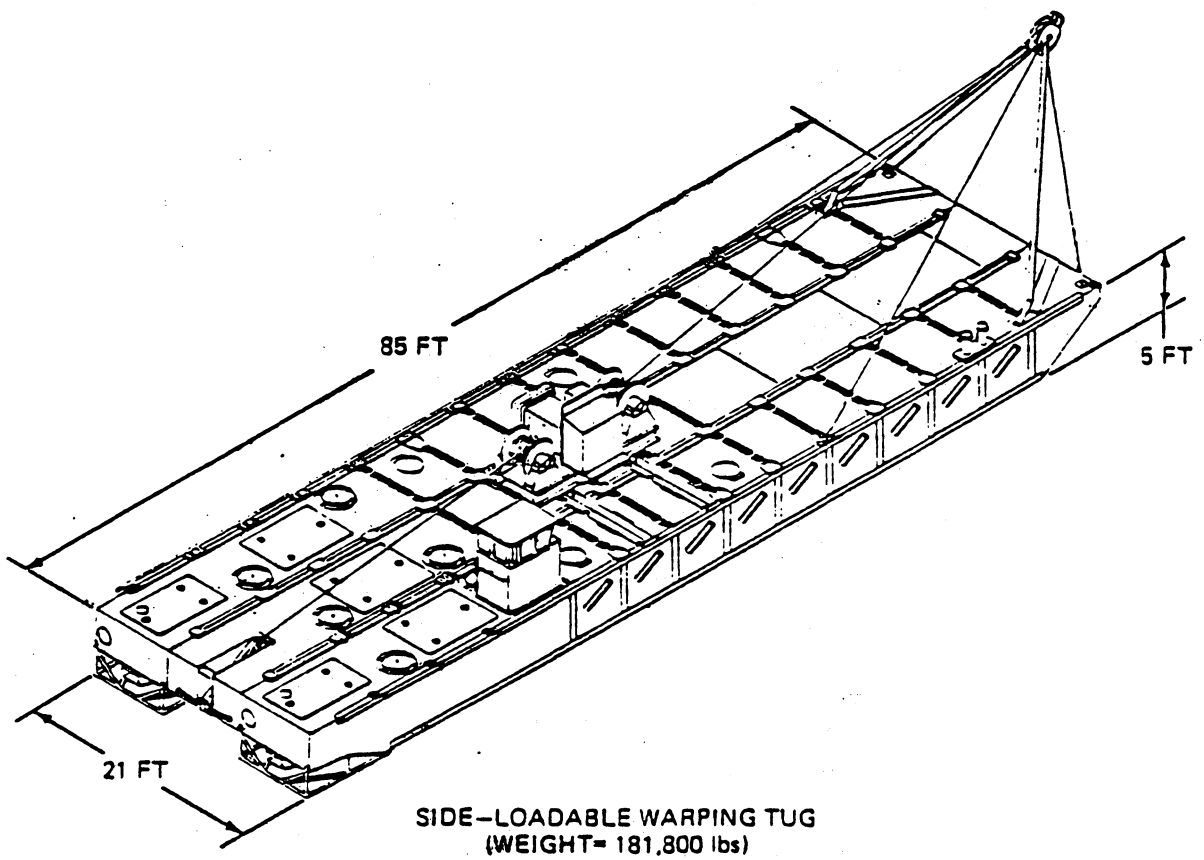


FIGURE 5.1. SIDE-LOADABLE WARPING TUG

represent exactly the present SLWT used by the Navy.

Being a seagoing vessel, the SLWT is virtually immune to fatal failure when maintained properly. An exception to this is when the SLWT is operating in a militarily hostile environment where an SLWT may be damaged irreparably by enemy attack. In Case Study 1 a population of 220 SLWT's is evaluated with some of the units operating in a support area environment that is removed from hostile military action, and the rest of the units operating in a combat zone.

The combat zone environment differs from the support area environment in two ways. The major difference between the two environments is in fatal failure probabilities. The SLWT's operating in the Support Area Environment have a very slim chance of fatal failure, which is represented by a mean time between fatal failures of 100,000 operating hours per unit. The cumulative rate of fatal failure for these SLWT's follows an Increasing Exponential function, which means that early in the life of these SLWT's there will be an almost nonexistent probability of fatal failure. On the other hand, SLWT's operating in the Combat Zone Environment have a mean time between fatal failures of 7200 operating hours per unit. The cumulative rate of fatal failure for these SLWT's follows a Uniform distribution, which means that there is an equal chance of fatal failure

within each time period. This type of fatal failure pattern is the result of the fact that fatal failure occurs mainly due to enemy attack.

The second major difference between the two environments is that SLWT's work a different number of hours per day in each environment. In the Support Area Environment, an SLWT works an average of 8 hours per day and in the Combat Zone Environment, an SLWT works an average of 12 hours per day. It is this difference in hours worked per day that gives the various components ("bottom of the tree" nodes) of the SLWT a different probability of failure in each environment.

As an example of the difference in failure rates created by the differing number of hours worked per day, consider the case of the "bottom of the tree" node representing a seawater pump. Keep in mind that the SLWT's are equipped identically regardless of which environment they operate in. It has been determined that the type of seawater pump used has an MTBF of 5000 hours.

$$\text{MTBF support} = \frac{5000 \text{ hours}}{8 \text{ hours/day}} = 625 \text{ days}$$

$$\text{MTBF combat} = \frac{5000 \text{ hours}}{12 \text{ hours/day}} = 417 \text{ days}$$

The logistics group that supports the SLWT's would like to know what kind of maintenance requirements the

total population of SLWT's will generate in 30 day periods over the course of a year. This length of model run of 1 year should cover the total length of the period over which the services of the population of SLWT's will be required. The population of SLWT's will be evaluated as if the 220 end items were procured in a block procurement at time = 0 and then operated for the next 12 consecutive periods.

When an SLWT requires repair, there are three levels of repair that will apply. Organizational level repairs will be performed on board the SLWT, with the spare parts necessary to perform these repairs carried on board. Intermediate repairs will be performed by a repair facility set up on shore, with that facility stocking the necessary spare parts. The third level of repair, depot level repair, will be necessary only in the event of serious failures. The depot stocks its own parts. The depot is a centrally located facility, so when a level 3 failure occurs, the SLWT must be transported to the depot for repair. It is assumed that, regardless of the environment in which the units operate, they require the same levels of repair dependent on the type of component failure that occurs.

In order to get some feel for the levels of spare parts to maintain at each repair level, the logistics support group will run the model in the EOQ mode. This

means the model will not only produce the period by period spare parts demand, but will generate an EOQ, a warehouse requirement based on the EOQ, and the total inventory cost of following the prescribed optimal EOQ policy.

Since the SLWT's are virtually indestructible due to natural causes, it has been determined that the retirement age of the units should be 500 periods or roughly 40 years. It has also been established that a total of 140 SLWT's must be operating in order for the Support Area operation to succeed and that a total of 60 units must be operating in order for the Combat Zone operation to succeed.

5.3 DATA BASE FOR CASE IMPLEMENTATION.

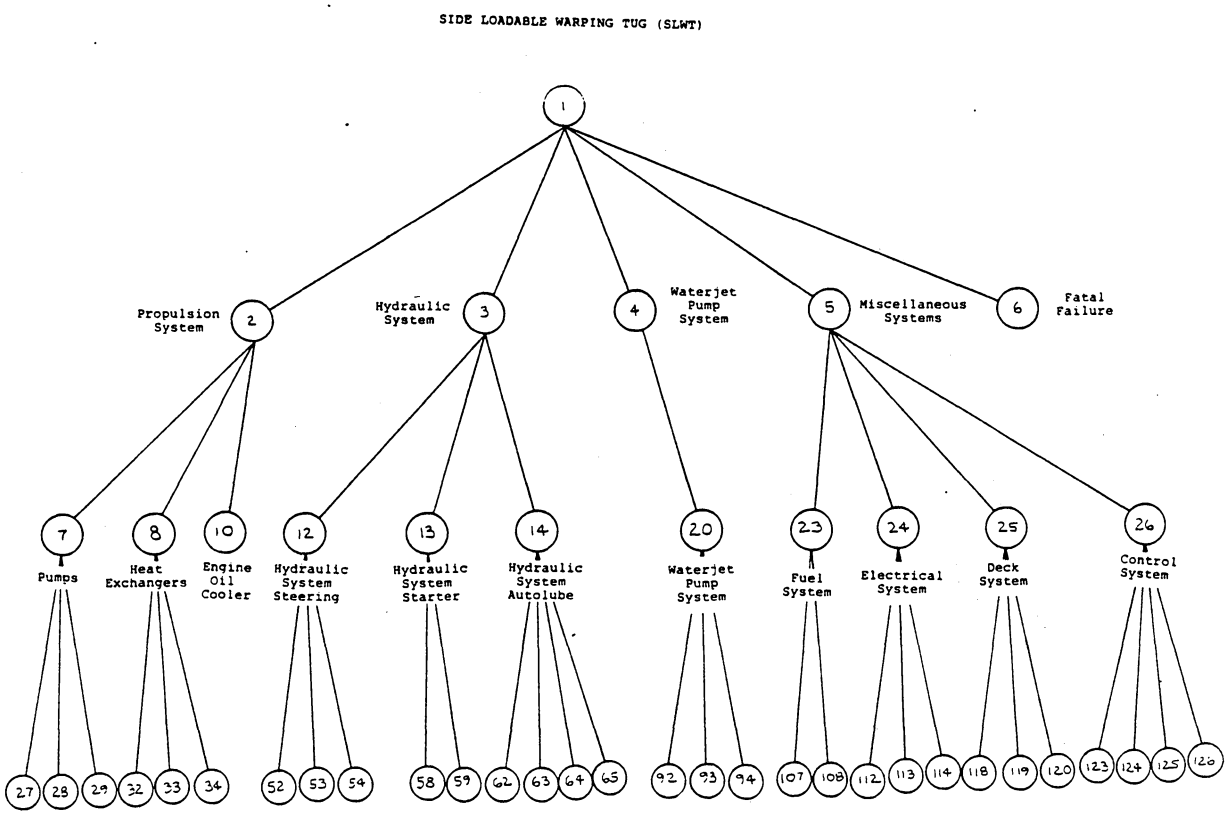
The set of information contained in Table 5.1 represents general population data that was input into the model for Case Study 1. Please keep in mind that good solid data was not available and a great deal of the data was fabricated around information that was available on the SLWT. The cost data may be significantly different from the actual costs involved with the SLWT.

The maintenance tree in Figure 5.2 represents the hierarchical structure of the SLWT. This tree is fairly representative of the actual SLWT. Tables 5.2 and 5.3 give the MTBF, MTTR, Repair Level, and Failure Curve data for each bottom of the tree node in each failure environment. Again, this particular set of data may be fairly

TABLE 5.1 GENERAL POPULATION DATA

Period Parameter	Days
Period Length	30
Length of Model Run	12
# of Failure Environments	2
First Cost per SLWT	$350,000
Salvage Value per SLWT (upon fatal failure)	$ 8000
Procurement Costs	$ 10000
Repair Costs per SLWT per Day	$ 500
Shortage Cost per Day	$ 30000
Time Value of Money	10%
Model Mode	Proposed Population
Spare parts and Warehouse Requirements	EOQ
Procurement Policy	one initial block procurement
Procurement Amount	220
Divide Procurement	Support: 144 Combat: 76

Failure Environments:	Support Area	Combat Zone
Retirement Age per SLWT	500 periods	500 periods
Retirement Book Value	$15000/SLWT	$15000/SLWT
Demand for SLWT's	140	60



27. freshwater pump
28. seawater pump
29. hydraulic pump
32. engine coolant heat exchanger
34. hydraulic fluid heat exchanger

52. pressure reducing valve
53. directional control valve
54. steering motor
58. filter
59. starter
62. automatic lubrication
63. timer
64. manifold
65. solenoid

92. waterjet pump
93. discharge elbow
94. steerable nozzle
107. booster pump
108. fuel/water separator
112. inverter
113. battery
114. alternator

118. hydraulic motor
119. hydraulic pump
120. cable

123. engine on/off
124. throttle
125. steering lever
126. clutch lever

FIGURE 5.2. MAINTENANCE TREE FOR CASE STUDY 1

TABLE 5.2 NODE DATA FOR SUPPORT AREA ENVIRONMENT

Node #	Node Name	Repair Level	MTBF (days)	MTTR (days)	Failure Curve
10	Engine Oil Cooler	1	1000	1.8	5
27	Freshwater Pump	2	250	1.0	3
28	Seawater Pump	2	625	1.2	3
29	Hydraulic Pump	1	1000	1.2	3
32	Eng.Cool.Heat Exch.	2	1000	1.8	4
33	Freshwa.Heat Exch.	1	1000	1.8	4
34	Hyd.Fluid HeatExch.	1	1000	1.8	4
52	Pres.Reducing Valve	1	500	0.8	3
53	Dir. Control Valve	1	125	1.2	3
54	Steering Motor	2	375	1.8	3
58	Filter	1	75	0.3	3
59	Starter	1	312	2.0	3
62	Auto. Lubrication	1	188	1.0	5
63	Timer	1	94	2.4	5
64	Manifold	2	1250	15.0	3
65	Solenoid	1	94	1.2	5
92	Waterjet Pump	3	625	12.0	3
93	Discharge Elbow	2	1250	11.0	5
94	Steerable Nozzle	2	625	12.0	3
107	Booster Pump	2	375	0.8	3
108	Fuel/Wa. Separator	1	188	1.0	5
112	Inverter	1	250	0.5	5
113	Battery	1	125	0.1	3
114	Alternator	1	250	2.4	5
118	Hydraulic Motor	2	125	0.8	5
119	Hydraulic Pump	2	125	1.6	3
120	Cable	1	6250	0.5	5
123	Engine On/Off	1	1250	0.1	5
124	Throttle	1	2625	1.0	5
125	Steering Lever	1	2625	0.8	5
126	Clutch Lever	1	2625	0.8	5

Repair level 1 = Organizational Level Repair
Repair level 2 = Intermediate Level Repair
Repair level 3 = Depot Level Repair

TABLE 5.3 NODE DATA FOR COMBAT ZONE ENVIRONMENT

Node #	Node Name	Repair Level	MTBF (days)	MTTR (days)	Failure Curve
10	Engine Oil Cooler	1	667	1.8	2
27	Freshwater Pump	2	167	1.0	2
28	Seawater Pump	2	417	1.2	2
29	Hydraulic Pump	1	667	1.2	2
32	Eng.Cool.Heat Exch.	2	667	1.8	2
33	Freshwa.Heat Exch.	1	667	1.8	2
34	Hyd.Fluid HeatExch.	1	667	1.8	2
52	Pres.Reducing Valve	1	333	0.8	3
53	Dir. Control Valve	1	83	1.2	3
54	Steering Motor	2	250	1.8	3
58	Filter	1	50	0.3	3
59	Starter	1	208	2.0	3
62	Auto. Lubrication	1	125	1.0	5
63	Timer	1	63	2.4	5
64	Manifold	2	833	15.0	3
65	Solenoid	1	63	1.2	5
92	Waterjet Pump	3	417	12.0	3
93	Discharge Elbow	2	833	11.0	5
94	Steerable Nozzle	2	417	12.0	3
107	Booster Pump	2	250	0.8	3
108	Fuel/Wa. Separator	1	125	1.0	5
112	Inverter	1	167	0.5	5
113	Battery	1	83	0.1	3
114	Alternator	1	167	2.4	5
118	Hydraulic Motor	2	83	0.8	5
119	Hydraulic Pump	2	83	1.6	3
120	Cable	1	4167	0.5	5
123	Engine On/Off	1	833	0.1	5
124	Throttle	1	1750	1.0	5
125	Steering Lever	1	1750	0.8	5
126	Clutch Lever	1	1750	0.8	5

Repair level 1 = Organizational Level Repair
Repair level 2 = Intermediate Level Repair
Repair level 3 = Depot Level Repair

representative of the SLWT, but can by no means be considered to be highly accurate. The failure data contained in Tables 5.2 and 5.3 represents the set of data that will prove to be the most difficult to obtain regardless of the type population of end items being evaluated. This data can be assumed to have been obtained from the performance of like components in other populations of end items, or in some cases, from data provided by the manufacturer of the component.

Tables 5.4 and 5.5 contain the inventory data that was input for each bottom of the tree node. This data, of course, is required for the generation of spare parts requirements using the EOQ model. If the model is extended to include other inventory lot sizing techniques, some different types of inputs may be required.

5.4 DESCRIPTION OF CASE STUDY 1 RESULTS.

The model outputs a wealth of information, as evidenced by the tables in Appendix 1, Tables 5.6 through 5.13, and Figures 5.3 through 5.6. These tables and figures will be analyzed in this section. Suggestions for performing sensitivity analysis and uses of the output will be discussed.

5.4.1 Maintenance Requirements Generated.

As defined in Section 5.2, the primary reason for running Case Study 1 was to determine the maintenance requirements generated by

TABLE 5.4 STOCKABLE ITEMS INPUT DATA

Node #	Part Name	Part #	QY #	WH cuft	PR $	PC $	HC $	LT
10	Eng.Oil Cooler	2101	1	.5	250.00	40	2.50	2
27	100 GPM Pump	27271	1	1.0	250.00	40	5.00	2
28	.5HP Pump Motor	27281	1	2.0	1800.00	40	10.00	2
28	Packing	27282	1	0.2	5.00	30	1.00	1
28	Impeller Wearri	27283	2	0.2	16.00	35	1.00	2
28	Casing Wearring	27284	2	0.2	14.00	35	1.00	2
29	100 PSI Pump	27291	1	1.0	600.00	40	5.00	2
32	EC Heat Exchang	28321	1	3.0	1000.00	40	15.00	2.5
33	FW Heat Exchang	28331	1	2.0	350.00	40	10.00	2
34	HF Heat Exchang	28341	1	2.0	450.00	40	10.00	2
52	3/8" PR Valve	312521	1	.1	23.00	35	.10	2
52	3/8" Hose Clamp	312522	2	.01	.45	30	.01	.5
53	3/8" DC Valve	312531	1	.1	30.00	35	.10	2
53	3/8" Hose Clamp	312532	2	.01	.45	30	.01	.5
54	1/4 HP Motor	312541	1	1.0	290.00	35	5.00	2
58	HSS Filter	313581	1	.2	11.00	35	.20	1
59	Starter	313591	1	1.0	190.00	35	1.00	1.5
62	Plunger Pump	314621	1	.5	80.00	35	.50	1.5
62	Oil Line Filter	314622	1	.1	4.50	30	.10	1.5

QY = # of stockable items that must be replaced when failure occurs.
WH = cubic feet of warehouse space required per stockable item.
PR = purchase price per stockable item.
PC = procurement cost per procurement of stockable items.
HC = storage cost per stockable item per period.
LT = order to receipt lead time for stockable items.

TABLE 5.5 STOCKABLE ITEMS INPUT DATA

Node #	Part Name	Part #	QY #	WH cuft	PR $	PC $	HC $	LT
63	HSAL Timer	314631	1	.25	63.00	40	.25	2
64	HSAL Manifold	314641	1	1.00	22.00	35	5.00	1
64	1/2-13 Bolts	314642	8	.01	.12	30	.05	.5
65	HSAL Solenoid	314651	1	.10	20.00	35	.50	1
92	200 HP Motor	420921	1	5.00	11000.0	90	25.00	3
93	WJ Dis. Elbow	420931	1	1.00	50.00	35	5.00	1
93	1/2-13 Bolts	420932	8	.01	.12	30	.05	.5
93	1/2-13 Nuts	420933	8	.005	.06	30	.03	.5
94	3"Steer.Nozzle	420941	1	1.00	190.00	40	5.00	2
107	.5HP Boost.Pump	5231071	1	.50	80.00	35	2.50	2
108	F/W Separator	5231081	1	.50	60.00	35	.50	1
112	Inverter	5241121	1	.10	18.00	35	.50	2
113	12Volt Battery	5241131	1	1.00	60.00	30	1.00	.5
113	Cable	5241132	2	.20	6.00	30	.20	.5
114	Isolation Diode	5241141	1	.05	5.00	30	.05	.5
114	Brushes	5241142	2	.05	2.00	30	.05	.5
114	VoltageRegulato	5241143	1	.10	21.00	30	.10	.5
118	5HP Motor	5251181	1	1.00	800.00	40	5.00	2
119	100PSI Hyd.Pump	5251191	1	1.00	600.00	40	5.00	2
120	1/8"Steel Cable	5251201	1	2.00	150.00	35	2.00	2
123	EOF Switch	5261231	1	.10	21.00	35	.10	1.5
124	Throttle	5261241	1	.20	15.00	35	.20	1.5
125	Steering Lever	5261251	1	.20	16.00	35	.20	1.5
126	Clutch Lever	5261261	1	.20	17.00	35	.20	1.5

QY = # of stockable items that must be replaced when failure occurs.
WH = cubic feet of warehouse space required per stockable item.
PR = purchase price per stockable item.
PC = procurement cost per procurement of stockable items.
HC = storage cost per stockable item per period.
LT = order to receipt lead time for stockable items.

the case population of SLWT's. This purpose is satisfied by the tables in Appendix 1, which represent end of the period summaries of the spare parts demand generated by the total population of SLWT's. Each table identifies the stockable item (spare part) by part number and by the bottom of the tree node to which that part is attached. The summary also displays the EOQ for each part, and the warehouse requirement and total cost generated if that EOQ policy is adhered to. The EOQ and Total Cost equations are defined in Section 3.3.5.6.

The location at which each stockable item should be stocked is identified by including the repair level number. At the bottom of each summary is the total cubic feet of warehouse space required at each repair level. The twelve period summary charts give the logistician a tool to use in planning his stockable item ordering schedule to assure that the proper repair facilities are stocked at a level that should satisfy the maintenance requirement for spare parts generated by the population of SLWT's.

Tables 5.6 and 5.7 are summaries of the spare part demand for stockable items attached to Nodes 107 and 108. Here, the demand is broken down by environment. A plot of the period by period demand (Figure 5.3) mirrors, to some extent, the plots of period by period point probabilities of failure of the respected nodes in their respective

TABLE 5.6 PERIOD SPARE PARTS DEMAND FOR NODE 107

Period	Stockable Item	Part #	Demand support	Demand combat	Repair Level
1	1/2HP Booster Pump	5231071	0	0	2
2	1/2HP Booster Pump	5231071	0	0	2
3	1/2HP Booster Pump	5231071	0	0	2
4	1/2HP Booster Pump	5231071	0	1	2
5	1/2HP Booster Pump	5231071	1	2	2
6	1/2HP Booster Pump	5231071	2	3	2
7	1/2HP Booster Pump	5231071	3	3	2
8	1/2HP Booster Pump	5231071	3	4	2
9	1/2HP Booster Pump	5231071	4	4	2
10	1/2HP Booster Pump	5231071	5	4	2
11	1/2HP Booster Pump	5231071	5	5	2
12	1/2HP Booster Pump	5231071	6	5	2

TABLE 5.7 PERIOD SPARE PARTS DEMAND FOR NODE 108

Period	Stockable Item	Part #	Demand support	Demand combat	Repair Level
1	Fuel/Wa. Separator	5231081	23	13	1
2	Fuel/Wa. Separator	5231081	4	3	1
3	Fuel/Wa. Separator	5231081	5	4	1
4	Fuel/Wa. Separator	5231081	5	5	1
5	Fuel/Wa. Separator	5231081	6	7	1
6	Fuel/Wa. Separator	5231081	7	8	1
7	Fuel/Wa. Separator	5231081	9	10	1
8	Fuel/Wa. Separator	5231081	10	12	1
9	Fuel/Wa. Separator	5231081	12	8	1
10	Fuel/Wa. Saparator	5231081	14	3	1
11	Fuel/Wa. Separator	5231081	17	3	1
12	Fuel/Wa. Separator	5231081	19	4	1

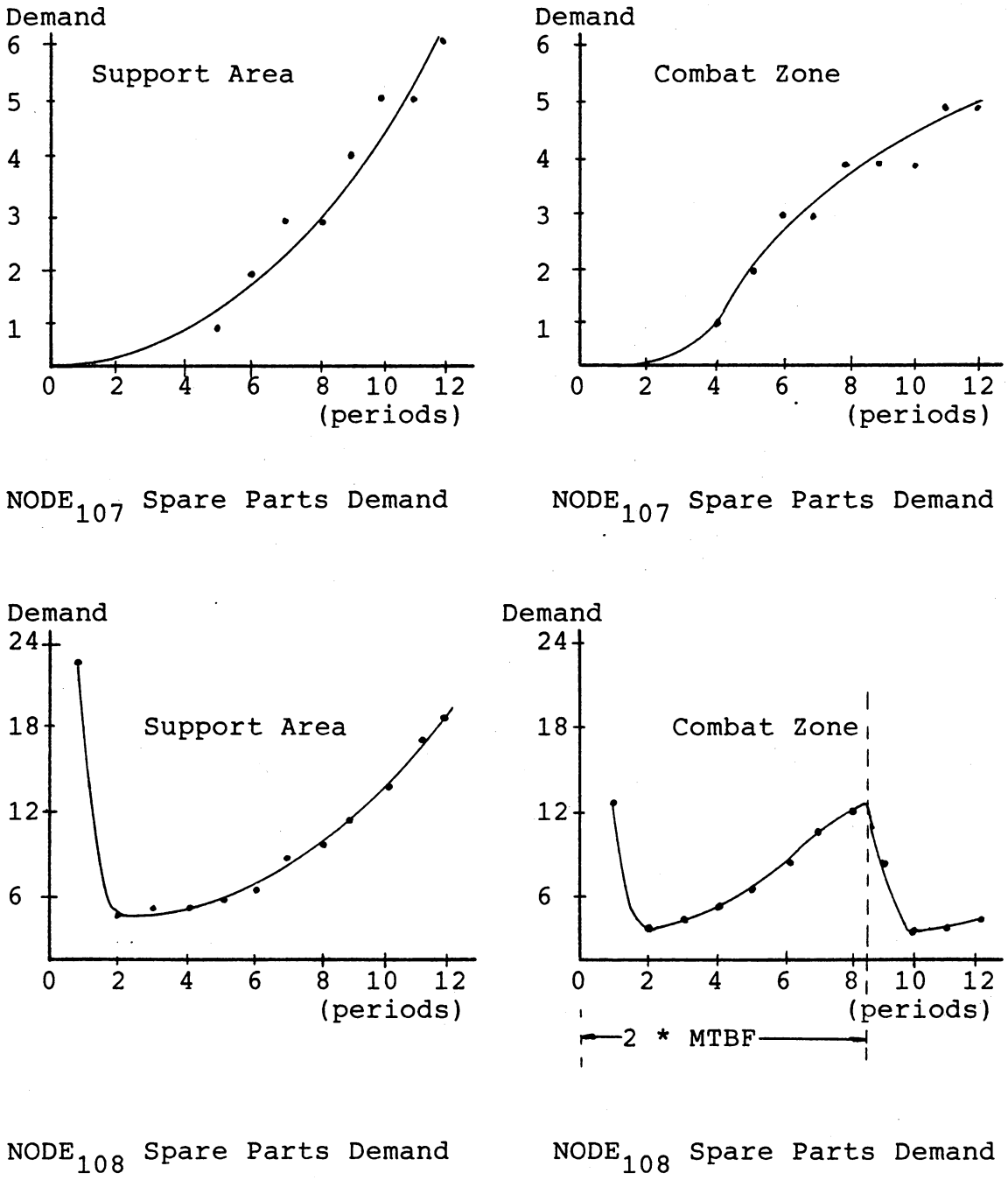


FIGURE 5.3. PERIOD SPARE PARTS DEMAND

environments. The plots of period by period point probabilities of failure are shown in Figures 5.4 and 5.5. The spare parts demand, for end items operating in the combat zone, will not exactly mirror point failure probabilities for the Combat Zone because there is a shrinking number of SLWT's in operation each period.

Table 5.8 contains information that will assist in the staffing of the repair facilities. Information supplied by the model includes the number of man-days of repair time per period required at each level of repair by the total population of SLWT's. As can be seen from the table, there are significant period to period fluctuations in the expected man-days required. Having some idea of expected repair time requirements will help prevent repair delays due to staffing shortages, or will help prevent overstaffing due to sudden drops in repair demand.

5.4.2 Failure Analysis. Tables 5.9 and 5.10 list the period by period probabilities of failure of selected nodes. As shown in the maintenance tree for the SLWT, Nodes 107 and 108 combine to give the failure probability for Node 23. Plots of the point probabilities and cumulative probabilities of failure for Nodes 107 and 108 (Figures 5.4 and 5.5) illustrate the effects that the choice of failure curve had on the failure patterns of these two particular nodes. The plot of the point

TABLE 5.8 PERIOD BY PERIOD MANPOWER REQUIREMENTS SUMMARY

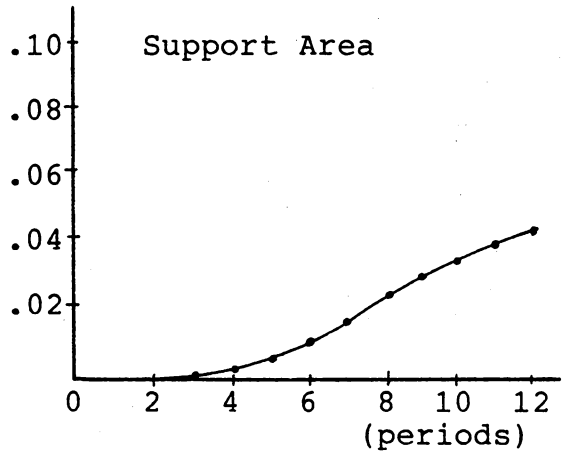
Period	Repair Level 1 (man-days)	Repair Level 2 (man-days)	Repair Level 3 (man-days)
1	190.02	157.06	.00
2	92.19	38.21	.00
3	113.54	45.04	.06
4	138.03	52.79	.58
5	131.35	64.81	1.99
6	150.69	64.61	5.19
7	143.78	61.24	8.47
8	137.41	65.87	10.07
9	120.07	63.61	13.42
10	118.10	70.15	16.53
11	137.93	85.24	19.55
12	160.70	76.04	23.04

TABLE 5.9 PERIOD POINT PROBABILITIES OF FAILURE FOR NODES 108 AND 107

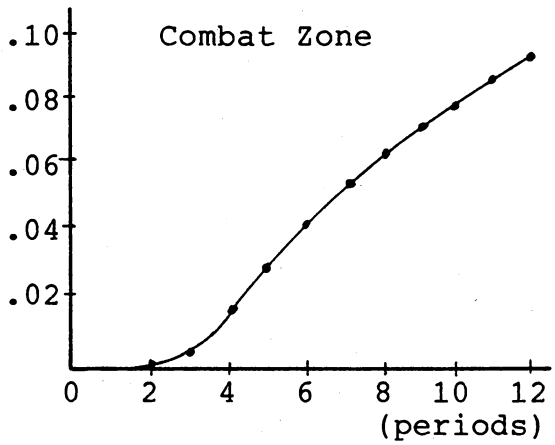
	Node 108		Node 107	
Environment -	support	combat	support	combat
Failure Curve -	3	3	5	5
MTBF (days) -	12.50	8.33	6.27	4.17
PF (period)				
1	0.15874	0.17201	0.00000	0.00000
2	0.02745	0.04662	0.00000	0.00027
3	0.03219	0.05925	0.00027	0.00437
4	0.03776	0.07531	0.00199	0.01511
5	0.04429	0.09571	0.00596	0.02848
6	0.05195	0.12165	0.01148	0.04101
7	0.06093	0.15462	0.01749	0.05165
8	0.07147	0.19653	0.02331	0.06055
9	0.08382	0.13126	0.02861	0.06815
10	0.09832	0.04297	0.03334	0.07489
11	0.11532	0.05461	0.03752	0.08112
12	0.13526	0.06941	0.04125	0.08712

TABLE 5.10 PERIOD POINT PROBABILITIES OF FAILURE FOR NODES 23 AND 1

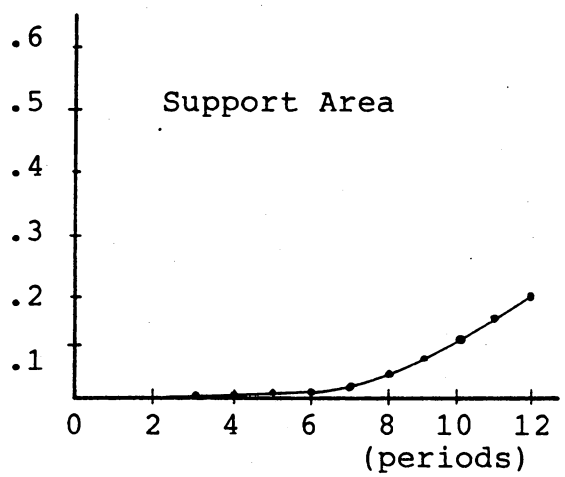
	Node 23		Node 1	
Environment -	support	combat	support	combat
Failure Curve -	n.a.	n.a.	n.a.	n.a.
MTBF (days) -	n.a.	n.a.	n.a.	n.a.
PF (period)				
1	0.15874	0.17201	0.90207	0.94478
2	0.02745	0.04688	0.38452	0.80305
3	0.03245	0.06336	0.57859	0.91690
4	0.03968	0.08972	0.71037	0.93035
5	0.04999	0.12147	0.80642	0.93837
6	0.06283	0.15767	0.78938	0.89042
7	0.07736	0.19829	0.79903	0.85582
8	0.09311	0.24518	0.82903	0.92819
9	0.11004	0.19046	0.71734	0.94100
10	0.12838	0.11464	0.74518	0.95634
11	0.14851	0.13130	0.71877	0.95404
12	0.17092	0.15048	0.83551	0.91572



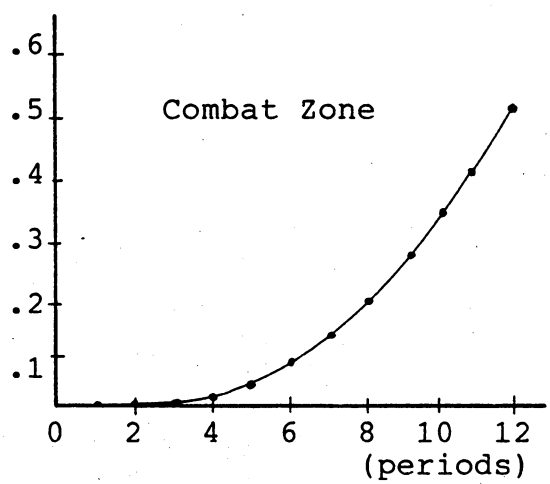
NODE$_{107}$ Point Probabilities of Failure



NODE$_{107}$ Point Probabilities of Failure

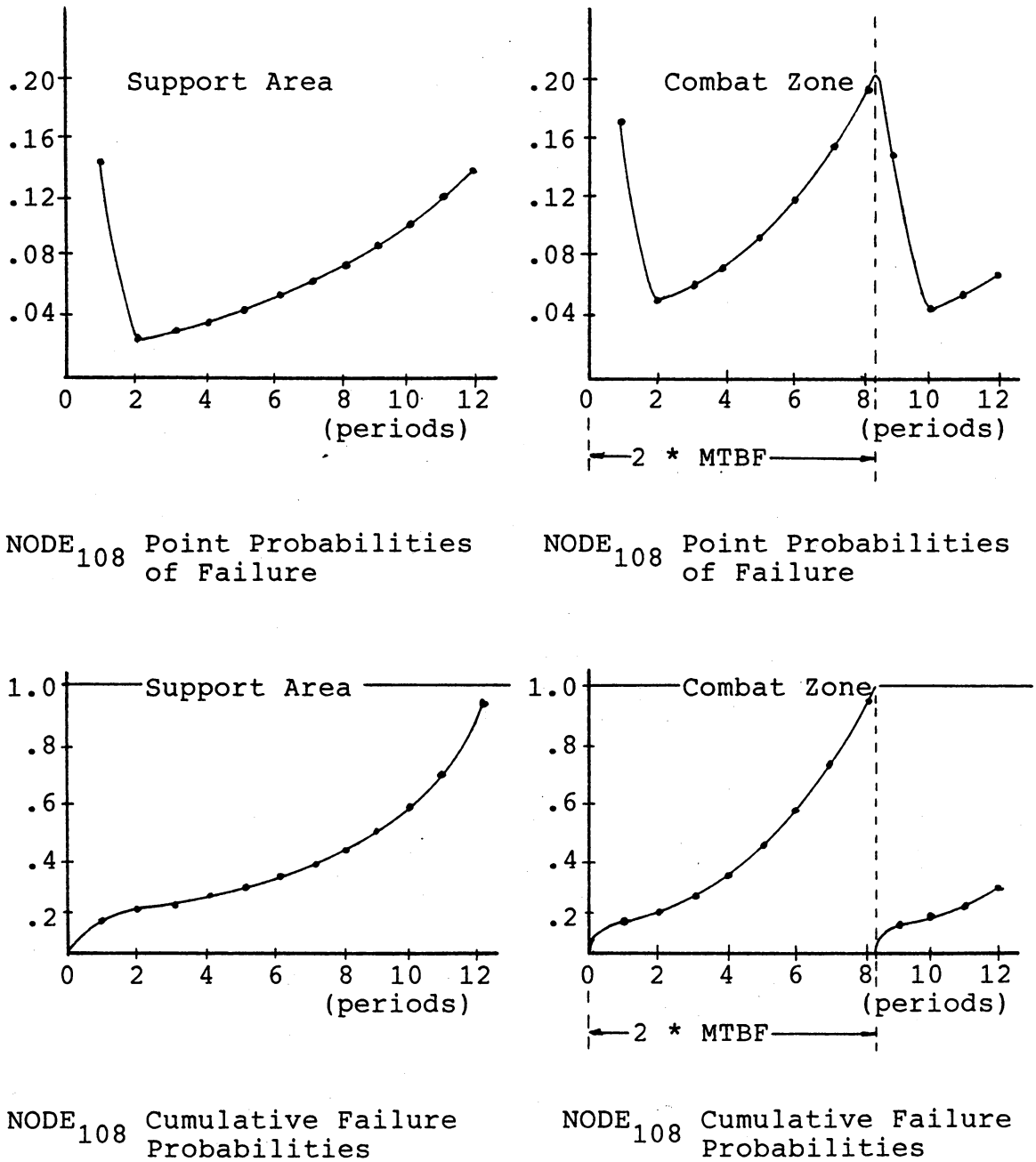


NODE$_{108}$ Cumulative Failure Probabilities



NODE$_{108}$ Cumulative Failure Probabilities

FIGURE 5.4. PROBABILITY OF FAILURE PLOTS



NODE$_{108}$ Point Probabilities of Failure

NODE$_{108}$ Point Probabilities of Failure

NODE$_{108}$ Cumulative Failure Probabilities

NODE$_{108}$ Cumulative Failure Probabilities

FIGURE 5.5. PROBABILITY OF FAILURE PLOTS

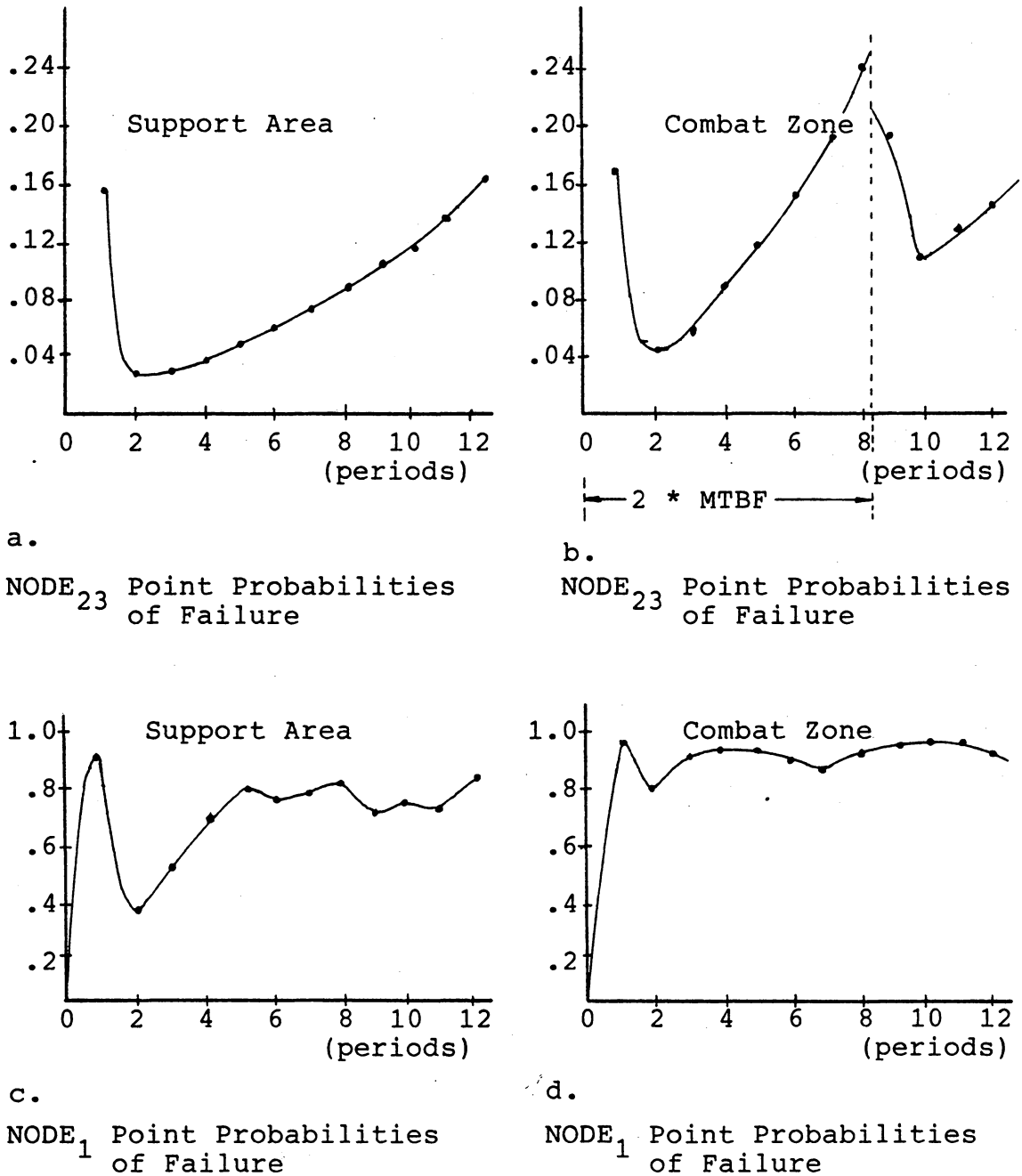


FIGURE 5.6. PROBABILITY OF FAILURE PLOTS

probabilities of failure for Node 23 (Figure 5.6) illustrates the additive effects of Nodes 107 and 108 on Node 23.

It is interesting to note in the plot of point probabilities of failure for Node 1 (the end item or SLWT) that the failure rate tends to approach a steady state. These plots are graphs c. and d. in Figure 5.6. Perturbations in the curve are caused when parts experience accelerating failure rates according to the failure distribution applied to those particular parts. For instance, many of the bottom of the tree components were identified as components whose failure characteristics were similar to a bathtub function. Thus, a high initial failure rate is experienced followed by a sizable drop in failure rate. Sudden rises and falls in the Node 1 plot can be attributed to the fact that many components went through similar phases of their failure rate life cycle at the same time.

In comparing the plots of Node 1 point probabilities of failure for each of the two failure environments (Figure 5.6), it can be seen that the failure environment with the higher failure rate, Combat Zone, seeks a steady state sooner and at a higher level of failure than does the environment with the lower rate of failure. Frisch [5] makes this point when comparing items with different

quality, i.e., different failure rates.

5.4.3 Availability of SLWT's. Additional information output by the model includes the availability of SLWT's in their respective environments. By viewing Table 5.11 it can be seen that by the 10th month the SLWT population in the combat zone is operable less than 14% of the time. This can be attributed to the fact that because of fatal failures, the number of SLWT's operating in the Combat Zone has dropped below the demand figure of 60 by the end of period 10. The number of SLWT's in operation per period is listed in Tables 5.12 and 5.13 under the headings "# Units Start" and "# Units End".

By referring to Table 5.12 it can be seen that the number of units failing fatally has no effect on the availability of units operating in the Support Area environment because no fatal failures occur during the model run. The initial number of SLWT's assigned to this environment (144) is very close to the demand number of 140. The low availability during certain periods can be attributed to SLWT failure due to minimal repair failure of components. By comparing the period by period availability to the period by period point probabilities of failure for Node 1 (Table 5.10) the correspondence between low availability and high failure rate may be seen.

TABLE 5.11 AVAILABILITY OF END ITEM (SLWT)

Period	Support	Combat	Total System
1	0.114165	0.999998	0.417532
2	0.936423	0.999998	0.957810
3	0.833255	0.999995	0.888318
4	0.716574	0.999880	0.808798
5	0.606586	0.999168	0.732508
6	0.428179	0.991981	0.605374
7	0.526298	0.953532	0.658464
8	0.583662	0.674056	0.611173
9	0.633159	0.511901	0.596870
10	0.557285	0.139997	0.433833
11	0.381189	0.000000	0.271068
12	0.309976	0.000000	0.222072

TABLE 5.12 SUPPORT AREA COST SUMMARY

Period	Capital Cost $	Repair Cost $	Shortage Cost $	Inventory Cost $	Total Cost $
1	50400000	114898	797251	37626	51349780
2	0	31928	57219	9922	99069
3	0	43532	150070	17439	211042
4	0	53288	255083	24426	332797
5	0	61686	354073	32322	448080
6	0	75777	514639	39196	629612
7	0	67835	426332	45336	539504
8	0	63425	374704	67350	505480
9	0	59674	330157	42657	432487
10	0	65440	398443	31774	495657
11	0	79889	556930	38579	675398
12	0	86737	621022	60099	767858

Period	Procurement Cost $	# Units Start	# Units End	Operating Cost/Unit $
1	10000	144	144	6596
2	0	144	144	688
3	0	144	144	1466
4	0	144	144	2311
5	0	144	144	3112
6	0	144	144	4372
7	0	144	144	3747
8	0	144	144	3510
9	0	144	144	3003
10	0	144	144	3442
11	0	144	144	4690
12	0	144	144	5332

TABLE 5.13 COMBAT ZONE COST SUMMARY

Period	Capital Cost $	Repair Cost $	Shortage Cost $	Inventory Cost $	Total Cost $
1	26584000	58642	2	39702	26682350
2	-16000	33274	2	31014	48290
3	-16000	35790	5	37009	56804
4	-8000	42400	108	43145	77653
5	-16000	37392	749	46095	68236
6	-16000	34468	7217	30493	56178
7	-8000	38914	41821	33448	106183
8	-16000	43247	293350	40253	360850
9	-8000	38870	439290	43198	513365
10	-16000	36948	774003	58022	852973
11	-8000	41470	900000	64711	998181
12	-16000	43153	900000	47884	975037

Period	Procurement Cost $	# Units Start	# Units End	Operating Cost/Unit $
1	10000	76	74	1294
2	0	74	72	869
3	0	72	70	1011
4	0	70	69	1224
5	0	69	67	1221
6	0	67	65	1077
7	0	65	64	1757
8	0	64	62	5888
9	0	62	61	8409
10	0	61	59	14245
11	0	59	58	17054
12	0	58	56	17087

5.4.4 Cost of Logistics Policy. Tables 5.12 and 5.13 summarize the period by period costs to operate the population of SLWT's in each environment. A period equivalent cost per environment is also given. As can be seen in the charts, shortage penalties played a significant role in the total costs and the operating cost per unit for the SLWT's operating in both environments. The negative capital cost figures represent salvage values received for SLWT's that failed fatally. While these cost figures, especially the shortage penalty, may not be exact, the relative costs of operation between the two environments serves as an index that tradeoffs may be evaluated against.

5.4.5 Sensitivity Analysis. Sensitivity analysis that might be performed on this case study would involve changing certain parameters in order to make improvements in availability and logistics policy costs. Very high costs are being incurred in each of the two failure environments because of shortage penalties. In this case study, the logistics decision maker could increase availability (and thus reduce shortage costs) by increasing the population size of SLWT's in either or both environments. Availability also could be improved if preventive maintenance policies could be implemented to improve component MTBF's. In the case of the Combat Zone Environment, if the fatal failure rate of SLWT's could be

reduced, availability would improve. Improvements in repair practices could reduce MTTR's and thereby increase the availability of the SLWT's.

Making changes in the procurement policy used could improve cost and availability. If the logistics decision maker has choices in quality of components installed on the SLWT, then better quality components should result in longer MTBF's which would result in better availability. The effects on cost would have to be evaluated.

The model is designed so that the decision maker can make sensitivity analyses involving many different parameters. As the model is set up now, sensitivity analysis will require the manual changing of the desired parameter and then a new run of the model. As described earlier in this thesis, some of these parameters will be under the total control of the logistics decision maker, some of these parameters have upper and/or lower limits within which the decision maker must work, and some of the parameters are set and cannot be changed.

5.4.6 Uses of the Model Output. The uses of model output is limited only by the imagination of the model user. Some of the more obvious uses of the particular output produced for Case Study 1 would include establishing stockable items ordering and stocking policies. The manpower requirements generated by the model could help in

the staffing of repair facilities and will warn of expected busy periods that are approaching, or of expected periods where personnel should be moved to other locations due to lack of work.

Plots of the period point probabilities of failure at each bottom of the tree node may be compared to actual repair data so that the model failure rates may be fine tuned to more closely approximate reality. The plots also show patterns in component failure rates that may lead to requested improvements in performance or to the use of different types of components that exhibit more favorable failure characteristics. Tracking the failure rates of nodes higher in the maintenance tree helps to point out systems of components that create the most failure problems on the SLWT. Paying special attention to preventive maintenance or offering better protection to these systems may prove to be worthwhile.

The availability data is especially crucial to the logistics decision maker. This data tells him whether or not the logistics support being provided is allowing the population of SLWT's to accomplish its designed mission. Various data output by the model may point to the area in which improvements can be made to increase availability. This is discussed in the previous subsection.

The model output could be used as a tool to provide data necessary to guide higher level decision makers. For instance, if the logistician were to demonstrate to the appropriate military personnel that the high fatal failure rate of the SLWT's in the Combat Zone resulted in low availabilities of the SLWT's in the latter stages of the 1 year time period evaluated, it may be possible to gain greater protection against hostile forces. Better protection would, in effect, reduce the fatal failure rate of the SLWT's in the Combat Zone to a more acceptable rate.

The model helps to determine what size of population must be supported and how much it will cost to field and support that population. Thus, the model output can serve as a budgeting aid for capital, as well as, operations budget planners. If extremely expensive populations, such as the SLWT's, are being evaluated, the output of this model could prove invaluable to high level decision makers who must appropriate the funds necessary to field and support such a system.

5.5 CASE STUDY 2.

A scenario for a second case study is described here to better illustrate some of the "what if" gaming capabilities of the model. This particular case study scenario also illustrates the model's value as a long range planning tool.

Assume we are evaluating the same population of SLWT's, operating in the same two failure environments, with the same number of units being assigned to each environment. The effects of minimal repair type failures on SLWT availability is assumed to be negligible. Only the probability of fatal failure will have any real effect on the ability of the two groups of SLWT's to perform their mission.

It is of utmost importance for the logistics planner to have knowledge of the expected ability of the SLWT's in each environment to perform their missions. An established availability must be met. Running the model versus this availability and playing "what if" games with the population size and the procurement policies used, will help the planner meet the availability requirement at an acceptable cost. Once this cost has been established, the planner has the information he needs to put together the budget package and support data that high level decision makers will use in appropriating the money necessary to field the SLWT's.

5.6 CASE STUDY 3.

A scenario for a third case study is described here to illustrate the micro-capabilities of the model. An astute reader may have noticed that in Case Study 1 there is no node assigned to an engine to propel the SLWT. The reason

for this is that the engine has a whole maintenance tree to itself. An entire case study could be built around the particular diesel engine that powers the SLWT, and possibly other Navy vessels.

Assume quality is also an issue in the evaluation of the diesel engine, as there are two manufacturers of diesel engines that are compatible with the SLWT. The engines have virtually identical maintenance trees, but one engine utilizes better quality components, in many cases. The better quality engine carries a higher first cost and higher stockable item costs. On the other hand, the quality engine has higher salvage and book values, and lower repair costs. The quality engine also has longer MTBF's and shorter MTTR's for many of its "bottom of the tree" node components.

The logistician's problem is deciding whether to spend the extra money up front to buy a better engine, or hope that the lesser priced engine will serve adequately without creating extremely high operations and maintenance costs. To evaluate this situation, the model could be run once for each brand of engine with the engines being placed in the same environment, and applying the same repair policy to each engine, i.e. similar components require the same repair level. The availability and logistics policy cost data produced should identify which engine will meet the

necessary requirements for the lowest cost. Remember, also that different quality engines could affect manpower requirements and inventory practices at repair facilities.

When a diesel engine has been selected for use in the SLWT, the engine should be added to the maintenance tree in Case Study 1 as a "bottom of the tree" node. The node should have period by period failure probabilities equal to the Node 1 (End Item) failure probabilities generated by the model in Case Study 3. Use the model in Case Study 3 to evaluate the specific maintenance requirements of the diesel engine and add these requirements to the other SLWT requirements generated by Case Study 1.

CHAPTER VI

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

6.1 SUMMARY.

It was stated in Chapter I of this thesis that a major weakness of existing maintenance or logistics models was their lack of ability to evaluate nonsteady-state phases of operation of various populations of items. From this problem arose the objective of this research, that being to develop a model that could evaluate the period by period maintenance requirements of repairable item populations over their entire life cycle. This life cycle could include transient, as well as steady-state phases of operation, where period to period failure rates could be distributed in a variety of configurations.

A literature review was conducted that summarized the types of models that had been produced in the logistics or maintenance modeling area. Concepts developed by Frisch [5] were found to be conducive to transient state evaluation of maintenance requirements, and were expanded upon and included in this research.

Chapter III of this thesis developed the concepts and computational methods that formed the basis of this model. Assumptions, decision variables, and other parameters were detailed. The many possible ways in which a logistics

decision maker could use this model were also expounded upon.

The limitations of the model due to limiting assumptions, computing power, and the lack of failure and repair data were discussed in Chapters III and IV. Finally, the model was validated by illustrating its application to a case population of Side Loadable Warping Tugs. The population evaluated was, in many ways, similar to an actual population of SLWT's presently used by the U.S. Navy. This case study demonstrated the spare part demand, inventory level, warehousing demand, availability, repair facility demand, and logistics policy cost outputs produced by the model.

6.2 CONCLUSIONS.

Upon completing this research project, it does appear to be feasible to attempt to produce expected maintenance requirements for transient phases of operation of repairable item populations. The model developed does meet the list of objectives stated in Section 1.2. The two major concepts applied in this model proved to be conducive to the generation of total life cycle maintenance requirements. The multi-stream concept meshes well with the idea of population end items experiencing different maintenance requirements at different stages of their life cycles. The maintenance tree concept recognizes that

different components of an end item can possess very different rates of failure and mean times to repair, and provides a structured method of evaluating the maintenance requirements generated by specific components. Using the structure of the tree, these specific requirements are summed to produce end item maintenance requirements.

This model illustrates the value of having one maintenance model that can produce as many aspects of the maintenance requirements of a population as possible. By having spare parts demand, inventory levels, warehousing demand, availability, repair facility demand, and logistics policy cost information provided by one run of one model, the logistics planner has the opportunity to study the interrelationships among the different maintenance requirements generated for the population under study. Taking this idea further could involve tying this model into a Material Requirements Planning System. The model would generate data that the MRP system could use in stocking parts and planning warehouse operations.

One major conclusion of this research is that failure rate and repair data for either end items or components is difficult to obtain. As discussed earlier in this thesis, the sources of such data is limited. It can only be hoped that this model, and other models, will demonstrate the value of possessing and utilizing failure and repair data.

Savings and increased success of missions resulting from good logistics planning should outweigh any difficulties involved with establishing data collection systems that track failure and repair data.

The data necessary for good logistics planning does, however in itself, represent a paradox. As was discovered while building this model, not only is a large data input required, but the data multiplies many times over as period by period maintenance requirements are generated. This means that the model is tracking a huge amount of data. This large quantity of data creates two major concerns for the model developer. First, a large computer memory must be provided in order to store the data. Second, the model developer must be very careful in choosing meaningful data to display to the user of the models, in order to avoid information saturation.

The SLWT case study illustrates exactly why it can be crucial to be able to evaluate transient state phases of operation, as well as, steady-state phases of operation. In observing the plots of end item period point probabilities of failure, it can be seen that the SLWT's began to exhibit a sort of steady-state failure pattern only after about five periods of operation (see Figure 5.6). The typical mission length for a population of SLWT's is one to five periods. Therefore, the SLWT's

mission is, in most cases, nearly over before any semblance of steady-state is reached. Furthermore, as evidenced by the period point probabilities of failure for Nodes 107 and 108 (see Figures 5.4 and 5.5), there can be component failure rates driving the end item failure rate that never reach a steady state. Nonsteady-state periods of operation can be evaluated in detail if the period lengths used in the model are of short duration, and if the failure data warrants such detail. Decreasing the period lengths evaluated by the model can produce a discrete approximation of continuous maintenance requirements. Figure 3.2 illustrated this concept.

As is the case with most research, several limiting assumptions were employed in the development of this model. Relieving as many of these assumptions as possible could do nothing but add to the accuracy of the expected maintenance requirements generated by this model. Extensions of the research that might relieve some of the assumptions are described in the next section.

6.3 RECOMMENDATIONS.

The research involved with the development of this model has only scratched the surface of estimating the maintenance requirements of repairable item populations operating under transient conditions. There are many computer related improvements that can be made to improve

the usability and the capabilities of the present model. In addition, research can be extended in many directions to give more depth and therefore more accuracy to expected maintenance requirements produced by transient state models.

Computer related improvements include:

1 - Make improvements to the input and display routines of the model in order to increase the speed with which the user can program the model. Displaying summaries of input data and making all input data easily accessible for change would be the types of improvements called for here.

2 - Increase the speed of model execution. This could be accomplished by compiling the existing program, reprogramming in more efficient computer languages, and/or using a computer with a larger random access memory.

3 - Build scenario generators into the model so that such things as changing costs and demands can be evaluated.

Extensions of this research, in the context of evaluating transient state operation, could take any of the following directions:

1 - Develop methods for evaluating the uncertainty present in failure and repair data.

2 - Refine the methods for calculating the times at which end items fail so as to gain a more exact estimation of the expected repair facility and spare parts demand.

3 - Incorporate the modeling of repair functions into the model, i.e., all of the events that transpire between an end item failure and the time that the end item returns to operation.

4 - Incorporate different methods for establishing inventory levels of spare parts.

5 - Evaluate the effects of redundant components and noncritical failure of components on maintenance requirements.

6 - Evaluate the effects of preventive maintenance on maintenance requirements.

7 - Develop the concept of shortage penalties and the manner in which they should be applied to the evaluation of maintenance requirements. Decision methods for evaluating tradeoffs involving shortage penalties would be of great value to the logistics planner as shortage penalties are generally difficult to quantify.

REFERENCES

1. Barlow, R.E. and Hunter, L., "Optimal Preventive Maintenance Policies", Operations Research, Vol. 8, 1960, pp. 90-100.

2. Barlow, R.E. and Proschan, F., Mathematical Theory of Reliability, John Wiley Sons, New York, 1967.

3. Derman, C., "On Sequential Decisions and Markov Chains", Management Science, Vol. 9, 1962, pp. 16-24.

4. Fabrycky, W.J. and Hart, J.T., "Economic Optimization of a Finite Population System Deployed to Meet a Demand", Proceedings of the Joint Conference, American Association of Cost Engineers/American Institute of Industrial Engineers, Houston, Texas, 1975.

5. Frisch, Franz A. P., "Mortality and Spareparts: A Conceptual Analysis", Proceedings of the 1983 Federal Acquisition Research Symposium.

6. Geisler, Murray A., editor, "Preface" in Logistics, North-Holland/TIMS Studies in the Management Science Series, Volume I, North-Holland, 1975.

7. Geisler, Murray A. and Murrie, Bruce L., "Assessment of Aircraft Logistics Planning Models", Omega, Vol. 9, #1, 1981, pp. 59-69.

8. Graves, Stephen C. and Keilson, Julian, "System Balance for Extended Logistic Systems", Operations Research, Mar/Apr, 1983, pp. 234-252.

9. Gross, D. and Harris, C.M., Fundamentals of Queueing Theory, John Wiley Sons, Inc., New York, 1974.

10. Gross, D. and Ince, J.F., "Spares Provisioning for Repairable Items: Cyclic Queues in Light Traffic", AIIE Transactions, Vol. 10, 1978, pp. 307-314.

11. Gross, D., Kahn, H.D., and Marsh, J.D., "Queueing Models for Spares Provisioning", Naval Research Logistics Quarterly, Vol. 24, #4, 1977.

12. Hart, J.T., "A Model for the Optimal Design of a Finite Population Queueing System Deployed to Meet a Demand", unpublished Master's Thesis, Virginia Tech, 1971.

13. Hayre, Lakhbir S., "A Note on Optimal Maintenance Policies for Deciding Whether to Repair or Replace", European Journal of Operational Research, Feb., 1983, pp. 171-175.

14. Hillestad, R.J., "Dyna-Metric: Dynamic Multi-Echelon Technique for Recoverable Item Control", RAND Corporation, Santa Monica, Cal., July, 1982.

15. Hillier, F.S. and Lieberman, G.J., Introduction to Operations Research, Holden-Day, Inc., San Francisco, 1970.

16. Jensen, P.A., "Optimum Repair Level Analysis: Some Extensions", in Logistics, edited by M.A. Geisler, North-Holland TIMS Studies in Management Sciences, 1975, pp. 87-104.

17. Kao, E., "Optimal Replacement Rules", Operations Research, Vol.21, 1973, pp. 1231-1249.

18. Kline, Melvin B., "A Survey of Logistics Analysis Models", Proceedings 17th Annual International Logistics Symposium, Society of Logistics Engineers, Boston, Mass., August, 1982.

19. Lie, C.H., Hwang, C.L., and Tillman, F.A., "Availability of Maintained Systems: A State-of-the-Art Survey", AIIE Transactions, Vol. 9, 1977, pp. 247-259.

20. Lohmar, J.W. and Mandel, D.B., "Guided Missile Frigate TIGER Assessment", 1981 Proceedings Annual Reliability and Maintainability Symposium, pp. 459-464.

21. McCall, J.J., "Maintenance Policies for Stochastically Failing Equipment: A Survey", Management Science, Vol. 11, 1965, pp. 493-524.

22. Muckstadt, John A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System", Management Science, Vol. 20, #4, Dec., 1973, pp.472-481.

23. Nahmias, S., "Managing Reparable Item Inventory Systems: A Review", in Multi-Level Production/Inventory Control Systems, edited by L.B. Schwarz, TIMS Studies in the Management Sciences, North-Holland, Amsterdam, Vol. 16, 1981, pp. 253-278.

24. Petrovic, R., Senborn, A., and Vujosevic, M., "Spares Allocation in the Presence of Uncertainty", European Journal of Operational Research, Sep., 1982, pp. 77-81.

25. Phelps, E.S., "Optimal Decision Rules for the Procurement, Repair or Disposal of Spare Parts", RAND Memorandum RS-2920PR, the Rand Corporation, Santa Monica, Cal., 1962.

26. Pierskalla, W.P. and Voelker, J.A., "A Survey of Maintenance Models: The Control and Surveillance of Deteriorating Systems", Naval Research Logistics Quarterly, #23, Sep., 1976, pp. 353-388.

27. Porteus, E.L. and Lansdowne, Z.F., "Optimal Design of a Multi-Item, Multi-Location, Multi-Repair Type Repair and Supply System", Naval Research Logistics Quarterly, #21, 1974, pp. 213-237.

28. Ross, Sheldon M., Applied Probability Models With Optimization Applications, Holden-Day, San Francisco, 1970.

29. Schrady, D.A., "A Deterministic Inventory Model for Repairable Items", Naval Research Logistics Quarterly, Vol. 14, No. 3, 1967, pp. 391-398.

30. Sethi, S., "Simultaneous Optimization of Preventive Maintenance and Replacement Policies for Machines: A Modern Control Theory Approach, AIIE Transactions, Vol. 5, 1973, pp. 156-163.

31. Sherbrooke, C.C., "METRIC: A Multi-Echelon Technique for Recoverable Item Control", Operations Research, Vol. 16, #1, 1968, pp.122-141.

32. Sherif, Y.S. and Smith, M.L., "Optimal Maintenance Models for Systems Subject to Failure - A Review", Naval Research Logistics Quarterly, March, 1981, pp. 47-74.

33. Taylor, J. and Jackson, R.R.P., "An Application of the Birth and Death Process to the Provision of Spare Machines", Operational Research Quarterly, 1954, pp. 95-108.

34. Thuesen, G.J. and Fabrycky, W.J., Engineering Economy, Prentice-Hall, Inc., New Jersey, 1984, Chapters 4 and 12.

35. White, J.A., Schmidt, J.W., and Bennett, G.K., Analysis of Queuing Systems, New York: Academic Press, 1975.

APPENDIX 1.

Case Study 1

Period by Period Spare Parts

and

Warehouse Summaries

PERIOD 1 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	23	13.50	27	5817.82	1
Freshwa.Pump	27271	12	14.00	14	3069.28	2
Seawater Pump	27281	5	12.00	6	9063.25	2
Seawater Pump	27282	5	3.40	17	42.32	2
Seawater Pump	27283	10	5.20	26	186.46	2
Seawater Pump	27284	10	5.20	26	166.46	2
Hyd. Pump	27291	3	7.00	7	1834.64	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	0	0.00	0	0.00	1
Pres.Red.Valve	312522	0	0.00	0	0.00	1
Dir.Con.Valve	312531	0	0.00	0	0.00	1
Dir.Con.Valve	312532	1	0.77	77	1.22	1
Steering Motor	312541	0	0.00	0	0.00	2
Filter	313581	5	8.40	42	63.37	1
Starter	313591	0	0.00	0	0.00	1
Auto. Lub.	314621	36	35.50	71	2915.50	1
Auto. Lub.	314622	36	14.70	147	176.70	1
Timer	314631	43	29.25	117	2738.33	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	43	7.80	78	898.79	1
Waterjet Pump	420921	0	0.00	0	0.00	3
Disch.Elbow	420931	31	21.00	21	1654.16	2
Disch.Elbow	420932	244	5.41	541	56.34	2
Disch.Elbow	420933	244	3.50	699	35.60	2
Steer.Nozzle	420941	0	0.00	0	0.00	2
BoosterPump	5231071	0	0.00	0	0.00	2
Fuel/Wa.Sep.	5231081	36	35.50	71	2195.50	1
Inverter	5241121	34	6.90	69	646.50	1
Battery	5241131	0	0.00	0	0.00	1
Battery	5241132	1	3.40	17	9.46	1
Alternator	5241141	34	10.10	202	180.10	1
Alternator	5241142	68	14.30	286	150.28	1
Alternator	5241143	34	14.30	143	728.28	1
Hyd. Motor	5251181	40	25.00	25	32126.49	2
Hyd. Pump	5251191	0	0.00	0	0.00	2
Cable	5251201	30	64.00	32	4564.81	1
Engine On/Off	5261231	31	14.70	147	665.73	1
Throttle	5261241	30	20.40	102	470.49	1
Steer.Lever	5261251	30	20.40	102	500.49	1
Clutch Lever	5261261	30	20.40	102	530.49	1

Total warehouse requirement repair level 1 = 361.32
Total warehouse requirement repair level 2 = 106.71
Total warehouse requirement repair level 3 = 0.00

PERIOD 2 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	4	5.50	11	1028.28	1
Freshwa.Pump	27271	10	13.00	13	2563.25	2
Seawater Pump	27281	5	12.00	6	9063.25	2
Seawater Pump	27282	5	3.40	17	42.32	2
Seawater Pump	27283	9	5.00	25	169.10	2
Seawater Pump	27284	9	5.00	25	151.10	2
Hyd. Pump	27291	3	7.00	7	1834.64	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	0	0.00	0	0.00	1
Pres.Red.Valve	312522	0	0.00	0	0.00	1
Dir.Con.Valve	312531	9	7.90	79	277.94	1
Dir.Con.Valve	312532	18	3.29	329	11.39	1
Steering Motor	312541	0	0.00	0	0.00	2
Filter	313581	38	23.00	115	441.07	1
Starter	313591	0	0.00	0	0.00	1
Auto. Lub.	314621	7	15.50	31	575.65	1
Auto. Lub.	314622	7	6.50	65	37.98	1
Timer	314631	20	20.00	80	1280.00	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	20	5.30	53	426.46	1
Waterjet Pump	420921	0	0.00	0	0.00	3
Disch.Elbow	420931	0	0.00	0	0.00	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	0	0.00	0	0.00	2
BoosterPump	5231071	0	0.00	0	0.00	2
Fuel/Wa.Sep.	5231081	7	15.50	31	435.65	1
Inverter	5241121	5	2.60	26	103.23	1
Battery	5241131	9	23.00	23	563.24	1
Battery	5241132	18	14.60	73	122.70	1
Alternator	5241141	5	3.85	77	28.87	1
Alternator	5241142	11	5.75	115	27.74	1
Alternator	5241143	5	5.50	55	110.48	1
Hyd. Motor	5251181	13	14.00	14	10472.11	2
Hyd. Pump	5251191	9	12.00	12	5460.00	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	0	0.00	0	0.00	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 184.79
Total warehouse requirement repair level 2 = 80.82
Total warehouse requirement repair level 3 = 0.00

PERIOD 3 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	4	5.50	11	1028.28	1
Freshwa.Pump	27271	9	12.00	12	2310.00	2
Seawater Pump	27281	4	12.00	6	7256.57	2
Seawater Pump	27282	4	3.00	15	35.49	2
Seawater Pump	27283	9	5.00	25	169.10	2
Seawater Pump	27284	9	5.00	25	151.10	2
Hyd. Pump	27291	3	7.00	7	1834.64	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	0	0.00	0	0.00	1
Pres.Red.Valve	312522	0	0.00	0	0.00	1
Dir.Con.Valve	312531	23	12.70	127	702.69	1
Dir.Con.Valve	312532	46	5.25	525	25.95	1
Steering Motor	312541	0	0.00	0	0.00	2
Filter	313581	65	30.20	151	745.17	1
Starter	313591	1	8.00	8	198.37	1
Auto. Lub.	314621	9	17.50	35	737.75	1
Auto. Lub.	314622	9	7.30	73	47.85	1
Timer	314631	29	24.00	96	1851.08	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	29	6.40	64	611.86	1
Waterjet Pump	420921	0	0.00	0	0.00	3
Disch.Elbow	420931	0	0.00	0	0.00	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	0	0.00	0	0.00	2
BoosterPump	5231071	0	0.00	0	0.00	2
Fuel/Wa.Sep.	5231081	9	17.50	35	557.75	1
Inverter	5241121	6	2.90	29	122.49	1
Battery	5241131	23	37.00	37	1417.15	1
Battery	5241132	46	23.40	117	299.49	1
Alternator	5241141	6	4.25	85	34.24	1
Alternator	5241142	11	5.75	115	27.74	1
Alternator	5241143	6	6.00	60	132.00	1
Hyd. Motor	5251181	18	17.00	17	14484.85	2
Hyd. Pump	5251191	23	19.00	19	13895.92	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	0	0.00	0	0.00	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 240.65
Total warehouse requirement repair level 2 = 89.42
Total warehouse requirement repair level 3 = 0.00

PERIOD 4 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	4	5.50	11	1028.28	1
Freshwa.Pump	27271	9	12.00	12	2310.00	2
Seawater Pump	27281	4	12.00	6	7256.57	2
Seawater Pump	27282	4	3.00	15	35.49	2
Seawater Pump	27283	8	4.80	24	151.66	2
Seawater Pump	27284	8	4.80	24	135.66	2
Hyd. Pump	27291	3	7.00	7	1834.64	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	0	0.00	0	0.00	1
Pres.Red.Valve	312522	1	0.77	77	1.22	1
Dir.Con.Valve	312531	33	15.20	152	1005.20	1
Dir.Con.Valve	312532	66	6.29	629	35.99	1
Steering Motor	312541	1	4.00	4	308.71	2
Filter	313581	47	25.60	128	542.65	1
Starter	313591	3	14.00	14	584.49	1
Auto. Lub.	314621	10	18.50	37	818.71	1
Auto. Lub.	314622	10	7.70	77	52.75	1
Timer	314631	43	29.25	117	2738.33	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	43	7.80	78	898.79	1
Waterjet Pump	420921	0	0.00	0	0.00	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	0	0.00	0	0.00	2
BoosterPump	5231071	1	2.50	5	93.23	2
Fuel/Wa.Sep.	5231081	10	18.50	37	618.71	1
Inverter	5241121	7	3.10	31	141.65	1
Battery	5241131	33	44.00	44	2024.50	1
Battery	5241132	66	28.20	141	424.14	1
Alternator	5241141	7	4.60	92	39.58	1
Alternator	5241142	13	6.25	125	32.25	1
Alternator	5241143	7	6.50	65	153.48	1
Hyd. Motor	5251181	23	19.00	19	18495.92	2
Hyd. Pump	5251191	33	23.00	23	19914.89	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 271.36
Total warehouse requirement repair level 2 = 105.52
Total warehouse requirement repair level 3 = 0.00

PERIOD 5 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	9	12.00	12	2310.00	2
Seawater Pump	27281	4	12.00	6	7256.57	2
Seawater Pump	27282	4	3.00	15	35.49	2
Seawater Pump	27283	7	4.40	22	134.14	2
Seawater Pump	27284	7	4.40	22	120.14	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	2	9.00	3	2048.99	2
Freshwa. HE	28331	2	8.00	4	740.00	1
Hyd.Fluid HE	28341	2	8.00	4	940.00	1
Pres.Red.Valve	332521	1	2.60	26	25.65	1
Pres.Red.Valve	312522	1	0.77	77	1.22	1
Dir.Con.Valve	312531	41	16.90	169	1246.94	1
Dir.Con.Valve	312532	82	7.01	701	43.91	1
Steering Motor	312541	3	6.00	6	902.40	2
Filter	313581	66	30.40	152	756.40	1
Starter	313591	5	19.00	19	968.71	1
Auto. Lub.	314621	13	21.50	43	1061.33	1
Auto. Lub.	314622	13	8.80	88	67.33	1
Timer	314631	38	27.50	110	2421.57	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	38	7.30	73	796.47	1
Waterjet Pump	420921	0	0.00	0	0.00	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	0	0.00	0	0.00	2
BoosterPump	5231071	3	4.50	9	262.91	2
Fuel/Wa.Sep.	5231081	13	21.50	43	801.33	1
Inverter	5241121	8	3.30	33	160.73	1
Battery	5241131	41	50.00	50	2509.60	1
Battery	5241132	82	31.40	157	523.37	1
Alternator	5241141	8	4.90	98	44.90	1
Alternator	5241142	15	6.70	134	36.71	1
Alternator	5241143	8	6.90	69	174.93	1
Hyd. Motor	5251181	31	22.00	22	24911.36	2
Hyd. Pump	5251191	41	26.00	26	24728.06	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 296.08
Total warehouse requirement repair level 2 = 111.72
Total warehouse requirement repair level 3 = 0.00

PERIOD 6 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	10	13.00	13	2563.25	2
Seawater Pump	27281	3	10.00	5	5448.99	2
Seawater Pump	27282	3	2.60	13	28.42	2
Seawater Pump	27283	6	4.00	20	116.49	2
Seawater Pump	27284	6	4.00	20	104.49	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	2	9.00	3	2048.99	2
Freshwa. HE	28331	2	8.00	4	740.00	1
Hyd.Fluid HE	28341	2	8.00	4	940.00	1
Pres.Red.Valve	332521	1	2.60	26	25.65	1
Pres.Red.Valve	312522	3	1.34	134	2.69	1
Dir.Con.Valve	312531	31	14.70	147	944.73	1
Dir.Con.Valve	312532	61	6.05	605	33.50	1
Steering Motor	312541	5	8.00	8	1491.83	2
Filter	313581	28	19.80	99	327.80	1
Starter	313591	7	22.00	22	1352.14	1
Auto. Lub.	314621	15	23.00	46	1222.91	1
Auto. Lub.	314622	15	9.50	95	76.99	1
Timer	314631	44	29.75	119	2801.66	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	44	7.80	78	919.24	1
Waterjet Pump	420921	0	0.00	0	0.00	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	0	0.00	0	0.00	2
BoosterPump	5231071	5	6.00	12	429.58	2
Fuel/Wa.Sep.	5231081	15	23.00	46	922.91	1
Inverter	5241121	9	3.50	35	179.75	1
Battery	5241131	31	43.00	43	1903.13	1
Battery	5241132	61	27.00	135	393.06	1
Alternator	5241141	9	5.20	104	50.20	1
Alternator	5241142	18	7.35	147	43.35	1
Alternator	5241143	9	7.30	73	196.35	1
Hyd. Motor	5251181	29	22.00	22	23307.70	2
Hyd. Pump	5251191	31	22.00	22	18711.36	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 282.49
Total warehouse requirement repair level 2 = 109.02
Total warehouse requirement repair level 3 = 0.00

PERIOD 7 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	11	13.00	13	2816.33	2
Seawater Pump	27281	3	10.00	5	5448.99	2
Seawater Pump	27282	3	2.60	13	28.42	2
Seawater Pump	27283	7	4.40	22	134.14	2
Seawater Pump	27284	7	4.40	22	120.14	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	2	9.00	3	2048.99	2
Freshwa. HE	28331	2	8.00	4	740.00	1
Hyd.Fluid HE	28341	2	8.00	4	940.00	1
Pres.Red.Valve	332521	3	4.60	46	73.58	1
Pres.Red.Valve	312522	5	1.73	173	3.98	1
Dir.Con.Valve	312531	30	14.50	145	914.49	1
Dir.Con.Valve	312532	59	5.95	595	32.50	1
Steering Motor	312541	6	9.00	9	1785.83	2
Filter	313581	32	21.20	106	373.17	1
Starter	313591	9	25.00	25	1735.10	1
Auto. Lub.	314621	19	26.00	52	1545.79	1
Auto. Lub.	314622	19	10.70	107	96.18	1
Timer	314631	34	26.00	104	2168.08	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	34	6.90	69	714.50	1
Waterjet Pump	420921	1	15.00	3	11067.08	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	1	4.00	4	210.00	2
BoosterPump	5231071	6	6.50	13	512.40	2
Fuel/Wa.Sep.	5231081	19	26.00	52	1165.79	1
Inverter	5241121	10	3.70	37	198.71	1
Battery	5241131	30	42.00	42	1842.43	1
Battery	5241132	59	26.60	133	380.61	1
Alternator	5241141	10	5.50	110	55.48	1
Alternator	5241142	20	7.75	155	47.75	1
Alternator	5241143	10	7.70	77	217.75	1
Hyd. Motor	5251181	26	20.00	20	20901.98	2
Hyd. Pump	5251191	30	22.00	22	18109.55	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 291.43
Total warehouse requirement repair level 2 = 113.32
Total warehouse requirement repair level 3 = 15.00

PERIOD 8 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	12	14.00	14	3069.28	2
Seawater Pump	27281	4	12.00	6	7256.57	2
Seawater Pump	27282	4	3.00	15	35.49	2
Seawater Pump	27283	6	4.00	20	116.49	2
Seawater Pump	27284	6	4.00	20	104.49	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	3	4.60	46	73.58	1
Pres.Red.Valve	312522	7	2.05	205	5.20	1
Dir.Con.Valve	312531	39	16.50	165	1186.52	1
Dir.Con.Valve	312532	79	6.88	688	42.43	1
Steering Motor	312541	7	10.00	10	2079.50	2
Filter	313581	39	23.40	117	452.37	1
Starter	313591	10	26.00	26	1926.46	1
Auto. Lub.	314621	22	27.50	55	1787.75	1
Auto. Lub.	314622	22	11.50	115	110.49	1
Timer	314631	29	24.00	96	1851.08	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	0	0.00	0	0.00	2
Solenoid	314651	29	6.40	64	611.86	1
Waterjet Pump	420921	2	20.00	4	22094.87	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	7	0.92	92	5.42	2
Disch.Elbow	420933	7	3.50	118	3.97	2
Steer.Nozzle	420941	2	6.00	6	408.28	2
BoosterPump	5231071	7	7.00	14	595.00	2
Fuel/Wa.Sep.	5231081	22	27.50	55	1347.75	1
Inverter	5241121	12	4.10	41	236.49	1
Battery	5241131	39	48.00	48	2388.37	1
Battery	5241132	79	30.80	154	504.79	1
Alternator	5241141	12	6.00	120	66.00	1
Alternator	5241142	24	8.50	170	56.49	1
Alternator	5241143	12	8.50	85	260.49	1
Hyd. Motor	5251181	34	23.00	23	27316.62	2
Hyd. Pump	5251191	39	25.00	25	23524.90	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 315.83
Total warehouse requirement repair level 2 = 128.42
Total warehouse requirement repair level 3 = 20.00

PERIOD 9 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	12	14.00	14	3069.28	2
Seawater Pump	27281	3	10.00	5	5448.99	2
Seawater Pump	27282	3	2.60	13	28.42	2
Seawater Pump	27283	7	4.40	22	134.14	2
Seawater Pump	27284	7	4.40	22	120.14	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	4	5.30	53	97.29	1
Pres.Red.Valve	312522	9	2.32	232	6.37	1
Dir.Con.Valve	312531	17	10.90	109	520.91	1
Dir.Con.Valve	312532	34	4.52	452	19.82	1
Steering Motor	312541	8	11.00	11	2372.92	2
Filter	313581	64	30.00	150	733.93	1
Starter	313591	12	29.00	29	2308.98	1
Auto. Lub.	314621	20	26.50	53	1626.46	1
Auto. Lub.	314622	20	11.00	110	100.95	1
Timer	314631	24	22.00	88	1533.91	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	1	0.35	35	1.85	2
Solenoid	314651	24	5.80	58	508.98	1
Waterjet Pump	420921	2	20.00	4	22094.87	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	8	0.98	98	5.86	2
Disch.Elbow	420933	8	3.50	126	4.27	2
Steer.Nozzle	420941	2	6.00	6	408.28	2
BoosterPump	5231071	8	7.50	15	677.42	2
Fuel/Wa.Sep.	5231081	20	26.50	53	1226.46	1
Inverter	5241121	13	4.30	43	255.33	1
Battery	5241131	17	32.00	32	1051.94	1
Battery	5241132	34	20.20	101	224.20	1
Alternator	5241141	13	6.25	125	71.24	1
Alternator	5241142	27	9.00	180	63.00	1
Alternator	5241143	13	8.80	88	281.83	1
Hyd. Motor	5251181	28	21.00	21	22505.83	2
Hyd. Pump	5251191	17	16.00	16	10282.46	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 287.99
Total warehouse requirement repair level 2 = 117.83
Total warehouse requirement repair level 3 = 20.00

PERIOD 10 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	13	14.00	14	3322.11	2
Seawater Pump	27281	3	10.00	5	5448.99	2
Seawater Pump	27282	3	2.60	13	28.42	2
Seawater Pump	27283	6	4.00	20	116.49	2
Seawater Pump	27284	6	4.00	20	104.49	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	5	5.90	59	120.92	1
Pres.Red.Valve	312522	10	2.45	245	6.95	1
Dir.Con.Valve	312531	17	10.90	109	520.91	1
Dir.Con.Valve	312532	34	4.52	452	19.82	1
Steering Motor	312541	9	11.00	11	2666.12	2
Filter	313581	83	34.00	170	947.09	1
Starter	313591	13	30.00	30	2500.17	1
Auto. Lub.	314621	17	24.50	49	1384.39	1
Auto. Lub.	314622	17	10.10	101	86.60	1
Timer	314631	25	22.25	89	1597.36	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	1	0.35	35	1.85	2
Solenoid	314651	25	5.90	59	529.58	1
Waterjet Pump	420921	3	25.00	5	33116.19	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	8	0.98	98	5.86	2
Disch.Elbow	420933	8	3.50	126	4.27	2
Steer.Nozzle	420941	3	7.00	7	604.64	2
BoosterPump	5231071	9	8.00	16	759.69	2
Fuel/Wa.Sep.	5231081	17	24.50	49	1044.39	1
Inverter	5241121	15	4.60	46	292.91	1
Battery	5241131	17	32.00	32	1051.94	1
Battery	5241132	34	20.20	101	224.20	1
Alternator	5241141	15	6.70	134	81.71	1
Alternator	5241142	31	9.65	193	71.64	1
Alternator	5241143	15	9.50	95	324.49	1
Hyd. Motor	5251181	18	17.00	17	14484.85	2
Hyd. Pump	5251191	17	16.00	16	10282.46	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 291.27
Total warehouse requirement repair level 2 = 114.43
Total warehouse requirement repair level 3 = 25.00

PERIOD 11 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	14	15.00	15	3574.83	2
Seawater Pump	27281	3	10.00	5	5448.99	2
Seawater Pump	27282	3	2.60	13	28.42	2
Seawater Pump	27283	7	4.40	22	134.14	2
Seawater Pump	27284	7	4.40	22	120.14	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	6	6.50	65	144.48	1
Pres.Red.Valve	312522	11	2.57	257	7.52	1
Dir.Con.Valve	312531	26	13.50	135	793.49	1
Dir.Con.Valve	312532	53	5.64	564	29.49	1
Steering Motor	312541	10	12.00	12	2959.16	2
Filter	313581	4	7.40	37	51.48	1
Starter	313591	15	32.00	32	2882.40	1
Auto. Lub.	314621	20	26.50	53	1626.46	1
Auto. Lub.	314622	20	11.00	110	100.95	1
Timer	314631	34	26.00	104	2168.08	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	1	0.35	35	1.85	2
Solenoid	314651	34	6.90	69	714.50	1
Waterjet Pump	420921	3	25.00	5	33116.19	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	8	0.98	98	5.86	2
Disch.Elbow	420933	8	3.50	126	4.27	2
Steer.Nozzle	420941	3	7.00	7	604.64	2
BoosterPump	5231071	10	8.50	17	841.83	2
Fuel/Wa.Sep.	5231081	20	26.50	53	1226.46	1
Inverter	5241121	17	4.90	49	330.39	1
Battery	5241131	26	39.00	39	1599.50	1
Battery	5241132	53	25.20	126	343.22	1
Alternator	5241141	17	7.15	143	92.14	1
Alternator	5241142	36	10.40	208	82.39	1
Alternator	5241143	17	10.10	101	367.10	1
Hyd. Motor	5251181	25	20.00	20	20100.00	2
Hyd. Pump	5251191	26	20.00	20	15701.98	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 294.86
Total warehouse requirement repair level 2 = 124.73
Total warehouse requirement repair level 3 = 25.00

PERIOD 12 SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node	Part#	Demand	W'house (cuft)	EOQ	Total Cost	Repair Level
Eng.Oil Cooler	2101	3	5.00	10	774.49	1
Freshwa.Pump	27271	21	18.00	18	5341.65	2
Seawater Pump	27281	4	12.00	6	7256.57	2
Seawater Pump	27282	4	3.00	15	35.49	2
Seawater Pump	27283	7	4.40	22	134.14	2
Seawater Pump	27284	7	4.40	22	120.14	2
Hyd. Pump	27291	2	6.00	6	1228.28	1
Eng.Coolant HE	28321	3	12.00	4	3060.00	2
Freshwa. HE	28331	3	10.00	5	1098.99	1
Hyd.Fluid HE	28341	3	10.00	5	1398.99	1
Pres.Red.Valve	332521	6	6.50	65	144.48	1
Pres.Red.Valve	312522	12	2.68	268	8.08	1
Dir.Con.Valve	312531	14	9.90	99	429.90	1
Dir.Con.Valve	312532	30	4.24	424	17.74	1
Steering Motor	312541	11	12.00	12	3252.05	2
Filter	313581	33	21.40	107	384.49	1
Starter	313591	15	32.00	32	2882.40	1
Auto. Lub.	314621	23	28.50	57	1868.37	1
Auto. Lub.	314622	23	11.70	117	115.25	1
Timer	314631	49	31.25	125	3118.31	1
Manifold	314641	0	0.00	0	0.00	2
Manifold	314642	3	0.60	60	3.36	2
Solenoid	314651	49	8.30	83	1021.41	1
Waterjet Pump	420921	4	25.00	5	44134.17	3
Disch.Elbow	420931	1	4.00	4	68.71	2
Disch.Elbow	420932	8	0.98	98	5.86	2
Disch.Elbow	420933	8	3.50	126	4.27	2
Steer.Nozzle	420941	4	8.00	8	800.00	2
BoosterPump	5231071	11	9.00	18	923.87	2
Fuel/Wa.Sep.	5231081	23	28.50	57	1408.37	1
Inverter	5241121	17	4.90	49	330.39	1
Battery	5241131	14	29.00	29	868.98	1
Battery	5241132	30	19.00	95	198.97	1
Alternator	5241141	17	7.15	143	92.14	1
Alternator	5241142	35	10.25	205	80.25	1
Alternator	5241143	17	10.10	101	367.10	1
Hyd. Motor	5251181	20	18.00	18	16089.44	2
Hyd. Pump	5251191	14	15.00	15	8474.83	2
Cable	5251201	0	0.00	0	0.00	1
Engine On/Off	5261231	1	2.60	26	23.65	1
Throttle	5261241	0	0.00	0	0.00	1
Steer.Lever	5261251	0	0.00	0	0.00	1
Clutch Lever	5261261	0	0.00	0	0.00	1

Total warehouse requirement repair level 1 = 298.97
Total warehouse requirement repair level 2 = 124.88
Total warehouse requirement repair level 3 = 25.00

APPENDIX 2.

Program Listing

```
10  '
20  '
30  '
40  '                TRANSIENT  STATE
50  '
60  '      MAINTENANCE  REQUIREMENTS  PLANNING
70  '
80  '      * * * * * * * * * * * * * * * * * *
90  '
100 '
110 '       BY:            KENNETH W. BRAMMER
120 '
130 '       ADVISOR:       DR. CHARLES J. MALMBORG
140 '
150 '       DATE:          MARCH 1985
160 '
170 '       DEPARTMENT:    IEOR
180 '
190 '       UNIVERSITY:    VIRGINIA TECH
200 '
210 '
220 '
230 '       DEFINITION OF VARIABLES
240 '       ______________________
250 '
260 ' A%,C%,N%,P%,Q%,V%,VT%,VV%..do loop counter variables
270 ' A$            ....inKey string variable
280 ' ACOST(e)      ....average operation cost per end item
290 '                    per period in environment e
300 ' AMT%(p)       ....number end items procured at
310 '                    procurement p
320 ' AVAIL         ....sum of environment availabilities
330 ' AVAILABILITY(e).availability of end items per period
340 '                    in environment e
350 ' AVGMTTR(e)    ....MTTR for an end item per period in
360 '                    environment e
370 ' AVGPF(e)      ....probability of end item failure per
380 '                    period in environment e
390 ' AVGSTREAM     ....average number of end items operating
400 '                    in a stream during a period
410 ' BV(e)         ....book value at retirement for an end
420 '                    item in environment e
430 ' C1(e)         ....capital costs for end items in
440 '                    environment e
450 ' C2            ....repair costs for end items in
460 '                    environment e
470 ' C3            ....shortage costs for end items in
480 '                    environment e
490 ' C4(e)         ....inventory costs for end items in
500 '                    environment e
510 ' C5(e), C6(e), C7(e), C8(e), C10(e)..present worth
520 '                    values of costs for environment e
530 ' CC            ....first cost of an end item
540 ' CCH(n)        ....holding cost for stockable item on
550 '                    node n
560 ' CCP(n)        ....procurement cost for stockable item
570 '                    on node n
580 ' CDF$, CDFP$,CDFP%..failure curve file variable
590 ' CDF%          ....failure curve number
600 ' CDF1%(n), CDF2%(n), CDF3%(n), CDF4%(n), CDF5%(n)
610 '               .......failure curves for node n in
620 '                    environments 1,2,3,4, or 5
630 ' CH            ....holding cost for stockable item
640 ' CL            ....# of days between EOQ orders
650 ' CNT1%,CNT2%,CNT3%,CNT4...counter variables
660 ' CODE%         ....repair level for stocking components
670 ' CP            ....procurement cost for stockable items
680 ' CT5, CT5$     ....time parameter used in probability of
690 '                    failure cumulative distributions
700 ' D%            ....signal to compute end item
710 '                    procurement cost if = 1
720 ' D2%           ....signal if = 1, the component
730 '                    cumulative probability function
740 '                    regenerates during period
750 ' DEMAND%(e)    ....demand for end items in environment e
```

```
760 ' F%            ....signal for chaining to designate spot
770 '                      in program to chain back to
780 ' Fn$           ....all Fn$ variables are print formats
790 ' FAC           ....variable for binomial availability
800 '                      computation
810 ' FILEn$        ....all FILEn$ variables are file names
820 ' FR2           ....fraction of period until component
830 '                      cumulative probability function
840 '                      regenerates
850 ' FRAC          ....fraction of period during which
860 '                      number of end items in operation
870 '                      exceeds demand for environment e
880 ' FT            ....cumulative probability of component
890 '                      failure during period
900 ' FT3(n)        ....warehouse requirement for stockable
910 '                      items at node n
920 ' H             ....# of 2*MTBF time spaces contained
930 '                      in one time period
940 ' I             ....per period interest rate
950 ' I$            ....stockable item name
960 ' IAP%, IIAP%   ....# components in initial procurement
970 ' IP%(s)        ....number of end items placed in
980 '                      stream s
990 ' KKEY%(s)      ....# of failure environment in which
1000 '                      stream s operates
1010 'L$            ....signal variable for procurement
1020 '                      = A infinite procurement
1030 '                      = B procure to upper limit
1040 'L1,L2,L3,L4,L5..summing variables used to calculate
1050 '                 union of node failure probabilities
1060 'LL,LLL,LLLL,RR,RRR..array subscripts used to calc.
1070 '                 union of node failure probabilities
1080 'LLT(n)        ....lead time for stockable item
1090 '                      on node n
1100 'LMR%          ....length of model run
1110 'LPRIME        ....EOQ minimum cost procurement level
1120 'LT            ....stockable item lead time
1130 'LTP%          ....length of time period
1140 'LX%           ....number of days in a time period
1150 'M( ),MM( )    ....holding arrays used to calculate
1160 '                 union of node failure probabilities
1170 'MTBF,MTBFP    ....mean time between failure for
1180 '                      components
1190 'MTBF1(n),MTBF2(n),MTBF3(n),MTBF4(n),MTBF5(n)
1200 '              ....MTBF for component n in
1210 '                      environments 1,2,3,4, or 5
1220 'MTTR,MTTRP    ....mean time to repair for components
1230 'MTTR1(n),MTTR2(n),MTTR3(n),MTTR4(n),MTTR5(n)
1240 '              ....MTTR for component n in
1250 '                      environments 1,2,3,4, or 5
1260 'MTTRAVG(s)    ....end item MTTR for stream s
1270 'MTTRSUM       ....weighted sum of component MTTR's in
1280 '                      stream s
1290 'NDE(n)        ....failure probability for component n
1300 'NN$(n)        ....node name for node n
1310 'NNAME$(n)     ....name for stockable item on node n
1320 'NO.ENVIR$(e)....name of environment e
1330 'NO.ENVIR%     ....number of failure environments
1340 'NO.RL%        ....number of repair levels
1350 'NOD(n)        ....previous period failure probability
1360 '                      for node n
1370 'NODE(n),NODE$,NODEP(n)...failure probability for
1380 '                              node n
1390 'NODE1%(n),NODE2%(n),NODE3%(n),NODE4%(n),NODE5%(n)
1400 '              ....failure probability for node n in
1410 '                      environments 1,2,3,4, or 5
1420 'NS%           ....# of streams of end items
1430 'OED%          ....# periods between EOQ orders
1440 'P(n)          ....end item failure probability during
1450 '                      period equal to MTTR
1460 'PA%           ....procurement # (for end items)
1470 'PART.NO$      ....stockable item part #
1480 'PB            ....present worth conversion factor or
1490 '                      period cost conversion factor
1500 'PC            ....end item procurement cost
```

```
1510 'PL%          ....# of procurement policy selected
1520 'PN$          ....stockable item part #
1530 'PPN$         ....stockable item part #
1540 'PPR(n)       ....price for stockable item on node n
1550 'PPSPR%(n)    ....spare part requirement for stockable
1560 '                  item on node n
1570 'PR           ....price for stockable item
1580 'PRC          ....total end item procurement costs
1590 'PSPR%        ....spare part requirement for
1600 '                  stockable item
1610 'QPRIME       ....economic order quantity of
1620 '                  stockable item
1630 'QY           ....number of stockable item needed to
1640 '                  support node
1650 'R1%          ....signal used to note when a
1660 '                  procurement of end items occurs
1670 'RA(e)        ....retirement age for end items in
1680 '                  environment e
1690 'RC           ....repair cost
1700 'RL%(n)       ....repair level for component at node n
1710 'RLW%         ....repair level
1720 'RRA(s)       ....retirement period for end items in
1730 '                  environment e
1740 'SC1,SC2,SC3,SC4,STC..period equivalent logistics
1750 '                     policy costs per environment
1760 'SIG%         ....signal to note when to cease
1770 '                  procurement of end items
1780 'SMTTR(e)     ....sum of weighted node MTTR's for
1790 '                  environment e
1800 'SP%(s)       ....number of end items in operation in
1810 '                  stream s at beginning of period
1820 'SP2%(s)      ....number of end items in operation in
1830 '                  stream s at end of period
1840 'SPR%         ....spare part requirement for stockable
1850 '                  item
1860 'SPWA%        ....# for choice of spare parts routine
1870 'SR           ....warehouse requirement for stockable
1880 '                  item
1890 'SSC1,SSC2,SSC3,SSC4,SSTC..total period equivalent
1900 '                          logistic policy costs
1910 'ST%(s)       ....# of end items in stream s
1920 'SUM          ....average # of end items in population
1930 '                  in operation during a period
1940 'SUM%         ....total # of end items in population
1950 '                  in operation at end of period
1960 'SUMA,SUMA$   ....cumulative probability of failure
1970 '                  through last period
1980 'SUMBINOM     ....availability for environment e
1990 'SUMPF(e)     ....sum of end item failure probabilities
2000 '                  for streams of end items operating
2010 '                  in environment e
2020 'SUMSP%(e)    ....# end items operating in environment
2030 '                  e at start of period
2040 'SUMSP2%(e)   ....# end items operating in environment
2050 '                  e at start of period
2060 'SUMSTREAM(e)....average # end items operating in
2070 '                  environment e during period
2080 'SV           ....salvage value for end item at fatal
2090 '                  failure'
2100 'SVC          ....total salvage value realized during
2110 '                  period
2120 'TC(E)        ....total cost of logistics policy for
2130 '                  period for environment e
2140 'TCPRIME      ....cost of optimal EOQ policy for
2150 '                  stockable item
2160 'TOTALAVAIL(t)...end item population availability for
2170 '                  period t
2180 'TT%          ....number of periods since procurement
2190 '                  of end items in stream s
2200 'TTIME%(s)    ....period in which end items in stream s
2210 '                  were procured
2211 'UP%          ....= 0 printer listing of all output
2212 '                 = 1 printer listing of summaries only
2220 'UP.LIMIT%    ....upper population limit on end items
2230 'UPA%         ....end item uniform procurement amount
```

```
2240 'W%            ....= 2 cease procurement at upper limit
2250 '                  = 1 maintain upper population limit
2260 'WH(r)         ....period warehouse requirement at
2270 '                      repair level r
2280 'XX,XXX,YY,YYY...array subscripts for calculating
2290 '                      union of failure probabilities
2300 '
2310 '
2320 '
2330 '       "MAIN" PROGRAM LISTING
2340 '
2350 '       This section of the program is the driver for
2360 '       the remainder of the model and represents the
2370 '       computational basis for the model developed.
2380 '
2390 '
2400 '       CHAIN TO THE INTRODUCTORY PROGRAM
2410 '       ________________________________
2420 '
2430 CHAIN "INTRO", 1, ALL
2440 '
2450 '       TURN OFF BASIC SCREEN KEYS
2460 '       __________________________
2470 '
2480 FOR I% = 1 TO 10:  KEY I%,"":  NEXT I%
2490 KEY OFF
2500 '
2510 '       CHAIN TO FIRST DATA INPUT PROGRAM
2520 '       ________________________________
2530 '
2540 CHAIN "INPUT1", 26, ALL
2550 '
2560 '      INITIALIZE VARIABLES
2570 '      ____________________
2580 '
2590 '      Note:  The majority of the variables are
2600 '             initialized in the input program,
2610 '             INPUT1.  All arrays are dimensioned
2620 '             in INPUT1.
2630 '
2640 F% = 0: NS% = 1: D% = 0: SIG%=0: PLEN%=0: PRC=0
2650 '
2660 '      IF AN EXISTING POPULATION IS BEING EVALUATED,
2670 '       CHAIN TO PROGRAM TO ALLOW USER TO SPLIT THE
2680 '                END ITEMS INTO STREAMS
2690 '      ____________________________________________
2700 '
2710 IF B$ = "A" THEN 2650
2720 CHAIN "EXISTING", 10, ALL
2730 '
2740 '      IF THERE IS MORE THAN ONE FAILURE ENVIRONMENT
2750 '      AND IF AN INITIAL PROCUREMENT HAS BEEN MADE,
2760 '      CHAIN TO PROGRAM TO SPLIT END ITEMS INTO STREAMS
2770 '      ________________________________________________
2780 '
2790 IF IP%(NS%) = 0 THEN 2940
2800 IF NO.ENVIR% = 1 THEN 2870
2810 CHAIN "DIVIDE", 400, ALL
2820 '
2830 '      INITIALIZE VARIABLES BASED ON INITIAL
2840 '              PROCUREMENT AMOUNT
2850 '      ____________________________________
2860 '
2870 KKEY%(NS%) = 1:  TTIME%(NS%) = T%
2880 IF IP%(NS%) > 0 THEN NS% = NS% + 1:   D%=1
2890 '
2900 '      BEGIN LOOP TO EVALUATE PERIOD BY PERIOD
2910 '                  REQUIREMENTS
2920 '      ___________________________________
2930 '
2940 T% = 1
2950    CNT1% = 0
2960   FOR N% = 1 TO 5
2970     C4(N%) = 0:  MP(N%) = 0
2980   NEXT N%
```

```
2990 '
3000 '      BEGIN LOOP TO EVALUATE STREAM BY STREAM
3010 '                    REQUIREMENTS
3020 '      ____________________________________
3030 '
3040   FOR C% = 1 TO NS%-1
3050     IF ST%(C%) = 0 THEN 4110
3060     TT% = T% - TTIME%(C%)
3070     G%(C%) = IP%(C%) - ST%(C%)
3080     IF TT% >= RRA(C%) + 1 THEN 4090
3090     FILE3$ = "0" + STR$(C%)
3100     FOR N% = 1 TO 126
3110       NODE(N%) = 0:  NDE(N%) = 0
3120     NEXT N%
3130 '
3140 '      LOAD ENVIRONMENT SPECIFIC NODE DATA INTO ARRAYS
3150 '      ___________________________________________________
3160 '
3170     ON KKEY%(C%) GOSUB 5360,5520,5680,5840,6000
3180     MTTRSUM = 0:  CNT4 = 0
3190 '
3200 '      BEGIN LOOP TO EVALUATE NODE BY NODE
3210 '             FAILURE PROBABILITIES
3220 '      ____________________________________
3230 '
3240     FOR N% = 1 TO 126
3250       IF NODEP%(N%) = 0 THEN 3790
3260       MTBF = MTBFP(N%):   A% = NODEP%(N%)
3270       CDF% = CDFP%(N%)
3280       FILE9$ = "0" + STR$(KKEY%(C%)) + STR$(N%)
3290       IF CDF% = 6 THEN 3520
3300       IF (2*MTBF) < 1 THEN GOSUB 7320
3310       IF TT% = 1 AND TT% <= (2*MTBF) THEN CT5 = 1
3320       GOTO 3450
3330       OPEN "r",#2,FILE3$,12
3340       FIELD #2,4 AS NODE$, 4 AS SUMA$, 4 AS CT5$
3350       GET #2,A%
3360       CT5 = CVS(CT5$) + 1
3370       IF CT5 <= (2*MTBF) THEN 3450
3380       FR2 = (2*MTBF) - CT5 + 1
3390       CT5 = 1 - FR2:  D2% = 1
3400 '
3410 '        based on failure curve selection gosub to
3420 '      calculate node cumulative probability of failure
3430 '      ...............................................
3440 '
3450       ON CDF% GOSUB 6850,6900,6950,7020,7070
3460       IF CT5 = (2*MTBF) THEN CT5 = 0
3470       IF FT > 1 THEN FT = 1
3480       IF TT% = 1 AND D2% = 0 THEN 3570
3490       NODE(A%) = CVS(NODE$):  SUMA = CVS(SUMA$)
3500       CLOSE 2
3510       GOTO 3580
3520       OPEN "r",#2,FILE9$,4
3530       FIELD #2,4 AS NODE$
3540       GET #2,TT%:  NODE(A%) = CVS(NODE$)
3550       CLOSE #2
3560       GOTO 3670
3570       SUMA = 0
3580       IF D2% = 1 THEN NODE(A%) = (FR2*(1-SUMA))
                 + ((1-FR2)*FT)
3590       GOTO 3620
3600       NODE(A%) = FT - SUMA
3610       GOTO 3670
3620       D2% = 0
3630 '
3640 '      gosub to sum weighted node MTTR's
3650 '      .................................
3660 '
3670       GOSUB 7090
3680       IF CDF% = 6 THEN 3790
3690       SUMA = FT
3700       IF CT5 = 0 THEN SUMA = 0
3710       OPEN "r",#2,FILE3$,12
3720       FIELD #2,4 AS NODE$, 4 AS SUMA$, 4 AS CT5$
```

```
3730        GET #2,A%
3740        LSET NODE$ = MKS$(NODE(A%))
3750        LSET SUMA$ = MKS$(SUMA)
3760        LSET CT5$ = MKS$(CT5)
3770        PUT #2,A%
3780        CLOSE 2
3790      NEXT N%
3800 '
3810 '      GOSUB TO COMPUTE END ITEM MTTR FOR STREAM
3820 '      ____________________________________________
3830 '
3840      GOSUB 7180
3850 '
3860 '      GOSUB TO COMPUTE #UNITS LEFT OPERATING IN STREAM
3870 '      ____________________________________________________
3880 '
3890      GOSUB 5010
3900 '
3910 '         GOSUB TO CALCULATE NODE BY NODE FAILURE
3920 '      PROBABILITIES FOR MAINTENANCE TREE FOR STREAM
3930 '      ____________________________________________
3940 '
3950      GOSUB 7460
3960      IF SPWA% = 3 THEN 4030
3970      CNT1% = CNT1% + 1
3980 '
3990 '      GOSUB TO COMPUTE STREAM SPAREPARTS DEMAND
4000 '      ____________________________________________
4010 '
4020      ON SPWA% GOSUB 9690,10100
4030      GOTO 4180
4040 '
4050 '      IF STREAM RETIRED OR ALL UNITS FAILED FATALLY,
4060 '                        PRINT MESSAGE
4070 '      ____________________________________________
4080 '
4090      ST%(C%) = 0:  CLS
4091      IF UP% = 1 THEN 4100
4092      LPRINT "All remaining units of stream(";C%;")"
                "have been retired.":   GOTO 4210
4100      PRINT "All remaining units of stream(";C%;")"
                "have been retired.":   GOTO 4180
4110      IF UP% = 1 THEN 4119
4111      LPRINT "Time Period"; T%
4112      LPRINT "Stream #"; C%
4113      LPRINT "# Units Initially"
4114      LPRINT "Procured"; IP%(C%)
4115      SP%(C%)=0:  SP2%(C%)=0
4116      LPRINT "Stream has been Retired":  GOTO 4210
4119      CLS:  COLOR 2,0:  LOCATE 1,40
4120      PRINT "Time Period"; T%
4130      LOCATE 2,40:  PRINT "Stream #"; C%
4140      LOCATE 3,40:  PRINT "# Units Initially"
4150      LOCATE 4,40:  PRINT "Procured"; IP%(C%)
4160      LOCATE 5,40:  SP%(C%)=0:  SP2%(C%)=0
4170      PRINT "Stream has been Retired"
4180      COLOR 3,0: LOCATE 20,50
4190      PRINT "Press any key to continue."
4200      A$ = INKEY$:  IF A$ = "" THEN 4200
4210    NEXT C%
4220 '
4230 '      GOSUB TO COMPUTE PERIOD SPAREPARTS AND
4240 '              WAREHOUSE REQUIREMENTS
4250 '      ____________________________________________
4260    IF SPWA% = 3 THEN 4320
4270    ON SPWA% GOSUB 9740,10240
4280 '
4290 '      GOSUB TO CALCULATE AVAILABILITY
4300 '      ____________________________________
4310 '
4320    GOSUB 8820
4330    IF T% = LMR% THEN 4830
4340 '
4350 '       IF PROCUREMENT POLICY IS BEING GENERATED
4360 '       GOTO 2310 OTHERWISE USER HAS OPPORTUNITY
```

```
4370 '      TO INPUT A PROCUREMENT AMOUNT OF END ITEMS
4380 '                    AT THIS POINT
4390 '
4400 '
4410   IF PL% <> 9 THEN 4690
4420   COLOR 3,0:   LOCATE 17,30
4430   PRINT "End of period; do you wish to"
            "make a procurement?"
4440   COLOR 2,0:  LOCATE 18,30
4450   INPUT "Enter yes or no >",A$
4460   IF A$ = "no" THEN 4830
4470 '
4480 '      GOSUB FOR MANUAL INPUT OF PROCUREMENT AMOUNT
4490 '
4500 '
4510   GOSUB 6690
4520   IF NO.ENVIR% = 1 THEN 4590
4530 '
4540 '      CHAIN TO PROGRAM TO DIVIDE PROCUREMENT
4550 '                  INTO STREAMS
4560 '
4570 '
4580   F% = 1:  CHAIN "DIVIDE", 400, ALL
4590   KKEY%(NS%) = 1:  TTIME%(NS%) = T%
4600   ST%(NS%) = IP%(NS%):  RRA(NS%) = RA(1)
4610   IF IP%(NS%) > 0 THEN NS% = NS% + 1: D%=1
4620   GOTO 4820
4630 '
4640 '      IF CNT3% INDICATES THAT IT IS TIME TO MAKE A
4650 '       PROCUREMENT OF END ITEMS THEN GOSUB TO THE
4660 '      ROUTINE THAT WILL GENERATE PROCUREMENT AMOUNT
4670 '
4680 '
4690   CNT3% = CNT3% + 1
4700   IF CNT3% < NO.P.PER% THEN 4830
4710   ON PL% GOSUB 6090,6090,6230,6230,6330,
                6330,6510,6510
4720 '
4730 '      CHAIN TO PROGRAM TO DIVIDE PROCUREMENT
4740 '                  INTO STREAMS
4750 '
4760 '
4770   IF NO.ENVIR% = 1 THEN 4790
4780   F% = 2:  CHAIN "DIVIDE", 400, ALL
4790   KKEY%(NS%) = 1:  TTIME%(NS%) = T%:
4800   ST%(NS%) = IP%(NS%):  RRA(NS%) = RA(1)
4810   IF IP%(NS%) > 0 THEN NS% = NS% + 1: D%=1
4820   CNT3% = 0
4830   IF T% = LMR% THEN 4900
4840 T% = T% + 1:  GOTO 2990
4850 '
4860 '      GOSUB TO CALCULATE PERIOD EQUIVALENT COST FOR
4870 '                      MODEL RUN
4880 '
4890 '
4900 GOSUB 12710
4910 IF UP% = 0 THEN 4940
4919 COLOR 3,0:  LOCATE 20,50
4920 PRINT "Press any key to continue.  "
4930 A$ = INKEY$:  IF A$ = "" THEN 4930
4940 '
4950 '      CHAIN TO PROGRAM TO GIVE PRINTER LISTING OF
4960 '                  MODEL RUN RESULTS
4970 '
4980 '
4990 CHAIN "PRINTOUT", 36000!, ALL
5000 END
5010 '
5020 '
5030 '
5040 '      E N D    O F    M A I N    P R O G R A M
5050 '
5060 '
5070 '      SUBROUTINE TO CALCULATE NUMBER OF UNITS THAT
5080 '              FAIL FATALLY DURING PERIOD
```

```
5090 '
5100 '        ____________________________________________
5110 SP%(C%) = ST%(C%)
5120 SUM6(C%) = SUM6(C%)
             + ((NODE(6)*ST%(C%))-INT(NODE(6)*ST%(C%)))
5130 ST%(C%) = ST%(C%) - INT(NODE(6)*ST%(C%))
5140 IF TRIG%(C%) = 1 THEN 5160
5150 IF SUM6(C%) >= .5 THEN ST%(C%) = ST%(C%) - 1:
                            TRIG%(C%) = 1
5160 IF SUM6(C%) >= 1! THEN SUM6(C%) = SUM6(C%) - 1:
                            TRIG%(C%) = 0
5170 IF TRIG%(C%) = 1 THEN 5190
5180 IF SUM6(C%) >= .5 THEN ST%(C%) = ST%(C%) - 1:
                            TRIG%(C%) = 1
5190 SP2%(C%)=ST%(C%)
5200 IF TT% = RRA(C%) THEN ST%(C%) = 0
5210 RETURN
5220 '
5230 '      SUBROUTINES TO SAVE NODE BY NODE INPUT DATA
5240 '            BY FAILURE ENVIRONMENT
5250 '      THESE SUBROUTINES ARE ALSO USED TO LOAD THE
5260 '      NODE BY NODE DATA BACK INTO THE PROPER ARRAY
5270 '      SO FAILURE PROBABILITIES MAY BE CALCULATED
5280 '        ____________________________________________
5290 '
5300 '      save failure environment #1 input
5310 '      ................................
5320 '
5330 MTBF1(A%) = MTBF:    NODE1%(A%) = A%
5340 CDF1%(A%) = CDF%:    MTTR1(A%) = MTTR
5350 RETURN
5360 '
5370 '      load failure environment #1 data into arrays
5380 '      ..........................................
5390 '
5400 FOR P% = 1 TO 126
5410   MTBFP(P%) = MTBF1(P%):  NODEP%(P%) = NODE1%(P%)
5420   CDFP%(P%) = CDF1%(P%):  MTTRP(P%) = MTTR1(P%)
5430 NEXT P%
5440 RETURN
5450 '
5460 '      save failure environment #2 input
5470 '      ................................
5480 '
5490 MTBF2(A%) = MTBF:    NODE2%(A%) = A%
5500 CDF2%(A%) = CDF%:    MTTR2(A%) = MTTR
5510 RETURN
5520 '
5530 '      load failure environment #2 data into arrays
5540 '      ..........................................
5550 '
5560 FOR P% = 1 TO 126
5570   MTBFP(P%) = MTBF2(P%):  NODEP%(P%) = NODE2%(P%)
5580   CDFP%(P%) = CDF2%(P%):  MTTRP(P%) = MTTR2(P%)
5590 NEXT P%
5600 RETURN
5610 '
5620 '      save failure environment #3 input
5630 '      ................................
5640 '
5650 MTBF3(A%) = MTBF:    NODE3%(A%) = A%
5660 CDF3%(A%) = CDF%:    MTTR3(A%) = MTTR
5670 RETURN
5680 '
5690 '      load failure environment #3 data into arrays
5700 '      ..........................................
5710 '
5720 FOR P% = 1 TO 126
5730   MTBFP(P%) = MTBF3(P%):   NODEP%(P%) = NODE3%(P%)
5740   CDFP%(P%) = CDF3%(P%):   MTTRP(P%) = MTTR3(P%)
5750 NEXT P%
5760 RETURN
5770 '
5780 '      save failure environment #4 input
5790 '      ................................
```

```
5800 '
5810 MTBF4(A%) = MTBF:   NODE4%(A%) = A%
5820 CDF4%(A%) = CDF%:   MTTR4(A%) = MTTR
5830 RETURN
5840 '
5850 '     load failure environment #4 data into arrays
5860 '     ..........................................
5870 '
5880 FOR P% = 1 TO 126
5890   MTBFP(P%) = MTBF4(P%):   NODEP%(P%) = NODE4%(P%)
5900   CDFP%(P%) = CDF4%(P%):   MTTRP(P%) = MTTR4(P%)
5910 NEXT P%
5920 RETURN
5930 '
5940 '     save failure environment #5 input
5950 '     ................................
5960 '
5970 MTBF5(A%) = MTBF:   NODE5%(A%) = A%
5980 CDF5%(A%) = CDF%:   MTTR5(A%) = MTTR
5990 RETURN
6000 '
6010 '     load failure environment #5 data into arrays
6020 '     ..........................................
6030 '
6040 FOR P% = 1 TO 126
6050   MTBFP(P%) = MTBF5(P%):   NODEP%(P%) = NODE5%(P%)
6060   CDFP%(P%) = CDF5%(P%):   MTTRP(P%) = MTTR5(P%)
6070 NEXT P%
6080 RETURN
6090 '
6100 '     SUBROUTINE TO PERFORM UNIFORM PROCUREMENT
6110 '     ________________________________________
6120 '
6130 SUM% = 0
6140 IF L$ = "A" THEN IP%(NS%) = UPA%:  RETURN
6150 IF SIG% = 1 THEN RETURN
6160 FOR P% = 1 TO NS%-1
6170   SUM% = SUM% + ST%(P%)
6180 NEXT P%
6190 IF SUM% + UPA% >= UP.LIMIT% THEN 6210
6200 IP%(NS%) = UPA%:  RETURN
6210 IF W% = 2 THEN SIG% = 1
6220 IP%(NS%) = UP.LIMIT% - SUM%:  RETURN
6230 '
6240 '     SUBROUTINE TO PERFORM BLOCK PROCUREMENT
6250 '     _______________________________________
6260 '
6270 SUM% = 0
6280 IF W% = 1 THEN RETURN
6290 FOR P% = 1 TO NS%-1
6300   SUM% = SUM% + ST%(P%)
6310 NEXT P%
6320 IP%(NS%) = AMT%(1) - SUM%:  RETURN
6330 '
6340 '     SUBROUTINE TO PERFORM INCREASING PROCUREMENT
6350 '     ____________________________________________
6360 '
6370 SUM% = 0:  PA% = PA% + 1
6380 IF PA% > 1 THEN 6400
6390 PA%=1:  GOTO 6500
6400 IF L$ = "A" THEN AMT%(PA%) =
                 CINT(AMT%(PA%-1)+AMT%(PA%-1)*UPA%/100):
                 GOTO 6500
6410 IF SIG% = 1 THEN RETURN
6420 FOR P% = 1 TO NS%-1
6430   SUM% = SUM% + ST%(P%)
6440 NEXT P%
6450 IF SUM% + CINT(AMT%(PA%-1)+AMT%(PA%-1)*UPA%/100) >=
                       UP.LIMIT% THEN 6480
6460 AMT%(PA%) = CINT(AMT%(PA%-1)+AMT%(PA%-1)*UPA%/100)
6470 GOTO 6500
6480 IF W% = 2 THEN SIG% = 1
6490 AMT%(PA%) = UP.LIMIT% - SUM%
6500 IP%(NS%) = AMT%(PA%):  RETURN
6510 '
```

```
6520 '      SUBROUTINE TO PERFORM DECREASING PROCUREMENT
6530 '      ____________________________________________
6540 '
6550 SUM% = 0:  PA% = PA% + 1
6560 IF PA% > 1 THEN 6580
6570 PA%=1:  GOTO 6680
6580 IF L$ = "A" THEN AMT%(PA%) =
                 CINT(AMT%(PA%-1)-AMT%(PA%-1)*UPA%/100):
                 GOTO 6680
6590 IF SIG% = 1 THEN RETURN
6600 FOR P% = 1 TO NS%-1
6610   SUM% = SUM% + ST%(P%)
6620 NEXT P%
6630 IF SUM% + CINT(AMT%(PA%-1)-AMT%(PA%-1)*UPA%/100) >=
                    UP.LIMIT% THEN 6660
6640 AMT%(PA%) = CINT(AMT%(PA%-1)-AMT%(PA%-1)*UPA%/100)
6650 GOTO 6680
6660 IF W% = 2 THEN SIG% = 1
6670 AMT%(PA%) = UP.LIMIT% - SUM%
6680 IP%(NS%) = AMT%(PA%):  RETURN
6690 '
6700 '      SUBROUTINE TO TAKE MANUAL PROCUREMENT INPUT
6710 '      ___________________________________________
6720 '
6730 CLS
6740 INPUT "Enter the number of units that"
                  "you wish to procure >",IP%(NS%)
6750 RETURN
6760 '
6770 '      SUBROUTINES TO CALCULATE CUMULATIVE FAILURE
6780 '      PROBABILITIES FOR BOTTOM OF THE TREE NODES
6790 '      ___________________________________________
6800 '
6810 '
6820 '      failure curve #1:  uniform distribution
6830 '      ......................................
6840 '
6850 FT = CT5 / (2*MTBF):  RETURN
6860 '
6870 '      failure curve #2:  exponential distribution
6880 '      ..........................................
6890 '
6900 FT = 1 - (1/(2.718282¢(CT5/MTBF))):  RETURN
6910 '
6920 '      failure curve #3:  increasing exponential
6930 '      .........................................
6940 '
6950 IF 2*MTBF-CT5 > 88 THEN FT = 0:  RETURN
6960 FT = (1/(2.718282¢((2*MTBF)/CT5))) *
                         (2.718282¢(CT5/(2*MTBF)))
6970 RETURN
6980 '
6990 '      failure curve #4:  ¬ normal distribution
7000 '      ........................................
7010 '
7020 FT = 1 - (1/(2.718282¢((CT5/MTBF)¢3))):  RETURN
7030 '
7040 '      failure curve #5:  bathtub distribution
7050 '      ......................................
7060 '
7070 FT = (1/(2.718282¢2)) * (2.718282¢(CT5/MTBF))
7080 RETURN
7090 '
7100 '      SUBROUTINE TO SUM THE WEIGHTED NODE MTTR'S
7110 '      ___________________________________________
7120 '
7130 IF N% = 6 THEN 7170
7140 MTTRSUM = MTTRSUM + (MTTRP(N%) * NODE(A%))
7150 CNT4 = CNT4 + NODE(A%)
7160 GOSUB 13210
7170 RETURN
7180 '
7190 '      SUBROUTINE TO COMPUTE MTTR FOR STREAM
7200 '      _____________________________________
7210 '
```

```
7220 IF CNT4 = 0 THEN MTTRAVG(C%) = 0: GOTO 7240
7230 MTTRAVG(C%) = MTTRSUM/CNT4
7240 RETURN
7250 '
7260 '       SUBROUTINE USED TO CALCULATE NODE FAILURE
7270 '        PROBABILITY IF THE EXPECTED LIFE OF THE
7280 '      COMPONENT REPRESENTED BY THAT NODE (2*MTBF)
7290 '             IS LESS THAN ONE PERIOD
7300 '      ____________________________________________
7310 '
7320 NODE(A%) = 1!
7330 IF TT% <> 1 THEN 7370
7340 H = INT(1/(2*MTBF)):  CT5 = 1 - (H*2*MTBF)
7350 ON CDF% GOSUB 6850,6900,6950,7020,7070
7360 NDE(A%) = H + FT - 1:  RETURN 3670
7370 OPEN "r",#2,FILE3$,12
7380 FIELD #2,4 AS NODE$, 4 AS SUMA$, 4 AS CT5$
7390 GET #2,A%
7400 CT5=CVS(CT5$):  SUMA=CVS(SUMA$)
7410 H = INT((1-(2*MTBF-CT5)) / (2*MTBF))
7420 CT5 = 1 - (H*2*MTBF) - (2*MTBF) + CT5
7430 ON CDF% GOSUB 6850,6900,6950,7020,7070
7440 CLOSE #2
7450 NDE(A%) = 1 - SUMA + H + FT:  RETURN 3670
7460 '
7470 '      SUBROUTINES TO CALCULATE NODE BY NODE FAILURE
7480 '       PROBABILITIES FROM THE BOTTOM TO THE TOP OF
7490 '                   THE MAINTENANCE TREE
7500 '      ____________________________________________
7510 '
7520 '
7530 '      print stream information on output screen
7540 '      ........................................
7550 '
7551 IF UP% = 1 THEN 7569
7552 LPRINT "Time Period";T%
7553 LPRINT "Stream #";C%
7554 LPRINT "# Units Initially"
7555 LPRINT "procured";IP%(C%)
7556 LPRINT "# Units in Operation-start";SP%(C%)
7557 LPRINT "# Units in Operation-end";SP2%(C%)
7558 LPRINT USING "\  \ ###.######";"MTTR";MTTRAVG(C%)
7559 LPRINT USING F8$;"node(6) =";NODE(6)
7560 GOTO 7680
7569 CLS:  COLOR 3,0,8
7570 LOCATE 1,50:   PRINT "Time Period";T%
7580 LOCATE 2,50:   PRINT "Stream #";C%
7590 LOCATE 3,50:   PRINT "# Units Initially"
7600 LOCATE 4,50:   PRINT "procured";IP%(C%)
7610 LOCATE 5,50
7620 PRINT "# Units in Operation-start";SP%(C%)
7630 LOCATE 6,50
7640 PRINT "# Units in Operation-end";SP2%(C%)
7650 LOCATE 7,50
7660 PRINT USING "\  \ ###.######";"MTTR";MTTRAVG(C%)
7670 LOCATE 15,48: PRINT USING F8$;"node(6) =";NODE(6)
7680 '
7690 '      driver for node summing subroutine
7700 '      ..................................
7710 '
7720 XX = 27:  XXX = 31:  YY = 7
7730 YYY = 11:  RR = -3:  RRR = 0
7740 FOR N% = 2 TO 5
7750    FOR P% = YY TO YYY
7760       L1 = 0: L2 = 0: L3 = 0: L4 = 0: L5 = 1
7770       LL = XX: LLL = XXX: CNT2% = 0
7780      FOR Q% = XX TO XXX
7790         M(Q%) = NODE(Q%)
7800         IF M(Q%) = 0 THEN CNT2% = CNT2% + 1
7810      NEXT Q%
7820      IF CNT2% = 5 THEN 7930
7830 '
7840 '      gosub to evaluate union of node probabilities
7850 '      .............................................
7860 '
```

```
7870      GOSUB 8500
7880      IF NODE(P%) = 0 THEN NODE(P%) = LLLL
7890 '
7900 '     print node failure probability
7910 '     ..............................
7920 '
7921      IF UP% = 1 THEN 7930
7922      LPRINT USING F9$;"node(";P%;") =";NODE(P%)
7923      GOTO 7950
7930      COLOR 2,0:  LOCATE P%-6+RRR,1
7940      PRINT USING F9$;"node(";P%;") =";NODE(P%)
7950      XX=XX+5:XXX=XXX+5
7960    NEXT P%
7970    RRR = RRR+1: L1 = 0: L2 = 0: L3 = 0: L4 = 0
7980    L5 = 1: LL = YY: LLL = YYY: CNT2% = 0
7990    FOR Q% = YY TO YYY
8000      M(Q%) = NODE(Q%)
8010      IF M(Q%) = 0 THEN CNT2% = CNT2% + 1
8020    NEXT Q%
8030    IF CNT2% = 5 THEN 8140
8040 '
8050 '     gosub to evaluate union of node probabilities
8060 '     ............................................
8070 '
8080    GOSUB 8500
8090    IF NODE(N%) = 0 THEN NODE(N%) = LLLL
8100 '
8110 '     print node failure probability
8120 '     ..............................
8130 '
8140   RR = RR+6: LOCATE RR,26
8141   IF UP% = 1 THEN 8150
8142   LPRINT USING F9$;"node(";N%;") =";NODE(N%)
8143   GOTO 8160
8150   PRINT USING F9$;"node(";N%;") =";NODE(N%)
8160   YY=YY+5:YYY=YYY+5
8170 NEXT N%
8180 L1 = 0: L2 = 0: L3 = 0: L4 = 0: L5 = 1: LL = 1
8190 LLL = 5: CNT2% = 0: M(6) = NODE(6): LL = 2: LLL = 6
8200 FOR Q% = 2 TO 5
8210   M(Q%) = NODE(Q%)
8220 NEXT Q%
8230 '
8240 '     gosub to evaluate union of node probabilities
8250 '     ............................................
8260 '
8270 GOSUB 8500
8280 IF NODE(1) = 0 THEN NODE(1) = LLLL
8290 '
8300 '     gosub to evaluate probability of end item
8310 '      failure without regard to fatal failure
8320 '       (for use in availability subroutine)
8330 '     ......................................
8340 '
8350 GOSUB 8710
8360 '
8370 '     print node 1 probability of failure
8380 '          (includes fatal failure)
8390 '     ................................
8400 '
8401 IF UP% = 1 THEN 8410
8402 LPRINT USING F8$;"node(1) =";NODE(1)
8403 GOTO 8440
8410 LOCATE 12,48: PRINT USING F8$;"node(1) =";NODE(1)
8420 LOCATE 22,40: PRINT "press any key to continue"
8430 A$=INKEY$:  IF A$ = "" THEN 8430
8440 RETURN
8450 '
8460 '     subroutine to evaluate the union of node
8470 '                failure probabilities
8480 '     ....................................
8490 '
8500 FOR Q% = LL TO LLL
8510   L1 = L1 + M(Q%)
8520   FOR V% = Q% TO LLL-1
```

```
8530      L2 = L2 + (M(Q%) * M(V%+1))
8540    NEXT V%
8550    MM(Q%) = M(Q%)
8560    IF M(Q%) = 0! THEN MM(Q%) = 1!: Z%=Z%+1
8570    L5 = L5 * MM(Q%)
8580 NEXT Q%
8590 IF Z% > 0 THEN L5 = 0
8600 Z% = 0
8610 IF CNT2% >= 3 THEN 8690
8620 L3 = M(LL) * M(LL+1) * M(LL+2) + M(LL+3) + M(LL+4))
8630 L3 = L3 + M(LL) * M(LL+2) * (M(LL+3) + M(LL+4))
             +  (M(LL)*M(LL+3)*M(LL+4))
8640 L3 = L3 + M(LL+1) * M(LL+2) * (M(LL+3)+M(LL+4))
8650 L3 = L3 + M(LL+3) * M(LL+4) * (M(LL+1) + M(LL+2))
8660 L4 = M(LL) * M(LL+1) * M(LL+2) * (M(LL+3)+M(LL+4))
8670 L4 = L4 + M(LL+2) * M(LL+3) * M(LL+4) *
                                (M(LL) + M(LL+1))
8680 L4 = L4 + M(LL) * M(LL+1) * M(LL+3) * M(LL+4)
8690 LLLL = L1 - L2 + L3 - L4 + L5
8700 RETURN
8710 '
8720 '      subroutine to subtract probability of fatal
8730 '       failure from node 1 probability of failure
8740 '      ..........................................
8750 '
8760 L1 = (NODE(2)+NODE(3)+NODE(4)+NODE(5)) * NODE(6)
8770 L2 = (((NODE(3)+NODE(4)+NODE(5))*NODE(2))
                 + ((NODE(4)+NODE(5))*NODE(3))
                 + (NODE(4)*NODE(5))) * NODE(6)
8780 L3 = (((NODE(4)+NODE(5))*NODE(2)*NODE(3))
        + ((NODE(3)+NODE(2))*NODE(4)*NODE(5))) * NODE(6)
8790 L4 = NODE(2) * NODE(3) * NODE(4) * NODE(5) * NODE(6)
8800 NOD(C%) = NODE(1) - NODE(6) + L1 - L2 + L3 - L4
8810 RETURN
8820 '
8830 '      SUBROUTINE TO CALCULATE AVAILABILITY
8840 '      ____________________________________
8850 '
8860 CLS:  COLOR 2,0,8:  LOCATE 12,30
8870 PRINT "one moment please"
8880 FOR N% = 1 TO 5
8890    SUMPF(N%) = 0: SUMSTREAM(N%) = 0: SMTTR(N%) = 0
8900    SUMSP2%(N%) = 0: SUMSP%(N%) = 0
8910    AVAILABILITY(N%) = 0
8920 NEXT N%
8930 FOR N% = 1 TO NS%-1
8940    SUMSP2%(KKEY%(N%)) = SUMSP2%(KKEY%(N%)) + SP2%(N%)
8950    SUMSP%(KKEY%(N%)) = SUMSP%(KKEY%(N%)) + SP%(N%)
8960    AVGSTREAM = (SP%(N%)+SP2%(N%)) / 2
8970    GOSUB 13280
8980    SUMPF(KKEY%(N%)) = SUMPF(KKEY%(N%))
                         + (AVGSTREAM * NOD(N%))
8990    SMTTR(KKEY%(N%)) = SMTTR(KKEY%(N%))
                         + (AVGSTREAM*MTTRAVG(N%)*NOD(N%))
9000    SUMSTREAM(KKEY%(N%)) = SUMSTREAM(KKEY%(N%))
                             + AVGSTREAM
9010 NEXT N%
9020 SUM = 0:  AVAIL = 0
9030 FOR N% = 1 TO NO.ENVIR%
9040    IF SUMSTREAM(N%)=0 THEN AVAILABILITY(N%)=0:
                                GOTO 9290
9050    AVGMTTR(N%) = SMTTR(N%) / SUMPF(N%)
9060    AVGPF(N%) = SUMPF(N%) / SUMSTREAM(N%)
9070    P(N%) = (AVGMTTR(N%) / LTP%) * AVGPF(N%)
9080    IF SUMSP%(N%) <= DEMAND%(N%) THEN 9240
9090    IF SUMSP2%(N%) >= DEMAND%(N%) THEN 9120
9100    FRAC = (SUMSP%(N%)-DEMAND%(N%)) /
             (DEMAND%(N%)-SUMSP2%(N%))
9110    SUMSTREAM(N%) = (SUMSP%(N%)+DEMAND%(N%)) / 2
9120    SUMBINOM = 0
9130    VT% = CINT(SUMSTREAM(N%) - DEMAND%(N%))
9140    FOR P% = 0 TO VT%
9150      FAC = 1
9160      IF P% = 0 THEN 9200
```

```
9170      FOR Q% = 1 TO P%
9180        FAC = FAC * ((CINT(SUMSTREAM(N%)) - Q% + 1) /
                        (P% - Q% + 1))*P(N%)
9190      NEXT Q%
9200      SUMBINOM = SUMBINOM + (FAC) *
                  ((1-P(N%))¢(CINT(SUMSTREAM(N%))-P%))
9210    NEXT P%
9220    AVAILABILITY(N%) = SUMBINOM
9230    IF SUMSP2%(N%) >= DEMAND%(N%) THEN 9270
9240    IF SUMSP%(N%) < DEMAND%(N%) THEN
                        AVAILABILITY(N%) = 0: GOTO 9270
9250    IF SUMSP%(N%) = DEMAND%(N%) THEN
              AVAILABILITY(N%) = (1-P(N%))¢(SUMSP%(N%)):
              GOTO 9270
9260    AVAILABILITY(N%) = (AVAILABILITY(N%)*FRAC) /
                          (FRAC+1)
9270    SUM = SUM + SUMSTREAM(N%)
9280    AVAIL = AVAIL + (AVAILABILITY(N%) * SUMSTREAM(N%))
9290 NEXT N%
9291 IF UP% = 1 THEN 9300
9292 LPRINT "        AVAILABILITY DURING TIME PERIOD ";T%
9293 LPRINT:  LPRINT
9294 LPRINT "    environment                    availability"
9295 LPRINT "    -----------                    ------------"
9296 FOR N% = 1 TO NO.ENVIR%
9297   LPRINT USING F1$;NO.ENVIR$(N%);AVAILABILITY(N%)
9298 NEXT N%
9299 GOTO 9410
9300 CLS: COLOR 3,0,8
9310 PRINT "        AVAILABILITY DURING TIME PERIOD ";T%
9320 PRINT: PRINT "Availability is the probability of"
                          "there being enough units"
9330 PRINT "in operation to meet the given demand."
9340 PRINT:  PRINT
9350 PRINT "    environment                    availability"
9360 PRINT "    -----------                    ------------"
9370 FOR N% = 1 TO NO.ENVIR%
9380   COLOR 2,0:  LOCATE 11+N%,1
9390   PRINT USING F1$;NO.ENVIR$(N%);AVAILABILITY(N%)
9400 NEXT N%
9410 IF SUM = 0 THEN TOTALAVAIL(T%) = 0:  GOTO 9430
9420 IF UP% =1 THEN 9430
9421 LPRINT "The total system availability was ";
9422                TOTALAVAIL(T%)
9423 GOTO 9450
9430 COLOR 10,0:  LOCATE 12+N%,1
9440 PRINT "The total system availability was ";
                TOTALAVAIL(T%)
9450 '
9460 '      gosub to save availability for printout
9470 '      ----------------------------------------
9480 '
9490 GOSUB 12930
9500 COLOR 3,0:  LOCATE 23,40
9510 PRINT "Press any key to continue"
9520 A$ = INKEY$:  IF A$ = "" THEN 9520
9530 '
9540 '     gosub to calculate period logistics policy cost
9550 '     ---------------------------------------------------
9560 '
9570 GOSUB 12210
9580 RETURN
9590 '
9600 '      SUBROUTINE TO CALCULATE STREAM SPAREPARTS
9610 '      DEMAND IF ONLY DEMAND FOR SPAREPARTS AND
9620 '          WAREHOUSING IS BEING CALCULATED
9630 '      ----------------------------------------
9640 '
9650 '
9660 '      gosub to calculate demand
9670 '      .........................
9680 '
9690 GOSUB 10640
9691 IF UP% = 0 THEN 9730
9700 COLOR 2,0:  LOCATE 22,50
```

```
9710 PRINT "press any key to continue"
9720 A$=INKEY$:  IF A$ = "" THEN 9720
9730 RETURN
9740 '
9750 '      SUBROUTINE TO PRODUCE PERIOD SUMMARY OF
9760 '       SPAREPARTS AND WAREHOUSE REQUIREMENTS
9770 '      ____________________________________________
9780 '
9781 IF UP% = 1 THEN 9790
9782 LPRINT "PERIOD ";T%;" SPAREPARTS AND WAREHOUSE"
                          "REQUIREMENT SUMMARY"
9784 LPRINT
9785 LPRINT "           Spareparts     Warehouse    Repair"
9786 LPRINT "Node    Stockable Item    Part#"
            "Requirements   Requirements   Level"
9788 LPRINT: LN% = 0
9789 GOTO 9850
9790 CLS:  COLOR 3,0
9800 PRINT "PERIOD ";T%;" SPAREPARTS AND WAREHOUSE"
                          "REQUIREMENT SUMMARY"
9810 PRINT
9820 PRINT "           Spareparts     Warehouse´   Repair"
9830 PRINT "Node    Stockable Item    Part#"
            "Requirements   Requirements   Level"
9840 PRINT: LN% = 0
9850 '
9860 '      gosub to calculate period spareparts and
9870 '                warehouse requirements
9880 '      ......................................
9890 '
9900 GOSUB 11530
9910 '
9920 '      gosub to print total warehouse requirement
9930 '                    by repair level
9940 '      ........................................
9950 '
9960 GOSUB 12030
9961 IF UP% = 0 THEN 10000
9970 COLOR 3,0:  LOCATE 22,50
9980 PRINT "press any key to continue"
9990 A$=INKEY$:  IF A$ = "" THEN 9990
10000 RETURN
10010 '
10020 '      SUBROUTINE TO CALCULATE STREAM SPAREPARTS
10030 '      DEMAND IF EOQ CALCULATIONS ARE CALLED FOR
10040 '      ____________________________________________
10050 '
10060 '
10070 '      gosub to calculate demand
10080 '      .........................
10090 '
10100 GOSUB 10640
10101 IF UP% = 0 THEN 10140
10110 COLOR 3,0:  LOCATE 22,30
10120 PRINT "Press any key to continue"
10130 A$=INKEY$:  IF A$ = "" THEN 10130
10140 '
10150 '      SUBROUTINE TO PRODUCE PERIOD SUMMARY OF
10160 '       SPAREPARTS AND WAREHOUSE REQUIREMENTS
10170 '                    INCLUDING EOQ
10180 '      ____________________________________
10190 '
10200 '
10210 '      gosub to give printer listing of headings
10220 '      .........................................
10230 '
10240 '
10250 GOSUB 13380
10251 IF UP% = 1 THEN 10265
10252 LPRINT "PERIOD ";T%;" SPAREPARTS AND WAREHOUSE"
                          "REQUIREMENT SUMMARY"
10254 LPRINT
10255 LPRINT "                                  Order  Opt."
10256 LPRINT "Stockable Part# Demand  W'house  Point  Proc."
```

```
                                        "EOQ  Total Repair"
10258 LPRINT "Node            Item"
10259 LPRINT "                              (cuft) (days) Level"
                                      "Cost   Level"
10261 LPRINT: LN% = 0
10262 GOTO 10340
10265 CLS:   COLOR 3,0
10270 PRINT "PERIOD ";T%;" SPAREPARTS AND WAREHOUSE"
                          "REQUIREMENT SUMMARY"
10280 PRINT
10290 PRINT "                                      Order  Opt."
10300 PRINT "Stockable Part# Demand  W'house  Point  Proc."
                                        "EOQ  Total Repair"
10310 PRINT "Node            Item"
10320 PRINT "                              (cuft) (days) Level"
                                      "Cost   Level"
10330 PRINT: LN% = 0
10340 '
10350 '     gosub to calculate spareparts and warehouse
10360 '              requirements and EOQ
10370 '     ..........................................
10380 '
10390 GOSUB 11530
10400 '
10410 '     gosub to print total warehouse requirement
10420 '                  by repair level
10430 '     ........................................
10440 '
10450 GOSUB 12030
10451 IF UP% = 0 THEN 10490
10460 COLOR 3,0:  LOCATE 22,30
10470 PRINT "press any key to continue"
10480 A$ = INKEY$:  IF A$ = "" THEN 10480
10490 RETURN
10500 '
10510 '     SUBROUTINE TO MAKE EOQ CALCULATIONS
10520 '     ___________________________________
10530 '
10540 QPRIME = CINT(((2*CP*SPR%(VT%)) / (CH))¢(1/2))
10550 LPRIME = SPR%(VT%) * LT
10560 IF SPR%(VT%) = 0 THEN 10620
10570 OED% = CINT((QPRIME*LX%*LTP%) / SPR%(VT%))
10580 CL = OED% / (LTP%*LX%)
10590 FT3(VT%) = FT3(VT%) * QPRIME
10600 TCPRIME = (PR*SPR%(VT%)) + (2*CP*CH*SPR%(VT%))¢(1/2)
10610 GOTO 10630
10620 OED% = 0:  TCPRIME = 0:  FT3(VT%) = 0
10630 RETURN
10640 '
10650 '     SUBROUTINE TO CALCULATE, FILE, AND PRINT
10660 '     SPAREPARTS AND WAREHOUSE DEMAND BY STREAM
10670 '     _________________________________________
10680 '
10681 IF UP% = 1 THEN 10690
10682 LPRINT "SPAREPARTS AND WAREHOUSE REQUIREMENTS,"
                          "STREAM ";C%;", PERIOD ";T%
10684 LPRINT
10685 LPRINT "                    Stockable       Part#"
                                "Quantity      Repair"
10687 LPRINT "Node                  Item          Level"
10688 LPRINT: LN% = 0
10689 GOTO 10750
10690 CLS:  COLOR 3,0
10700 PRINT "SPAREPARTS AND WAREHOUSE REQUIREMENTS,"
                          "STREAM ";C%;", PERIOD ";T%
10710 PRINT
10720 PRINT "                    Stockable       Part#"
                                "Quantity      Repair"
10730 PRINT "Node                  Item          Level"
10740 PRINT: LN% = 0
10750 FOR V% = 1 TO 126
10760   IF V% = 6 THEN 11510
10770   IF NODE1%(V%) = 0 THEN 11510
10780   FILE4$ = "node" + STR$(V%)
10790   FILE5$ = "sums" + STR$(V%)
```

```
10800   OPEN "i", #1, FILE4$:  VV% = 1
10810   IF EOF(1) THEN 11080
10820   ON SPWA% GOTO 10830,10840,10850
10830   INPUT #1,I$,PN$,QY,SR,PR:  GOTO 10860
10840   INPUT #1,I$,PN$,QY,SR,PR,CP,CH,LT: GOTO 10860
10850   INPUT #1,I$,PN$,QY,SR,PR,CODE%,IAP%,CP,CH,SC
10860   NNAME$(VV%) = I$: PPN$(VV%) = PN$: SSC(VV%) = SC
10870   CCP(VV%) = CP: CCH(VV%) = CH: LLT(VV%) = LT
10880   PPR(VV%) = PR: IIAP%(VV%) = IAP%
10890   SPR%(VV%) = CINT((NODE(V%)+NDE(V%)) * QY *
                  ((SP%(C%)+SP2%(C%))/2))
10900   FT3(VV%) = SR
10910   C4(KKEY%(C%)) = C4(KKEY%(C%)) + (PR*SPR%(VV%))
                  + (2*CP*CH*SPR%(VV%))¢(1/2)
10920   IF LN% < 13 THEN 11040
10921 IF UP% = 0 THEN 10960
10930   COLOR 3,0: LOCATE 22,5
10940   PRINT "press any key to view remainder of"
              "Spareparts and Warehouse Requirements"
10950   A$=INKEY$: IF A$ = "" THEN 10950
10960 '
10970 '     print stream spareparts and warehouse demand
10980 '     ..........................................
10990 '
10991   IF UP% = 1 THEN 11000
10992   LPRINT "                          Stockable        Part#"
                                         "Quantity       Repair"
10994   LPRINT "Node                        Item           Level"
10995   LPRINT
10996   LPRINT USING F2$;NN$(V%);I$;PN$;SPR%(VV%);RL%(V%)
10997   GOTO 11060
11000   CLS: LN%=0: LOCATE 3,1
11010   PRINT "                          Stockable        Part#"
                                         "Quantity       Repair"
11020   PRINT "Node                         Item           Level"
11030   PRINT
11040   COLOR 2,0
11050   PRINT USING F2$;NN$(V%);I$;PN$;SPR%(VV%);RL%(V%)
11060   LN% = LN%+1
11070   VV% = VV% + 1:  GOTO 10810
11080   CLOSE 1
11090   IF CNT1% <> 1 THEN 11240
11100   OPEN "o", #2, FILE5$
11110   FOR Q% = 1 TO VV%-1
11120     I$ = NNAME$(Q%):  PSPR% = SPR%(Q%): PR = PPR(Q%)
11130     SR = FT3(Q%):   PN$ = PPN$(Q%)
11140     CP = CCP(Q%):   CH = CCH(Q%):   LT = LLT(Q%)
11150     SC = SSC(Q%):   IAP% = IIAP%(Q%)
11160     ON SPWA% GOTO 11170,11180,11200
11170     PRINT#2,I$;",";PN$;",";PSPR%;SR:  GOTO 11210
11180     PRINT#2,I$;",";PN$;",";PSPR%;SR;PR;IAP%;CP;CH;LT
11190     GOTO 11210
11200     PRINT#2,I$;",";PART.NO$;",";PSPR%;SR;PR;IAP%;
                CP;CH;SC;CODE%
11210   NEXT Q%
11220   CLOSE 2
11230   GOTO 11510
11240   OPEN "i", #2, FILE5$:  Q% = 1
11250   IF EOF(2) THEN 11360
11260   ON SPWA% GOTO 11270,11280,11300
11270   INPUT#2, I$,PN$,PSPR%,SR:  GOTO 11310
11280   INPUT#2,I$,PN$,PSPR%,SR,PR,IAP%,CP,CH,LT
11290   GOTO 11310
11300   INPUT#2,I$,PN$,PSPR%,SR,PR,IAP%,CP,CH,SC,CODE%
11310   NNAME$(Q%) = I$: PPSPR%(Q%) = PSPR%: LLT(Q%) = LT
11320   FT3(Q%) = SR:  PPR(Q%) = PR: SSC(Q%) = SC
11330   CCP(Q%) = CP:  CCH(Q%) = CH: PPN$(Q%) = PN$
11340   IIAP%(Q%) = IAP%
11350   Q% = Q% + 1:  GOTO 11250
11360   CLOSE 2
11370   OPEN "o", #2, FILE5$
11380   FOR VT% = 1 TO Q%-1
11390     I$ = NNAME$(VT%): PSPR% = PPSPR%(VT%)+SPR%(VT%)
11400     PR = PPR(VT%):  PN$ = PPN$(VT%)
11410     IAP% = IIAP%(VT%):  SR = FT3(VT%)
```

```
11420     CP = CCP(VT%): CH = CCH(VT%)
11430     LT = LLT(VT%): SC = SSC(VT%)
11440     ON SPWA% GOTO 11450,11460,11480
11450     PRINT#2,I$;",";PN$;",";PSPR%;SR: GOTO 11490
11460     PRINT#2,I$;",";PN$;",";PSPR%;SR;PR;IAP%;CP;CH;LT
11470     GOTO 11490
11480     PRINT#2,I$;",";PN$;",";PSPR%;SR;PR;IAP%;CP;
11481              CH;SC;CODE%
11490   NEXT VT%
11500   CLOSE 2
11510 NEXT V%
11520 RETURN
11530 '
11540 '      SUBROUTINE FOR SUMMING AND PRINTING PERIOD
11550 '          SUMMARY OF INVENTORY REQUIREMENTS
11560 '
11570 '
11580 FOR VV% = 1 TO 126
11590   IF VV% = 6 THEN 12010
11600   IF NODE1%(VV%) = 0 THEN 12010
11610   FILE5$ = "sums" + STR$(VV%)
11620   FILE6$ = "ssums" + STR$(VV%)
11630   OPEN "i", #1, FILE5$:  VT% = 1
11640   IF EOF(1) THEN 12000
11650   ON SPWA% GOTO 11660,11670,11690
11660   INPUT #1, I$,PN$,PSPR%,SR:  GOTO 11700
11670   INPUT #1, I$,PN$,PSPR%,SR,PR,IAP%,CP,CH,LT
11680   GOTO 11700
11690   INPUT #1, I$,PN$,PSPR%,SR,PR,IAP%,CP,CH,SC,CODE%
11700   SPR%(VT%) = PSPR%:  FT3(VT%) = SR
11710   NNAME$(VT%) = I$: CCH(VT%) = CH: LLT(VT%) = LT
11720   PPN$(VT%) = PN$: PPR(VT%) = PR
11730   IIAP%(VT%) = IAP%: CCP(VT%) = CP: SSC(VT%) = SC
11740   IF SPWA% = 2 THEN GOSUB 10500
11750   RLW% = RL%(VV%)
11760   IF SPWA%=1 THEN FT3(VT%) = FT3(VT%) * SPR%(VT%)
11770   WH(RLW%) = WH(RLW%) + FT3(VT%)
11780   IF LN% < 13 THEN 11910
11785   IF UP% = 0 THEN 11820
11790   COLOR 3,0:  LOCATE 22,5
11800   PRINT "press any key to continue with"
              "Spareparts and Warehouse Summary"
11810   A$ = INKEY$: IF A$ = "" THEN 11810
11820   IF SPWA% = 2 THEN 11870
11830   CLS:  LN% = 0
11831   IF UP% = 1 THEN 11840
11832   LPRINT "  Spareparts     Warehouse    Repair"
11833   LPRINT "Node              Stockable Item     Part#"
                    "Requirements   Requirements   Level"
11835   GOTO 11860
11840   PRINT "  Spareparts     Warehouse    Repair"
11850   PRINT "Node              Stockable Item     Part#"
                    "Requirements   Requirements   Level"
11860   PRINT:  GOTO 11910
11870   CLS:  LN% = 0
11871   IF UP% = 1 THEN 11880
11872   LPRINT "                                Order  Opt."
11873   LPRINT "                    Stockable  Part#  Demand"
              "W'house  Point  Proc.  EOQ  Total Repair"
11875   LPRINT "Node                Item"
              "(cuft)   (days) Level        Cost   Level"
11877   GOTO 11910
11880   PRINT "                                Order  Opt."
11890   PRINT "                    Stockable  Part#  Demand"
              "W'house  Point  Proc.  EOQ  Total Repair"
11900   PRINT "Node                Item"
              "(cuft)   (days) Level        Cost   Level"
11910   ON SPWA% GOTO 11920,11950,11990
11920   COLOR 2,0
11921   IF UP% = 1 THEN 11930
11922   LPRINT USING F6$;NN$(VV%);I$;PN$;PSPR%;FT3(VT%);RLW%
11923   GOTO 11940
11930   PRINT USING F6$;NN$(VV%);I$;PN$;PSPR%;FT3(VT%);RLW%
11940   LN% = LN% + 1:  GOTO 11990
11950   COLOR 2,0
```

```
11951   IF UP% = 1 THEN 11960
11952   LPRINT USING F3$; NN$(VV%);I$;PN$;PSPR%;FT3(VT%);
                          OED%;LPRIME;QPRIME;TCPRIME;RLW%
11954   GOTO 11970
11960   PRINT USING F3$; NN$(VV%);I$;PN$;PSPR%;FT3(VT%);
                          OED%;LPRIME;QPRIME;TCPRIME;RLW%
11970   LN% = LN% + 1
11980   GOSUB 13500
11990   VT% = VT% + 1:  GOTO 11640
12000   CLOSE 1
12010 NEXT VV%
12020 RETURN
12030 '
12040 '     SUBROUTINE TO PRINT TOTAL PERIOD WAREHOUSE
12050 '            REQUIREMENT BY REPAIR LEVEL
12060 '     ____________________________________________
12070 '
12071 IF UP% = 1 THEN 12080
12072 LPRINT
12073 FOR N% = 1 TO NO.RL%
12074   LPRINT "Total warehouse requirement repair level ";
                          N%;"=";WH(N%)
12076 NEXT N%
12077 GOTO 12130
12080 PRINT
12090 FOR N% = 1 TO NO.RL%
12100   COLOR 6,0
12110   PRINT "Total warehouse requirement repair level ";
                          N%;"=";WH(N%)
12120 NEXT N%
12130 GOSUB 13570
12140 FOR N% = 1 TO NO.RL%:  WH(N%) = 0:  NEXT
12150 RETURN
12160 '
12170 '     SUBROUTINE TO CALCULATE AND PRINT PERIOD
12180 '        BY PERIOD COST OF LOGISTICS POLICY
12190 '     ____________________________________________
12200 '
12210 FOR N% = 1 TO NO.ENVIR%
12220   TC(N%) = 0:  C1(N%) = 0
12230   ACOST(N%) = 0:  C2 = 0:  C3 = 0
12240 NEXT N%
12250 IF R1% < NS%-1 THEN D% = 1
12260 FOR N% = 1 TO NS% - 1
12270   IF SP%(N%) = 0 THEN 12350
12280   TT% = T% - TTIME%(N%)
12290   SVC = (SP%(N%)-SP2%(N%)) * SV
12300   IF TT% = RRA(N%) THEN SVC =
                  SVC + SP2%(N%) * BV(KKEY%(N%))
12310   IF SVC < 0 THEN SVC = 0
12320   IF N% > R1% THEN 12340
12330   C1(KKEY%(N%)) = C1(KKEY%(N%)) - SVC:  GOTO 12350
12340   C1(KKEY%(N%)) = C1(KKEY%(N%)) + (IP%(N%)*CC) - SVC
12350 NEXT N%
12351 IF UP% = 1 THEN 12360
12352 LPRINT "PERIOD COST SUMMARY"
12353 LPRINT "Environment     Capital Cost  Repair Cost"
                "Short. Cost  Inven. Cost  Total Cost"
12355 LPRINT "Environment     # Units Start     #Units End"
                          "Operating Cost Per Unit"
12357 GOTO 12400
12360 CLS:  COLOR 3,0:  PRINT "PERIOD COST SUMMARY"
12370 PRINT "Environment     Capital Cost  Repair Cost"
                "Short. Cost  Inven. Cost  Total Cost"
12380 LOCATE 11,1
12390 PRINT "Environment     # Units Start     #Units End"
                          "Operating Cost Per Unit"
12400 FOR N% = 1 TO NO.ENVIR%
12410   C2=0:  C3=0
12420   IF SUMSP%(N%) = 0 THEN 12490
12430   C2 = SMTTR(N%) * RC
12440   C3 = (1-AVAILABILITY(N%)) * SC * LTP%
12450   TC(N%) = C1(N%) + C2 + C3 + C4(N%)
12460   ACOST(N%) = (TC(N%)-C1(N%)) / SUMSP%(N%)
```

```
12461   IF UP% = 1 THEN 12470
12462   LPRINT USING F7$;NO.ENVIR$(N%);C1(N%);C2;
                        C3;C4(N%);TC(N%)
12464   LPRINT USING F30$;NO.ENVIR$(N%);SUMSP%(N%);
                         SUMSP2%(N%);ACOST(N%)
12466   GOTO 12510
12470   COLOR 2,0:  LOCATE 4+N%,1
12480   PRINT USING F7$;NO.ENVIR$(N%);C1(N%);C2;
                        C3;C4(N%);TC(N%)
12490   LOCATE 12+N%,1
12500   PRINT USING F30$;NO.ENVIR$(N%);SUMSP%(N%);
                         SUMSP2%(N%);ACOST(N%)
12510   GOSUB 13040
12520   PB = 1 / ((1+I)¢T%)
12530   C5(N%) = C5(N%) + C1(N%) * PB
12540   C6(N%) = C6(N%) + C2 * PB
12550   C7(N%) = C7(N%) + C3 * PB
12560   C8(N%) = C8(N%) + C4(N%) * PB
12570   C9(N%) = C9(N%) + TC(N%) * PB
12580 NEXT N%
12581 IF UP% = 1 THEN 12590
12582 LPRINT
12583 LPRINT "Procurement costs for the period =";D%*PC
12584 GOTO 12610
12590 PRINT
12600 PRINT "Procurement costs for the period =";D%*PC
12610 PRC = PRC + ((D%*PC)*PB):  PCOST(T%) = D% * PC
12611 IF UP% = 0 THEN 12640
12620 LOCATE 20,30: PRINT "press any key to continue"
12630 A$=INKEY$: IF A$="" THEN 12630
12640 R1% =NS% - 1:  D% = 0
12650 RETURN
12660 '
12670 '     SUBROUTINE TO CALCULATE AND PRINT PERIOD
12680 '          EQUIVALENT COST FOR MODEL RUN
12690 '
12700 '     ____________________________________
12710 SSC1 = 0: SSC2 = 0: SSC3 = 0: SSC4 = 0: SSTC = 0
12711 IF UP% = 1 THEN 12720
12712 LPRINT "TOTAL  COST  SUMMARY"
12713 LPRINT "Environment     Capital Cost  Repair Cost"
                "Short. Cost  Inven. Cost  Total Cost"
12715 GOTO 12740
12720 CLS:  COLOR 3,0:  PRINT "TOTAL  COST  SUMMARY"
12730 PRINT "Environment     Capital Cost  Repair Cost"
                "Short. Cost  Inven. Cost  Total Cost"
12740 PB = (I*((1+I)¢T%)) / (((1+I)¢T%)-1)
12750 FOR N% = 1 TO NO.ENVIR%
12760   SC1 = C5(N%) * PB:   SC2 = C6(N%) * PB
12770   SC3 = C7(N%) * PB:   SC4 = C8(N%) * PB
12780   STC = C9(N%) * PB:   SSC1 = SSC1 + SC1
12790   SSC2 = SSC2 + SC2:   SSC3 = SSC3 + SC3
12800   SSC4 = SSC4 + SC4:   SSTC = SSTC + STC
12801   IF UP% = 1 THEN 12810
12802   LPRINT USING F7$;NO.ENVIR$(N%);SC1;SC2;SC3;SC4;STC
12803   GOTO 12820
12810   PRINT USING F7$;NO.ENVIR$(N%);SC1;SC2;SC3;SC4;STC
12820 NEXT N%
12821 IF UP% = 1 THEN 12830
12822 LPRINT "Procurement costs per period =";PRC*PB
12823 GOTO 12840
12830 PRINT "Procurement costs per period =";PRC*PB
12840 SSTC = SSTC + (PRC*PB)
12841 IF UP% = 1 THEN 12850
12842 LPRINT
12843 LPRINT USING F7$;"TOTALS";SSC1;SSC2;SSC3;SSC4;SSTC
12844 GOTO 12870
12850 PRINT: COLOR 10,0
12860 PRINT USING F7$;"TOTALS";SSC1;SSC2;SSC3;SSC4;SSTC
12870 RETURN
12880 '
12890 '     SUBROUTINE TO SAVE PERIOD BY PERIOD
12900 '      AVAILABILITY FOR PRINTER LISTING
12910 '     ____________________________________
12920 '
```

```
12930 FOR N% = 1 TO NO.ENVIR%
12940   GA1(T%,N%) = AVAILABILITY(N%)
12950 NEXT N%
12960 RETURN
12970 '
12980 '          SUBROUTINE TO SAVE PERIOD BY PERIOD
12990 '     LOGISTICS POLICY COST DATA FOR PRINTER LISTING
13000 '     (presently set up to handle 12 periods and 2
13010 '                 failure environments)
13020 '
13030 '     ____________________________________________
13040 IF N% = 2 THEN 13100
13050 GCOST(T%,1) = C1(N%): GCOST(T%,2) = C2
13060 GCOST(T%,3) = C3:     GCOST(T%,4) = C4(N%)
13070 GCOST(T%,5) = TC(N%): GCOST(T%,6) = SUMSP%(N%)
13080 GCOST(T%,7) = SUMSP2%(N%)
13090 GCOST(T%,8) = ACOST(N%):  GOTO 13150
13100 GCOST(T%+12,1) = C1(N%): GCOST(T%+12,2) = C2
13110 GCOST(T%+12,3) = C3:     GCOST(T%+12,4) = C4(N%)
13120 GCOST(T%+12,5) = TC(N%): GCOST(T%+12,6) = SUMSP%(N%)
13130 GCOST(T%+12,7) = SUMSP2%(N%)
13140 GCOST(T%+12,8) = ACOST(N%)
13150 RETURN
13160 '
13170 '     SUBROUTINE TO SUM PER END ITEM DEMAND ON
13180 '                 REPAIR FACILITIES
13190 '     ____________________________________
13200 '
13210 MP(RL%(A%)) = MP(RL%(A%)) + (MTTRP(N%) * NODE(A%))
13220 RETURN
13230 '
13240 '     SUBROUTINE TO CALCULATE PERIOD DEMAND ON
13250 '                 REPAIR FACILITIES
13260 '     ____________________________________
13270 '
13280 FOR VT% = 1 TO NO.RL%
13290   GMP(T%,VT%) = GMP(T%,VT%) + (MP(VT%) * AVGSTREAM)
13300 NEXT VT%
13310 RETURN
13320 '
13330 '     SUBROUTINE TO GIVE PRINTER LISTING OF PERIOD
13340 '     SPAREPARTS AND WAREHOUSE REQUIREMENT HEADING
13350 '             IF EOQ MODE IS IN EFFECT
13360 '     ____________________________________________
13370 '
13380 LPRINT "                PERIOD ";T%;
        " SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY"
13390 LPRINT
13400 LPRINT "                Node            Part#   Demand"
          "W'house  EOQ   Total Repair"
13410 LPRINT "    (cuft)          Cost   Level"
13420 LPRINT
13430 RETURN
13440 '
13450 '     SUBROUTINE TO GIVE PRINTER LISTING OF PERIOD
13460 '     SPAREPARTS AND WAREHOUSE REQUIREMENT SUMMARY
13470 '             IF EOQ MODE IS IN EFFECT
13480 '     ____________________________________________
13490 '
13500 LPRINT USING F56$;NN$(VV%);PN$;PSPR%;FT3(VT%);
                        QPRIME;TCPRIME;RLW%
13510 RETURN
13520 '
13530 '     SUBROUTINE FOR PRINTER LISTING OF WAREHOUSE
13540 '             DEMAND TOTALS FOR PERIOD
13550 '     ____________________________________________
13560 '
13570 LPRINT
13580 FOR N% = 1 TO NO.RL%
13590   LPRINT USING F57$;
          "Total warehouse requirement repair level ";
          N%;"=";WH(N%)
13600 NEXT N%
13610 FOR N% = 1 TO 12: LPRINT: NEXT
13620 RETURN
```

```
1 '       "INTRO" PROGRAM LISTING
2 '
3 '       This section of the program familiarizes the
4 '       user with the operation and concepts of the model
5 '
6 '
7 '       CALL SUBROUTINES THAT DISPLAY "INTRO" SCREENS
8 '       ____________________________________________
9 '
10 REQ%=1:GOSUB 11000
20 REQ%=4:GOSUB 11000
30 IF SCR$ = "R" THEN 10
40 REQ%=1:GOSUB 5000
50 REQ%=4:GOSUB 5000
60 IF SCR$ = "R" THEN 10
70 REQ%=1:GOSUB 6000
80 REQ%=4:GOSUB 6000
90 IF SCR$ = "R" THEN 40
100 REQ%=1:GOSUB 7000
110 REQ%=4:GOSUB 7000
120 IF SCR$ = "R" THEN 70
130 REQ%=1:GOSUB 8000
140 REQ%=4:GOSUB 8000
150 IF SCR$ = "R" THEN 100
160 REQ%=1:GOSUB 2000
170 REQ%=4:GOSUB 2000
180 IF SCR$ = "R" THEN 130
195 '
196 '      CHAIN TO FIRST INPUT PROGRAM, "INPUT1"
197 '      ______________________________________
198 '
200 CHAIN "INPUT1",26,ALL
220 '
225 '      END OF MAIN PROGRAM (INTRO)
230 '
240 '
250 '
680 '
682 '
683 '                SCREEN DRIVER SUBROUTINE
685 '      ALL OTHER SCREEN SUBROUTINES REFERENCE THIS
686 '      SUBROUTINE DURING INTERACTIVE SESSIONS WITH
687 '                THE USER OF THE MODEL
688 '      ____________________________________________
690 '
700 'save "SGSUB.bas",a
701 REM  PARAMETER  REQ/OPT    PURPOSE
702 REM  SG.ROW%        REQ   Row    (Not Altered)
703 REM  SG.COL%        REQ   Column (Not Altered)
704 REM  SG.LNG%        REQ   Length (Not Altered)
705 REM  SG.UP%=1       OPT   Uppercases input (default=0)
706 REM  SG.DF$         OPT   String for default input
710 REM  ---------------Entry------------------
711 REM     Numeric only   gosub 770
712 REM     Alphameric     gosub 775
713 REM ------------Output parms--------------
714 REM      SG.REPLY$  - alphameric input
715 REM  VAL(SG.REPLY$) - numeric input
720 REM ------------------------------------
770 SG.HI%=57:SG.LO%=46:SG.MASK$=CHR$(219):GOTO 780
775 SG.HI%=126:SG.LO%=32:SG.MASK$=CHR$(178)
780 LOCATE SG.ROW%,SG.COL%,1:SG.CO%=SG.COL%
785 SG.INS%=0:SG.BTAB%=0
800 PRINT STRING$(SG.LNG%,SG.MASK$);
820 IF SG.DF$<>"" THEN LOCATE SG.ROW%,SG.COL%,1:
                       PRINT LEFT$(SG.DF$,SG.LNG%);
840 LOCATE SG.ROW%,SG.COL%,1
860 SG.REPLY$=INKEY$ SG.REPLY$="" THEN GOTO 860
880 IF LEN(SG.REPLY$) = 2 THEN GOTO 1080
900 SG.CHAR%=ASC(SG.REPLY$)
901 IF SG.CHAR%=13 THEN SOUND 40,2:GOTO 1260
920 IF SG.CO%=SG.COL%+SG.LNG% THEN BEEP:GOTO 860
930 IF SG.CHAR%=45 THEN GOTO 981
940 IF SG.CHAR%=47 AND SG.LO%=46 THEN SG.CHAR%=127
960 IF SG.CHAR%>SG.HI% OR SG.CHAR%<SG.LO% THEN BEEP:
```

```
                        GOTO 860
979 IF SG.CHAR%>96 AND SG.CHAR%<123 AND SG.UP%=1 THEN
                    SG.REPLY$=CHR$(ASC(SG.REPLY$)-32)
981 SG.Y$="" NOT SG.INS% THEN GOTO 1000
982 FOR SG.I%=SG.CO% TO SG.COL%+SG.LNG%-1
983 SG.CHAR%=SCREEN(SG.ROW%,SG.I%)
985 SG.Y$=SG.Y$+CHR$(SG.CHAR%):NEXT SG.I%
986 SG.Y$=LEFT$(SG.REPLY$+SG.Y$,SG.COL%+SG.LNG%-SG.CO%)
987 PRINT SG.Y$;
988 LOCATE SG.ROW%,SG.CO%+1,1:GOTO 1020
1000 PRINT SG.REPLY$;
1020 SG.CO%=SG.CO%+1
1040 IF SG.CO%=SG.COL%+SG.LNG% THEN LOCATE ,,0
1060 GOTO 860
1080 SG.CHAR%=ASC(RIGHT$(SG.REPLY$,1))
1082 IF SG.CHAR%=79 THEN SG.CO%=SG.COL%+SG.LNG%-1:
                         LOCATE ,SG.CO%,1:GOTO 860
1084 IF SG.CHAR%<>117 THEN GOTO 1100
1086 IF SG.CO%=SG.COL%+SG.LNG% THEN BEEP:GOTO 860
1088 FOR SG.I%=SG.CO% TO SG.COL%+SG.LNG%-1:PRINT SG.MASK$:
                      NEXT SG.I%:LOCATE ,SG.CO%,1:GOTO 860
1100 IF SG.CHAR%<> 77 THEN GOTO 1160
1120 IF SG.CO%+1>SG.COL%+SG.LNG%-1 THEN BEEP:GOTO 860
1140 SG.CO%=SG.CO%+1:PRINT CHR$(28);:SG.INS%=0:GOTO 860
1160 IF SG.CHAR%<>75 THEN GOTO 1220
1180 IF SG.CO%-1<SG.COL% THEN BEEP:GOTO 860
1200 SG.CO%=SG.CO%-1:PRINT CHR$(29);
1201 SG.INS%=0:LOCATE ,,1:GOTO 860
1220 IF SG.CHAR%=71 THEN GOTO 780
1221 IF SG.CHAR%=82 THEN SG.INS% = NOT SG.INS%:GOTO 860
1222 IF SG.CHAR%=73 THEN SG.BTAB%=1:
                         SG.REPLY$=CHR$(13):GOTO 900
1224 IF SG.CHAR%=81 THEN SG.BTAB%=2:
                         SG.REPLY$=CHR$(13):GOTO 900
1226 IF SG.CHAR%=68 THEN SG.BTAB%=3:
                         SG.REPLY$=CHR$(13):GOTO 900
1228 IF SG.CHAR%=59 THEN SG.BTAB%=4:
                         SG.REPLY$=CHR$(13):GOTO 900
1230 IF SG.CHAR%=67 THEN SG.BTAB%=5:
                         SG.REPLY$=CHR$(13):GOTO 900
1232 IF SG.CHAR%=66 THEN SG.BTAB%=6:
                         SG.REPLY$=CHR$(13):GOTO 900
1240 IF SG.CHAR%<>83 THEN BEEP:GOTO 860
1241 SG.Y$=""
1242 IF SG.CO%=SG.COL%+SG.LNG% THEN BEEP:GOTO 860
1243 FOR SG.I%=SG.CO% TO SG.COL%+SG.LNG%-1
1244 SG.CHAR%=SCREEN(SG.ROW%,SG.I%)
1245 SG.Y$=SG.Y$+CHR$(SG.CHAR%):NEXT SG.I%
1246 SG.Y$=SG.Y$+SG.MASK$:PRINT RIGHT$(SG.Y$,LEN(SG.Y$)-1)
1247 LOCATE SG.ROW%,SG.CO%,1:GOTO 860
1260 SG.REPLY$="":SG.UP%=0
1280 FOR SG.I%=0 TO SG.LNG%-1
1281 SG.CHAR%=SCREEN(SG.ROW%,SG.COL%+SG.I%)
1300 IF SG.CHAR%=46 AND SG.UP%=1 AND SG.LO%=46 THEN
                                          GOTO 1380
1320 IF SG.CHAR%=46 THEN SG.UP%=1
1340 IF SG.CHAR%=ASC(SG.MASK$) THEN GOTO 1400
1360 SG.REPLY$=SG.REPLY$+CHR$(SG.CHAR%)
1380 NEXT SG.I%
1400 IF SG.HI%=126 THEN LOCATE SG.ROW%,SG.COL%,0:
      RINT LEFT$(SG.REPLY$+SPACE$(SG.LNG%),SG.LNG%);:
                                          GOTO 1440
1420 LOCATE SG.ROW%,SG.COL%,0
1421 PRINT RIGHT$(SPACE$(SG.LNG%)+SG.REPLY$,SG.LNG%);
1440 COLOR 7,0
1450 SG.DF$="":SG.UP%=0:RETURN
1460 REM ---------End of Data Input Subroutine--------
1461 IF NOT SG.DEFU% THEN DEF FNSG$(FLD#)
                =RIGHT$(STR$(FLD#),LEN(STR$(FLD#))-1)
1463 KEY OFF
1464 SG.DEFU%=-1:SG.BTAB%=0
1466 LOCATE 25,1:COLOR 15,0
1468 IF MSG$="" THEN PRINT TAB(79); ELSE PRINT MSG$;
          TAB(79);:SOUND 340,2:SOUND 32767,1:SOUND 340,2
1470 MSG$="":COLOR 7,0:RETURN
```

```
1471 COLOR 0,0:LOCATE 25,60:PRINT SPACE$(19);:LOCATE 25,60
1472 IF LEN(SG.MSK$)>19 THEN SG.MSK$=LEFT$(SG.MSK$,19)
1473 PRINT USING SG.MSK$;SG.VV#;
1474 SG.DF$="":SG.CC%=59
1476 SG.CC%=SG.CC%+1 SG.CC%>78 THEN GOTO 1486
1478 SG.VR%=SCREEN(25,SG.CC%)
1480 IF SG.VR%=32 AND SG.DF$="" THEN GOTO 1476
1482 IF SG.VR%=32 THEN GOTO 1486
1484 SG.DF$=SG.DF$+CHR$(SG.VR%):GOTO 1476
1486 COLOR 7,0:RETURN
1990 '
1991 '       INTRO SCREEN SUBROUTINE
1992 '       ________________________
1993 '
2000 ON REQ% GOSUB 2052,2046,2042,2024,2034,2030,2038
2001 RETURN
2002 ' ENTRY------------ENTRY TYPE ------------SUBROUTINE-
2003 '  1    DISPLAY SCREEN ONLY (ERASE DATA)   2052
2004 '  2    DISPLAY SCREEN & DATA ELEMENTS     2046
2006 '  3    DISPLAY DATA ELEMENTS ONLY         2042
2008 '  4    INPUT ALL DATA ELEMENTS            2024
2010 '  5    INPUT ONE ELEMENT BY RELATIVE FLD  2034
2012 '  6    UPDATE ALL DATA ELEMENTS           2030
2014 '  7    UPDATE ONE DATA ELEMENT BY REL FLD 2038
2016 ' ---------------------------------------------------
2018 SG.DF$=SCR$
2020 SG.UP%=1:SG.ROW%=24:SG.COL%=74:SG.LNG%=1
2021 COLOR 15,1  :GOSUB 775
2022 SCR$=SG.REPLY$:RETURN
2024 GOSUB 1462:GOSUB 2053
2026 IF SG.BTAB%=0 THEN GOSUB 2020  ELSE GOSUB 2018  'SCR$
2028 RETURN
2030 GOSUB 1462
2031 GOSUB 2018    'SCR$
2032 IF SG.BTAB%>2 THEN RETURN
2033 RETURN
2034 GOSUB 1462
2035 IF FLD%<31 THEN ON FLD% GOSUB 2020
2037 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
2038 GOSUB 1462
2039 IF FLD%<31 THEN ON FLD% GOSUB 2018
2041 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
2042 GOSUB 1462
2043 LOCATE 24,74:COLOR 15,1:PRINT USING "!";SCR$;
2045 RETURN
2046 GOSUB 2052
2048 GOSUB 2043
2050 RETURN
2051 LOCATE 25,1:COLOR , 0:PRINT" ";
2052 COLOR ,, 8:CLS:GOSUB 1462
2053 COLOR  2,0
2054 LOCATE 1,1:PRINT "                                        ";
2056 COLOR 24,0:LOCATE 2,40:PRINT "|";
2058 COLOR 8,0:LOCATE 3,10:PRINT " .";
2060 COLOR 8,0:LOCATE 3,31:PRINT ".";
2062 COLOR 24,0:LOCATE 3,51:PRINT ".";
2064 COLOR 8,0:LOCATE 3,71:PRINT ".";
2066 COLOR 8,0:LOCATE 3,80:PRINT ".";
2068 COLOR 3,0:LOCATE 4,11:PRINT " ";
2070 COLOR 4,0:LOCATE 4,31:PRINT " ";
2072 COLOR 7,0:LOCATE 4,51:PRINT " ";
2074 COLOR 14,0:LOCATE 4,71:PRINT " ";
2076 COLOR 5,0:LOCATE 4,80:PRINT " ";
2078 COLOR 8,0:LOCATE 5,11:PRINT "|";
2080 COLOR 8,0:LOCATE 5,31:PRINT "|";
2082 COLOR 24,0:LOCATE 5,51:PRINT "|";
2084 COLOR 8,0:LOCATE 5,71:PRINT "|";
2086 COLOR 8,0:LOCATE 6,3:PRINT ".";
2088 COLOR 8,0:LOCATE 6,7:PRINT ".";
2090 COLOR 8,0:LOCATE 6,11:PRINT ".";
2092 COLOR 8,0:LOCATE 6,15:PRINT ".";
2094 COLOR 8,0:LOCATE 6,19:PRINT ".";
2096 COLOR 8,0:LOCATE 6,23:PRINT ".";
2098 COLOR 8,0:LOCATE 6,27:PRINT ".";
2100 COLOR 8,0:LOCATE 6,31:PRINT ".";
```

```
2102 COLOR 8,0:LOCATE 6,35:PRINT ".";
2104 COLOR 8,0:LOCATE 6,39:PRINT ".";
2106 COLOR 8,0:LOCATE 6,43:PRINT ".";
2108 COLOR 24,0:LOCATE 6,47:PRINT ".";
2110 COLOR 8,0:LOCATE 6,51:PRINT ".";
2112 COLOR 8,0:LOCATE 6,55:PRINT ".";
2114 COLOR 8,0:LOCATE 6,59:PRINT ".";
2116 COLOR 8,0:LOCATE 6,62:PRINT " .";
2118 COLOR 8,0:LOCATE 6,67:PRINT ".";
2120 COLOR 8,0:LOCATE 6,71:PRINT ".";
2122 COLOR 8,0:LOCATE 6,75:PRINT ".";
2124 COLOR 8,0:LOCATE 6,79:PRINT ".";
2126 COLOR 3,0:LOCATE 7,3:PRINT " ";
2128 COLOR 3,0:LOCATE 7,7:PRINT " ";
2130 COLOR 3,0:LOCATE 7,11:PRINT " ";
2132 COLOR 3,0:LOCATE 7,15:PRINT " ";
2134 COLOR 3,0:LOCATE 7,19:PRINT " ";
2136 COLOR 4,0:LOCATE 7,23:PRINT " ";
2138 COLOR 4,0:LOCATE 7,27:PRINT " ";
2140 COLOR 4,0:LOCATE 7,31:PRINT " ";
2142 COLOR 4,0:LOCATE 7,35:PRINT " ";
2144 COLOR 4,0:LOCATE 7,39:PRINT " ";
2146 COLOR 7,0:LOCATE 7,43:PRINT " ";
2148 COLOR 7,0:LOCATE 7,47:PRINT " ";
2150 COLOR 7,0:LOCATE 7,51:PRINT " ";
2152 COLOR 7,0:LOCATE 7,55:PRINT " ";
2154 COLOR 7,0:LOCATE 7,59:PRINT " ";
2156 COLOR 14,0:LOCATE 7,63:PRINT " ";
2158 COLOR 14,0:LOCATE 7,67:PRINT " ";
2160 COLOR 14,0:LOCATE 7,71:PRINT " ";
2162 COLOR 14,0:LOCATE 7,75:PRINT " ";
2164 COLOR 14,0:LOCATE 7,79:PRINT " ";
2166 COLOR 8,0:LOCATE 8,3:PRINT "|";
2168 COLOR 8,0:LOCATE 8,7:PRINT "|";
2170 COLOR 8,0:LOCATE 8,11:PRINT "|";
2172 COLOR 8,0:LOCATE 8,15:PRINT "|";
2174 COLOR 8,0:LOCATE 8,19:PRINT "|";
2176 COLOR 8,0:LOCATE 8,23:PRINT "|";
2178 COLOR 8,0:LOCATE 8,27:PRINT "|";
2180 COLOR 8,0:LOCATE 8,31:PRINT "|";
2182 COLOR 8,0:LOCATE 8,34:PRINT " |";
2184 COLOR 8,0:LOCATE 8,39:PRINT "|";
2186 COLOR 8,0:LOCATE 8,43:PRINT "|";
2188 COLOR 24,0:LOCATE 8,47:PRINT "|";
2190 COLOR 8,0:LOCATE 8,51:PRINT "|";
2192 COLOR 8,0:LOCATE 8,55:PRINT "|";
2194 COLOR 8,0:LOCATE 8,59:PRINT "|";
2196 COLOR 8,0:LOCATE 8,63:PRINT "|";
2198 COLOR 8,0:LOCATE 8,67:PRINT "|";
2200 COLOR 8,0:LOCATE 8,71:PRINT "|";
2202 COLOR 8,0:LOCATE 8,75:PRINT "|";
2204 COLOR 8,0:LOCATE 8,79:PRINT "|";
2206 COLOR 8,0:LOCATE 9,3:PRINT "|";
2208 COLOR 8,0:LOCATE 9,7:PRINT "|";
2210 COLOR 8,0:LOCATE 9,11:PRINT "|";
2212 COLOR 8,0:LOCATE 9,15:PRINT "|";
2214 COLOR 8,0:LOCATE 9,19:PRINT "|";
2216 COLOR 8,0:LOCATE 9,23:PRINT "|";
2218 COLOR 8,0:LOCATE 9,27:PRINT "|";
2220 COLOR 8,0:LOCATE 9,31:PRINT "|";
2222 COLOR 8,0:LOCATE 9,35:PRINT "|";
2224 COLOR 8,0:LOCATE 9,39:PRINT "|";
2226 COLOR 8,0:LOCATE 9,43:PRINT "|";
2228 COLOR 24,0:LOCATE 9,47:PRINT "|";
2230 COLOR 8,0:LOCATE 9,51:PRINT "|";
2232 COLOR 8,0:LOCATE 9,55:PRINT "|";
2234 COLOR 8,0:LOCATE 9,59:PRINT "|";
2236 COLOR 8,0:LOCATE 9,63:PRINT "|";
2238 COLOR 8,0:LOCATE 9,67:PRINT "|";
2240 COLOR 8,0:LOCATE 9,71:PRINT "|";
2242 COLOR 8,0:LOCATE 9,75:PRINT "|";
2244 COLOR 8,0:LOCATE 9,79:PRINT "|";
2246 COLOR 8,0:LOCATE 10,1:PRINT ".....";
2248 COLOR 8,0:LOCATE 10,7:PRINT "|";
2250 COLOR 8,0:LOCATE 10,11:PRINT "|";
```

```
2252 COLOR 8,0:LOCATE 10,15:PRINT "|";
2254 COLOR 8,0:LOCATE 10,19:PRINT "|";
2256 COLOR 8,0:LOCATE 10,21:PRINT ".....";
2258 COLOR 8,0:LOCATE 10,27:PRINT "|";
2260 COLOR 8,0:LOCATE 10,31:PRINT "|";
2262 COLOR 8,0:LOCATE 10,35:PRINT "|";
2264 COLOR 8,0:LOCATE 10,39:PRINT "|";
2266 COLOR 8,0:LOCATE 10,41:PRINT ".....";
2268 COLOR 24,0:LOCATE 10,47:PRINT "|";
2270 COLOR 8,0:LOCATE 10,51:PRINT "|";
2272 COLOR 8,0:LOCATE 10,55:PRINT "|";
2274 COLOR 8,0:LOCATE 10,59:PRINT "|";
2276 COLOR 8,0:LOCATE 10,61:PRINT ".....";
2278 COLOR 8,0:LOCATE 10,67:PRINT "|";
2280 COLOR 8,0:LOCATE 10,71:PRINT "|";
2282 COLOR 8,0:LOCATE 10,75:PRINT "|";
2284 COLOR 8,0:LOCATE 10,79:PRINT "|";
2286 COLOR 3,0:LOCATE 11,1:PRINT "     ";
2288 COLOR 8,0:LOCATE 11,7:PRINT "|";
2290 COLOR 8,0:LOCATE 11,11:PRINT "|";
2292 COLOR 8,0:LOCATE 11,15:PRINT "|";
2294 COLOR 8,0:LOCATE 11,19:PRINT "|";
2296 COLOR 4,0:LOCATE 11,20:PRINT "       ";
2298 COLOR 8,0:LOCATE 11,27:PRINT "|";
2300 COLOR 8,0:LOCATE 11,31:PRINT "|";
2302 COLOR 8,0:LOCATE 11,35:PRINT "|";
2304 COLOR 8,0:LOCATE 11,39:PRINT "|";
2306 COLOR 7,0:LOCATE 11,41:PRINT "     ";
2308 COLOR 24,0:LOCATE 11,47:PRINT "|";
2310 COLOR 8,0:LOCATE 11,51:PRINT "|";
2312 COLOR 8,0:LOCATE 11,55:PRINT "|";
2314 COLOR 8,0:LOCATE 11,59:PRINT "|";
2316 COLOR 14,0:LOCATE 11,61:PRINT "     ";
2318 COLOR 8,0:LOCATE 11,67:PRINT "|";
2320 COLOR 8,0:LOCATE 11,71:PRINT "|";
2322 COLOR 8,0:LOCATE 11,75:PRINT "|";
2324 COLOR 8,0:LOCATE 11,79:PRINT "|";
2326 COLOR 8,0:LOCATE 12,5:PRINT ".....";
2328 COLOR 8,0:LOCATE 12,11:PRINT "|";
2330 COLOR 8,0:LOCATE 12,15:PRINT "|";
2332 COLOR 8,0:LOCATE 12,19:PRINT "|";
2334 COLOR 8,0:LOCATE 12,25:PRINT ".....";
2336 COLOR 8,0:LOCATE 12,31:PRINT "|";
2338 COLOR 8,0:LOCATE 12,35:PRINT "|";
2340 COLOR 8,0:LOCATE 12,39:PRINT "|";
2342 COLOR 24,0:LOCATE 12,45:PRINT ".....";
2344 COLOR 8,0:LOCATE 12,51:PRINT "|";
2346 COLOR 8,0:LOCATE 12,55:PRINT "|";
2348 COLOR 8,0:LOCATE 12,59:PRINT "|";
2350 COLOR 8,0:LOCATE 12,65:PRINT ".....";
2352 COLOR 8,0:LOCATE 12,71:PRINT "|";
2354 COLOR 8,0:LOCATE 12,75:PRINT "|";
2356 COLOR 8,0:LOCATE 12,79:PRINT "|";
2358 COLOR 3,0:LOCATE 13,5:PRINT "     ";
2360 COLOR 8,0:LOCATE 13,11:PRINT "|";
2362 COLOR 8,0:LOCATE 13,15:PRINT "|";
2364 COLOR 8,0:LOCATE 13,19:PRINT "|";
2366 COLOR 4,0:LOCATE 13,25:PRINT "     ";
2368 COLOR 8,0:LOCATE 13,31:PRINT "|";
2370 COLOR 8,0:LOCATE 13,35:PRINT "|";
2372 COLOR 8,0:LOCATE 13,39:PRINT "|";
2374 COLOR 7,0:LOCATE 13,45:PRINT "     ";
2376 COLOR 8,0:LOCATE 13,51:PRINT "|";
2378 COLOR 8,0:LOCATE 13,55:PRINT "|";
2380 COLOR 8,0:LOCATE 13,59:PRINT "|";
2382 COLOR 14,0:LOCATE 13,65:PRINT "     ";
2384 COLOR 8,0:LOCATE 13,71:PRINT "|";
2386 COLOR 8,0:LOCATE 13,75:PRINT "|";
2388 COLOR 8,0:LOCATE 13,79:PRINT "|";
2390 COLOR 8,0:LOCATE 14,9:PRINT ".....";
2392 COLOR 8,0:LOCATE 14,15:PRINT "|";
2394 COLOR 8,0:LOCATE 14,19:PRINT "|";
2396 COLOR 8,0:LOCATE 14,29:PRINT ".....";
2398 COLOR 8,0:LOCATE 14,35:PRINT "|";
2400 COLOR 8,0:LOCATE 14,39:PRINT "|";
```

```
2402 COLOR 8,0:LOCATE 14,49:PRINT ".....";
2404 COLOR 8,0:LOCATE 14,55:PRINT "|";
2406 COLOR 8,0:LOCATE 14,59:PRINT "|";
2408 COLOR 8,0:LOCATE 14,69:PRINT ".....";
2410 COLOR 8,0:LOCATE 14,75:PRINT "|";
2412 COLOR 8,0:LOCATE 14,79:PRINT "|";
2414 COLOR 3,0:LOCATE 15,9:PRINT "     ";
2416 COLOR 8,0:LOCATE 15,15:PRINT "|";
2418 COLOR 8,0:LOCATE 15,19:PRINT "|";
2420 COLOR 4,0:LOCATE 15,29:PRINT "     ";
2422 COLOR 8,0:LOCATE 15,35:PRINT "|";
2424 COLOR 8,0:LOCATE 15,39:PRINT "|";
2426 COLOR 7,0:LOCATE 15,49:PRINT "     ";
2428 COLOR 8,0:LOCATE 15,55:PRINT "|";
2430 COLOR 8,0:LOCATE 15,59:PRINT "|";
2432 COLOR 14,0:LOCATE 15,69:PRINT "     ";
2434 COLOR 8,0:LOCATE 15,75:PRINT "|";
2436 COLOR 8,0:LOCATE 15,79:PRINT "|";
2438 COLOR 8,0:LOCATE 16,13:PRINT ".....";
2440 COLOR 8,0:LOCATE 16,19:PRINT "|";
2442 COLOR 8,0:LOCATE 16,33:PRINT ".....";
2444 COLOR 8,0:LOCATE 16,39:PRINT "|";
2446 COLOR 8,0:LOCATE 16,53:PRINT ".....";
2448 COLOR 8,0:LOCATE 16,59:PRINT "|";
2450 COLOR 8,0:LOCATE 16,73:PRINT ".....";
2452 COLOR 8,0:LOCATE 16,79:PRINT "|";
2454 COLOR 3,0:LOCATE 17,13:PRINT "     ";
2456 COLOR 8,0:LOCATE 17,19:PRINT "|";
2458 COLOR 4,0:LOCATE 17,33:PRINT "     ";
2460 COLOR 8,0:LOCATE 17,39:PRINT "|";
2462 COLOR 7,0:LOCATE 17,53:PRINT "     ";
2464 COLOR 8,0:LOCATE 17,59:PRINT "|";
2466 COLOR 14,0:LOCATE 17,73:PRINT "     ";
2468 COLOR 8,0:LOCATE 17,79:PRINT "|";
2470 COLOR 8,0:LOCATE 18,17:PRINT ".....";
2472 COLOR 8,0:LOCATE 18,37:PRINT ".....";
2474 COLOR 8,0:LOCATE 18,57:PRINT ".....";
2476 COLOR 8,0:LOCATE 18,76:PRINT ".....";
2478 COLOR 3,0:LOCATE 19,17:PRINT "     ";
2480 COLOR 4,0:LOCATE 19,37:PRINT "     ";
2482 COLOR 7,0:LOCATE 19,57:PRINT "     ";
2484 COLOR 14,0:LOCATE 19,76:PRINT "     ";
2486 COLOR 3,0:LOCATE 24,1:PRINT "Enter one of the"
     "following (r to reverse or f to go forward)";
2488 COLOR 15,1:LOCATE 24,74:PRINT " ";
2490 COLOR 2,0:RETURN
4990 '
4991 '      INTRO SCREEN SUBROUTINE
4992 '
4993 '      ----------------------
5000 ON REQ% GOSUB 5052,5046,5042,5024,5034,5030,5038
5001 RETURN
5002 ' ENTRY------------ENTRY TYPE ------------SUBROUTINE-
5003 '  1    DISPLAY SCREEN ONLY (ERASE DATA)    5052
5004 '  2    DISPLAY SCREEN & DATA ELEMENTS      5046
5006 '  3    DISPLAY DATA ELEMENTS ONLY          5042
5008 '  4    INPUT ALL DATA ELEMENTS             5024
5010 '  5    INPUT ONE ELEMENT BY RELATIVE FLD   5034
5012 '  6    UPDATE ALL DATA ELEMENTS            5030
5014 '  7    UPDATE ONE DATA ELEMENT BY REL FLD  5038
5016 ' ---------------------------------------------------
5018 SG.DF$=SCR$
5020 SG.UP%=1:SG.ROW%=23:SG.COL%=73:SG.LNG%=1
5021 COLOR 15,1  :GOSUB 775
5022 SCR$=SG.REPLY$:RETURN
5024 GOSUB 1462:GOSUB 5053
5026 IF SG.BTAB%=0 THEN GOSUB 5020  ELSE GOSUB 5018  'SCR$
5028 RETURN
5030 GOSUB 1462
5031 GOSUB 5018    'SCR$
5032 IF SG.BTAB%>2 THEN RETURN
5033 RETURN
5034 GOSUB 1462
5035 IF FLD%<31 THEN ON FLD% GOSUB 5020
5037 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
```

```
5038 GOSUB 1462
5039 IF FLD%<31 THEN ON FLD% GOSUB 5018
5041 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
5042 GOSUB 1462
5043 LOCATE 23,73:COLOR 15,1:PRINT USING "!";SCR$;
5045 RETURN
5046 GOSUB 5052
5048 GOSUB 5043
5050 RETURN
5051 LOCATE 25,1:COLOR , 0:PRINT" ";
5052 COLOR ,, 8:CLS:GOSUB 1462
5053 COLOR  2,0
5054 COLOR 3,0:LOCATE 2,28:PRINT "MULTI - STREAM  CONCEPT"
5056 COLOR 13,0:LOCATE 4,19:PRINT "T1  T2  T3  T4  T5";
5058 COLOR 13,0:LOCATE 7,6:PRINT "AGE3";
5060 COLOR 7,0:LOCATE 7,40:PRINT " .";
5062 COLOR 15,1:LOCATE 7,42:PRINT "1a";
5064 COLOR 7,0:LOCATE 7,44:PRINT ".";
5066 COLOR 15,1:LOCATE 7,45:PRINT "1b";
5068 COLOR 7,0:LOCATE 7,52:PRINT " .";
5070 COLOR 15,1:LOCATE 7,54:PRINT "2a";
5072 COLOR 7,0:LOCATE 7,56:PRINT ".";
5074 COLOR 15,1:LOCATE 7,57:PRINT "2b";
5076 COLOR 7,0:LOCATE 7,65:PRINT " .";
5078 COLOR 15,1:LOCATE 7,67:PRINT "3a";
5080 COLOR 7,0:LOCATE 7,69:PRINT ".";
5082 COLOR 15,1:LOCATE 7,70:PRINT "3b";
5084 COLOR 7,0:LOCATE 8,32:PRINT ".";
5086 COLOR 7,0:LOCATE 8,35:PRINT ".";
5088 COLOR 7,0:LOCATE 8,44:PRINT ".";
5090 COLOR 7,0:LOCATE 8,47:PRINT ".";
5092 COLOR 7,0:LOCATE 8,56:PRINT ".";
5094 COLOR 7,0:LOCATE 8,59:PRINT ".";
5096 COLOR 7,0:LOCATE 8,69:PRINT ".";
5098 COLOR 7,0:LOCATE 8,72:PRINT ".";
5100 COLOR 13,0:LOCATE 9,6:PRINT "AGE2";
5102 COLOR 7,0:LOCATE 9,28:PRINT " .";
5104 COLOR 15,1:LOCATE 9,30:PRINT "1a";
5106 COLOR 7,0:LOCATE 9,32:PRINT ".";
5108 COLOR 15,1:LOCATE 9,33:PRINT "1b";
5110 COLOR 7,0:LOCATE 9,39:PRINT "  .";
5112 COLOR 15,1:LOCATE 9,42:PRINT "2a";
5114 COLOR 7,0:LOCATE 9,44:PRINT ".";
5116 COLOR 15,1:LOCATE 9,45:PRINT "2b";
5118 COLOR 7,0:LOCATE 9,52:PRINT " .";
5120 COLOR 15,1:LOCATE 9,54:PRINT "3a";
5122 COLOR 15,1:LOCATE 9,57:PRINT "3b";
5124 COLOR 7,0:LOCATE 9,65:PRINT " .";
5126 COLOR 15,1:LOCATE 9,67:PRINT "4a";
5128 COLOR 7,0:LOCATE 9,69:PRINT ".";
5130 COLOR 15,1:LOCATE 9,70:PRINT "4b";
5132 COLOR 7,0:LOCATE 10,19:PRINT ".";
5134 COLOR 7,0:LOCATE 10,22:PRINT ".";
5136 COLOR 7,0:LOCATE 10,32:PRINT ".";
5138 COLOR 7,0:LOCATE 10,35:PRINT ".";
5140 COLOR 7,0:LOCATE 10,44:PRINT ".";
5142 COLOR 7,0:LOCATE 10,47:PRINT ".";
5144 COLOR 7,0:LOCATE 10,56:PRINT ".";
5146 COLOR 7,0:LOCATE 10,59:PRINT ".";
5148 COLOR 7,0:LOCATE 10,69:PRINT ".";
5150 COLOR 7,0:LOCATE 10,72:PRINT ".";
5152 COLOR 13,0:LOCATE 11,6:PRINT "AGE1";
5154 COLOR 15,1:LOCATE 11,17:PRINT "1a";
5156 COLOR 15,1:LOCATE 11,20:PRINT "1b";
5158 COLOR 15,1:LOCATE 11,30:PRINT "2a";
5160 COLOR 15,1:LOCATE 11,33:PRINT "2b";
5162 COLOR 15,1:LOCATE 11,42:PRINT "3a";
5164 COLOR 15,1:LOCATE 11,45:PRINT "3b";
5166 COLOR 15,1:LOCATE 11,54:PRINT "4a";
5168 COLOR 15,1:LOCATE 11,57:PRINT "4b";
5170 COLOR 15,1:LOCATE 11,67:PRINT "5a";
5172 COLOR 15,1:LOCATE 11,70:PRINT "5b";
5173 COLOR 3,0:LOCATE 15,6
5174 PRINT "T1,T2,.......  = time period"
```

```
          "AGE1,AGE2,...  = age of end items";
5176 COLOR 3,0:LOCATE 17,6:PRINT "1a 1b........";
5178 COLOR 3,0:LOCATE 17,21
5179 PRINT "= end items operating in failure";
5180 COLOR 3,0:LOCATE 18,23:PRINT "environments A and B";
5181 COLOR 3,0:LOCATE 20,6
5182 PRINT "1a.1a.1a.....  = a stream"
       "(A stream is no more when either all end";
5184 COLOR 3,0:LOCATE 21,23:PRINT "items have failed"
                "or all end items are retired.)";
5186 LOCATE 23,1:PRINT " Enter one of the following"
5187    "(R to go back one screen, F to go forward)";
5188 COLOR 15,1:LOCATE 23,73:PRINT " ";
5190 COLOR 2,0:RETURN
5990 '
5991 '      INTRO SCREEN SUBROUTINE
5992 '      ----------------------
5993 '
6000 ON REQ% GOSUB 6052,6046,6042,6024,6034,6030,6038
6001 RETURN
6002 ' ENTRY------------ENTRY TYPE ------------SUBROUTINE-
6003 '  1    DISPLAY SCREEN ONLY (ERASE DATA)   6052
6004 '  2    DISPLAY SCREEN & DATA ELEMENTS     6046
6006 '  3    DISPLAY DATA ELEMENTS ONLY         6042
6008 '  4    INPUT ALL DATA ELEMENTS            6024
6010 '  5    INPUT ONE ELEMENT BY RELATIVE FLD  6034
6012 '  6    UPDATE ALL DATA ELEMENTS           6030
6014 '  7    UPDATE ONE DATA ELEMENT BY REL FLD 6038
6016 ' ---------------------------------------------------
6018 SG.DF$=SCR$
6020 SG.UP%=1:SG.ROW%=23:SG.COL%=74:SG.LNG%=1
6021 COLOR 15,1  :GOSUB 775
6022 SCR$=SG.REPLY$:RETURN
6024 GOSUB 1462:GOSUB 6053
6026 IF SG.BTAB%=0 THEN GOSUB 6020  ELSE GOSUB 6018  'SCR$
6028 RETURN
6030 GOSUB 1462
6031 GOSUB 6018    'SCR$
6032 IF SG.BTAB%>2 THEN RETURN
6033 RETURN
6034 GOSUB 1462
6035 IF FLD%<31 THEN ON FLD% GOSUB 6020
6037 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
6038 GOSUB 1462
6039 IF FLD%<31 THEN ON FLD% GOSUB 6018
6041 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
6042 GOSUB 1462
6043 LOCATE 23,74:COLOR 15,1:PRINT USING "!";SCR$;
6045 RETURN
6046 GOSUB 6052
6048 GOSUB 6043
6050 RETURN
6051 LOCATE 25,1:COLOR , 0:PRINT" ";
6052 COLOR ,, 8:CLS:GOSUB 1462
6053 COLOR  2,0
6054 COLOR 3,0:LOCATE 2,26
6055 PRINT "MAINTENANCE  TREE  CONCEPT";
6056 LOCATE 4,1:PRINT "   The repairable items being"
             "evaluated by this logistics model are";
6058 LOCATE 5,1:PRINT "   referred to as end items."
       "Each end item is broken into repairable or";
6060 LOCATE 6,1:PRINT "   replaceable components and"
            "then into subcomponents and so on until";
6062 LOCATE 7,1:PRINT "   a spare part level is reached."
                  "This breakdown produces a maintenance";
6064 LOCATE 8,1:PRINT "   tree.";
6066 COLOR 10,0:LOCATE 8,11:PRINT "See the next screen"
                        "for a sample maintenance tree.";
6068 COLOR 6,0:LOCATE 10,4:PRINT "Maintenance Tree Terminology:";
6070 COLOR 6,0:LOCATE 12,4:PRINT "node - each"
         "component on the tree represents a node";
6072 COLOR 6,0:LOCATE 14,4:PRINT "bottom of the tree node"
       "- this is a node where failure and repair data is";
6074 COLOR 6,0:LOCATE 15,30:PRINT "Known";
6076 COLOR 6,0:LOCATE 17,4:PRINT "stockable items"
```

```
     "- parts needed to repair the component listed at a";
6078 COLOR 6,0:LOCATE 18,30:PRINT "bottom of the tree node";
6080 COLOR 6,0:LOCATE 20,4:PRINT "repair level"
     "- maintenance facility which must perform the repair";
6082 COLOR 3,0:LOCATE 23,3:PRINT "Enter one of the"
     "following (R to go back one screen, F to go forward)";
6084 COLOR 15,1:LOCATE 23,74:PRINT " ";
6086 COLOR 2,0:RETURN
6990 '
6991 '      INTRO SCREEN SUBROUTINE
6992 '      ----------------------
6993 '
7000 ON REQ% GOSUB 7052,7046,7042,7024,7034,7030,7038
7001 RETURN
7002 ' ENTRY------------ENTRY TYPE ------------SUBROUTINE-
7003 '  1     DISPLAY SCREEN ONLY (ERASE DATA)   7052
7004 '  2     DISPLAY SCREEN & DATA ELEMENTS     7046
7006 '  3     DISPLAY DATA ELEMENTS ONLY         7042
7008 '  4     INPUT ALL DATA ELEMENTS            7024
7010 '  5     INPUT ONE ELEMENT BY RELATIVE FLD  7034
7012 '  6     UPDATE ALL DATA ELEMENTS           7030
7014 '  7     UPDATE ONE DATA ELEMENT BY REL FLD 7038
7016 ' ---------------------------------------------------
7018 SG.DF$=SCR$
7020 SG.UP%=1:SG.ROW%=24:SG.COL%=74
7021 SG.LNG%=1:COLOR 15,1  :GOSUB 775
7022 SCR$=SG.REPLY$:RETURN
7024 GOSUB 1462:GOSUB 7053
7026 IF SG.BTAB%=0 THEN GOSUB 7020  ELSE GOSUB 7018  'SCR$
7028 RETURN
7030 GOSUB 1462
7031 GOSUB 7018    'SCR$
7032 IF SG.BTAB%>2 THEN RETURN
7033 RETURN
7034 GOSUB 1462
7035 IF FLD%<31 THEN ON FLD% GOSUB 7020
7037 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
7038 GOSUB 1462
7039 IF FLD%<31 THEN ON FLD% GOSUB 7018
7041 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
7042 GOSUB 1462
7043 LOCATE 24,74:COLOR 15,1:PRINT USING "!";SCR$;
7045 RETURN
7046 GOSUB 7052
7048 GOSUB 7043
7050 RETURN
7051 LOCATE 25,1:COLOR , 0:PRINT" ";
7052 COLOR ,, 8:CLS:GOSUB 1462
7053 COLOR  2,0
7054 LOCATE 1,1:PRINT "                IMMMMMMMMMM;";
7056 LOCATE 2,1:PRINT "                           :";
7058 COLOR 10,0:LOCATE 2,40:PRINT "car";
7060 COLOR 2,0:LOCATE 2,43:PRINT "   :";
7062 LOCATE 3,1:PRINT "                MMMMMMMMMMM";
7064 LOCATE 4,1:PRINT ".     .      .     .         .";
7066 LOCATE 5,1:PRINT ".     .      .     .         .";
7068 LOCATE 6,1:PRINT "_____ _________  _________";
7070 LOCATE 7,1:PRINT "     |";
7072 COLOR 10,0:LOCATE 7,7:PRINT "cooling";
7074 COLOR 2,0:LOCATE 7,14:PRINT "|   |";
7076 COLOR 10,0:LOCATE 7,19:PRINT "drivetrain";
7078 COLOR 2,0:LOCATE 7,29:PRINT "|     |";
7080 COLOR 10,0:LOCATE 7,36:PRINT "electrical";
7082 COLOR 2,0:LOCATE 7,46:PRINT "|     |";
7084 COLOR 10,0:LOCATE 7,55:PRINT "other";
7086 COLOR 2,0:LOCATE 7,60:PRINT "  |     |";
7088 COLOR 10,0:LOCATE 7,70:PRINT "fatal";
7090 COLOR 2,0:LOCATE 7,75:PRINT " |";
7092 LOCATE 8,1:PRINT "     |";
7094 COLOR 10,0:LOCATE 8,7:PRINT "system";
7096 COLOR 2,0:LOCATE 8,13:PRINT " |   |";
7098 COLOR 10,0:LOCATE 8,19:PRINT "combustion";
7100 COLOR 2,0:LOCATE 8,29:PRINT "|     |";
7102 COLOR 10,0:LOCATE 8,38:PRINT "system";
7104 COLOR 2,0:LOCATE 8,44:PRINT "  |     |";
```

```
7106 COLOR 10,0:LOCATE 8,54:PRINT "failures";
7108 COLOR 2,0:LOCATE 8,62:PRINT "|      |";
7110 COLOR 10,0:LOCATE 8,69:PRINT "failure";
7112 COLOR 2,0:LOCATE 8,76:PRINT "|";
7114 LOCATE 9,1:PRINT "-------  ----------    ----------";
7116 LOCATE 10,1:PRINT "                          .   .";
7118 LOCATE 11,1:PRINT "                     .        .";
7120 LOCATE 12,1:PRINT "              _________  _________";
7122 LOCATE 13,1:PRINT "                                |";
7124 COLOR 10,0:LOCATE 13,31:PRINT "alternator";
7126 COLOR 2,0:LOCATE 13,41:PRINT "| |";
7128 COLOR 10,0:LOCATE 13,44:PRINT "ignition";
7130 COLOR 2,0:LOCATE 13,52:PRINT "|";
7132 COLOR 2,0:LOCATE 14,1:PRINT "----------   --------";
7134 COLOR 6,0:LOCATE 15,26:PRINT "MTTR=1,RL=2,MTBF=3";
7136 LOCATE 16,1:PRINT "                              .";
7138 LOCATE 17,1:PRINT "                     alternator";
7140 LOCATE 18,1:PRINT "                        brushes";
7142 LOCATE 19,1:PRINT "                          bolts";
7144 COLOR 3,0:LOCATE 24,3:PRINT "Enter one of the"
     "following (R to go back, F to go forward)";
7146 COLOR 15,1:LOCATE 24,74:PRINT " ";
7148 COLOR 2,0:RETURN
7990 '
7991 '      INTRO SCREEN SUBROUTINE
7992 '      ____________________________
7993 '
8000 ON REQ% GOSUB 8052,8046,8042,8024,8034,8030,8038
8001 RETURN
8002 ' ENTRY------------ENTRY TYPE ------------SUBROUTINE-
8003 '  1    DISPLAY SCREEN ONLY (ERASE DATA)   8052
8004 '  2    DISPLAY SCREEN & DATA ELEMENTS     8046
8006 '  3    DISPLAY DATA ELEMENTS ONLY         8042
8008 '  4    INPUT ALL DATA ELEMENTS            8024
8010 '  5    INPUT ONE ELEMENT BY RELATIVE FLD  8034
8012 '  6    UPDATE ALL DATA ELEMENTS           8030
8014 '  7    UPDATE ONE DATA ELEMENT BY REL FLD 8038
8016 ' ---------------------------------------------------
8018 SG.DF$=SCR$
8020 SG.UP%=1:SG.ROW%=23:SG.COL%=73:SG.LNG%=1
8021 COLOR 15,1  :GOSUB 775
8022 SCR$=SG.REPLY$:RETURN
8024 GOSUB 1462:GOSUB 8053
8026 IF SG.BTAB%=0 THEN GOSUB 8020  ELSE GOSUB 8018  'SCR$
8028 RETURN
8030 GOSUB 1462
8031 GOSUB 8018    'SCR$
8032 IF SG.BTAB%>2 THEN RETURN
8033 RETURN
8034 GOSUB 1462
8035 IF FLD%<31 THEN ON FLD% GOSUB 8020
8037 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
8038 GOSUB 1462
8039 IF FLD%<31 THEN ON FLD% GOSUB 8018
8041 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
8042 GOSUB 1462
8043 LOCATE 23,73:COLOR 15,1:PRINT USING "!";SCR$;
8045 RETURN
8046 GOSUB 8052
8048 GOSUB 8043
8050 RETURN
8051 LOCATE 25,1:COLOR , 0:PRINT" ";
8052 COLOR ,, 8:CLS:GOSUB 1462
8053 COLOR  2,0
8054 COLOR 3,0:LOCATE 2,6:PRINT "The previous screen"
     "was a partial maintenance tree for an automobile.";
8056 COLOR 3,0:LOCATE 4,6:PRINT "explanations:";
8058 COLOR 10,0:LOCATE 6,6:PRINT "alternator and ignition";
8060 COLOR 3,0:LOCATE 6,37
8061 PRINT "- were bottom of the tree nodes";
8062 LOCATE 8,1:PRINT "    alternator, brushes, and bolts";
8064 COLOR 3,0:LOCATE 8,37:PRINT "- were";
8066 COLOR 3,0:LOCATE 8,44:PRINT "stockable items";
8068 COLOR 6,0:LOCATE 10,6:PRINT "MTTR";
8070 COLOR 3,0:LOCATE 10,37:PRINT "- mean time to repair";
```

```
8072 COLOR 6,0:LOCATE 12,6:PRINT "MTBF";
8074 COLOR 3,0:LOCATE 12,37
8075 PRINT "- mean time between failures";
8076 COLOR 6,0:LOCATE 14,6:PRINT "RL";
8078 COLOR 3,0:LOCATE 14,37:PRINT "- repair level";
8080 COLOR 3,0:LOCATE 18,6:PRINT "The next screen shows"
            "the maintenance tree limitations for this";
8082 COLOR 3,0:LOCATE 18,70:PRINT "model.";
8084 LOCATE 20,1:PRINT "     max. # nodes = 126"
                "max. levels of indenture = 5";
8086 COLOR 3,0:LOCATE 23,2:PRINT "Enter one of the"
     "following (R to go back one screen, F to go forward)"
8088 COLOR 15,1:LOCATE 23,73:PRINT " ";
8090 COLOR 2,0:RETURN
10990 '
10991 '     INTRO SCREEN SUBROUTINE
10992 '
10993 '
10999 ON REQ% GOSUB 11052,11046,11042,11024,
                        11034,11030,11038
11001 RETURN
11002 ' ENTRY------------ENTRY TYPE ------------SUBROUTINE-
11003 '  1    DISPLAY SCREEN ONLY (ERASE DATA)   11052
11004 '  2    DISPLAY SCREEN & DATA ELEMENTS     11046
11006 '  3    DISPLAY DATA ELEMENTS ONLY         11042
11008 '  4    INPUT ALL DATA ELEMENTS            11024
11010 '  5    INPUT ONE ELEMENT BY RELATIVE FLD  11034
11012 '  6    UPDATE ALL DATA ELEMENTS           11030
11014 '  7    UPDATE ONE DATA ELEMENT BY REL FLD 11038
11016 ' ---------------------------------------------------
11018 SG.DF$=SCR$
11020 SG.UP%=1:SG.ROW%=23:SG.COL%=61:SG.LNG%=1
11021 COLOR 0,7  :GOSUB 775
11022 SCR$=SG.REPLY$:RETURN
11024 GOSUB 1462:GOSUB 11053
11026 IF SG.BTAB%=0 THEN GOSUB 11020  ELSE GOSUB 11018  'SCR$
11028 RETURN
11030 GOSUB 1462
11031 GOSUB 11018    'SCR$
11032 IF SG.BTAB%>2 THEN RETURN
11033 RETURN
11034 GOSUB 1462
11035 IF FLD%<31 THEN ON FLD% GOSUB 11020
11037 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
11038 GOSUB 1462
11039 IF FLD%<31 THEN ON FLD% GOSUB 11018
11041 RETURN   'FIELDS HIGHER THAN 60 NOT INCLUDED ABOVE
11042 GOSUB 1462
11043 LOCATE 23,61:COLOR 0,7:PRINT USING "!";SCR$;
11045 RETURN
11046 GOSUB 11052
11048 GOSUB 11043
11050 RETURN
11051 LOCATE 25,1:COLOR , 7:PRINT" ";
11052 COLOR ,, 8:CLS:GOSUB 1462
11053 COLOR  0,7
11054 LOCATE 1,1:PRINT SPACE$(80);
11056 LOCATE 2,1:PRINT "";
11058 COLOR 2,7:LOCATE 2,1:PRINT SPACE$(80);
11060 LOCATE 3,1:PRINT SPACE$(80);
11062 LOCATE 4,1:PRINT "";
11064 COLOR 0,7:LOCATE 4,1:PRINT SPACE$(80);
11066 LOCATE 5,1:PRINT SPACE$(80);
11068 LOCATE 6,1:PRINT "MAINTENANCE REQUIREMENTS PLANNING"
11070 LOCATE 7,1:PRINT SPACE$(80);
11072 LOCATE 8,1:PRINT "                FOR";
11074 LOCATE 9,1:PRINT SPACE$(80);
11076 LOCATE 10,1:PRINT "  REPAIRABLE ITEM POPULATIONS";
11078 LOCATE 11,1:PRINT SPACE$(80);
11080 LOCATE 12,1:PRINT SPACE$(80);
11082 LOCATE 13,1:PRINT SPACE$(80);
11086 LOCATE 14,1:PRINT "                  ";
11088 LOCATE 14,1:PRINT "             MARCH 1, 1985";
11090 LOCATE 16,1:PRINT "               REVISIONS:";
11180 LOCATE 17,1:PRINT SPACE$(80);
```

```
11182 LOCATE 18,1:PRINT "                                    ";
11184 COLOR 12,7:LOCATE 18,35:PRINT "MMMMMMMMM ";
11186 COLOR 0,7:LOCATE 18,45:PRINT SPACE$(36);
11188 LOCATE 19,1:PRINT "                                    ";
11190 COLOR 9,7:LOCATE 19,35:PRINT "MMMMMMMMM";
11192 COLOR 12,7:LOCATE 19,44:PRINT "  ";
11194 COLOR 0,7:LOCATE 19,46:PRINT SPACE$(35);
11196 LOCATE 20,1:PRINT "                                    ";
11198 COLOR 12,7:LOCATE 20,37:PRINT "          ";
11200 COLOR 0,7:LOCATE 20,47:PRINT SPACE$(34);
11204 LOCATE 21,1:PRINT SPACE$(80);
11206 LOCATE 22,1:PRINT SPACE$(80);
11208 LOCATE 23,1
11209 PRINT "Enter F to go forward to the next screen";
11210 LOCATE 24,1:PRINT SPACE$(79);
11212 COLOR 2,0:RETURN
```

```
1 '
2 '     PROGRAM LISTING FOR "INPUT1"
3 '
4 '
5 '     DIMENSION ARRAYS
6 '     ----------------
7 '
26 DIM NODE(126),IP%(40),ST%(40),NO.ENVIR$(5),NN$(126)
27 DIM RL%(126),RA(5),MTTRP(126),AA%(126),NDE(126),AMT%(40)
30 DIM M(126),M1(126),KKEY%(40),TTIME%(40),CCP(50),CCH(50)
31 DIM LLT(50),PPR(50),G%(40),SSC(50),WH(5),PPSPR%(50)
32 DIM SUMPF(5),SUMSTREAM(5),AVGMTTR(5),AVGPF(5),P(5)
33 DIM AVAILABILITY(5),SP%(40),SP2%(40),NOD(40),RRA(126)
34 DIM PPN$(50),IIAP%(50)
35 DIM NNAME$(50),SPR%(50),FT3(50),NO.ITEMS%(10),DA(10)
36 DIM SMTTR(5),C1(5),TC(5),SUMSP%(5),SUMSP2%(5),C5(5)
37 DIM C6(5),C7(5),C8(5),C9(5),C10(5),ACOST(5),C4(40)
39 DIM MTTRAVG(20),TOTALAVAIL(40)
40 DIM MTBFP(126),NODEP%(126),CDFP%(126),MTBF1(126)
41 DIM NODE1%(126),CDF1%(126),MTTR1(126),MTBF2(126)
42 DIM NODE2%(126),CDF2%(126),MTTR2(126),DEMAND%(5)
50 DIM MTBF3(126),NODE3%(126),CDF3%(126),MTTR3(126)
51 DIM MTBF4(126),NODE4%(126),CDF4%(126)
55 DIM I$(10),PN$(10),QY(10),SR(10),PR(10)
56 DIM CP(10),CH(10),LT(10)
59 DIM GA1(12,2),GCOST(24,8),GMP(12,3),GNP(12,8)
60 DIM G108(12,2),G107(12,2),MP(5),PCOST(12)
65 '
66 '     GIVE USER CHOICE OF WORKING AN EXAMPLE, SAVING
67 '     THE DATA FROM A PREVIOUS MODEL RUN, BEGINNING,
68 '     A MODEL RUN WITH ALL NEW DATA, OR ENDING PROGRAM.
69 '     ------------------------------------------------
70 '
71 SCREEN 0,1
80 CNT6%=0: T%=0: COLOR 3,0,8: CLS
81 LOCATE 2,30: PRINT "DATA  CHOICE"
84 COLOR 2,0: LOCATE 6,24
85 PRINT "1...........Work Example"
86 LOCATE 8,24: PRINT "2...........Save Data"
87 LOCATE 10,24:PRINT "3..........Continue-All New Inputs"
88 LOCATE 12,24: PRINT "4...........End Program"
89  COLOR 10,0: LOCATE 16,24
90 INPUT "Enter 1,2,3, or 4 >",UB%
91 IF UB% = 4 THEN END
93 '
94 '     USER INPUTS CHOICE OF PRINTER LISTINGS
95 '     --------------------------------------
96 '
102 CLS: COLOR 3,0,8
103 LOCATE 2,30: PRINT "PRINT OUT CHOICE"
104 COLOR 2,0: LOCATE 6,24
105 PRINT "1...Give Printer Listing of all Output Screens"
106 LOCATE 10,24
107 PRINT "2.....Give Printer Listing of End of"
                          "Model Summaries Only"
109 COLOR 10,0: LOCATE 17,24: INPUT "Enter 1 or 2 >",UP%
110 IF UB% <> 3 THEN 130
115 '
116 '     INITIALIZE VARIABLES
117 '     --------------------
118 '
120 FOR N% = 1 TO 126
121   MTBFP(N%)=0: NODEP%(N%)=0: CDFP%(N%)=0
122   MTBF1(N%)=0: NODE1%(N%)=0: CDF1%=0
123   MTBF2(N%)=0: NODE2%(N%)=0: CDF2%(N%)=0
124 NEXT
130 IF UB% = 2 OR UB% = 3 THEN 138
131 '
132 '     DATA FOR RUNNING EXAMPLE
133 '     ------------------------
134 '
135 LTP$="days": LTP%=30: LMR%=12: NO.ENVIR%=2: CC=350000!
136 SV=8000: PC=10000: RC=500: SC=30000: II=10: B$="A"
137 SPWA%=2: NO.RL%=3
138 '
```

```
139 '      INITIALIZE VARIABLES
140 '      ____________________
141 '
142 FOR N% = 1 TO 40
143   ST%(N%)=0:IP%(N%)=0:SP%(N%)=0:SP2%(N%)=0
144 NEXT
145 R1%=0:PA%=1:D%=0:SIG%=0:CNT3%=0
146 FOR N% = 1 TO 4
147   C5(N%)=0:C6(N%)=0:C7(N%)=0:C8(N%)=0
148   C9(N%)=0:SUMSTREAM(N%)=0
149 NEXT
151 '
152 '          GOSUB TO SCREEN SUBROUTINE
153 '      INPUT PERIOD UNIT, LENGTH OF PERIOD,
154 '             LENGTH OF MODEL RUN,
155 '         AND # OF FAILURE ENVIRONMENTS
156 '      ____________________________________
157 '
158 REQ%=2:GOSUB 4000
159 REQ%=6:GOSUB 4000
160 '
161 '        SET VALUE FOR LX% DEPENDING ON
162 '      PERIOD UNIT CHOSEN BY USER OF MODEL
163 '      ___________________________________
164 '
169 IF ASC(LTP$)=68 OR ASC(LTP$)=100 THEN 200
170 IF ASC(LTP$)=87 OR ASC(LTP$)=119 THEN 210
180 IF ASC(LTP$)=77 OR ASC(LTP$)=109 THEN 220
190  LX%=365: LTP$="year":GOTO 230
200 LX%=1:  LTP$="day":GOTO 230
210 LX%=7:  LTP$="week":GOTO 230
220 LX%=30: LTP$="month":GOTO 230
230 IF SCR$ = "R" THEN 158
232 '
233 '        GOSUB SCREEN SUBROUTINE
234 '      INPUT COST DATA FOR END ITEM
235 '      ____________________________
236 '
240 REQ%=2:GOSUB 5000
250 REQ%=6:GOSUB 5000
260 I=II/100
270 IF SCR$ = "R" THEN 158
272 '
273 '             GOSUB SCREEN SUBROUTINE
274 '      INPUT TYPE POPULATION BEING EVALUATED,
275 '         INVENTORY LOT SIZING METHOD, AND
276 '               # OF REPAIR LEVELS
277 '      ____________________________________
278 '
280 REQ%=2:GOSUB 8000
290 REQ%=6:GOSUB 8000
295 FOR N% = 1 TO NO.RL%
296  FOR VT% = 1 TO LMR%:  GMP(VT%,N%) = 0: NEXT
297 NEXT
300 IF SPWA% = 3 THEN NS%=1
310 IF SCR$ = "R" THEN 240
315 IF UB% = 1 OR UB% = 2 THEN 370
316 '
317 '          GOSUB SCREEN SUBROUTINE
318 '      DISPLAY LIBRARY OF FAILURE CURVES
319 '      _________________________________
320 '
321 GOSUB 7000
330 SCREEN 0,1
340 IF SCR$ = "R" OR SCR$ = "r" THEN SCR$="R"
350 IF SCR$ = "F" OR SCR$ = "f" THEN SCR$="F"
360 IF SCR$ = "R" THEN 280
364 '
365 '             GOSUB SCREEN SUBROUTINE
366 '        INPUT AND STORE NODE FAILURE DATA
367 '      LOOP THROUGH ONCE FOR EACH ENVIRONMENT
368 '      ______________________________________
369 '
370 FOR N% = 1 TO NO.ENVIR%
374   IF UB% = 1 THEN 396
```

```
375   IF UB% = 2 THEN 405
376 '
377 '      input data
378 '      ----------
379 '
380   GOSUB 7500
385   SCREEN 0,1
390   IF UB% = 3 THEN 405
391 '
392 '      more input data for example
393 '      ---------------------------
394 '
396   GOSUB 2000
400 '
401 '      store data
402 '      ----------
403 '
405   ON N% GOSUB 11050,11150,11250,11350
408 '
409 '      gosub screen subroutines for period by period
410 '           probability of component failure if
411 '                      data exists
412 '      ----------------------------------------------
413 '
415   GOSUB 20000
425   REQ%=2:GOSUB 21000
426   REQ%=6:GOSUB 21000
428 '
429 '                gosub screen subroutines
430 '      input demand, end item retirement age, and
431 '          end item book value at retirement
432 '      ----------------------------------------------
433 '
435   GOSUB 20500
445   REQ%=2:GOSUB 17000
446   REQ%=6:GOSUB 17000
450 NEXT
480 '
481 '      CHAIN TO "INPUT2"
482 '      -----------------
483 '
490 CHAIN "INPUT2",200,ALL
700 '
710 '      END OF MAIN PROGRAM (INPUT1)
720 '
730 '
740 '
750 '      SCREEN SUBROUTINE DRIVER
760 '      ------------------------
770 '
780 '      note:  not included - see Screen Subroutine
790 '             Driver, lines 700-1486 in INTRO
800 '
810 '
1990 '
1992 '      SUBROUTINE CONTAINING FAILURE DATA FOR
1994 '                  EXAMPLE
1995 '      ---------------------------------------
1997 '
2000 NODE1%(27)=27: NN$(27)="Freshwa.Pump": CDF1%(27)=3
2001 RL(27)=2: MTBF1(27)=8.33: MTTR1(27)=1!
2002 DEMAND%(1)=140: BV(1)=15000: RA(1)=500
2003 NN$(6)="fatal failure": NO.ENVIR$(1)="SUPPORT"
2004 NO.ENVIR$(2)="COMBAT ZONE"
2010 CDF2%(27)=2: MTBF2(27)=5.57: MTTR2(27)=1!
2011 DEMAND%(2)=60: BV(2)=15000: RA(2)=500: NODE2%(27)=27
2015 NODE1%(28)=28: NN$(28)="Seawater Pump": CDF1%(28)=3
2016 RL%(28)=2: MTBF1(28)=20.83: MTTR1(28)=1.2
2017 CDF2%(28)=2: MTBF2(28)=13.9: MTTR2(28)=1.2
2018 NODE2%(28)=28
2020 NODE1%(29)=29: NN$(29)="Hyd. Pump": CDF1%(29)=3
2021 RL%(29)=1: MTBF1(29)=33.33: MTTR1(29)=1.2
2022 CDF2%(29)=2: MTBF2(29)=22.23: MTTR2(29)=1.2
2023 NODE2%(29)=29
2025 NODE1%(32)=32: NN$(32)="Eng.Coolant HE": CDF1%(32)=4
```

```
2026 RL%(32)=2: MTBF1(32)=33.33: MTTR1(32)=1.8
2027 CDF2%(32)=2: MTBF2(32)=22.23: MTTR2(32)=1.8
2028 NODE2%(32)=32
2030 NODE1%(33)=33: NN$(33)="Freshwa. HE": CDF1%(33)=4
2031 RL%(33)=1: MTBF1(33)=33.33: MTTR1(33)=1.8: CDF2%(33)=2
2032 MTBF2(33)=22.23: MTTR2(33)=1.8: NODE2%(33)=33
2035 NODE1%(34)=34: NN$(34)="Hyd.Fluid HE": CDF1%(34)=4
2036 RL%(34)=1: MTBF1(34)=33.33: MTTR1(34)=1.8
2037 CDF2%(34)=2: MTBF2(34)=22.23: MTTR2(34)=1.8
2038 NODE2%(34)=34
2040 NODE1%(10)=10: NN$(10)="Eng.Oil Cooler": CDF1%(10)=5
2041 MTBF1(10)=33.33: RL%(10)=1: MTTR1(10)=1.8: CDF2%(10)=2
2042 MTBF2(10)=22.23: MTTR2(10)=1.8: NODE2%(10)=10
2045 NODE1%(52)=52: NN$(52)="Pres.Red.Valve": CDF1%(52)=3
2046 RL%(52)=1: MTBF1(52)=16.67: MTTR1(52)=.8: CDF2%(52)=3
2047 MTBF2(52)=11.1: MTTR2(52)=.8: NODE2%(52)=52
2050 NODE1%(6)=6: CDF1%(6)=3: MTBF1(6)=416.67: CDF2%(6)=1
2051 MTBF2(6)=20: NODE2%(6)=6
2055 NODE1%(53)=53: NN$(53)="Dir.Con.Valve": CDF1%(53)=3
2056 RL%(53)=1: MTBF1(53)=4.17: MTTR1(53)=1.2: CDF2%(53)=3
2057 MTBF2(53)=2.77: MTTR2(53)=1.2: NODE2%(53)=53
2060 NODE1%(54)=54: NN$(54)="Steering Motor": CDF1%(54)=3
2061 RL%(54)=2: MTBF1(54)=12.5: MTTR1(54)=1.8: CDF2%(54)=3
2062 MTBF2(54)=8.33: MTTR2(54)=1.8: NODE2%(54)=54
2065 NODE1%(58)=58: NN$(58)="Filter": CDF1%(58)=3
2066 RL%(58)=1: MTBF1(58)=2.5: MTTR1(58)=.3: CDF2%(58)=3
2067 MTBF2(58)=1.67: MTTR2(58)=.3: NODE2%(58)=58
2070 NODE1%(59)=59: NN$(59)="Starter": CDF1%(59)=3
2071 RL%(59)=1: MTBF1(59)=10.4: MTTR1(59)=2!: CDF2%(59)=3
2072 MTBF2(59)=6.93: MTTR2(59)=2!: NODE2%(59)=59
2075 NODE1%(62)=62: NN$(62)="Auto. Lub.": CDF1%(62)=5
2076 RL%(62)=1: MTBF1(62)=6.27: MTTR1(62)=1!: CDF2%(62)=5
2077 MTBF2(62)=4.17: MTTR2(62)=1!: NODE2%(62)=62
2080 NODE1%(63)=63:NN$(63)="Timer": CDF1%(63)=5: RL%(63)=1
2081 MTBF1(63)=3.13: MTTR1(63)=2.4: CDF2%(63)=5
2082 MTBF2(63)=2.1: MTTR2(63)=2.4: NODE2%(63)=63
2085 NODE1%(64)=64: NN$(64)="Manifold": CDF1%(64)=3
2086 RL%(64)=2: MTBF1(64)=41.67: MTTR1(64)=15: CDF2%(64)=3
2087 MTBF2(64)=27.77: MTTR2(64)=15: NODE2%(64)=64
2090 NODE1%(65)=65: NN$(65)="Solenoid": CDF1%(65)=5
2091 RL%(65)=1: MTBF1(65)=3.13: MTTR1(65)=1.2: CDF2%(65)=5
2092 MTBF2(65)=2.1: MTTR2(65)=1.2: NODE2%(65)=65
2095 NODE1%(92)=92: NN$(92)="Waterjet Pump": CDF1%(92)=3
2096 RL%(92)=3: MTBF1(92)=20.83: MTTR1(92)=12: CDF2%(92)=3
2097 MTBF2(92)=13.9: MTTR2(92)=12: NODE2%(92)=92
2100 NODE1%(93)=93: NN$(93)="Disch.Elbow": CDF1%(93)=5
2101 RL%(93)=2: MTBF1(93)=41.67: MTTR1(93)=11: CDF2%(93)=5
2102 MTBF2(93)=27.77: MTTR2(93)=11: NODE2%(93)=93
2105 NODE1%(94)=94: NN$(94)="Steer.Nozzle": CDF1%(94)=3
2106 RL%(94)=2: MTBF1(94)=20.83: MTTR1(94)=12: CDF2%(94)=3
2107 MTBF2(94)=13.9: MTTR2(94)=12: NODE2%(94)=94
2110 NODE1%(107)=107: NN$(107)="BoosterPump":CDF1%(107)=3
2111 RL%(107)=2: MTBF1(107)=12.5:MTTR1(107)=.8:CDF2%(107)=3
2112 MTBF2(107)=8.33:MTTR2(107)=.8:NODE2%(107)=107
2115 NODE1%(108)=108:NN$(108)="Fuel/Wa.Sep.":CDF1%(108)=5
2116 RL%(108)=1:MTBF1(108)=6.27:MTTR1(108)=1:CDF2%(108)=5
2117 MTBF2(108)=4.17: MTTR2(108)=1: NODE2%(108)=108
2120 NODE1%(112)=112:NN$(112)="Inverter":CDF1%(112)=5
2121 RL%(112)=1: MTBF1(112)=8.33:MTTR1(112)=.5:CDF2%(112)=5
2122 MTBF2(112)=5.57: MTTR2(112)=.5: NODE2%(112)=112
2125 NODE1%(113)=113: NN$(113)="Battery": CDF1%(113)=3
2126 RL%(113)=1: MTBF1(113)=4.17:MTTR1(113)=.1:CDF2%(113)=3
2127 MTBF2(113)=2.77:MTTR2(113)=.1:NODE2%(113)=113
2130 NODE1%(114)=114:NN$(114)="Alternator":CDF1%(114)=5
2131 RL%(114)=1:MTBF1(114)=8.33:MTTR1(114)=2.4:CDF2%(114)=5
2132 MTBF2(114)=5.57:MTTR2(114)=2.4:NODE2%(114)=114
2135 NODE1%(118)=118: NN$(118)="Hyd. Motor": CDF1%(118)=5
2136 RL%(118)=2: MTBF1(118)=4.17:MTTR1(118)=.8:CDF2%(118)=5
2137 MTBF2(118)=2.77:MTTR2(118)=.8:NODE2%(118)=118
2140 NODE1%(119)=119: NN$(119)="Hyd. Pump":CDF1%(119)=3
2141 RL%(119)=2:MTBF1(119)=4.17:MTTR1(119)=1.6:CDF2%(119)=3
2142 MTBF2(119)=2.77:MTTR2(119)=1.6:NODE2%(119)=119
2145 NODE1%(120)=120: NN$(120)="Cable": CDF1%(120)=5
2146 RL%(120)=1: MTBF1(120)=208.33:MTTR1(120)=.5
2147 CDF2%(120)=5:MTBF2(120)=138.9:MTTR2(120)=.5
```

```
2148 NODE2%(120)=120
2150 NODE1%(123)=123:NN$(123)="Engine On/Off":CDF1%(123)=5
2151 RL%(123)=1:MTBF1(123)=41.67:MTTR1(123)=.1:CDF2%(123)=5
2152 MTBF2(123)=27.77:MTTR2(123)=.1:NODE2%(123)=123
2155 NODE1%(124)=124: NN$(124)="Throttle": CDF1%(124)=5
2156 RL%(124)=1: MTBF1(124)=87.5:MTTR1(124)=1:CDF2%(124)=5
2157 MTBF2(124)=58.33:MTTR2(124)=1:NODE2%(124)=124
2160 NODE1%(125)=125: NN$(125)="Steer.Lever":CDF1%(125)=5
2161 RL%(125)=1:MTBF1(125)=87.5:MTTR1(125)=.8:CDF2%(125)=5
2162 MTBF2(125)=58.33:MTTR2(125)=.8:NODE2%(125)=125
2165 NODE1%(126)=126: NN$(126)="Clutch Lever":CDF1%(126)=5
2166 RL%(126)=1:MTBF1(126)=87.5:MTTR1(126)=.8:CDF2%(126)=5
2167 MTBF2(126)=58.33:MTTR2(126)=.8:NODE2%(126)=126
2200 RETURN
3990 '
3991 '
3992 '                    SCREEN SUBROUTINE
3993 '      INPUT PERIOD UNIT, LENGTH OF PERIOD, LENGTH OF
3994 '         MODEL RUN, AND # OF FAILURE ENVIRONMENTS
3995 '
3996 '      ____________________________________________
4000 ON REQ% GOSUB 4130,4124,4110,4052,4098,4074,4104
4002 RETURN
4022 SG.DF$=LTP$
4024 SG.UP%=1:SG.ROW%=4:SG.COL%=68:SG.LNG%=6
4025 COLOR 15,1  :GOSUB 775
4026 LTP$=SG.REPLY$:RETURN
4028 SG.VV#=LTP%:SG.MSK$="######":GOSUB 1471
4030 SG.ROW%=6:SG.COL%=68:SG.LNG%=6:COLOR 15,1  :GOSUB 770
4032 LTP%=VAL(SG.REPLY$):RETURN
4034 SG.VV#=LMR%:SG.MSK$="######":GOSUB 1471
4036 SG.ROW%=8:SG.COL%=68:SG.LNG%=6:COLOR 15,1  :GOSUB 770
4038 LMR%=VAL(SG.REPLY$):RETURN
4040 SG.VV#=NO.ENVIR%:SG.MSK$="#":GOSUB 1471
4042 SG.ROW%=18:SG.COL%=44:SG.LNG%=1:COLOR 15,1  :GOSUB 770
4044 NO.ENVIR%=VAL(SG.REPLY$):RETURN
4046 SG.DF$=SCR$
4048 SG.UP%=1:SG.ROW%=23:SG.COL%=73:SG.LNG%=1
4049 COLOR 15,1  :GOSUB 775
4050 SCR$=SG.REPLY$:RETURN
4074 GOSUB 1462
4076 GOSUB 4022    'LTP$
4078 IF SG.BTAB%>2 THEN GOTO 4168
4080 GOSUB 4028  SG.BTAB%=1 THEN GOTO 4076    'LTP%
4082 IF SG.BTAB%>2 THEN GOTO 4168
4084 GOSUB 4034  SG.BTAB%=1 THEN GOTO 4080    'LMR%
4086 IF SG.BTAB%>2 THEN GOTO 4168
4088 GOSUB 4040  SG.BTAB%=1 THEN GOTO 4084    'NO.ENVIR%
4090 IF SG.BTAB%>2 THEN GOTO 4168
4092 GOSUB 4046  SG.BTAB%=1 THEN GOTO 4088    'SCR$
4094 IF SG.BTAB%>2 THEN RETURN
4096 RETURN
4110 GOSUB 1462
4112 LOCATE 4,68:COLOR 15,1:PRINT USING "\     \";LTP$;
4114 LOCATE 6,68:COLOR 15,1:PRINT USING "######";LTP%;
4116 LOCATE 8,68:COLOR 15,1:PRINT USING "######";LMR%;
4118 LOCATE 18,44:COLOR 15,1:PRINT USING "#";NO.ENVIR%;
4120 LOCATE 23,73:COLOR 15,1:PRINT USING "!";SCR$;
4122 RETURN
4124 GOSUB 4130
4126 GOSUB 4112
4128 RETURN
4130 LOCATE 25,1:COLOR , 0:PRINT" ";
4131 COLOR ,, 8:CLS:GOSUB 1462
4132 COLOR  3,0
4134 LOCATE 2,1:PRINT "This model will produce period"
                "by period maintenance requirements.";
4136 COLOR 2,0:LOCATE 4,1:PRINT "Will a period be defined"
                "in days, weeks, months, or years?";
4138 COLOR 15,1:LOCATE 4,68:PRINT "      ";
4140 COLOR 2,0:LOCATE 6,1:PRINT "How long will each time"
                "period be (# of dys, wks, mths, yrs)?";
4142 COLOR 15,1:LOCATE 6,68:PRINT "      ";
4144 COLOR 2,0:LOCATE 8,1
4145 PRINT "How many periods should the model run?";
```

```
4146 COLOR 15,1:LOCATE 8,68:PRINT "      ";
4148 COLOR 3,0:LOCATE 12,1
4149 PRINT "Operating environment information:";
4150 LOCATE 14,1:PRINT "It is possible for units of the"
            "same age to have different failure rates";
4152 LOCATE 15,1:PRINT "due to the respective environments"
                          "in which the units operate.";
4154 COLOR 2,0:LOCATE 17,1:PRINT "In how many different"
       "failure environments will your population of";
4156 LOCATE 18,1
4157 PRINT "units operate? (limit of 5 environments)";
4158 COLOR 15,1:LOCATE 18,44:PRINT " ";
4160 COLOR 3,0:LOCATE 23,1:PRINT "Enter one of the"
      "following (r to go back, f to go forward)";
4162 COLOR 15,1:LOCATE 23,73:PRINT " ";
4164 ' GOSUB 25083
4166 COLOR 2,0:RETURN
4168 IF SG.BTAB%<5 THEN RETURN ELSE GOTO 4092
4990 '
4991 '        SCREEN SUBROUTINE
4992 '     INPUT END ITEM COST DATA
4993 '     ________________________
4994 '
4995 '     note: subroutine not listed -
4996 '           This subroutine is very similar to the
4997 '           screen subroutine in lines 4000 - 4168.
4998 '
4999 '
6990 '
6991 '     SUBROUTINE TO DISPLAY LIBRARY OF FAILURE CURVES
6992 '     ___________________________________________________
6993 '
7000 ' KEY OFF
7010 PI=3.141593
7020 CLS
7030 SCREEN 2,0
7040 LOCATE 1,28:  PRINT "LIBRARY  OF  FAILURE  CURVES"
7050 LOCATE 2,1: PRINT "1":  LOCATE 8,1
7051 PRINT "0                t"
7060 LINE (10,10)-(10,60): LINE (10,60)-(90,60)
7070 LINE (13,15)-(90,15),,,&HCCCC
7071 LINE (10,60)-(90,15),,,&HCCCC
7080 LINE (140,10)-(140,60): LINE (140,60)-(220,60)
7090 LINE (143,15)-(220,15),,,&HCCCC
7100 CIRCLE (220,60),80,,PI/2,PI,7/13
7110 LINE (280,10)-(280,60): LINE (280,60)-(360,60)
7120 LINE (283,15)-(360,15),,,&HCCCC
7130 CIRCLE (280,17),80,,3*PI/2,0,8/15
7140 LINE (410,10)-(410,60): LINE (410,60)-(490,60)
7150 LINE (413,15)-(490,15),,,&HCCCC
7160 CIRCLE (410,37),40,,3*PI/2,0,8/15
7161 CIRCLE (490,38),40,,PI/2,PI,7/13
7170 LINE (540,10)-(540,60): LINE (540,60)-(620,60)
7180 LINE (543,15)-(620,15),,,&HCCCC
7190 CIRCLE (580,16),40,,3*PI/2,0,8/15
7191 CIRCLE (580,59),40,,PI/2,PI,7/13
7200 LOCATE 10,1:  PRINT " 1) Uniform      2) Exponential"
        "3) Exp. Increase   4) ~Normal      5) Bathtub"
7205 IF YM=1 THEN RETURN
7210 PRINT:  PRINT "      The above curves are cumulative"
                          "distributions of failure rate"
7212 PRINT "probabilities.  To interpret the curves read"
                       "the x axis as time and the y axis"
7220 PRINT "as the probability of a failure.  So the"
                 "curves represent the probability of"
7222 PRINT "failure of an item during that item's"
                                   "life cycle."
7230 PRINT:  PRINT "If one of these curves appears to"
       "represent the failure characteristics of the"
7240 PRINT "item that you are evaluating, enter either"
                  "1, 2, 3, 4, or 5 on the next screen"
7242 PRINT "when you are asked for the failure rate"
            "curve shape and the model will compute"
7245 PRINT " period by period failure probabilities"
```

```
                                        "automatically"
7250 PRINT "If you have period by period failure data"
             "available for certain items, enter 6 "
7252 PRINT "for failure rate curve shape and 0 for MTBF."
                                "Input failure data later."
7270 PRINT:  INPUT "Enter one of the following (r to go"
               "back one screen, f to go forward) > ",SCR$
7290 RETURN
7490 '
7491 '     SUBROUTINE FOR LISTING FAILURE INPUT EXPLANATION
7492 '     ------------------------------------------------
7493 '
7500 YM=1:  GOSUB 7000
7510 LOCATE 12,17
7511 PRINT "INPUT MAINTENANCE TREE DATA BY FAILURE"
                                          "ENVIRONMENT"
7520 PRINT: PRINT " Enter a 1 or 2 word description of"
                 "the operating environment for units"
7522 PRINT " operating in failure environment #";N%
7525 LOCATE 15,40:  LINE INPUT "> ",NO.ENVIR$(N%)
7530 PRINT: PRINT " You will be asked to enter known"
            "failure rate information about the item"
7540 PRINT " being evaluated.  The information will"
              "be entered by maintenance tree node."
7550 PRINT: PRINT " You must input the following"
                "information when prompted:"
7560 PRINT:  PRINT " Node #                    Node Name"
                            "Failure Rate Curve Shape"
7570 PRINT " Mean Time Between Failures   Mean Time to"
                              "Repair   Repair Level"
7575 LOCATE 24,50:  PRINT "press any key to continue"
7577 A$=INKEY$: IF A$ = "" THEN 7577
7580 CLS:  GOSUB 7000
7590 LOCATE 12,1:  PRINT " Node 6 represents the chance"
                       "of fatal failure of the end item"
7592 PRINT " Data for this node must be provided -"
            "input as prompted, then continue with"
7600 PRINT " the input of information for other bottom"
                                  "of the tree nodes."
7610 LOCATE 16,1:  PRINT " Node # 6        Fatal Failure"
7620 LOCATE 16,50:INPUT "Failure rate curve shape > ",CDF%
7625 PRINT USING "              \   \";LTP$
7630 LOCATE 17,1: INPUT " MTBF > ",MTBF
7632 MTBF = MTBF/LTP%: A%=6: MTTR=0
7640 LOCATE 17,24
7641 PRINT "MTTR and Repair Level not applicable": CNT6%=2
7650 ON N% GOSUB 11000,11100,11200,11300,11400
7660 IF N% > 1 THEN 7720
7670 PRINT: INPUT " Node # >",A%
7680 IF A% = 0 THEN 7900
7690 LOCATE CSRLIN-1,24: INPUT "Node Name >",NN$(A%)
7700 GOSUB 10500
7710 GOTO 7790
7720 FOR A% = 1 TO 126
7730   IF NODE1%(A%) = 0 THEN 7770
7735   IF A% = 6 THEN 7770
7740   PRINT:  PRINT "Node # ";NODE1%(A%)
7750   LOCATE CSRLIN-1,24: PRINT "Node Name:   ";NN$(A%)
7760   GOSUB 10500
7770 NEXT
7780 GOTO 7900
7790 GOTO 7670
7900 RETURN
7991 '
7992 '                    SCREEN SUBROUTINE
7993 '     INPUT TYPE OF POPULATION, INVENTORY LOT SIZING
7994 '            CHOICE, AND # OF REPAIR LEVELS
7995 '     ------------------------------------------------
7996 '
7997 '     note:  subroutine not included
7998 '            This screen subroutine is similar to
7999 '            subroutine in lines 4000 - 4168.
8000 '
8001 '
10490 '
```

```
10491 '        SCREEN SUBROUTINE
10492 '     INPUT NODE FAILURE DATA
10493 '     ------------------------
10494 '
10500 ' input cdf$,mtbf,b,mttr subroutine
10510 LOCATE CSRLIN-1,50
10511 INPUT "Failure rate curve shape > ", CDF%
10515 PRINT USING "                    \     \";LTP$
10520 LOCATE CSRLIN-1,1:  INPUT " MTBF >", MTBF
10525 MTBF = MTBF/LTP%
10565 LOCATE CSRLIN-1,24
10566 PRINT USING "                        \     \";LTP$
10570 LOCATE CSRLIN-1,24:  INPUT "MTTR >",MTTR
10572 IF N%>1 THEN LOCATE CSRLIN-1,50
                PRINT "Repair Level > ";RL%(A%):GOTO 10580
10575 LOCATE CSRLIN-1,50:  INPUT "Repair Level > ",RL%(A%)
10580 ON N% GOSUB 11000,11100,11200,11300,11400
10585 CNT6% = CNT6%+1
10587 IF CNT6% <= 3 THEN 10590
10588 CNT6% = 1:  CLS:  GOSUB 7000
10590 RETURN
10990 '
10991 '     SUBROUTINES FOR SAVING NODE FAILURE DATA
10992 '     ----------------------------------------
10993 '
11000 ' input for no.envir = 1
11010 MTBF1(A%)=MTBF: NODE1%(A%)=A%
11011 CDF1%(A%)=CDF%: MTTR1(A%)=MTTR
11020 RETURN
11050 '
11060 FOR P% = 1 TO 126
11070   MTBFP(P%)=MTBF1(P%): NODEP%(P%)=NODE1%(P%)
11071   CDFP%(P%)=CDF1%(P%):  MTTRP(P%) = MTTR1(P%)
11080 NEXT
11090 RETURN
11100 ' input for no.envir = 2
11110 MTBF2(A%)=MTBF: NODE2%(A%)=A%
11111 CDF2%(A%)=CDF%: MTTR2(A%)=MTTR
11120 RETURN
11150 '
11160 FOR P% = 1 TO 126
11170   MTBFP(P%)=MTBF2(P%): NODEP%(P%)=NODE2%(P%)
11171   CDFP%(P%)=CDF2%(P%):  MTTRP(P%) = MTTR2(P%)
11180 NEXT
11190 RETURN
11200 ' input for no.envir = 3
11210 MTBF3(A%)=MTBF: NODE3%(A%)=A%
11211 CDF3%(A%)=CDF%: MTTR3(A%)=MTTR
11220 RETURN
11250 '
11260 FOR P% = 1 TO 126
11270   MTBFP(P%)=MTBF3(P%): NODEP%(P%)=NODE3%(P%)
11271   CDFP%(P%)=CDF3%(P%):   MTTRP(P%) = MTTR3(P%)
11280 NEXT
11290 RETURN
11300 ' input for no.envir = 4
11310 MTBF4(A%)=MTBF: NODE4%(A%)=A%
11311 CDF4%(A%)=CDF%: MTTR4(A%)=MTTR
11320 RETURN
11350 '
11360 FOR P% = 1 TO 126
11370   MTBFP(P%)=MTBF4(P%): NODEP%(P%)=NODE4%(P%)
11371   CDFP%(P%)=CDF4%(P%): MTTRP(P%) = MTTR4(P%)
11380 NEXT
11390 RETURN
16990 '
16991 '                SCREEN SUBROUTINE
16992 '     INPUT DEMAND, END ITEM RETIREMENT AGE,
16993 '     AND END ITEM BOOK VALUE AT RETIREMENT
16994 '     ------------------------------------
16995 '
16996 '     note:  subroutine not listed -
16997 '            This screen subroutine is similar to
16998 '            the subroutine in lines 4000 - 4168.
16999 '
```

```
17000 '
19990 '
19991 '      SUBROUTINE FOR CONTROLLING THE INPUT OF
19992 '            PERIOD BY PERIOD NODE DATA
19993 '      ____________________________________
19994 '
20000 V%=1
20010 FOR VT% = 1 TO 126
20020   IF NODEP%(VT%) = 0 THEN 20050
20030   AA%(V%) = NODEP%(VT%): V% = V% + 1
20035   IF V% < 16 THEN 20050
20040   REQ%=2:GOSUB 21000
20042   REQ%=6:GOSUB 21000
20045   GOSUB 20500
20047   V%=1
20050 NEXT VT%
20070 RETURN
20500 IF VT% > 126 THEN VT% = 126
20505 FOR A% = 1 TO VT%
20510   IF CDFP%(A%) = 0 THEN 20540
20520   MTBF = MTBFP(A%):  CDF% = CDFP%(A%)
20521   MTTR = MTTRP(A%)
20530   ON N% GOSUB 11000,11100,11200,11300
20540 NEXT
20545 FOR A% = 1 TO 15:  AA%(A%)=0:  NEXT
20550 RETURN
20990 '
20991 '                SCREEN SUBROUTINE
20992 '      INPUT PERIOD BY PERIOD NODE FAILURE DATA
20993 '      ____________________________________
20994 '
20995 '      note:  subroutine not listed -
20996 '             This screen subroutine is similar to
20997 '             the subroutine in lines 4000 - 4168.
20998 '
20999 '
```

```
10 '
20 '      PROGRAM LISTING FOR "INPUT2"
30 '
40 '
50 '
60 '      DISPLAY FAILURE DATA IF PREVIOUS MODEL
70 '            RUN DATA HAS BEEN SAVED
80 '
90 '            DATA CHANGES MAY BE MADE
100 '
110 '      OUTER LOOP ONCE FOR EACH ENVIRONMENT
115 '          INNER LOOP ONCE FOR EACH NODE
120 '      ____________________________________
130 '
200 FOR N% = 1 TO NO.ENVIR%
201 '
202 '      retrieve data
203 '      _____________
204 '
210   ON N% GOSUB 11050,11150,11250,11350
220   FOR P% = 1 TO 126
230     IF CDFP%(P%) <> 6 THEN 320
231     FOR A% = 1 TO 40:  MTBFP(A%)=0:  NEXT
233 '
234 '              gosub screen subroutine
235 '      review and allow changes to failure data
236 '      ________________________________________
237 '
240     REQ%=2:GOSUB 25000
250     REQ%=6:GOSUB 25000
260     FILE9$ = "0" + STR$(N%) + STR$(P%)
270     OPEN "r",#2,FILE9$,4
280     FIELD #2, 4 AS NODE$
290     FOR A% = 1 TO LMR%
300       LSET NODE$ = MKS$(MTBFP(A%)):  PUT #2,A%
310     NEXT A%
315     CLOSE #2
320   NEXT P%
330 NEXT N%
400 '
410 '          GOSUB SCREEN SUBROUTINE
420 '      INPUT PROCUREMENT POLICY CHOICE
430 '      ______________________________
440 '
442 '
443 '                GOSUB SCREEN SUBROUTINE
444 '      INPUT DATA FOR THE PROCUREMENT POLICY CHOSEN
445 '      ____________________________________________
446 '
450 REQ%=2:ON PL% GOSUB 9400,9200,9800,9600,19200,19000,19600,19400,19800
460 REQ%=6:ON PL% GOSUB 9400,9200,9800,9600,19200,19000,19600,19400,19800
470 IF SCR$ = "R" THEN 430
480 ST%(1)=IP%(1):  AMT%(1)=IP%(1)
482 '
483 '        CHAIN TO STOCKABLE ITEM INPUT PROGRAMS
484 '
485 '      IF EOQ INVENTORY LOT SIZING IS BEING USED,
486 '                    CHAIN TO "IV2"
487 '
488 '      IF NO INVENTORY LOT SIZING IS BEING USED,
489 '                    CHAIN TO "IV1"
490 '      ____________________________________________
491 '
495 IF SPWA% = 1 THEN CHAIN "IV1",10,ALL
500 IF SPWA% = 2 THEN CHAIN "IV2",10,ALL
510 IF SPWA% = 3 THEN CHAIN"b:finall",100,ALL
700 '
710 '      END OF MAIN PROGRAM (INPUT2)
720 '
730 '
740 '
750 '
760 '      SCREEN SUBROUTINE DRIVER
770 '      ________________________
780 '
```

```
790 '      note:  subroutine not listed -
800 '             See INTRO program for a listing of
810 '             this subroutine.
820 '
830 '
8000 '
8010 '                SCREEN SUBROUTINE
8020 '       INPUT PROCUREMENT POLICY SELECTION
8030 '       ____________________________________
8040 '
9000 ON REQ% GOSUB 9064,9058,9052,9030,9044,9038,9048
9001 RETURN
9018 SG.VV#=PL%:SG.MSK$="#":GOSUB 1471
9020 SG.ROW%=22:SG.COL%=75:SG.LNG%=1
9021 COLOR 15,1  :GOSUB 770
9022 PL%=VAL(SG.REPLY$):RETURN
9024 SG.DF$=SCR$
9026 SG.UP%=1:SG.ROW%=24:SG.COL%=75:SG.LNG%=1
9027 COLOR 15,1  :GOSUB 775
9028 SCR$=SG.REPLY$:RETURN
9038 GOSUB 1462
9039 GOSUB 9018    'PL%
9040 IF SG.BTAB%>2 THEN GOTO 9128
9041 GOSUB 9024  SG.BTAB%=1 THEN GOTO 9039    'SCR$
9042 IF SG.BTAB%>2 THEN RETURN
9043 RETURN
9052 GOSUB 1462
9053 LOCATE 22,75:COLOR 15,1:PRINT USING "#";PL%;
9055 LOCATE 24,75:COLOR 15,1:PRINT USING "!";SCR$;
9057 RETURN
9058 GOSUB 9064
9060 GOSUB 9053
9062 RETURN
9064 LOCATE 25,1:COLOR , 0:PRINT" ";
9065 COLOR ,, 8:CLS:GOSUB 1462
9066 COLOR 3,0:LOCATE 2,23
9067 PRINT "PROCUREMENT  POLICY  SELECTION"
9068 LOCATE 4,1:PRINT " This will be the policy that"
9069     "you wish to follow in procuring end items.";
9070 COLOR 2,0:LOCATE 6,1:PRINT " 1.....Continuous";
9072 COLOR 10,0:LOCATE 6,19
9073 PRINT "Uniform Procurement"
9074 LOCATE 8,1:PRINT " 2....."
9075 LOCATE 8,8:PRINT "Uniform Procurement";
9078 COLOR 2,0:LOCATE 8,27
9079 PRINT " Until Upper Limit is Reached"
9080 LOCATE 10,1:PRINT " 3.....One Initial"
9081 LOCATE 12,1:PRINT " 4....."
9082 LOCATE 12,8:PRINT "Block Procurement";
9083 COLOR 10,0:LOCATE 10,20:PRINT "Block Procurement"
9084 LOCATE 12,8:PRINT "Block Procurement"
9085 LOCATE 14,19:PRINT "Increasing Procurement";
9090 COLOR 2,0:LOCATE 12,25
9091 PRINT " with Replenishment as Units Fail Fatally"
9092 LOCATE 14,1:PRINT " 5.....Continuous"
9093 LOCATE 14,41:PRINT " Amounts"
9095 LOCATE 16,1:PRINT " 6.....";
9098 COLOR 10,0:LOCATE 16,8:PRINT "Increasing Procurement"
9099 LOCATE 18,19:PRINT "Decreasing Procurement"
9100 LOCATE 20,8:PRINT "Decreasing Procurement";
9102 COLOR 2,0:LOCATE 16,30
9103 PRINT " Until Upper Limit is Reached"
9104 LOCATE 18,1:PRINT " 7.....Continuous"
9105 LOCATE 18,42:PRINT "Amounts"
9106 LOCATE 20,1:PRINT " 8.....";
9112 COLOR 2,0:LOCATE 20,30
9113 PRINT " Until Upper Limit is Reached"
9114 LOCATE 22,1:PRINT " 9.....Other";
9117 COLOR 3,0:LOCATE 22,38
9118 PRINT "Enter either 1,2,3,4,5,6,7,8,or 9"
9119 LOCATE 24,2:PRINT "Enter F to go forward";
9120 COLOR 15,1:LOCATE 22,75:PRINT " "
9121 LOCATE 24,75:PRINT " ";
9126 COLOR 2,0:RETURN
9128 IF SG.BTAB% <5 THEN RETURN ELSE GOTO 9041
```

```
9175 '
9176 '                    SCREEN SUBROUTINE
9177 '            INPUT DATA FOR PROCUREMENT POLICY
9188 '     UNIFORM PROCUREMENT UNTIL UPPER LIMIT IS REACHED
9189 '     ____________________________________________________
9190 '
9191 '     note:  subroutine not listed -
9192 '            This screen subroutine is similar to
9193 '            the subroutine in lines 9000 - 9128.
9194 '
9195 '
9370 '
9371 '             SCREEN SUBROUTINE
9372 '     INPUT DATA FOR PROCUREMENT POLICY
9373 '       CONTINUOUS UNIFORM PROCUREMENT
9374 '     ____________________________________
9375 '
9376 '     note:  subroutine not listed -
9378 '            This screen subroutine is similar to
9379 '            the subroutine in lines 9000 - 9128.
9380 '
9381 '
9580 '
9581 '                SCREEN SUBROUTINE
9582 '      INPUT DATA FOR PROCUREMENT POLICY
9583 '     BLOCK PROCUREMENT WITH REPLENISHMENT
9584 '     ____________________________________________
9585 '
9586 '     note:  subroutine not listed -
9587 '            This screen subroutine is similar to
9588 '            the subroutine in lines 9000 - 9128.
9589 '
9590 '
9780 '
9781 '            SCREEN SUBROUTINE
9782 '     INPUT DATA FOR PROCUREMENT POLICY
9783 '          ONE BLOCK PROCUREMENT
9784 '     ____________________________________
9785 '
9786 '     note:  subroutine not listed -
9787 '            This screen subroutine is similar to
9788 '            the subroutine in lines 9000 - 9128.
9789 '
9790 '
10980 '
10981 '      SUBROUTINES FOR RETRIEVING NODE FAILURE DATA
10982 '      ____________________________________________________
10983 '
11050 FOR P% = 1 TO 126
11070   MTBFP(P%)=MTBF1(P%): NODEP%(P%)=NODE1%(P%)
11071   CDFP%(P%)=CDF1%(P%):  MTTRP(P%) = MTTR1(P%)
11080 NEXT
11090 RETURN
11150 FOR P% = 1 TO 126
11170   MTBFP(P%)=MTBF2(P%): NODEP%(P%)=NODE2%(P%)
11171   CDFP%(P%)=CDF2%(P%):  MTTRP(P%) = MTTR2(P%)
11180 NEXT
11190 RETURN
11250 FOR P% = 1 TO 126
11270   MTBFP(P%)=MTBF3(P%): NODEP%(P%)=NODE3%(P%)
11271   CDFP%(P%)=CDF3%(P%):   MTTRP(P%) = MTTR3(P%)
11280 NEXT
11290 RETURN
11350 FOR P% = 1 TO 126
11370   MTBFP(P%)=MTBF4(P%): NODEP%(P%)=NODE4%(P%)
11371   CDFP%(P%)=CDF4%(P%): MTTRP(P%) = MTTR4(P%)
11380 NEXT
11390 RETURN
18980 '
18981 '                  SCREEN SUBROUTINE
18982 '          INPUT DATA FOR PROCUREMENT POLICY
18983 '       INCREASING PROCUREMENT TO UPPER LIMIT
18984 '       ____________________________________________
18985 '
18986 '       note:  subroutine not listed -
```

```
18987 '            This screen subroutine is similar to
18988 '            the subroutine in lines 9000 - 9128.
18989 '
18990 '
19180 '
19181 '              SCREEN SUBROUTINE
19182 '      INPUT DATA FOR PROCUREMENT POLICY
19183 '       INCREASING PROCUREMENT AMOUNTS
19184 '
19185 '      ____________________________________
19186 '      note:  subroutine not listed -
19187 '             This screen subroutine is similar to
19188 '             the subroutine in lines 9000 - 9128.
19189 '
19190 '
19380 '
19381 '                SCREEN SUBROUTINE
19382 '        INPUT DATA FOR PROCUREMENT POLICY
19383 '      DECREASING PROCUREMENT TO UPPER LIMIT
19384 '
19385 '      ____________________________________
19386 '      note:  subroutine not listed -
19387 '             This screen subroutine is similar to
19388 '             the subroutine in lines 9000 - 9128.
19389 '
19390 '
19580 '
19581 '              SCREEN SUBROUTINE
19582 '      INPUT DATA FOR PROCUREMENT POLICY
19583 '       DECREASING PROCUREMENT AMOUNTS
19584 '
19585 '      ____________________________________
19586 '      note:  subroutine not listed -
19587 '             This screen subroutine is similar to
19588 '             the subroutine listed in lines 9000 - 9128.
19589 '
19590 '
19780 '
19781 '              SCREEN SUBROUTINE
19782 '      INPUT DATA FOR PROCUREMENT POLICY
19783 '          OTHER PROCUREMENT POLICY
19784 '
19785 '      ____________________________________
19786 '      note:  subroutine not listed -
19787 '             This screen subroutine is similar to
19788 '             the subroutine in lines 9000 - 9128.
19789 '
19790 '
20000 '
20001 '                    SCREEN SUBROUTINE
20002 '      DISPLAY AND ALLOW CHANGES TO NODE FAILURE DATA
20003 '
20004 '      ____________________________________________
20005 '      note:  subroutine not listed -
20006 '             This screen subroutine is similar to
20007 '             the subroutine in lines 9000 - 9128.
20008 '
20009 '
```

```
1 '
2 '       PROGRAM LISTING FOR "IV1"
3 '
4 '
5 '
6 '            GOSUB IF THE EXAMPLE IS BEING RUN OR
7 '       IF DATA FROM A PREVIOUS MODEL RUN IS BEING USED
8 '       ____________________________________________
9 '
10 IF UB% = 1 OR UB% = 2 THEN GOSUB 400
12 '
13 '      INPUT STOCKABLE ITEM DATA NECESSARY TO CALCULATE
14 '         SPARE PART AND WAREHOUSE REQUIREMENTS ONLY
15 '
16 '                   LOOP ONCE FOR EACH NODE
17 '        ____________________________________________
18 '
20 FOR V% = 1 TO 126
30   IF V% = 6 THEN 170
40   IF NODE1%(V%) = 0 THEN 170
42 '
43 '       gosub screen subroutine
44 '      input stockable item data
45 '      _________________________
46 '
50   REQ%=2: GOSUB 21000
60   REQ%=6: GOSUB 21000
61 '
62 '      store input data in disk files
63 '      ______________________________
64 '
65   FILE4$ = "node" + STR$(V%)
70   OPEN "o",#1,FILE4$
90   FOR N% = 1 TO 10
95     IF QY(N%) = 0 THEN 110
100    PRINT#1, I$(N%);",";PN$(N%);",";QY(N%);SR(N%);PR(N%)
110  NEXT N%
160  CLOSE 1
161 '
162 '      initialize variables
163 '      ____________________
164 '
165 NEXT VT%
166 FOR VT% = 1 TO 10
167   I$(VT%)="":  PN$(VT%)="": QY(VT%)=0
168   SR(VT%)=0: PR(VT%)=0
169 NEXT VT%
170 NEXT V%
180 '
181 '       CHAIN TO PROGRAM "MAIN"
182 '      BEGIN MODEL CALCULATIONS
183 '      ________________________
184 '
200 CHAIN "MAIN",100,ALL
210 '
211 '      END OF MAIN PROGRAM (IV1)
212 '      _________________________
213 '
380 '
381 '       SUBROUTINE TO DISPLAY AND ALLOW CHANGES TO
382 '      NODE FAILURE DATA FOR EITHER EXAMPLE PROBLEM
383 '            OR DATA FROM PREVIOUS MODEL RUN
384 '      ____________________________________________
385 '
400 FOR V% = 1 TO 126
410   IF V% = 6 THEN 580
420   IF NODE1%(V%) = 0 THEN 580
430   FILE4$ = "node" + STR$(V%)
440   OPEN "i",#1,FILE4$
450   FOR N% = 1 TO 10
455     IF EOF(1) THEN 470
460     INPUT#1, I$(N%),PN$(N%),QY(N%),SR(N%),PR(N%)
470   NEXT N%
480   REQ%=2:GOSUB 21000
490   REQ%=6:GOSUB 21000
```

```
500   FOR N% = 1 TO 10
510    IF QY(N%) = 0 THEN 530
520    PRINT#1, I$(N%);",";PN$(N%);",";QY(N%);SR(N%);PR(N%)
530   NEXT N%
540   CLOSE 1
550   FOR VT% = 1 TO 10
560     I$(VT%)="":  PN$(VT%)="": QY(VT%)=0: SR(VT%)=0: PR(VT%)=0
570   NEXT VT%
580 NEXT V%
590 RETURN 200
680 '
681 '      SCREEN SUBROUTINE DRIVER
682 '      ________________________
683 '
684 '      note:  subroutine not listed -
685 '             This subroutine is listed in lines
686 '             700 - 1486 in program INTRO.
687 '
688 '
20900 '
20901 '                    SCREEN SUBROUTINE
20902 '      INPUT, REVIEW, OR CHANGE STOCKABLE ITEM DATA
20903 '      ____________________________________________
20904 '
20905 '
21000 ON REQ% GOSUB 21784 , 21778 , 21652 , 21390,
                           21644 , 21518 , 21648
21002 RETURN
21018 SG.DF$=NN$(V%)
21020 SG.ROW%=6:SG.COL%=16:SG.LNG%=14:COLOR 10,0 :GOSUB 775
21022 NN$(V%)=SG.REPLY$:RETURN
21024 SG.DF$=I$(1)
21026 SG.ROW%=10:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21028 I$(1)=SG.REPLY$:RETURN
21030 SG.DF$=PN$(1)
21032 SG.ROW%=10:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21034 PN$(1)=SG.REPLY$:RETURN
21036 SG.VV#=QY(1):SG.MSK$="#######":GOSUB 1471
21038 SG.ROW%=10:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21040 QY(1)=VAL(SG.REPLY$):RETURN
21042 SG.VV#=SR(1):SG.MSK$="#####.##":GOSUB 1471
21044 SG.ROW%=10:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21046 SR(1)=VAL(SG.REPLY$):RETURN
21048 SG.VV#=PR(1):SG.MSK$="######.##":GOSUB 1471
21050 SG.ROW%=10:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21052 PR(1)=VAL(SG.REPLY$):RETURN
21054 SG.DF$=I$(2)
21056 SG.ROW%=11:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21058 I$(2)=SG.REPLY$:RETURN
21060 SG.DF$=PN$(2)
21062 SG.ROW%=11:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21064 PN$(2)=SG.REPLY$:RETURN
21066 SG.VV#=QY(2):SG.MSK$="#######":GOSUB 1471
21068 SG.ROW%=11:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21070 QY(2)=VAL(SG.REPLY$):RETURN
21072 SG.VV#=SR(2):SG.MSK$="#####.##":GOSUB 1471
21074 SG.ROW%=11:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21076 SR(2)=VAL(SG.REPLY$):RETURN
21078 SG.VV#=PR(2):SG.MSK$="######.##":GOSUB 1471
21080 SG.ROW%=11:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21082 PR(2)=VAL(SG.REPLY$):RETURN
21084 SG.DF$=I$(3)
21086 SG.ROW%=12:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21088 I$(3)=SG.REPLY$:RETURN
21090 SG.DF$=PN$(3)
21092 SG.ROW%=12:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21094 PN$(3)=SG.REPLY$:RETURN
21096 SG.VV#=QY(3):SG.MSK$="#######":GOSUB 1471
21098 SG.ROW%=12:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21100 QY(3)=VAL(SG.REPLY$):RETURN
21102 SG.VV#=SR(3):SG.MSK$="#####.##":GOSUB 1471
21104 SG.ROW%=12:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21106 SR(3)=VAL(SG.REPLY$):RETURN
21108 SG.VV#=PR(3):SG.MSK$="######.##":GOSUB 1471
21110 SG.ROW%=12:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
```

```
21112 PR(3)=VAL(SG.REPLY$):RETURN
21114 SG.DF$=I$(4)
21116 SG.ROW%=13:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21118 I$(4)=SG.REPLY$:RETURN
21120 SG.DF$=PN$(4)
21122 SG.ROW%=13:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21124 PN$(4)=SG.REPLY$:RETURN
21126 SG.VV#=QY(4):SG.MSK$="#######":GOSUB 1471
21128 SG.ROW%=13:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21130 QY(4)=VAL(SG.REPLY$):RETURN
21132 SG.VV#=SR(4):SG.MSK$="#####.##":GOSUB 1471
21134 SG.ROW%=13:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21136 SR(4)=VAL(SG.REPLY$):RETURN
21138 SG.VV#=PR(4):SG.MSK$="######.##":GOSUB 1471
21140 SG.ROW%=13:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21142 PR(4)=VAL(SG.REPLY$):RETURN
21144 SG.DF$=I$(5)
21146 SG.ROW%=14:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21148 I$(5)=SG.REPLY$:RETURN
21150 SG.DF$=PN$(5)
21152 SG.ROW%=14:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21154 PN$(5)=SG.REPLY$:RETURN
21156 SG.VV#=QY(5):SG.MSK$="#######":GOSUB 1471
21158 SG.ROW%=14:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21160 QY(5)=VAL(SG.REPLY$):RETURN
21162 SG.VV#=SR(5):SG.MSK$="#####.##":GOSUB 1471
21164 SG.ROW%=14:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21166 SR(5)=VAL(SG.REPLY$):RETURN
21168 SG.VV#=PR(5):SG.MSK$="######.##":GOSUB 1471
21170 SG.ROW%=14:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21172 PR(5)=VAL(SG.REPLY$):RETURN
21174 SG.DF$=I$(6)
21176 SG.ROW%=15:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21178 I$(6)=SG.REPLY$:RETURN
21180 SG.DF$=PN$(6)
21182 SG.ROW%=15:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21184 PN$(6)=SG.REPLY$:RETURN
21186 SG.VV#=QY(6):SG.MSK$="#######":GOSUB 1471
21188 SG.ROW%=15:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21190 QY(6)=VAL(SG.REPLY$):RETURN
21192 SG.VV#=SR(6):SG.MSK$="#####.##":GOSUB 1471
21194 SG.ROW%=15:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21196 SR(6)=VAL(SG.REPLY$):RETURN
21198 SG.VV#=PR(6):SG.MSK$="######.##":GOSUB 1471
21200 SG.ROW%=15:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21202 PR(6)=VAL(SG.REPLY$):RETURN
21204 SG.DF$=I$(7)
21206 SG.ROW%=16:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21208 I$(7)=SG.REPLY$:RETURN
21210 SG.DF$=PN$(7)
21212 SG.ROW%=16:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21214 PN$(7)=SG.REPLY$:RETURN
21216 SG.VV#=QY(7):SG.MSK$="#######":GOSUB 1471
21218 SG.ROW%=16:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21220 QY(7)=VAL(SG.REPLY$):RETURN
21222 SG.VV#=SR(7):SG.MSK$="#####.##":GOSUB 1471
21224 SG.ROW%=16:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21226 SR(7)=VAL(SG.REPLY$):RETURN
21228 SG.VV#=PR(7):SG.MSK$="######.##":GOSUB 1471
21230 SG.ROW%=16:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21232 PR(7)=VAL(SG.REPLY$):RETURN
21234 SG.DF$=I$(8)
21236 SG.ROW%=17:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21238 I$(8)=SG.REPLY$:RETURN
21240 SG.DF$=PN$(8)
21242 SG.ROW%=17:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21244 PN$(8)=SG.REPLY$:RETURN
21246 SG.VV#=QY(8):SG.MSK$="#######":GOSUB 1471
21248 SG.ROW%=17:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21250 QY(8)=VAL(SG.REPLY$):RETURN
21252 SG.VV#=SR(8):SG.MSK$="#####.##":GOSUB 1471
21254 SG.ROW%=17:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21256 SR(8)=VAL(SG.REPLY$):RETURN
21258 SG.VV#=PR(8):SG.MSK$="######.##":GOSUB 1471
21260 SG.ROW%=17:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
```

```
21262 PR(8)=VAL(SG.REPLY$):RETURN
21264 SG.DF$=I$(9)
21266 SG.ROW%=18:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21268 I$(9)=SG.REPLY$:RETURN
21270 SG.DF$=PN$(9)
21272 SG.ROW%=18:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21274 PN$(9)=SG.REPLY$:RETURN
21276 SG.VV#=QY(9):SG.MSK$="#######":GOSUB 1471
21278 SG.ROW%=18:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21280 QY(9)=VAL(SG.REPLY$):RETURN
21282 SG.VV#=SR(9):SG.MSK$="#####.##":GOSUB 1471
21284 SG.ROW%=18:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21286 SR(9)=VAL(SG.REPLY$):RETURN
21288 SG.VV#=PR(9):SG.MSK$="######.##":GOSUB 1471
21290 SG.ROW%=18:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21292 PR(9)=VAL(SG.REPLY$):RETURN
21294 SG.DF$=I$(10)
21296 SG.ROW%=19:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
21298 I$(10)=SG.REPLY$:RETURN
21300 SG.DF$=PN$(10)
21302 SG.ROW%=19:SG.COL%=24:SG.LNG%=7:COLOR 6,0  :GOSUB 775
21304 PN$(10)=SG.REPLY$:RETURN
21306 SG.VV#=QY(10):SG.MSK$="#######":GOSUB 1471
21308 SG.ROW%=19:SG.COL%=34:SG.LNG%=7:COLOR 6,0  :GOSUB 770
21310 QY(10)=VAL(SG.REPLY$):RETURN
21312 SG.VV#=SR(10):SG.MSK$="#####.##":GOSUB 1471
21314 SG.ROW%=19:SG.COL%=52:SG.LNG%=8:COLOR 6,0  :GOSUB 770
21316 SR(10)=VAL(SG.REPLY$):RETURN
21318 SG.VV#=PR(10):SG.MSK$="######.##":GOSUB 1471
21320 SG.ROW%=19:SG.COL%=71:SG.LNG%=9:COLOR 6,0  :GOSUB 770
21322 PR(10)=VAL(SG.REPLY$):RETURN
21384 SG.DF$=SCR$
21386 SG.UP%=1:SG.ROW%=23:SG.COL%=56:SG.LNG%=1
21387 COLOR 15,1  :GOSUB 775
21388 SCR$=SG.REPLY$:RETURN
21518 GOSUB 1462
21519 GOSUB 21018    'NN$(V%)
21520 IF SG.BTAB%>2 THEN GOTO 21828
21521 GOSUB 21024  SG.BTAB%=1 THEN GOTO 21519    'I$(1)
21522 IF SG.BTAB%>2 THEN GOTO 21828
21523 GOSUB 21030  SG.BTAB%=1 THEN GOTO 21521    'PN$(1)
21524 IF SG.BTAB%>2 THEN GOTO 21828
21525 GOSUB 21036  SG.BTAB%=1 THEN GOTO 21523    'QY(1)
21526 IF SG.BTAB%>2 THEN GOTO 21828
21527 GOSUB 21042  SG.BTAB%=1 THEN GOTO 21525    'SR(1)
21528 IF SG.BTAB%>2 THEN GOTO 21828
21529 GOSUB 21048  SG.BTAB%=1 THEN GOTO 21527    'PR(1)
21530 IF SG.BTAB%>2 THEN GOTO 21828
21531 GOSUB 21054  SG.BTAB%=1 THEN GOTO 21529    'I$(2)
21532 IF SG.BTAB%>2 THEN GOTO 21828
21533 GOSUB 21060  SG.BTAB%=1 THEN GOTO 21531    'PN$(2)
21534 IF SG.BTAB%>2 THEN GOTO 21828
21535 GOSUB 21066  SG.BTAB%=1 THEN GOTO 21533    'QY(2)
21536 IF SG.BTAB%>2 THEN GOTO 21828
21537 GOSUB 21072  SG.BTAB%=1 THEN GOTO 21535    'SR(2)
21538 IF SG.BTAB%>2 THEN GOTO 21828
21539 GOSUB 21078  SG.BTAB%=1 THEN GOTO 21537    'PR(2)
21540 IF SG.BTAB%>2 THEN GOTO 21828
21541 GOSUB 21084  SG.BTAB%=1 THEN GOTO 21539    'I$(3)
21542 IF SG.BTAB%>2 THEN GOTO 21828
21543 GOSUB 21090  SG.BTAB%=1 THEN GOTO 21541    'PN$(3)
21544 IF SG.BTAB%>2 THEN GOTO 21828
21545 GOSUB 21096  SG.BTAB%=1 THEN GOTO 21543    'QY(3)
21546 IF SG.BTAB%>2 THEN GOTO 21828
21547 GOSUB 21102  SG.BTAB%=1 THEN GOTO 21545    'SR(3)
21548 IF SG.BTAB%>2 THEN GOTO 21828
21549 GOSUB 21108  SG.BTAB%=1 THEN GOTO 21547    'PR(3)
21550 IF SG.BTAB%>2 THEN GOTO 21828
21551 GOSUB 21114  SG.BTAB%=1 THEN GOTO 21549    'I$(4)
21552 IF SG.BTAB%>2 THEN GOTO 21828
21553 GOSUB 21120  SG.BTAB%=1 THEN GOTO 21551    'PN$(4)
21554 IF SG.BTAB%>2 THEN GOTO 21828
21555 GOSUB 21126  SG.BTAB%=1 THEN GOTO 21553    'QY(4)
21556 IF SG.BTAB%>2 THEN GOTO 21828
21557 GOSUB 21132  SG.BTAB%=1 THEN GOTO 21555    'SR(4)
```

```
21558 IF SG.BTAB%>2 THEN GOTO 21828
21559 GOSUB 21138  SG.BTAB%=1 THEN GOTO 21557    'PR(4)
21560 IF SG.BTAB%>2 THEN GOTO 21828
21561 GOSUB 21144  SG.BTAB%=1 THEN GOTO 21559    'I$(5)
21562 IF SG.BTAB%>2 THEN GOTO 21828
21563 GOSUB 21150  SG.BTAB%=1 THEN GOTO 21561    'PN$(5)
21564 IF SG.BTAB%>2 THEN GOTO 21828
21565 GOSUB 21156  SG.BTAB%=1 THEN GOTO 21563    'QY(5)
21566 IF SG.BTAB%>2 THEN GOTO 21828
21567 GOSUB 21162  SG.BTAB%=1 THEN GOTO 21565    'SR(5)
21568 IF SG.BTAB%>2 THEN GOTO 21828
21569 GOSUB 21168  SG.BTAB%=1 THEN GOTO 21567    'PR(5)
21570 IF SG.BTAB%>2 THEN GOTO 21828
21571 GOSUB 21174  SG.BTAB%=1 THEN GOTO 21569    'I$(6)
21572 IF SG.BTAB%>2 THEN GOTO 21828
21573 GOSUB 21180  SG.BTAB%=1 THEN GOTO 21571    'PN$(6)
21574 IF SG.BTAB%>2 THEN GOTO 21828
21575 GOSUB 21186  SG.BTAB%=1 THEN GOTO 21573    'QY(6)
21576 IF SG.BTAB%>2 THEN GOTO 21828
21577 GOSUB 21192  SG.BTAB%=1 THEN GOTO 21575    'SR(6)
21578 IF SG.BTAB%>2 THEN GOTO 21828
21579 GOSUB 21198  SG.BTAB%=1 THEN GOTO 21577    'PR(6)
21580 IF SG.BTAB%>2 THEN GOTO 21828
21581 GOSUB 21204  SG.BTAB%=1 THEN GOTO 21579    'I$(7)
21582 IF SG.BTAB%>2 THEN GOTO 21828
21583 GOSUB 21210  SG.BTAB%=1 THEN GOTO 21581    'PN$(7)
21584 IF SG.BTAB%>2 THEN GOTO 21828
21585 GOSUB 21216  SG.BTAB%=1 THEN GOTO 21583    'QY(7)
21586 IF SG.BTAB%>2 THEN GOTO 21828
21587 GOSUB 21222  SG.BTAB%=1 THEN GOTO 21585    'SR(7)
21588 IF SG.BTAB%>2 THEN GOTO 21828
21589 GOSUB 21228  SG.BTAB%=1 THEN GOTO 21587    'PR(7)
21590 IF SG.BTAB%>2 THEN GOTO 21828
21591 GOSUB 21234  SG.BTAB%=1 THEN GOTO 21589    'I$(8)
21592 IF SG.BTAB%>2 THEN GOTO 21828
21593 GOSUB 21240  SG.BTAB%=1 THEN GOTO 21591    'PN$(8)
21594 IF SG.BTAB%>2 THEN GOTO 21828
21595 GOSUB 21246  SG.BTAB%=1 THEN GOTO 21593    'QY(8)
21596 IF SG.BTAB%>2 THEN GOTO 21828
21597 GOSUB 21252  SG.BTAB%=1 THEN GOTO 21595    'SR(8)
21598 IF SG.BTAB%>2 THEN GOTO 21828
21599 GOSUB 21258  SG.BTAB%=1 THEN GOTO 21597    'PR(8)
21600 IF SG.BTAB%>2 THEN GOTO 21828
21601 GOSUB 21264  SG.BTAB%=1 THEN GOTO 21599    'I$(9)
21602 IF SG.BTAB%>2 THEN GOTO 21828
21603 GOSUB 21270  SG.BTAB%=1 THEN GOTO 21601    'PN$(9)
21604 IF SG.BTAB%>2 THEN GOTO 21828
21605 GOSUB 21276  SG.BTAB%=1 THEN GOTO 21603    'QY(9)
21606 IF SG.BTAB%>2 THEN GOTO 21828
21607 GOSUB 21282  SG.BTAB%=1 THEN GOTO 21605    'SR(9)
21608 IF SG.BTAB%>2 THEN GOTO 21828
21609 GOSUB 21288  SG.BTAB%=1 THEN GOTO 21607    'PR(9)
21610 IF SG.BTAB%>2 THEN GOTO 21828
21611 GOSUB 21294  SG.BTAB%=1 THEN GOTO 21609    'I$(10)
21612 IF SG.BTAB%>2 THEN GOTO 21828
21613 GOSUB 21300  SG.BTAB%=1 THEN GOTO 21611    'PN$(10)
21614 IF SG.BTAB%>2 THEN GOTO 21828
21615 GOSUB 21306  SG.BTAB%=1 THEN GOTO 21613    'QY(10)
21616 IF SG.BTAB%>2 THEN GOTO 21828
21617 GOSUB 21312  SG.BTAB%=1 THEN GOTO 21615    'SR(10)
21618 IF SG.BTAB%>2 THEN GOTO 21828
21619 GOSUB 21318  SG.BTAB%=1 THEN GOTO 21617    'PR(10)
21620 IF SG.BTAB%>2 THEN GOTO 21828
21641 GOSUB 21384  SG.BTAB%=1 THEN GOTO 21639    'SCR$
21642 IF SG.BTAB%>2 THEN RETURN
21643 RETURN
21652 GOSUB 1462
21653 LOCATE 6,16:COLOR 10,0
21654 PRINT USING "\              \";NN$(V%):COLOR 6,0
21714 FOR Q% = 1 TO 10
21715   LOCATE Q%+9,5:PRINT USING "\              \";I$(Q%);
21717   LOCATE Q%+9,24:PRINT USING "\      \";PN$(Q%);
21719   LOCATE Q%+9,34:PRINT USING "#######";QY(Q%);
21721   LOCATE Q%+9,52:PRINT USING "#####.##";SR(Q%);
21723   LOCATE Q%+9,71:PRINT USING "######.##";PR(Q%);
```

```
21724 NEXT
21775 LOCATE 23,56:COLOR 15,1:PRINT USING "!";SCR$;
21777 RETURN
21778 GOSUB 21784
21780 GOSUB 21653
21782 RETURN
21783 LOCATE 25,1:COLOR , 0:PRINT" ";
21784 COLOR ,, 8:CLS:GOSUB 1462
21785 COLOR  2,0
21786 COLOR 3,0:LOCATE 2,1:PRINT "    LIST STOCKABLE ITEMS"
      "SO THAT SPARE PARTS REQUIREMENTS MAY BE CALCULATED";
21788 COLOR 3,0:LOCATE 4,10:PRINT "When needed, press the"
                                "F9 key to advance to bottom";
21790 COLOR 3,0:LOCATE 4,61:PRINT "of screen.";
21792 COLOR 2,0:LOCATE 6,1:PRINT "    Node name:";
21794 LOCATE 8,1:PRINT "    Stockable item      Part#"
                "Quantity   Space Requirement(cu.ft.)";
21796 COLOR 2,0:LOCATE 8,72:PRINT " Price";
21798 FOR Q% = 1 TO 10
21800   LOCATE Q%+9,1:PRINT USING "##.";Q%
21802 NEXT
21822 COLOR 3,0:LOCATE 23,1:PRINT " Enter F to continue";
21824 COLOR 15,1:LOCATE 23,56:PRINT " ";
21826 COLOR 2,0:RETURN
21828 IF SG.BTAB% < 5 THEN RETURN ELSE GOTO 21641
```

```
1 '
2 '       PROGRAM LISTING FOR "IV2"
3 '
4 '
5 '
6 '      GOSUB TO REVIEW OR MAKE CHANGES TO STOCKABLE ITEM
7 '       DATA IF EXAMPLE IS BEING RUN OR IF DATA FROM A
8 '             PREVIOUS MODEL RUN HAS BEEN SAVED
9 '       ____________________________________________________
10 '
12 IF UB%=1 OR UB%=2 THEN GOSUB 400
14 '
15 '          INPUT STOCKABLE ITEM DATA FOR CALCULATING
16 '      SPARE PARTS AND WAREHOUSING REQUIREMENTS USING
17 '                      EOQ LOT SIZING
18 '
19 '                 LOOP ONCE FOR EACH NODE
20 '       ____________________________________________________
21 '
25 FOR V% = 1 TO 126
30   IF V% = 6 THEN 170
40   IF NODE1%(V%) = 0 THEN 170
42 '
43 '       gosub screen subroutine
44 '             input data
45 '       ____________________________
46 '
50   REQ%=2:GOSUB 23000
60   REQ%=6:GOSUB 23000
61 '
62 '       store data in disk files
63 '       ____________________________
64 '
65   FILE4$ = "node" + STR$(V%)
70   OPEN "o",#1,FILE4$
120  FOR N% = 1 TO 10
130    IF QY(N%) = 0 THEN 150
140    PRINT#1, I$(N%);",";PN$(N%);",";QY(N%);
               SR(N%);PR(N%);CP(N%);CH(N%);LT(N%)
150  NEXT N%
160  CLOSE 1
162 '
163 '       initialize variables
164 '       ____________________
165 '
166 FOR VT% = 1 TO 10
167   I$(VT%)="": PN$(VT%)="": QY(VT%)=0: SR(VT%)=0
168   PR(VT%)=0: CP(VT%)=0: CH(VT%)=0: LT(VT%)=0
169 NEXT VT%
170 NEXT V%
180 '
181 '       CHAIN TO PROGRAM "MAIN"
182 '        BEGIN CALCULATIONS
183 '       ____________________
184 '
200 CHAIN "MAIN",100,ALL
300 '
310 '
320 '       END OF MAIN PROGRAM (IV2)
330 '
340 '
350 '       SUBROUTINE FOR DISPLAYING AND CHANGING STOCKABLE
360 '         ITEM DATA FROM EXAMPLE OR PREVIOUS MODEL RUN
370 '       ____________________________________________________
380 '
400 FOR V% = 1 TO 126
410  IF V% = 6 THEN 580
420  IF NODE1%(V%) = 0 THEN 580
430  FILE4$ = "node" + STR$(V%)
440  OPEN "i",#1,FILE4$
450  FOR N% = 1 TO 10
460    IF EOF(1) THEN 470
465    INPUT#1, I$(N%),PN$(N%),QY(N%),SR(N%),
               PR(N%),CP(N%),CH(N%),LT(N%)
470  NEXT N%
```

```
475 CLOSE 1
480  REQ%=2:GOSUB 23000
490  REQ%=6:GOSUB 23000
495  OPEN "o",#1,FILE4$
500  FOR N% = 1 TO 10
510    IF QY(N%) = 0 THEN 530
520    PRINT#1, I$(N%);",";PN$(N%);",";QY(N%);
                SR(N%);PR(N%);CP(N%);CH(N%);LT(N%)
530  NEXT N%
540  CLOSE 1
550  FOR VT% = 1 TO 10
560    I$(VT%)="": PN$(VT%)="": QY(VT%)=0: SR(VT%)=0
561    PR(VT%)=0: CP(VT%)=0: CH(VT%)=0: LT(VT%)=0
570  NEXT VT%
580 NEXT V%
590 RETURN 200
660 '
661 '      SCREEN SUBROUTINE DRIVER
662 '      ------------------------
663 '
664 '      note:  subroutine not listed -
665 '             See screen subroutine driver listing in
666 '             lines 700 - 1486 of program INTRO.
667 '
668 '
22000 '
22001 '                    SCREEN SUBROUTINE
22002 '      INPUT STOCKABLE ITEM DATA FOR EOQ LOT SIZING
22003 '      ---------------------------------------------
22004 '
23000 ON REQ% GOSUB 24216 , 24210 , 24012 , 23606,
                           24004 , 23806 , 24008
23002 RETURN
23018 SG.DF$=NN$(V%)
23020 SG.ROW%=6:SG.COL%=16:SG.LNG%=14:COLOR 10,0 :GOSUB 775
23022 NN$(V%)=SG.REPLY$:RETURN
23024 SG.DF$=I$(1)
23026 SG.ROW%=11:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23028 I$(1)=SG.REPLY$:RETURN
23030 SG.DF$=PN$(1)
23032 SG.ROW%=11:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23034 PN$(1)=SG.REPLY$:RETURN
23036 SG.VV#=QY(1):SG.MSK$="#######":GOSUB 1471
23038 SG.ROW%=11:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23040 QY(1)=VAL(SG.REPLY$):RETURN
23042 SG.VV#=SR(1):SG.MSK$="#####.##":GOSUB 1471
23044 SG.ROW%=11:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23046 SR(1)=VAL(SG.REPLY$):RETURN
23048 SG.VV#=PR(1):SG.MSK$="#####.##":GOSUB 1471
23050 SG.ROW%=11:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23052 PR(1)=VAL(SG.REPLY$):RETURN
23054 SG.VV#=CP(1):SG.MSK$="####.##":GOSUB 1471
23056 SG.ROW%=11:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23058 CP(1)=VAL(SG.REPLY$):RETURN
23060 SG.VV#=CH(1):SG.MSK$="####.##":GOSUB 1471
23062 SG.ROW%=11:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23064 CH(1)=VAL(SG.REPLY$):RETURN
23066 SG.VV#=LT(1):SG.MSK$="##.#":GOSUB 1471
23068 SG.ROW%=11:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23070 LT(1)=VAL(SG.REPLY$):RETURN
23072 SG.DF$=I$(2)
23074 SG.ROW%=12:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23076 I$(2)=SG.REPLY$:RETURN
23078 SG.DF$=PN$(2)
23080 SG.ROW%=12:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23082 PN$(2)=SG.REPLY$:RETURN
23084 SG.VV#=QY(2):SG.MSK$="#######":GOSUB 1471
23086 SG.ROW%=12:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23088 QY(2)=VAL(SG.REPLY$):RETURN
23090 SG.VV#=SR(2):SG.MSK$="#####.##":GOSUB 1471
23092 SG.ROW%=12:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23094 SR(2)=VAL(SG.REPLY$):RETURN
23096 SG.VV#=PR(2):SG.MSK$="#####.##":GOSUB 1471
23098 SG.ROW%=12:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23100 PR(2)=VAL(SG.REPLY$):RETURN
```

```
23102 SG.VV#=CP(2):SG.MSK$="####.##":GOSUB 1471
23104 SG.ROW%=12:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23106 CP(2)=VAL(SG.REPLY$):RETURN
23108 SG.VV#=CH(2):SG.MSK$="####.##":GOSUB 1471
23110 SG.ROW%=12:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23112 CH(2)=VAL(SG.REPLY$):RETURN
23114 SG.VV#=LT(2):SG.MSK$="##.#":GOSUB 1471
23116 SG.ROW%=12:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23118 LT(2)=VAL(SG.REPLY$):RETURN
23120 SG.DF$=I$(3)
23122 SG.ROW%=13:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23124 I$(3)=SG.REPLY$:RETURN
23126 SG.DF$=PN$(3)
23128 SG.ROW%=13:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23130 PN$(3)=SG.REPLY$:RETURN
23132 SG.VV#=QY(3):SG.MSK$="#######":GOSUB 1471
23134 SG.ROW%=13:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23136 QY(3)=VAL(SG.REPLY$):RETURN
23138 SG.VV#=SR(3):SG.MSK$="#####.##":GOSUB 1471
23140 SG.ROW%=13:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23142 SR(3)=VAL(SG.REPLY$):RETURN
23144 SG.VV#=PR(3):SG.MSK$="#####.##":GOSUB 1471
23146 SG.ROW%=13:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23148 PR(3)=VAL(SG.REPLY$):RETURN
23150 SG.VV#=CP(3):SG.MSK$="####.##":GOSUB 1471
23152 SG.ROW%=13:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23154 CP(3)=VAL(SG.REPLY$):RETURN
23156 SG.VV#=CH(3):SG.MSK$="####.##":GOSUB 1471
23158 SG.ROW%=13:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23160 CH(3)=VAL(SG.REPLY$):RETURN
23162 SG.VV#=LT(3):SG.MSK$="##.#":GOSUB 1471
23164 SG.ROW%=13:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23166 LT(3)=VAL(SG.REPLY$):RETURN
23168 SG.DF$=I$(4)
23170 SG.ROW%=14:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23172 I$(4)=SG.REPLY$:RETURN
23174 SG.DF$=PN$(4)
23176 SG.ROW%=14:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23178 PN$(4)=SG.REPLY$:RETURN
23180 SG.VV#=QY(4):SG.MSK$="#######":GOSUB 1471
23182 SG.ROW%=14:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23184 QY(4)=VAL(SG.REPLY$):RETURN
23186 SG.VV#=SR(4):SG.MSK$="#####.##":GOSUB 1471
23188 SG.ROW%=14:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23190 SR(4)=VAL(SG.REPLY$):RETURN
23192 SG.VV#=PR(4):SG.MSK$="#####.##":GOSUB 1471
23194 SG.ROW%=14:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23196 PR(4)=VAL(SG.REPLY$):RETURN
23198 SG.VV#=CP(4):SG.MSK$="####.##":GOSUB 1471
23200 SG.ROW%=14:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23202 CP(4)=VAL(SG.REPLY$):RETURN
23204 SG.VV#=CH(4):SG.MSK$="####.##":GOSUB 1471
23206 SG.ROW%=14:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23208 CH(4)=VAL(SG.REPLY$):RETURN
23210 SG.VV#=LT(4):SG.MSK$="##.#":GOSUB 1471
23212 SG.ROW%=14:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23214 LT(4)=VAL(SG.REPLY$):RETURN
23216 SG.DF$=I$(5)
23218 SG.ROW%=15:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23220 I$(5)=SG.REPLY$:RETURN
23222 SG.DF$=PN$(5)
23224 SG.ROW%=15:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23226 PN$(5)=SG.REPLY$:RETURN
23228 SG.VV#=QY(5):SG.MSK$="#######":GOSUB 1471
23230 SG.ROW%=15:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23232 QY(5)=VAL(SG.REPLY$):RETURN
23234 SG.VV#=SR(5):SG.MSK$="#####.##":GOSUB 1471
23236 SG.ROW%=15:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23238 SR(5)=VAL(SG.REPLY$):RETURN
23240 SG.VV#=PR(5):SG.MSK$="#####.##":GOSUB 1471
23242 SG.ROW%=15:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23244 PR(5)=VAL(SG.REPLY$):RETURN
23246 SG.VV#=CP(5):SG.MSK$="####.##":GOSUB 1471
23248 SG.ROW%=15:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23250 CP(5)=VAL(SG.REPLY$):RETURN
```

```
23252 SG.VV#=CH(5):SG.MSK$="####.##":GOSUB 1471
23254 SG.ROW%=15:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23256 CH(5)=VAL(SG.REPLY$):RETURN
23258 SG.VV#=LT(5):SG.MSK$="##.#":GOSUB 1471
23260 SG.ROW%=15:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23262 LT(5)=VAL(SG.REPLY$):RETURN
23264 SG.DF$=I$(6)
23266 SG.ROW%=16:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23268 I$(6)=SG.REPLY$:RETURN
23270 SG.DF$=PN$(6)
23272 SG.ROW%=16:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23274 PN$(6)=SG.REPLY$:RETURN
23276 SG.VV#=QY(6):SG.MSK$="#######":GOSUB 1471
23278 SG.ROW%=16:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23280 QY(6)=VAL(SG.REPLY$):RETURN
23282 SG.VV#=SR(6):SG.MSK$="#####.##":GOSUB 1471
23284 SG.ROW%=16:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23286 SR(6)=VAL(SG.REPLY$):RETURN
23288 SG.VV#=PR(6):SG.MSK$="#####.##":GOSUB 1471
23290 SG.ROW%=16:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23292 PR(6)=VAL(SG.REPLY$):RETURN
23294 SG.VV#=CP(6):SG.MSK$="####.##":GOSUB 1471
23296 SG.ROW%=16:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23298 CP(6)=VAL(SG.REPLY$):RETURN
23300 SG.VV#=CH(6):SG.MSK$="####.##":GOSUB 1471
23302 SG.ROW%=16:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23304 CH(6)=VAL(SG.REPLY$):RETURN
23306 SG.VV#=LT(6):SG.MSK$="##.#":GOSUB 1471
23308 SG.ROW%=16:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23310 LT(6)=VAL(SG.REPLY$):RETURN
23312 SG.DF$=I$(7)
23314 SG.ROW%=17:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23316 I$(7)=SG.REPLY$:RETURN
23318 SG.DF$=PN$(7)
23320 SG.ROW%=17:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23322 PN$(7)=SG.REPLY$:RETURN
23324 SG.VV#=QY(7):SG.MSK$="#######":GOSUB 1471
23326 SG.ROW%=17:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23328 QY(7)=VAL(SG.REPLY$):RETURN
23330 SG.VV#=SR(7):SG.MSK$="#####.##":GOSUB 1471
23332 SG.ROW%=17:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23334 SR(7)=VAL(SG.REPLY$):RETURN
23336 SG.VV#=PR(7):SG.MSK$="#####.##":GOSUB 1471
23338 SG.ROW%=17:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23340 PR(7)=VAL(SG.REPLY$):RETURN
23342 SG.VV#=CP(7):SG.MSK$="####.##":GOSUB 1471
23344 SG.ROW%=17:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23346 CP(7)=VAL(SG.REPLY$):RETURN
23348 SG.VV#=CH(7):SG.MSK$="####.##":GOSUB 1471
23350 SG.ROW%=17:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23352 CH(7)=VAL(SG.REPLY$):RETURN
23354 SG.VV#=LT(7):SG.MSK$="##.#":GOSUB 1471
23356 SG.ROW%=17:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23358 LT(7)=VAL(SG.REPLY$):RETURN
23360 SG.DF$=I$(8)
23362 SG.ROW%=18:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23364 I$(8)=SG.REPLY$:RETURN
23366 SG.DF$=PN$(8)
23368 SG.ROW%=18:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23370 PN$(8)=SG.REPLY$:RETURN
23372 SG.VV#=QY(8):SG.MSK$="#######":GOSUB 1471
23374 SG.ROW%=18:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23376 QY(8)=VAL(SG.REPLY$):RETURN
23378 SG.VV#=SR(8):SG.MSK$="#####.##":GOSUB 1471
23380 SG.ROW%=18:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23382 SR(8)=VAL(SG.REPLY$):RETURN
23384 SG.VV#=PR(8):SG.MSK$="#####.##":GOSUB 1471
23386 SG.ROW%=18:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23388 PR(8)=VAL(SG.REPLY$):RETURN
23390 SG.VV#=CP(8):SG.MSK$="####.##":GOSUB 1471
23392 SG.ROW%=18:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23394 CP(8)=VAL(SG.REPLY$):RETURN
23396 SG.VV#=CH(8):SG.MSK$="####.##":GOSUB 1471
23398 SG.ROW%=18:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23400 CH(8)=VAL(SG.REPLY$):RETURN
```

```
23402 SG.VV#=LT(8):SG.MSK$="##.#":GOSUB 1471
23404 SG.ROW%=18:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23406 LT(8)=VAL(SG.REPLY$):RETURN
23408 SG.DF$=I$(9)
23410 SG.ROW%=19:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23412 I$(9)=SG.REPLY$:RETURN
23414 SG.DF$=PN$(9)
23416 SG.ROW%=19:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23418 PN$(9)=SG.REPLY$:RETURN
23420 SG.VV#=QY(9):SG.MSK$="#######":GOSUB 1471
23422 SG.ROW%=19:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23424 QY(9)=VAL(SG.REPLY$):RETURN
23426 SG.VV#=SR(9):SG.MSK$="#####.##":GOSUB 1471
23428 SG.ROW%=19:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23430 SR(9)=VAL(SG.REPLY$):RETURN
23432 SG.VV#=PR(9):SG.MSK$="#####.##":GOSUB 1471
23434 SG.ROW%=19:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23436 PR(9)=VAL(SG.REPLY$):RETURN
23438 SG.VV#=CP(9):SG.MSK$="####.##":GOSUB 1471
23440 SG.ROW%=19:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23442 CP(9)=VAL(SG.REPLY$):RETURN
23444 SG.VV#=CH(9):SG.MSK$="####.##":GOSUB 1471
23446 SG.ROW%=19:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23448 CH(9)=VAL(SG.REPLY$):RETURN
23450 SG.VV#=LT(9):SG.MSK$="##.#":GOSUB 1471
23452 SG.ROW%=19:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23454 LT(9)=VAL(SG.REPLY$):RETURN
23456 SG.DF$=I$(10)
23458 SG.ROW%=20:SG.COL%=5:SG.LNG%=15:COLOR 6,0  :GOSUB 775
23460 I$(10)=SG.REPLY$:RETURN
23462 SG.DF$=PN$(10)
23464 SG.ROW%=20:SG.COL%=22:SG.LNG%=7:COLOR 6,0  :GOSUB 775
23466 PN$(10)=SG.REPLY$:RETURN
23468 SG.VV#=QY(10):SG.MSK$="#######":GOSUB 1471
23470 SG.ROW%=20:SG.COL%=31:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23472 QY(10)=VAL(SG.REPLY$):RETURN
23474 SG.VV#=SR(10):SG.MSK$="#####.##":GOSUB 1471
23476 SG.ROW%=20:SG.COL%=40:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23478 SR(10)=VAL(SG.REPLY$):RETURN
23480 SG.VV#=PR(10):SG.MSK$="#####.##":GOSUB 1471
23482 SG.ROW%=20:SG.COL%=50:SG.LNG%=8:COLOR 6,0  :GOSUB 770
23484 PR(10)=VAL(SG.REPLY$):RETURN
23486 SG.VV#=CP(10):SG.MSK$="####.##":GOSUB 1471
23488 SG.ROW%=20:SG.COL%=59:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23490 CP(10)=VAL(SG.REPLY$):RETURN
23492 SG.VV#=CH(10):SG.MSK$="####.##":GOSUB 1471
23494 SG.ROW%=20:SG.COL%=68:SG.LNG%=7:COLOR 6,0  :GOSUB 770
23496 CH(10)=VAL(SG.REPLY$):RETURN
23498 SG.VV#=LT(10):SG.MSK$="##.#":GOSUB 1471
23500 SG.ROW%=20:SG.COL%=76:SG.LNG%=3:COLOR 6,0  :GOSUB 770
23502 LT(10)=VAL(SG.REPLY$):RETURN
23600 SG.DF$=SCR$
23602 SG.UP%=1:SG.ROW%=24:SG.COL%=56:SG.LNG%=1
23603 COLOR 15,1  :GOSUB 775
23604 SCR$=SG.REPLY$:RETURN
23806 GOSUB 1462
23807 GOSUB 23018    'NN$(V%)
23808 IF SG.BTAB%>2 THEN GOTO 24264
23809 GOSUB 23024  SG.BTAB%=1 THEN GOTO 23807    'I$(1)
23810 IF SG.BTAB%>2 THEN GOTO 24264
23811 GOSUB 23030  SG.BTAB%=1 THEN GOTO 23809    'PN$(1)
23812 IF SG.BTAB%>2 THEN GOTO 24264
23813 GOSUB 23036  SG.BTAB%=1 THEN GOTO 23811    'QY(1)
23814 IF SG.BTAB%>2 THEN GOTO 24264
23815 GOSUB 23042  SG.BTAB%=1 THEN GOTO 23813    'SR(1)
23816 IF SG.BTAB%>2 THEN GOTO 24264
23817 GOSUB 23048  SG.BTAB%=1 THEN GOTO 23815    'PR(1)
23818 IF SG.BTAB%>2 THEN GOTO 24264
23819 GOSUB 23054  SG.BTAB%=1 THEN GOTO 23817    'CP(1)
23820 IF SG.BTAB%>2 THEN GOTO 24264
23821 GOSUB 23060  SG.BTAB%=1 THEN GOTO 23819    'CH(1))
23822 IF SG.BTAB%>2 THEN GOTO 24264
23823 GOSUB 23066  SG.BTAB%=1 THEN GOTO 23821    'LT(1)
23824 IF SG.BTAB%>2 THEN GOTO 24264
23825 GOSUB 23072  SG.BTAB%=1 THEN GOTO 23823    'I$(2)
```

```
23826 IF SG.BTAB%>2 THEN GOTO 24264
23827 GOSUB 23078  SG.BTAB%=1 THEN GOTO 23825     'PN$(2)
23828 IF SG.BTAB%>2 THEN GOTO 24264
23829 GOSUB 23084  SG.BTAB%=1 THEN GOTO 23827     'QY(2)
23830 IF SG.BTAB%>2 THEN GOTO 24264
23831 GOSUB 23090  SG.BTAB%=1 THEN GOTO 23829     'SR(2)
23832 IF SG.BTAB%>2 THEN GOTO 24264
23833 GOSUB 23096  SG.BTAB%=1 THEN GOTO 23831     'PR(2)
23834 IF SG.BTAB%>2 THEN GOTO 24264
23835 GOSUB 23102  SG.BTAB%=1 THEN GOTO 23833     'CP(2)
23836 IF SG.BTAB%>2 THEN GOTO 24264
23837 GOSUB 23108  SG.BTAB%=1 THEN GOTO 23835     'CH(2)
23838 IF SG.BTAB%>2 THEN GOTO 24264
23839 GOSUB 23114  SG.BTAB%=1 THEN GOTO 23837     'LT(2)
23840 IF SG.BTAB%>2 THEN GOTO 24264
23841 GOSUB 23120  SG.BTAB%=1 THEN GOTO 23839     'I$(3)
23842 IF SG.BTAB%>2 THEN GOTO 24264
23843 GOSUB 23126  SG.BTAB%=1 THEN GOTO 23841     'PN$(3)
23844 IF SG.BTAB%>2 THEN GOTO 24264
23845 GOSUB 23132  SG.BTAB%=1 THEN GOTO 23843     'QY(3)
23846 IF SG.BTAB%>2 THEN GOTO 24264
23847 GOSUB 23138  SG.BTAB%=1 THEN GOTO 23845     'SR(3)
23848 IF SG.BTAB%>2 THEN GOTO 24264
23849 GOSUB 23144  SG.BTAB%=1 THEN GOTO 23847     'PR(3)
23850 IF SG.BTAB%>2 THEN GOTO 24264
23851 GOSUB 23150  SG.BTAB%=1 THEN GOTO 23849     'CP(3)
23852 IF SG.BTAB%>2 THEN GOTO 24264
23853 GOSUB 23156  SG.BTAB%=1 THEN GOTO 23851     'CH(3)
23854 IF SG.BTAB%>2 THEN GOTO 24264
23855 GOSUB 23162  SG.BTAB%=1 THEN GOTO 23853     'LT(3)
23856 IF SG.BTAB%>2 THEN GOTO 24264
23857 GOSUB 23168  SG.BTAB%=1 THEN GOTO 23855     'I$(4)
23858 IF SG.BTAB%>2 THEN GOTO 24264
23859 GOSUB 23174  SG.BTAB%=1 THEN GOTO 23857     'PN$(4)
23860 IF SG.BTAB%>2 THEN GOTO 24264
23861 GOSUB 23180  SG.BTAB%=1 THEN GOTO 23859     'QY(4)
23862 IF SG.BTAB%>2 THEN GOTO 24264
23863 GOSUB 23186  SG.BTAB%=1 THEN GOTO 23861     'SR(4)
23864 IF SG.BTAB%>2 THEN GOTO 24264
23865 GOSUB 23192  SG.BTAB%=1 THEN GOTO 23863     'PR(4)
23866 IF SG.BTAB%>2 THEN GOTO 24264
23867 GOSUB 23198  SG.BTAB%=1 THEN GOTO 23865     'CP(4)
23868 IF SG.BTAB%>2 THEN GOTO 24264
23869 GOSUB 23204  SG.BTAB%=1 THEN GOTO 23867     'CH(4)
23870 IF SG.BTAB%>2 THEN GOTO 24264
23871 GOSUB 23210  SG.BTAB%=1 THEN GOTO 23869     'LT(4)
23872 IF SG.BTAB%>2 THEN GOTO 24264
23873 GOSUB 23216  SG.BTAB%=1 THEN GOTO 23871     'I$(5)
23874 IF SG.BTAB%>2 THEN GOTO 24264
23875 GOSUB 23222  SG.BTAB%=1 THEN GOTO 23873     'PN$(5)
23876 IF SG.BTAB%>2 THEN GOTO 24264
23877 GOSUB 23228  SG.BTAB%=1 THEN GOTO 23875     'QY(5)
23878 IF SG.BTAB%>2 THEN GOTO 24264
23879 GOSUB 23234  SG.BTAB%=1 THEN GOTO 23877     'SR(5)
23880 IF SG.BTAB%>2 THEN GOTO 24264
23881 GOSUB 23240  SG.BTAB%=1 THEN GOTO 23879     'PR(5)
23882 IF SG.BTAB%>2 THEN GOTO 24264
23883 GOSUB 23246  SG.BTAB%=1 THEN GOTO 23881     'CP(5)
23884 IF SG.BTAB%>2 THEN GOTO 24264
23885 GOSUB 23252  SG.BTAB%=1 THEN GOTO 23883     'CH(5)
23886 IF SG.BTAB%>2 THEN GOTO 24264
23887 GOSUB 23258  SG.BTAB%=1 THEN GOTO 23885     'LT(5)
23888 IF SG.BTAB%>2 THEN GOTO 24264
23889 GOSUB 23264  SG.BTAB%=1 THEN GOTO 23887     'I$(6)
23890 IF SG.BTAB%>2 THEN GOTO 24264
23891 GOSUB 23270  SG.BTAB%=1 THEN GOTO 23889     'PN$(6)
23892 IF SG.BTAB%>2 THEN GOTO 24264
23893 GOSUB 23276  SG.BTAB%=1 THEN GOTO 23891     'QY(6)
23894 IF SG.BTAB%>2 THEN GOTO 24264
23895 GOSUB 23282  SG.BTAB%=1 THEN GOTO 23893     'SR(6)
23896 IF SG.BTAB%>2 THEN GOTO 24264
23897 GOSUB 23288  SG.BTAB%=1 THEN GOTO 23895     'PR(6)
23898 IF SG.BTAB%>2 THEN GOTO 24264
23899 GOSUB 23294  SG.BTAB%=1 THEN GOTO 23897     'CP(6)
23900 IF SG.BTAB%>2 THEN GOTO 24264
```

```
23901 GOSUB 23300  SG.BTAB%=1 THEN GOTO 23899    'CH(6)
23902 IF SG.BTAB%>2 THEN GOTO 24264
23903 GOSUB 23306  SG.BTAB%=1 THEN GOTO 23901    'LT(6)
23904 IF SG.BTAB%>2 THEN GOTO 24264
23905 GOSUB 23312  SG.BTAB%=1 THEN GOTO 23903    'I$(7)
23906 IF SG.BTAB%>2 THEN GOTO 24264
23907 GOSUB 23318  SG.BTAB%=1 THEN GOTO 23905    'PN$(7)
23908 IF SG.BTAB%>2 THEN GOTO 24264
23909 GOSUB 23324  SG.BTAB%=1 THEN GOTO 23907    'QY(7)
23910 IF SG.BTAB%>2 THEN GOTO 24264
23911 GOSUB 23330  SG.BTAB%=1 THEN GOTO 23909    'SR(7)
23912 IF SG.BTAB%>2 THEN GOTO 24264
23913 GOSUB 23336  SG.BTAB%=1 THEN GOTO 23911    'PR(7)
23914 IF SG.BTAB%>2 THEN GOTO 24264
23915 GOSUB 23342  SG.BTAB%=1 THEN GOTO 23913    'CP(7)
23916 IF SG.BTAB%>2 THEN GOTO 24264
23917 GOSUB 23348  SG.BTAB%=1 THEN GOTO 23915    'CH(7)
23918 IF SG.BTAB%>2 THEN GOTO 24264
23919 GOSUB 23354  SG.BTAB%=1 THEN GOTO 23917    'LT(7)
23920 IF SG.BTAB%>2 THEN GOTO 24264
23921 GOSUB 23360  SG.BTAB%=1 THEN GOTO 23919    'I$(8)
23922 IF SG.BTAB%>2 THEN GOTO 24264
23923 GOSUB 23366  SG.BTAB%=1 THEN GOTO 23921    'PN$(8)
23924 IF SG.BTAB%>2 THEN GOTO 24264
23925 GOSUB 23372  SG.BTAB%=1 THEN GOTO 23923    'QY(8)
23926 IF SG.BTAB%>2 THEN GOTO 24264
23927 GOSUB 23378  SG.BTAB%=1 THEN GOTO 23925    'SR(8)
23928 IF SG.BTAB%>2 THEN GOTO 24264
23929 GOSUB 23384  SG.BTAB%=1 THEN GOTO 23927    'PR(8)
23930 IF SG.BTAB%>2 THEN GOTO 24264
23931 GOSUB 23390  SG.BTAB%=1 THEN GOTO 23929    'CP(8)
23932 IF SG.BTAB%>2 THEN GOTO 24264
23933 GOSUB 23396  SG.BTAB%=1 THEN GOTO 23931    'CH(8)
23934 IF SG.BTAB%>2 THEN GOTO 24264
23935 GOSUB 23402  SG.BTAB%=1 THEN GOTO 23933    'LT(8)
23936 IF SG.BTAB%>2 THEN GOTO 24264
23937 GOSUB 23408  SG.BTAB%=1 THEN GOTO 23935    'I$(9)
23938 IF SG.BTAB%>2 THEN GOTO 24264
23939 GOSUB 23414  SG.BTAB%=1 THEN GOTO 23937    'PN$(9)
23940 IF SG.BTAB%>2 THEN GOTO 24264
23941 GOSUB 23420  SG.BTAB%=1 THEN GOTO 23939    'QY(9)
23942 IF SG.BTAB%>2 THEN GOTO 24264
23943 GOSUB 23426  SG.BTAB%=1 THEN GOTO 23941    'SR(9)
23944 IF SG.BTAB%>2 THEN GOTO 24264
23945 GOSUB 23432  SG.BTAB%=1 THEN GOTO 23943    'PR(9)
23946 IF SG.BTAB%>2 THEN GOTO 24264
23947 GOSUB 23438  SG.BTAB%=1 THEN GOTO 23945    'CP(9)
23948 IF SG.BTAB%>2 THEN GOTO 24264
23949 GOSUB 23444  SG.BTAB%=1 THEN GOTO 23947    'CH(9)
23950 IF SG.BTAB%>2 THEN GOTO 24264
23951 GOSUB 23450  SG.BTAB%=1 THEN GOTO 23949    'LT(9)
23952 IF SG.BTAB%>2 THEN GOTO 24264
23953 GOSUB 23456  SG.BTAB%=1 THEN GOTO 23951    'I$(10)
23954 IF SG.BTAB%>2 THEN GOTO 24264
23955 GOSUB 23462  SG.BTAB%=1 THEN GOTO 23953   'PN$(10)
23956 IF SG.BTAB%>2 THEN GOTO 24264
23957 GOSUB 23468  SG.BTAB%=1 THEN GOTO 23955    'QY(10)
23958 IF SG.BTAB%>2 THEN GOTO 24264
23959 GOSUB 23474  SG.BTAB%=1 THEN GOTO 23957    'SR(10)
23960 IF SG.BTAB%>2 THEN GOTO 24264
23961 GOSUB 23480  SG.BTAB%=1 THEN GOTO 23959    'PR(10)
23962 IF SG.BTAB%>2 THEN GOTO 24264
23963 GOSUB 23486  SG.BTAB%=1 THEN GOTO 23961    'CP(10)
23964 IF SG.BTAB%>2 THEN GOTO 24264
23965 GOSUB 23492  SG.BTAB%=1 THEN GOTO 23963    'CH(10)
23966 IF SG.BTAB%>2 THEN GOTO 24264
23967 GOSUB 23498  SG.BTAB%=1 THEN GOTO 23965    'LT(10)
23968 IF SG.BTAB%>2 THEN GOTO 24264
24001 GOSUB 23600  SG.BTAB%=1 THEN GOTO 23999    'SCR$
24002 IF SG.BTAB%>2 THEN RETURN
24003 RETURN
24012 GOSUB 1462
24013 LOCATE 6,16:COLOR 10,0
24014 PRINT USING "\              \";NN$(V%):COLOR 6,0
24157 FOR Q% = 1 TO 10
```

```
24159 LOCATE 10+Q%,5:PRINT USING "\                    \";I$(Q%);
24161 LOCATE 10+Q%,22:PRINT USING "\      \";PN$(Q%);
24163 LOCATE 10+Q%,31:PRINT USING "#######";QY(Q%);
24165 LOCATE 10+Q%,40:PRINT USING "#####.##";SR(Q%);
24167 LOCATE 10+Q%,50:PRINT USING "#####.##";PR(Q%);
24169 LOCATE 10+Q%,59:PRINT USING "####.##";CP(Q%);
24171 LOCATE 10+Q%,68:PRINT USING "####.##";CH(Q%);
24173 LOCATE 10+Q%,76:PRINT USING "##.#";LT(Q%);
24175 NEXT
24207 LOCATE 24,56:COLOR 15,1:PRINT USING "!";SCR$;
24209 RETURN
24210 GOSUB 24216
24212 GOSUB 24013
24214 RETURN
24215 LOCATE 25,1:COLOR , 0:PRINT" ";
24216 COLOR ,, 8:CLS:GOSUB 1462
24217 COLOR  2,0
24218 COLOR 3,0:LOCATE 2,1:PRINT "    LIST STOCKABLE ITEMS"
      "SO THAT SPARE PARTS REQUIREMENTS MAY BE CALCULATED";
24220 COLOR 3,0:LOCATE 4,10:PRINT "When needed, press the"
                                "F9 key to advance to bottom";
24222 COLOR 3,0:LOCATE 4,61:PRINT "of screen.";
24224 COLOR 2,0:LOCATE 6,1:PRINT "    Node name:";
24226 LOCATE 7,1:PRINT "                                Quantity"
            "W'house    Unit     Proc.  Holding  Lead";
24228 LOCATE 8,1:PRINT "    Stockable item    Part#"
          "Req.     Price    Cost     Cost   Time";
24230 LOCATE 9,1:PRINT "(#parts) (cuft/part)   $      $";
24232 COLOR 2,0:LOCATE 9,70:PRINT "$    (#per)";
24234 FOR Q% = 1 TO 10
24236   LOCATE 10+Q%,1: PRINT USING "##.";Q%
24238 NEXT
24258 COLOR 3,0:LOCATE 24,34:PRINT "Enter F to continue";
24260 COLOR 15,1:LOCATE 24,56:PRINT " ";
24262 COLOR 2,0:RETURN
24264 IF SG.BTAB% < 5 THEN RETURN ELSE GOTO 24001
```

```
1 '
2 '       PROGRAM LISTING FOR "EXISTING"
3 '
4 '
5 '                GOSUB SCREEN SUBROUTINE
6 '       GROUP ANY EXISTING END ITEMS INTO STREAMS
7 '
8 '             EXPLANATION AND FIRST STREAM
9 '
10 '
11 REQ%=2:GOSUB 19000
20 REQ%=6:GOSUB 19000
30 IF SCR$ = "R" THEN 11
32 IF AGE% = 1 THEN 97
33 '
34 '                GOSUB SCREEN SUBROUTINE
35 '       GROUP ANY EXISTING END ITEMS INTO STREAMS
36 '
37 '               SECOND AND THIRD STREAMS
38 '
39 '
40 REQ%=2:GOSUB 19350
50 REQ%=6:GOSUB 19350
60 IF SCR$ = "R" THEN 11
62 IF AGE% <= 3 THEN 97
63 '
64 '                GOSUB SCREEN SUBROUTINE
65 '       GROUP ANY EXISTING END ITEMS INTO STREAMS
66 '
67 '               FOURTH AND FIFTH STREAMS
68 '
69 '
70 REQ%=2:GOSUB 19900
80 REQ%=6:GOSUB 19900
90 IF SCR$ = "R" THEN 40
91 '
92 '      INITIALIZE VARIABLES
93 '
94 '
97 V% = 2
100 FOR N% = 1 TO 5
110   FOR VT% = 1 TO 5
120     KKEY%(V%) = N%:  TTIME%(V%) = XD%(N%): V% = V% + 1
130   NEXT VT%
140 NEXT N%
150 V% = 2
160 FOR N% = 2 TO 26
170   IF IP%(N%) = 0 THEN 190
180   IP%(V%)=IP%(N%): TTIME%(V%)=TTIME%(N%)
181   KKEY%(V%)=KKEY%(N%): RRA(V%)=RA(KKEY%(N%))
182   ST%(V%)=IP%(N%)
188 V% = V% + 1
190 NEXT N%
191 NS% = V%
192 IF IP%(1) = 0 THEN 210
193 '
194 '      CHAIN BACK TO PROGRAM "MAIN"
195 '            BEGIN CALCULATIONS
196 '
197 '
200 CHAIN "MAIN",200,ALL
203 '
204 '      INITIALIZE VARIABLES
205 '
206 '
210 FOR N% = 2 TO NS%-1
220   IP%(N%-1)=IP%(N%): TTIME%(N%-1)=TTIME%(N%)
221   KKEY%(N%-1)=KKEY%(N%): RRA(N%-1)=RRA(N%)
222   ST%(N%-1)=ST%(N%)
230 NEXT N%
240 GOTO 200
300 '
310 '      END OF MAIN PROGRAM (EXISTING)
320 '
330 '
```

```
680 '
681 '       SCREEN SUBROUTINE DRIVER
682 '       ________________________
683 '
684 '      note:  subroutine not listed -
685 '             See screen subroutine driver listing in
686 '             lines 700 - 1486 in program "INTRO"
687 '
688 '
18980 '
18981 '                 SCREEN SUBROUTINE
18982 '      INPUT DATA FOR FIRST EXISTING STREAM
18983 '      ____________________________________
18984 '
19000 ON REQ% GOSUB 19268 , 19262 , 19222 , 19132,
                            19214 , 19174 , 19218
19002 RETURN
19018 SG.VV#=AGE%:SG.MSK$="###":GOSUB 1471
19020 SG.ROW%=11:SG.COL%=60:SG.LNG%=3:COLOR 15,1 :GOSUB 770
19022 AGE%=VAL(SG.REPLY$):RETURN
19024 SG.VV#=XD%(1):SG.MSK$="#####":GOSUB 1471
19026 SG.ROW%=14:SG.COL%=58:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19028 XD%(1)=VAL(SG.REPLY$):RETURN
19030 SG.VV#=XF%(1):SG.MSK$="#####":GOSUB 1471
19032 SG.ROW%=16:SG.COL%=58:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19034 XF%(1)=VAL(SG.REPLY$):RETURN
19048 SG.VV#=IP%(2):SG.MSK$="#####":GOSUB 1471
19050 SG.ROW%=18:SG.COL%=72:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19052 IP%(2)=VAL(SG.REPLY$):RETURN
19066 SG.VV#=IP%(3):SG.MSK$="#####":GOSUB 1471
19068 SG.ROW%=19:SG.COL%=72:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19070 IP%(3)=VAL(SG.REPLY$):RETURN
19084 SG.VV#=IP%(4):SG.MSK$="#####":GOSUB 1471
19086 SG.ROW%=20:SG.COL%=72:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19088 IP%(4)=VAL(SG.REPLY$):RETURN
19102 SG.VV#=IP%(5):SG.MSK$="#####":GOSUB 1471
19104 SG.ROW%=21:SG.COL%=72:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19106 IP%(5)=VAL(SG.REPLY$):RETURN
19120 SG.VV#=IP%(6):SG.MSK$="#####":GOSUB 1471
19122 SG.ROW%=22:SG.COL%=72:SG.LNG%=5:COLOR 15,1 :GOSUB 770
19124 IP%(6)=VAL(SG.REPLY$):RETURN
19126 SG.DF$=SCR$
19128 SG.UP%=1:SG.ROW%=24:SG.COL%=74:SG.LNG%=1
19129 COLOR 15,1  :GOSUB 775
19130 SCR$=SG.REPLY$:RETURN
19174 GOSUB 1462
19175 GOSUB 19018    'AGE%
19176 IF SG.BTAB%>2 THEN GOTO 19320
19177 GOSUB 19024  SG.BTAB%=1 THEN GOTO 19175    'T%
19178 IF SG.BTAB%>2 THEN GOTO 19320
19179 GOSUB 19030  SG.BTAB%=1 THEN GOTO 19177    'IP%
19180 IF SG.BTAB%>2 THEN GOTO 19320
19185 GOSUB 19048  SG.BTAB%=1 THEN GOTO 19179    'IP%(2)
19186 IF SG.BTAB%>2 THEN GOTO 19320
19187 IF NO.ENVIR% = 1 THEN GOTO 19211
19191 GOSUB 19066  SG.BTAB%=1 THEN GOTO 19185    'IP%(3)
19192 IF SG.BTAB%>2 THEN GOTO 19320
19193 IF NO.ENVIR% = 2 THEN GOTO 19211
19197 GOSUB 19084  SG.BTAB%=1 THEN GOTO 19191    'IP%(4)
19198 IF SG.BTAB%>2 THEN GOTO 19320
19199 IF NO.ENVIR% = 3 THEN GOTO 19211
19203 GOSUB 19102  SG.BTAB%=1 THEN GOTO 19197    'IP%(4)
19204 IF SG.BTAB%>2 THEN GOTO 19320
19205 IF NO.ENVIR% = 4 THEN GOTO 19211
19209 GOSUB 19120  SG.BTAB%=1 THEN GOTO 19203    'IP%(6)
19210 IF SG.BTAB%>2 THEN GOTO 19320
19211 GOSUB 19126                                   'SCR$
19212 IF SG.BTAB%=1 AND NO.ENVIR%=1 THEN GOTO 19185
19213 IF SG.BTAB%=1 AND NO.ENVIR%=2 THEN GOTO 19191
19214 IF SG.BTAB%=1 AND NO.ENVIR%=3 THEN GOTO 19197
19215 IF SG.BTAB%=1 AND NO.ENVIR%=4 THEN GOTO 19203
19216 IF SG.BTAB%=1 AND NO.ENVIR%=5 THEN GOTO 19209
19218 IF SG.BTAB%>2 THEN RETURN
19219 RETURN
19222 GOSUB 1462
```

```
19223 LOCATE 11,60:COLOR 15,1:PRINT USING "###";AGE%;
19225 LOCATE 14,58:COLOR 15,1:PRINT USING "#####";XD%(1);
19227 LOCATE 16,58:COLOR 15,1:PRINT USING "#####";XF%(1);
19229 LOCATE 18,41:COLOR 2,0
19230 PRINT USING "\              \";NO.ENVIR$(1);
19231 LOCATE 18,64:COLOR 2,0:PRINT USING"#####";DEMAND%(1);
19233 LOCATE 18,72:COLOR 15,1:PRINT USING "#####";IP%(2);
19234 IF NO.ENVIR% < 2 THEN GOTO 19259
19235 LOCATE 19,41:COLOR 2,0
19236 PRINT USING "\              \";NO.ENVIR$(2);
19237 LOCATE 19,64:COLOR 2,0:PRINT USING"#####";DEMAND%(2);
19239 LOCATE 19,72:COLOR 15,1:PRINT USING "#####";IP%(3);
19240 IF NO.ENVIR% < 3 THEN GOTO 19259
19241 LOCATE 20,41:COLOR 2,0
19242 PRINT USING "\              \";NO.ENVIR$(3);
19243 LOCATE 20,64:COLOR 2,0:PRINT USING"#####";DEMAND%(3);
19245 LOCATE 20,72:COLOR 15,1:PRINT USING "#####";IP%(4);
19246 IF NO.ENVIR% < 4 THEN GOTO 19259
19247 LOCATE 21,41:COLOR 2,0
19248 PRINT USING "\              \";NO.ENVIR$(4);
19249 LOCATE 21,64:COLOR 2,0:PRINT USING"#####";DEMAND%(4);
19251 LOCATE 21,72:COLOR 15,1:PRINT USING "#####";IP%(5);
19252 IF NO.ENVIR% < 5 THEN GOTO 19259
19253 LOCATE 22,41:COLOR 2,0
19254 PRINT USING "\              \";NO.ENVIR$(5);
19255 LOCATE 22,64:COLOR 2,0:PRINT USING"#####";DEMAND%(5);
19257 LOCATE 22,72:COLOR 15,1:PRINT USING "#####";IP%(6);
19259 LOCATE 24,74:COLOR 15,1:PRINT USING "!";SCR$;
19261 RETURN
19262 GOSUB 19268
19264 GOSUB 19223
19266 RETURN
19267 LOCATE 25,1:COLOR , 0:PRINT" ";
19268 COLOR ,, 8:CLS:GOSUB 1462
19269 COLOR  2,0
19270 COLOR 3,0:LOCATE 2,24
19271 PRINT "EXISTING  END  ITEMS  DATA  INPUT";
19272 COLOR 3,0:LOCATE 4,2:PRINT "Existing end items"
      "should be divided into groups depending on how many"
19274 COLOR 3,0:LOCATE 5,1:PRINT " periods these end items"
      "have been in operation.  You will be asked to input"
19276 LOCATE 6,1:PRINT " the age of the end items and the"
            "total # of end items in each age group.";
19278 COLOR 3,0:LOCATE 8,2:PRINT "You will then divide the"
            "end items in each age group into different";
19280 COLOR 3,0:LOCATE 9,1
19281 PRINT " operating environments where applicable.";
19282 COLOR 10,0:LOCATE 11,2:PRINT "How many age groups"
                  "of existing end items do you have?";
19284 COLOR 15,1:LOCATE 11,60:PRINT "   ";
19286 COLOR 2,0:LOCATE 14,1
19287 PRINT " Age of group 1 in periods";
19288 COLOR 15,1:LOCATE 14,58:PRINT "     ";
19290 COLOR 2,0:LOCATE 16,1
19291 PRINT " Total # end items in age group 1";
19292 COLOR 15,1:LOCATE 16,58:PRINT "     ";
19294 COLOR 2,0:LOCATE 18,1
19295 PRINT " Divide total # end items   :";
19298 COLOR 2,0:LOCATE 19,1
19299 PRINT " of age group 1 into         >";
19302 COLOR 2,0:LOCATE 20,1
19303 PRINT " failure environments       :";
19304 FOR X% = 1 TO NO.ENVIR%
19305 COLOR 2,0:LOCATE 17+X%,56:PRINT "(demand=     )";
19306 COLOR 15,1:LOCATE 17+X%,72:PRINT "     ";
19307 NEXT X%
19314 COLOR 3,0:LOCATE 24,2:PRINT "Enter one of the"
      "following (R to go back, F to go forward)";
19316 COLOR 15,1:LOCATE 24,74:PRINT " ";
19318 COLOR 2,0:RETURN
19320 IF SG.BTAB% < 5 THEN RETURN ELSE GOTO 19211
19330 '
19331 '                SCREEN SUBROUTINE
19332 '      INPUT DATA FOR SECOND AND THIRD STREAM
19333 '      ____________________________________
```

```
19334 '
19335 '     note:  subroutine not listed -
19336 '            This screen subroutine is similar to
19337 '            the subroutine in lines 19000 - 19320.
19338 '
19339 '
19730 '
19731 '                SCREEN SUBROUTINE
19732 '     INPUT DATA FOR FOURTH AND FIFTH STREAM
19733 '     ______________________________________
19734 '
19735 '     note:  subroutine not listed -
19736 '            This screen subroutine is similar to
19737 '            the subroutine in lines 19000 - 19320.
19738 '
19739 '
```

```
10 '
20 '      PROGRAM LISTING "DIVIDE"
30 '
40 '
50 '
380 '
385 '      INITIALIZE VARIABLES
386 '      ____________________
387 '
400 R%=NS%:  SUM%=0:  KK%=IP%(NS%)
410 FOR N% = 1 TO NO.ENVIR%
420   SUMSTREAM(N%) = 0
430 NEXT N%
440 FOR N% = 1 TO NS%
450   SUMSTREAM(KKEY%(N%))=SUMSTREAM(KKEY%(N%))+SP2%(N%)
460 NEXT N%
470 '
480 '                GOSUB SCREEN SUBROUTINE
490 '      INPUT NUMBER OF END ITEMS TO BE PLACED IN EACH
495 '                        ENVIRONMENT
496 '      ____________________________________________
497 '
500 REQ%=2:GOSUB 25000
505 REQ%=6:GOSUB 25000
506 '
507 '      INITIALIZE VARIABLES
508 '      ____________________
509 '
510 '
512 V% = R%: VT%=1
513 FOR N% = R% TO R%+4
514   IF IP%(N%) = 0 THEN 516
515   IP%(V%)=IP%(N%):  KKEY%(V%)=VT%: V% = V% + 1
516 VT% = VT% + 1
517 NEXT N%
519 NS%=V%-1:  V% = 0
520 FOR N% = R% TO NS%
530   TTIME%(N%)=T%: ST%(N%)=IP%(N%)
531   RRA(N%)=RA(KKEY%(N%)): SUM%=SUM%+IP%(N%): V%=V%+1
540 NEXT N%
550 IF IP%(NS%) > 0 THEN NS% = NS% + 1:  D% = 1
560 IF SUM% = KK% THEN 650
565 '
566 '      CHECK USER'S ADDITION
567 '      _____________________
568 '
570 CLS:COLOR 2,0:LOCATE 10,30:PRINT"Check your addition."
571 PRINT"The # of units assigned to failure environments"
580 PRINT "    does not equal the # of units procured."
590 COLOR 3,0:LOCATE 20,20
591 PRINT "press any key to review previous input"
600 A$=INKEY$ A$ = "" THEN GOTO 600
610 NS% = R%:  SUM% = 0:  GOTO 500
630 '
635 '      CHAIN TO APPROPRIATE PLACE IN PROGRAM "MAIN"
640 '      _____________________________________________
645 '
650 IF F%=0 THEN CHAIN "MAIN",630,ALL
660 IF F%=1 THEN CHAIN "MAIN",1130,ALL
670 IF F%=2 THEN CHAIN "MAIN",1180,ALL
680 '
685 '      END OF MAIN PROGRAM (DIVIDE)
686 '
687 '
690 '      SCREEN SUBROUTINE DRIVER
691 '      ________________________
692 '
693 '      note:  subroutine not listed -
694 '             See the screen subroutine driver in lines
695 '             700 - 1486 in program "INTRO".
696 '
697 '
25000 ON REQ% GOSUB 25364 , 25358 , 25302 , 25180,
                            25294 , 25238 , 25298
25002 RETURN
```

```
25048 SG.VV#=IP%(NS%):SG.MSK$="#####":GOSUB 1471
25050 SG.ROW%=12:SG.COL%=66:SG.LNG%=5:COLOR 15,1 :GOSUB 770
25052 IP%(NS%)=VAL(SG.REPLY$):RETURN
25078 SG.VV#=IP%(NS%+1):SG.MSK$="#####":GOSUB 1471
25080 SG.ROW%=14:SG.COL%=66:SG.LNG%=5:COLOR 15,1 :GOSUB 770
25082 IP%(NS%+1)=VAL(SG.REPLY$):RETURN
25108 SG.VV#=IP%(NS%+2):SG.MSK$="#####":GOSUB 1471
25110 SG.ROW%=16:SG.COL%=66:SG.LNG%=5:COLOR 15,1 :GOSUB 770
25112 IP%(NS%+2)=VAL(SG.REPLY$):RETURN
25138 SG.VV#=IP%(NS%+3):SG.MSK$="#####":GOSUB 1471
25140 SG.ROW%=18:SG.COL%=66:SG.LNG%=5:COLOR 15,1 :GOSUB 770
25142 IP%(NS%+3)=VAL(SG.REPLY$):RETURN
25168 SG.VV#=IP%(NS%+4):SG.MSK$="#####":GOSUB 1471
25170 SG.ROW%=20:SG.COL%=66:SG.LNG%=5:COLOR 15,1 :GOSUB 770
25172 IP%(NS%+4)=VAL(SG.REPLY$):RETURN
25174 SG.DF$=SCR$
25176 SG.UP%=1:SG.ROW%=23:SG.COL%=53:SG.LNG%=1
25177 COLOR 15,1  :GOSUB 775
25178 SCR$=SG.REPLY$:RETURN
25238 GOSUB 1462
25249 GOSUB 25048      'IP%(NS%)
25250 IF SG.BTAB%>2 THEN GOTO 25408
25254 IF NO.ENVIR% = 1 THEN GOTO 25291
25259 GOSUB 25078  SG.BTAB%=1 THEN GOTO 25249'IP%(NS%+1)
25260 IF SG.BTAB%>2 THEN GOTO 25408
25264 IF NO.ENVIR% = 2 THEN GOTO 25291
25269 GOSUB 25108  SG.BTAB%=1 THEN GOTO 25259'IP%(NS%+2)
25270 IF SG.BTAB%>2 THEN GOTO 25408
25274 IF NO.ENVIR% = 3 THEN GOTO 25291
25279 GOSUB 25138  SG.BTAB%=1 THEN GOTO 25269'IP%(NS%+3)
25280 IF SG.BTAB%>2 THEN GOTO 25408
25284 IF NO.ENVIR% = 4 THEN GOTO 25291
25289 GOSUB 25168  SG.BTAB%=1 THEN GOTO 25279'IP%(NS%+4)
25290 IF SG.BTAB%>2 THEN GOTO 25408
25291 GOSUB 25174                                    'SCR$
25292 IF SG.BTAB%=1 AND NO.ENVIR%=1 THEN GOTO 25249
25293 IF SG.BTAB%=1 AND NO.ENVIR%=2 THEN GOTO 25259
25294 IF SG.BTAB%=1 AND NO.ENVIR%=3 THEN GOTO 25269
25295 IF SG.BTAB%=1 AND NO.ENVIR%=4 THEN GOTO 25279
25296 IF SG.BTAB%=1 AND NO.ENVIR%=5 THEN GOTO 25289
25297 IF SG.BTAB%>2 THEN RETURN
25298 RETURN
25302 GOSUB 1462
25303 LOCATE 4,20:COLOR 10,0:PRINT USING "######";KK%;
25305 LOCATE 12,2:COLOR 6,0
25306 PRINT USING "\              \";NO.ENVIR$(1);
25307 LOCATE 12,25:COLOR 6,0:PRINT USING "#####";RA(1);
25309 LOCATE 12,37:COLOR 6,0
25310 PRINT USING "#####";DEMAND%(1)  ;
25311 LOCATE 12,49:COLOR 6,0
25312 PRINT USING "#####";SUMSTREAM(1);
25313 LOCATE 12,66:COLOR 15,1:PRINT USING "#####";IP%(NS%);
25314 IF NO.ENVIR%=1 THEN GOTO 25355
25315 LOCATE 14,2:COLOR 6,0
25316 PRINT USING "\              \";NO.ENVIR$(2);
25317 LOCATE 14,25:COLOR 6,0:PRINT USING "#####";RA(2);
25319 LOCATE 14,37:COLOR 6,0
25320 PRINT USING "#####";DEMAND%(2);
25321 LOCATE 14,49:COLOR 6,0
25322 PRINT USING "#####";SUMSTREAM(2);
25323 LOCATE 14,66:COLOR 15,1
25324 PRINT USING "#####";IP%(NS%+1);
25325 IF NO.ENVIR%=2 THEN GOTO 25355
25326 LOCATE 16,2:COLOR 6,0
25327 PRINT USING "\              \";NO.ENVIR$(3);
25328 LOCATE 16,25:COLOR 6,0:PRINT USING "#####";RA(3);
25329 LOCATE 16,37:COLOR 6,0
25330 PRINT USING "#####";DEMAND%(3);
25331 LOCATE 16,49:COLOR 6,0
25332 PRINT USING "#####";SUMSTREAM(3);
25333 LOCATE 16,66:COLOR 15,1
25334 PRINT USING "#####";IP%(NS%+2);
25335 IF NO.ENVIR%=3 THEN GOTO 25355
25336 LOCATE 18,2:COLOR 6,0
25337 PRINT USING "\              \";NO.ENVIR$(4);
```

```
25338 LOCATE 18,25:COLOR 6,0:PRINT USING "#####";RA(4);
25339 LOCATE 18,37:COLOR 6,0
25340 PRINT USING "#####";DEMAND%(4);
25341 LOCATE 18,49:COLOR 6,0
25342 PRINT USING "#####";SUMSTREAM(4);
25343 LOCATE 18,66:COLOR 15,1
25344 PRINT USING "#####";IP%(NS%+3);
25345 IF NO.ENVIR%=4 THEN GOTO 25355
25346 LOCATE 20,2:COLOR 6,0
25347 PRINT USING "\                    \";NO.ENVIR$(5);
25348 LOCATE 20,25:COLOR 6,0:PRINT USING "#####";RA(5);
25349 LOCATE 20,37:COLOR 6,0
25350 PRINT USING "#####";DEMAND%(5);
25351 LOCATE 20,49:COLOR 6,0
25352 PRINT USING "#####";SUMSTREAM(5);
25353 LOCATE 20,66:COLOR 15,1
25354 PRINT USING "#####";IP%(NS%+4);
25355 LOCATE 23,53:COLOR 15,1:PRINT USING "!";SCR$;
25357 RETURN
25358 GOSUB 25364
25360 GOSUB 25303
25362 RETURN
25363 LOCATE 25,1:COLOR , 0:PRINT" ";
25364 COLOR ,, 8:CLS:GOSUB 1462
25365 COLOR  2,0
25366 COLOR 3,0:LOCATE 2,16:PRINT "DIVIDE  PROCUREMENT"
                          "INTO  FAILURE  ENVIRONMENTS";
25368 COLOR 3,0:LOCATE 4,26
25369 PRINT " units have just been procured.";
25370 LOCATE 6,1:PRINT " Divide this procurement"
                      "into failure environments:";
25372 COLOR 2,0:LOCATE 8,2:PRINT "  Retirement    Demand"
                                    "# Units";
25374 COLOR 2,0:LOCATE 8,63:PRINT "# Units to be";
25376 COLOR 2,0:LOCATE 9,2:PRINT "Failure Environment";
25378 COLOR 2,0:LOCATE 9,27:PRINT "Age";
25380 COLOR 2,0:LOCATE 9,45:PRINT " Presently in";
25382 COLOR 2,0:LOCATE 9,63:PRINT "Placed in This";
25384 COLOR 2,0:LOCATE 10,22:PRINT "(in periods)";
25386 COLOR 2,0:LOCATE 10,36:PRINT "(#units)";
25388 COLOR 2,0:LOCATE 10,48:PRINT "Operation";
25390 COLOR 2,0:LOCATE 10,64:PRINT "Environment";
25392 FOR X% = 1 TO NO.ENVIR%
25394 COLOR 15,1:LOCATE 10+X%*2,66:PRINT "     ";
25396 NEXT X%
25402 COLOR 3,0:LOCATE 23,1
25403 PRINT "Press return to continue";
25404 COLOR 15,1:LOCATE 23,53:PRINT " ";
25406 COLOR 2,0:RETURN
25408 IF SG.BTAB% < 5 THEN RETURN ELSE GOTO 25291
```

```
10 '
20 '     PROGRAM LISTING "PRINTOUT"
30 '
40 '
50 '
36022 '
36023 '     PRINT PERIOD BY PERIOD AVAILABILITY OF END ITEM
36024 '     ____________________________________________________
36025 '
36030 LPRINT "                    AVAILABILITY OF END ITEM"
        "(SIDE-LOADABLE WARPING TUG)":  LPRINT:  LPRINT
36040 LPRINT "                Period          Support"
                      "Combat          Total System"
36050 LPRINT "                  ------          -------"
                 "------          ------------":  LPRINT
36060 F52$ = "                  ##            #.######"
                      " #.######            #.######"
36070 FOR N% = 1 TO LMR%
36080   LPRINT USING F52$;N%;GA1(N%,1);
                          GA1(N%,2);TOTALAVAIL(N%)
36090 NEXT
36100 CLS:  LOCATE 10,20
36101 PRINT "set printer:  press any key to continue"
36110 A$ = INKEY$:  IF A$ = "" THEN 36110
36192 '
36193 '     PRINT PERIOD BY PERIOD COST SUMMARY
36194 '     ____________________________________
36195 '
36200 LPRINT "SUPPORT AREA PERIOD BY PERIOD COST SUMMARY"
36201 LPRINT:  LPRINT
36210 LPRINT "                Period   Capital   Repair"
                      "Shortage   Inventory   Total"
36220 LPRINT "                            Cost $    Cost $"
                       "Cost $     Cost $    Cost $"
36230 LPRINT "                  ------   -------   ------"
                 "--------   ---------   -----":  LPRINT
36240 F53$ = "                  ##    ########   #######"
                      "#######    #######  ########"
36250 FOR N% = 1 TO 12
36260   LPRINT USING F53$;N%;GCOST(N%,1);GCOST(N%,2);
        GCOST(N%,3);GCOST(N%,4);GCOST(N%,5)
36270 NEXT
36280 LPRINT:  LPRINT:  LPRINT:  LPRINT "          Period"
        "Procurement    # Units    # Units    Operating"
36290 LPRINT "                              Cost $       Start"
                                 "End    Cost/Unit $"
36300 LPRINT "                ------    -----------    -------"
                  "-------   ----------":  LPRINT
36310 F54$ = "                  ##         ######       ####"
                                 "####     ########"
36320 FOR N% = 1 TO 12
36330   LPRINT USING F54$;N%;PCOST(N%);GCOST(N%,6);
                          GCOST(N%,7);GCOST(N%,8)
36340 NEXT
36400 CLS:  LOCATE 10,20
36401 PRINT "set printer:  press any key to continue"
36410 A$ = INKEY$:  IF A$ = "" THEN 36410
36500 LPRINT "COMBAT ZONE PERIOD BY PERIOD COST SUMMARY"
36510 LPRINT "                Period   Capital   Repair"
                      "Shortage   Inventory   Total"
36520 LPRINT "                            Cost $    Cost $"
                       "Cost $     Cost $    Cost $"
36530 LPRINT "                  ------   -------   ------"
                 "--------   ---------   -----":  LPRINT
36550 FOR N% = 13 TO 24
36560   LPRINT USING F53$;N%;GCOST(N%,1);GCOST(N%,2);
                 GCOST(N%,3);GCOST(N%,4);GCOST(N%,5)
36570 NEXT
36580 LPRINT:  LPRINT:  LPRINT:  LPRINT "          Period"
        "Procurement    # Units    # Units    Operating"
36590 LPRINT "                              Cost $       Start"
                                  END    COST/UNIT $"
36595 LPRINT "                ------    -----------    -------"
```

```
                          -------   -----------":  LPRINT
36600 FOR N% = 13 TO 24
36610   LPRINT USING F54$;N%;PCOST(N%);GCOST(N%,6);
                                GCOST(N%,7);GCOST(N%,8)
36620 NEXT
36630 CLS:  LOCATE 10,20
36631 PRINT "set printer:  press any key to continue"
36640 A$ = INKEY$:  IF A$ = "" THEN 36640
36680 '
36681 '     PRINT PERIOD BY PERIOD MANPOWER REQUIREMENTS
36682 '     ---------------------------------------------
36683 '
36700 LPRINT "PERIOD BY PERIOD MANPOWER REQUIREMENTS"
36710 LPRINT "Period       Repair      Repair      Repair"
36720 LPRINT "            Level 1     Level 2     Level 3"
36730 F55$ = "            \     \     \     \     \     \"
36740 LPRINT USING F55$;LTP$;LTP$;LTP$
36750 LPRINT "------      -------     -------     -------"
36760 F56$ = "  ##      ######.##   ######.##  ######.##"
36770 FOR N% = 1 TO 12
36780   LPRINT USING F56$;N%;GMP(N%,1);GMP(N%,2);GMP(N%,3)
36790 NEXT
36791 '
36792 '     CHAIN TO "INPUT2" SO THAT USER MAY CHOOSE
36793 '     WHETHER OR NOT TO BEGIN ANOTHER MODEL RUN
36794 '     ------------------------------------------
36795 '
36800 CHAIN "INPUT2",90,ALL
```

APPENDIX 3.

Operator's Guide

OPERATOR'S GUIDE FOR

MRPM

MAINTENANCE REQUIREMENTS PLANNING MODEL

for the IBM-PC

Developed by:

Department of Industrial Engineering and Operations Research

Virginia Tech

Blacksburg, Virginia 24060

TABLE OF CONTENTS

Section 1: INTRODUCTION

MRPM is a user friendly computer software package designed for use by Logistics Planners. MRPM will assist in the evaluation of maintenance requirements generated by repairable item populations. The population may be operating under conditions where steady-state or nonsteady-state maintenance requirements are being generated.

This operator's guide" will illustrate all introductory, input, and output screens that the user of the model may deal with. Key selections available for use by the user of the model are detailed. The "Options" section of this guide gives further information on options that the user may select when using the model. Finally, the model Output Summaries are described.

Section 2: GETTING STARTED

MRPM is programmed in Microsoft Advanced Basic (Basica), and is compatible with an IBM-PC with a color monitor. A PC with two 5 1/4" disk drives is required. Follow these steps to get MRPM running:

1) "Boot" your PC with DOS 2.1 (only versions of DOS that will support Basica may be used)

2) Place the **MRPM** diskette in disk drive b: (right hand or bottom disk drive).

3) Place the **DATA** diskette in disk drive a: (left hand or top disk drive).

4) If the PC has been "booted" there should be an "A>" showing on the CRT screen. Beside "A>", type BASICA and <return>.

5) When the Basica prompt "OK" appears, type LOAD "B:MAIN" and <return>.

6) Type RUN and <return>.

MRPM is now running. The first screen that you will see is screen 1, described in the next section.

Section 3: INTRODUCTORY SCREENS

There are 6 introductory screens included in MRPM. The screens are intended to give the user of the model an overview of the terminology and concepts employed by MRPM. The 6 screens are shown, beginning with Screen 1 on this page, and should be self explanatory. Key selection available for use with the introductory screens is detailed on the next page.

```
MAINTENANCE  REQUIREMENTS  PLANNING
                FOR
REPAIRABLE  ITEM  POPULATIONS

MARCH  1985
REVISIONS:
_____________
_____________

Enter F to go forward to the next screen ___
```

INTRO SCREEN 1

```
                MULTI - STREAM CONCEPT
          T1        T2        T3        T4        T5

AGE3                        .1a.1b    .2a.2b    .3a.3b
AGE2              .1a.1b    .2a.2b    .3a.3b    .4a.4b
AGE1    1a 1b      2a 2b     3a 3b     4a 4b     5a 5b

T1,T2,.......  = time period    AGE1,AGE2,...  = age of end items
1a 1b........  = end items of the same age, operating in failure
                 environments A and B
1a.1a.1a.....  = a stream  (A stream is no more when either all end
                 items have failed or all end items are retired.)
Enter one of the following (R to go back one screen, F to go forward) __
```

INTRO SCREEN 2

MAINTENANCE TREE CONCEPT

The repairable items being evaluated by this logistics model are referred to as end items. Each end item is broken into repairable or replaceable components and then into subcomponents and so on until a spare part level is reached. This "breakdown" produces a maintenance tree. See the next screen for a sample maintenance tree.

Maintenance Tree Terminology:

node - each component on the tree represents a node

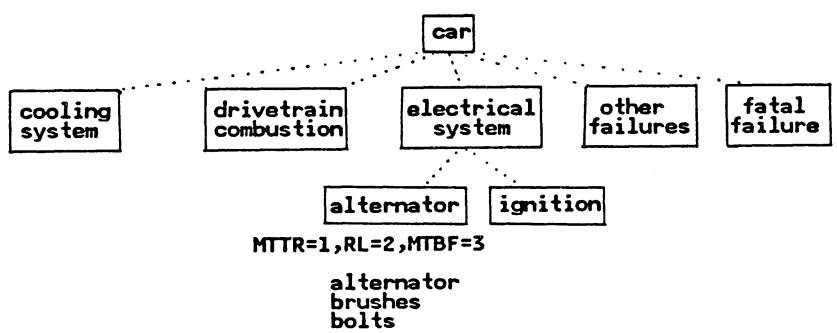
bottom of tree node - this is the node where failure and repair data is known

stockable items - parts needed to repair the component listed at bottom of the tree node

repair level - maintenance facility which must perform the repair

Enter one of the following (R to go back one screen, F to go forward) __

INTRO SCREEN 3



Enter one of the following (R to go back one screen, F to go forward) __

INTRO SCREEN 4

Key Selection for all INTRO Screens:

* R <return> .....display previous screen

* F <return> .....display next screen

```
The previous screen was a partial maintenance tree for an automobile
explanations:
alternator and ignition        - were bottom of the tree nodes
alternator, brushes, and bolts - were stockable items
MTTR                           - mean time to repair
MTBF                           - mean time between failures
RL                             - repair level

The next screen shows the maintenance tree limitations for this model.
max. # nodes = 126                max. levels of indenture = 5

Enter one of the following (R to go back one screen, F to go forward) __
```

INTRO SCREEN 5

```
                 graphical depiction of maintenance tree
                          (not shown here)

Enter one of the following (R to go back one screen, F to go forward)__
```

INTRO SCREEN 6

Section 4: INPUT SCREENS

The input screens are the screens of MRPM designed to accept input from the model user. The input may be data concerning the population being evaluated or model option choices.

It is important that the model user follow the directions displayed on each input screen.

Each input screen is illustrated in this section. Key selection for each screen is also detailed.

MRPM displays the screens in the same order that they are displayed in this chapter.

```
DATA  CHOICE

1...........Work Example
2...........Save Data
3...........Continue - All New Inputs
4...........Review Intro Screens
5...........End Program

Enter 1,2,3,4, or 5 >
```

INPUT SCREEN 1

```
PRINTER  CHOICE

1........Printer Listings of all Model Outputs
2........Printer Listings of Summaries Only -
         Viewing of Screen Output Allowed
3........Printer Listings of Summaries Only -
         Viewing of Screen Output Not Allowed

Enter 1, 2, or 3 > __
```

INPUT SCREEN 2

See Section 6 of this guide for further explanation of the model options for input screens 1 and 2.

Key Selection for input screens 1 and 2:

* type the number of your selection and <return>.

```
This model will produce period by period maintenance requirements.
Will a period be defined in days, weeks, months, or years? ________
How long will each time period be (# of dys, wks, mths, yrs)? ______
How many periods should the model run?                        ______

Operating environment information:
It is possible for units of the same age to have different failure rates
due to the respective environments in which the units operate.
In how many different failure environments will your population of
repairable units operate? (limit of 4 environments)            ______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 3

Input data may be entered only in the areas marked as blank lines on the above illustration. On the color monitor of your IBM-PC, the input areas will appear as bright blue boxes. The cursor may not be moved to any point outside these boxes.

Key Selection for Input Screen 3:

```
* R <return> .........display previous screen
* F <return> .........display next screen
*   <F9>    .........move cursor to last input area on
                       screen
*   <F10>   .........execute command contained in last
                       input area
*   <pg up> .........move to previous input area
*   <pg dn> .........move to next input area
*   <enter> .........move to next input area
```

note: Always type <return> after data is input in an input area.

This set of key selections will be referred to as **Key Selection A** as it will apply to other input screens.

```
                    COST  DATA  INPUT

The following cost information will be required in order to calculate the
costs involved with implementing the logistics policies you choose:

1) First cost (of an end item)                                  $_______

2) Salvage value (of an end item)                               $_______

3) Procurement costs (i.e. the administrative costs incurred
     each time a procurement of end items occurs)               $_______

4) Repair cost per day (include labor, transportation, and
   overhead costs that are incurred, on the average, each time
     an end item is serviced by a repair facility)              $_____/day

5) Shortage cost per day   (the cost incurred when there are
     not enough end items in operation to satisfy the demand) $_____/day

6) Interest rate per period  (time value of money)              _______%

Enter one of the following (r to go back one screen, f to go forward) __
```

INPUT SCREEN 4

```
                    CHOICE  OF  MODEL  MODE

A....The model will be used to build and evaluate a proposed population.
B....The model will be used to evaluate a population in which some units
     are already in operation.

                    Enter either A or B     ____

       CHOICE  OF  SPARE PARTS  AND  WAREHOUSE  REQUIREMENTS  OUTPUT

1......Compute spare parts and warehouse requirements
2......Compute spare parts and warehouse requirements and replenish
       inventory using EOQ
3......Do not compute spare parts and warehouse requirements

                    Enter either 1,2, or 3     ___

          NUMBER  OF  REPAIR  LEVELS  IN  MAINTENANCE  SYSTEM

          Enter # of repair levels (maximum 5)     ___

Enter one of the following (r to go back one screen, f to go forward) __
```

INPUT SCREEN 5

See Section 6 of this guide for a discussion of the Model Mode and Spare Parts and Warehouse options.

Key Selection for Input Screens 4 and 5:

* use Key Selection A

If the Data Choice input by the model user on Input Screen 1 was 1 or 2 then the next Input Screen displayed will be Input Screen 6a shown below. This Input Screen will be shown in lieu of Input Screens 6 and 7.1 - 7.3 which are described on the following pages.

Input Screen 6a summarizes the "bottom of the tree" node data that is already in MRPM's memory whether the data remains from a previous model run or whether the data is the data from the example problem that the model can run. The user may make changes to the data by typing over the displayed data on Input Screen 6a.

```
NODE DATA FOR FAILURE ENVIRONMENT nnnnnnn

Node      Node Name        Failure Curve      MTBF      MTTR      RL

#         nnnnnnnnnnnn          #             ###.##    ###.##    #
#         nnnnnnnnnnnn          #             ###.##    ###.##    #
.            .                  .               .         .       .
.            .                  .               .         .       .
.            .                  .               .         .       .

                    Enter F to go forward
```

INPUT SCREEN 6a

Key Selection for Input Screen 6a:
* Use Key Selection A. The one exception is that the R command may not be used. It is not possible to return to a previous screen.

```
                    LIBRARY  OF  FAILURE  CURVES


1) Uniform   2) Exponential   3) Exp. Increase   4) Normal   5) Bathtub

     The above curves are cumulative distributions of failure rate
probabilities.  To interpret the curves read the x axis as time and the
y axis as the probability of failure.  Therefore, the curves represent
the probability of failure of an item during that item's life cycle.

If one of these curves appears to represent the failure characteristics
of the item that you are evaluating, enter either 1,2,3,4, or 5 on the
next screen when you are asked for the failure rate curve shape.  The
model will compute period by period failure probabilities automatically.
If you have period by period failure data available for certain items,
enter 6 for failure rate curve shape and 0 for MTBF.  Input failure data
later.

Enter one of the following (r to go back one screen, f to go forward) __
```

INPUT SCREEN 6

Input screen 6 gives a description of the user's choices in describing the failure rate of each of the "bottom of the tree" node components that make up the end item.

Before proceeding through this section of input curves, the model user should construct a maintenance tree and identify each node of the tree by component name. Failure data must be gathered for each "bottom of the tree node" so that it may be input on the next series of screens.

Input screen 7.1 lists the input data that will be required for each "bottom of the tree" node. More detail on the structure of the maintenance tree is contained in the Introductory Screens. (See Section 3 of this guide.)

Key Selection for Input Screen 6:

R <return> .....display previous screen
F <return> .....display next screen

note: When you enter F and move to the next screen, you will not be able to further access Input Screens 1-6 during this model run.

```
                LIBRARY  OF  FAILURE  CURVES



1) Uniform   2) Exponential   3) Exp. Increase   4)  Normal   5) Bathtub

              INPUT MAINTENANCE TREE DATA BY FAILURE ENVIRONMENT

Enter a 1 or 2 word description of the operating environment for units
operating in failure environment # n   >  ____________

You will be asked to enter known failure rate information about the item
being evaluated. The information will be entered by maintenance tree node

You must input the following information when prompted:

Node #                          Node Name              Failure Rate Curve Shape
Mean Time Between Failures      Mean Time to Repair    Repair Level

                                                   press any key to continue
```

INPUT SCREEN 7.1

```
                LIBRARY  OF  FAILURE  CURVES



1) Uniform   2) Exponential   3) Exp. Increase   4)  Normal   5) Bathtub

Node 6 represents the chance of fatal failure of the end item.
Data for this node must be provided - input as prompted, then continue
with the input of information for other bottom of the tree nodes.

Node# 6                 Fatal Failure          Failure rate curve shape > __
MTBF > ____/per.unit       MTTR and Repair Level not applicable

Node# > ____        Node Name > _____________  Failure rate curve shape > __
MTBF > ____/p.u.    MTTR > _____/p.u.          Repair Level > ____
```

INPUT SCREEN 7.2

Key Selection for Input Screens 7.1 and 7.2:

* type in data and <return>
* 0 <return> .....If a zero is entered as the value for "Node #" the model will move to Input Screen 7.3.

note: If an entry error is made for any "bottom of the tree" node, retype the node # beside "Node #" and <return>. Then reenter the Node Name, Failure rate curve shape, MTBF, MTTR, and Repair Level for that node.

```
                    FAILURE ENVIRONMENT # n  DATA INPUT (continued)
How many periods before end items are retired from operation?  ____
What is the book value of an end item at retirement?          $____
What is the demand for units operating in this environment?    ____


When you press return, data input for the next failure environment will
begin if there is another failure environment.  Otherwise, you will
advance to the next data input screen.

                          Press return to continue.
```

INPUT SCREEN 7.3

Key Selection for Input Screen 7.3:

* <pg up> .....move to previous input area

* <pg dn> .....move to next input area

* <F9> <return> .....move to next input screen

* <F10> .....move to next input screen

note: <return> after each input.

If the model user specified that the population of repairable items is operating in more than one environment, then the model will loop through Input Screens 7.1 - 7.3 until input has been received for each environment.

Each end item must have an identically structured maintenance tree, so on subsequent loops through Input Screens 7.1 - 7.3, the model will provide the node #, the node name, and the repair level. The user of the model inputs the Failure curve shape, MTBF, and MTTR for each "bottom of the tree" node.

```
                    PROCUREMENT  POLICY  SELECTION
This will be the policy that you wish to follow in procuring end items.
1.....Continuous Uniform Procurement
2.....Uniform Procurement Until Upper Limit is Reached
3.....One Initial Block Procurement
4.....Block Procurement with Replenishment as Units Fail Fatally
5.....Continuous Increasing Procurement Amounts
6.....Increasing Procurement Until Upper Limit is Reached
7.....Continuous Decreasing Procurement Amounts
8.....Decreasing Procurement Until Upper Limit is Reached
9.....Other                    Enter either 1,2,3,4,5,6,7,8,or 9 ____
Enter F to go forward __
```

INPUT SCREEN 8

Key Selection for Input Screens 8 and 9.1 - 9.9:

* Use Key Selection A.

Depending on the procurement policy selected, one of the Input Screens 9.1 - 9.9 will be displayed following Input Screen 8. If you first select a procurement policy and move to the appropriate Input Screen 9.i and then decide to select a different procurement policy, type R <return> to return to Input Screen 8. Then select a different procurement policy.

```
                    CONTINUOUS  UNIFORM  PROCUREMENT

    # End Items to be Procured per Procurement          _______

    # Periods Between Procurements                      _______

    # End Items in the Initial Procurement              _______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 9.1

```
          UNIFORM  PROCUREMENT  UNTIL  UPPER  LIMIT  IS  REACHED
1.....Maintain Population at Upper Limit
2.....Cease Procuring Completely When Upper Limit is Reached and Let
      Population Die Out
                                  Enter either 1 or 2  ___

Upper Population Limit                                  _______
# End Items to be Procured per Procurement              _______
# Periods Between Procurements                          _______
# End Items in Initial Procurement                      _______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 9.2

ONE INITIAL BLOCK PROCUREMENT

End Items in the Initial Block Procurement _____

Enter one of the following (R to go back one screen, F to go forward) __

INPUT SCREEN 9.3

BLOCK PROCUREMENT WITH REPLENISHMENT AS UNITS FAIL FATALLY

Periods Between Procurements _____

End Items in Initial Procurement _____

Enter one of the following (R to go back one screen, F to go forward) __

INPUT SCREEN 9.4

```
                    INCREASING  PROCUREMENT  AMOUNTS

        % Increase in Procurement Amounts          _______%

        # Periods Between Procurements              _______

        # End Items in Initial Procurement          _______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 9.5

```
       INCREASING  PROCUREMENT  UNTIL  UPPER  LIMIT  IS  REACHED
1.....Maintain Population at Upper Limit
2.....Cease Procuring Completely When Upper Limit is Reached and Let
      Population Die Out
                                Enter either 1 or 2  ___

Upper Population Limit                                 _______

% Increase in Procurement Amounts                      _______%

# Periods Between Procurements                         _______

# End Items in Initial Procurement                     _______

Enter one of the following (R to go back one screen, F to go forward)  __
```

INPUT SCREEN 9.6

```
                    DECREASING  PROCUREMENT  AMOUNTS

      % Decrease in Procurement Amounts              _______%

      # Periods Between Procurements                 _______

      # End Items in Initial Procurement             _______


Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 9.7

```
      DECREASING  PROCUREMENT  UNTIL  UPPER  LIMIT  IS  REACHED
1.....Maintain Population at Upper Limit
2.....Cease Procuring Completely When Upper Limit is Reached and Let
      Population Die Out
                                Enter either 1 or 2   ___

Upper Population                                     _______
% Decrease in Procurement Amounts                    _______%
# Periods Between Procurements                       _______
# End Items in Initial Procurement                   _______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 9.8

```
                    OTHER  PROCUREMENT  POLICY

Selecting this procurement policy means that you will be asked, at
the end of each period, if you wish to make a procurement.

If you answer yes, then you will be allowed to input the amount of
end items that you wish to procure at that time.

How many end items do you wish to procure initially?        ______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 9.9

When all procurement policy data has been entered, the model will display a stockable items input screen.

If the model is computing only the spare parts and warehouse demand, then Input Screen 10.1 will be displayed. If the model is computing spare parts and warehouse maintenance requirements using EOQ theory, then Input Screen 10.2 will be displayed.

An input screen will be displayed for each "bottom of the tree" node. MRPM will supply the node name at the top of the screen. Be sure to list the node component as a stockable item if it is a stockable item.

If MTRP is not asked to compute any spare parts and warehouse demands, then neither Input Screen 10.1 or 10.2 will be displayed.

LIST STOCKABLE ITEMS SO THAT SPARE PARTS REQUIREMENTS MAY BE CALCULATED

When needed, press the F9 key to advance to bottom of screen.

Node name: _____

	Stockable item	Part#	Quantity	Space Requirement(cu.ft)	Price
1.					
2.					
3.					
4.					
5.					
6.					
7.					
8.					
9.					
10.					

Enter F to continue __

INPUT SCREEN 10.1

LIST STOCKABLE ITEMS SO THAT SPARE PARTS REQUIREMENTS MAY BE CALCULATED

When needed, press the F9 key to advance to bottom of screen.

Node name: _____

	Stockable item	Part#	Quantity (#parts)	W'house Req. (cuft/part)	Unit Price $	Proc. Cost $	Holding Cost $	Lead Time (#per)
1.								
2.								
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								

Enter F to continue __

INPUT SCREEN 10.2

Key Selection for Input Screens 10.1 and 10.2:

* Use Key Selection A. There is one exception to Key Selection A. The command R may not be used as it is not possible to review a previous screen. Make sure all data is correct before leaving an input screen.

If the user specified that the repairable item population operates in more than one environment, then Input Screen 11 will be displayed next. MRPM will provide all data marked as n on the screen shown below.

This screen will be displayed again every time a procurement of end items is made and the model user will be given the opportunity to divide the newly procured end items into environments.

DIVIDE PROCUREMENT INTO FAILURE ENVIRONMENTS

n units have just been procured

Divide this procurement into failure environments:

Failure Environment	Retirement Age (in periods)	Demand (#units)	# Units Presently in Operation	# Units to be Placed in this Environment
nnnnnn	n	n	n	_____
nnnnnn	n	n	n	_____
nnnnnn	n	n	n	_____
nnnnnn	n	n	n	_____

Press return to continue

INPUT SCREEN 11

Key Selection for Input Screen 11:

* Use Key Selection A. There is one exception to Key Selection A. The command R may not be used as it is not possible to review a previous screen.

If the Model Mode selected was A, i.e., the model is being used to evaluate an all new population, then the input phase of the model ends here and model operation begins. If Model Mode B is being used, then Input Screens 12.1 - 12.3 will be displayed.

```
                    EXISTING  END  ITEMS  DATA  INPUT

Existing end items should be divided into groups depending on how many
periods these end items have been in operation.  You will be asked to
input the age of the end items and the total end items in each age
group.

You will then divide the end items in each age group into different
operating environments where applicable.

How many age groups of existing end items do you have?      _______

Age of group 1 in periods                                   _______

Total end items in age group 1                              _______

Divide total # end items          nnnnn   (demand=n)        _______
of age group 1 into               nnnnn   (demand=n)        _______
failure environments              nnnnn   (demand=n)        _______
                                  nnnnn   (demand=n)        _______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 12.1

```
                    EXISTING  END  ITEMS  DATA  INPUT  (cont.)

Age of group 2 in periods  _____   Total # end items in age group 2 _____

Divide total # end items          nnnnn   (demand=n)        _______
of age group 2 into               nnnnn   (demand=n)        _______
failure environments              nnnnn   (demand=n)        _______
                                  nnnnn   (demand=n)        _______

Age of group 3 in periods  _____   Total # end items in age group 3 _____

Divide total # end items          nnnnn   (demand=n)        _______
of age group 3 into               nnnnn   (demand=n)        _______
failure environments              nnnnn   (demand=n)        _______
                                  nnnnn   (demand=n)        _______

Enter one of the following (R to go back one screen, F to go forward) __
```

INPUT SCREEN 12.2

Key Selection for Input Screens 12.1 - 12.3:

* Use Key Selection A.

note: Using the R command while on Input Screen 12.1 will only review Input Screen 12.1. It is not possible to review any screens previous to that Input Screen.

note: Input Screen 12.3 is not shown, but is identical to Input Screen 12.2, except for the fact that it accepts input data for age groups 4 and 5. If the user does not specify that there are more than 3 age groups, then Input Screen 12.3 will not be needed and will not be displayed.

Section 5: OUTPUT SCREENS

After all data has been input by the model user, MRPM enters the operation phase. During the computations, certain output data is displayed on the screen to the model user.

If the model user chose Printer Option 2 on Input Screen 2, then MRPM will halt execution when each output screen is displayed. The model user must prompt MRPM to continue operation. If Printer Options 1 or 3 were chosen, then the only interaction between MRPM and the user, that may be required during the operation phase, is if end items are procured and must be divided into environments.

The following pages contain examples of the screen output generated by MRPM.

```
node( 7) = 0.00000                                Time Period #
node( 8) = 0.00000                                Stream #
node( 9) = 0.00000     node(2) = 0.00000          # Units Initially
node(10) = 0.00000                                procured #
node(11) = 0.00000                                # Units in Operation-start #
                                                  # Units in Operation-end   #
node(12) = 0.00000                                MTTR   1.00000
node(13) = 0.00000
node(14) = 0.00000     node(3) = 0.00000
node(15) = 0.00000
node(16) = 0.00000
                                                  node(1) = 0.00000
node(17) = 0.00000
node(18) = 0.00000
node(19) = 0.00000     node(4) = 0.00000          node(6) = 0.00000
node(20) = 0.00000
node(21) = 0.00000

node(22) = 0.00000
node(23) = 0.00000
node(24) = 0.00000     node(5) = 0.00000
node(25) = 0.00000
node(26) = 0.00000                        press any key to continue
```

OUTPUT SCREEN 1

This Output Screen is displayed every period for each stream of end items. See Intro Screen 2 for information on streams.

```
SPARE PARTS AND WAREHOUSE REQUIREMENTS, STREAM n , PERIOD n

Node        Stockable      Part#       Quantity       Repair
              Item                                     Level

 n          nnnnnnnnn       n            n             n
 n          nnnnnnnnn       n            n             n
 .            .             .            .             .
 .            .             .            .             .
 .            .             .            .             .

                                  press any key to continue
```

OUTPUT SCREEN 2

This Output Screen is displayed every period for each stream of end items.

```
PERIOD n  SPARE PARTS AND WAREHOUSE REQUIREMENT SUMMARY

                                          Spare Parts    Warehouse     Repair
Node      Stockable Item    Part#         Requirements   Requirements   Level

 n        nnnnnnnnn         n             n              n              n
 n        nnnnnnnnn         n             n              n              n
 .          .               .             .              .              .
 .          .               .             .              .              .
 .          .               .             .              .              .


  Total warehouse requirement repair level n = nn.nn
  Total warehouse requirement repair level n = nn.nn
          .
          .
          .


                                   press any key to continue
```

OUTPUT SCREEN 3.1

This Output Screen is displayed once at the end of each period if the model has been instructed to output only spare parts and warehouse demands.

```
PERIOD n SPARE PARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node    Part#   Demand   W'house   EOQ   Order   Order   Total   Repair
                         (cuft)         Level   (days)  Cost    Level
____    _____   ______   ______    ___   _____   _____   _____   _____

nnnnn   n       n        n         n     n       n       nn.nn   n
nnnnn   n       n        n         n     n       n       nn.nn   n
  .     .       .        .         .     .       .       .       .
  .     .       .        .         .     .       .       .       .
  .     .       .        .         .     .       .       .       .

Total warehouse requirement repair level n =   n
Total warehouse requirement repair level n =   n
          .
          .
          .
```

OUTPUT SCREEN 3.2

This Output Screen is displayed once at the end of each period if the model has been instructed to output spare parts and warehouse requirements using EOQ theory.

```
                    AVAILABILITY  DURING  TIME  PERIOD  #

Availability is the probability of there being enough units
in operation to meet the given demand.

    environment                          availability
    ___________                          ____________

      nnnnn                               #.######
      nnnnn                               #.######
      nnnnn                               #.######
      nnnnn                               #.######

The total system availability was #.######


                                         press any key to continue
```

OUTPUT SCREEN 4

```
PERIOD COST SUMMARY

Environment   Capital      Repair      Short.      Inven.      Total
               Cost        Cost        Cost        Cost        Cost

nnnnn         ##.##       ##.##       ##.##       ##.##       ##.##
nnnnn         ##.##       ##.##       ##.##       ##.##       ##.##
  .             .           .           .           .           .


Environment   Units Start   Units End   Operating Cost per Unit

nnnnn            ##            ##             ##.##
nnnnn            ##            ##             ##.##
  .               .             .               .


                           press any key to continue
```

OUTPUT SCREEN 5

Both of these Output Screens are displayed once at the end of each period.

```
                TOTAL  COST  SUMMARY
Environment     Capital     Repair     Shortage     Inventory     Total
                  Cost       Cost       Cost         Cost         Cost

nnnn            ##.##      ##.##      ##.##        ##.##        ##.##
nnnn            ##.##      ##.##      ##.##        ##.##        ##.##
 .                .          .          .            .            .
 .                .          .          .            .            .

Procurement costs per period =  ##.##

TOTALS          ##.##      ##.##      ##.##        ##.##        ##.##

                          press any key to continue
```

OUTPUT SCREEN 6

This Output Screen is displayed once at the end of the model run. After leaving this screen, the model begins to print Summaries 2-5.

Section 6: OPTIONS

Explanation of Options on Input Screen 1:

1.....Work Example. MRPM will analyze a sample population in order to familiarize the user with the model input and output screens. No summaries will be printed.

2.....Save Data. If a model run has just been completed and the user wishes to analyze the same population but only change a few variables, then this option may be chosen.

3.....Continue - All New Inputs. The user must input all data if this option is chosen. This option must be chosen for the first model run of any population evaluation other than the example population referred to in option 1.

4.....Review Intro Screens. This option may be used to review the Intro Screens.

5.....End Program. Use option 5 when all model runs have been completed. When the program ends, any data left from previous runs is "forgotten" by MRPM.

Explanation of Options on Input Screen 2:

1.....Printer Listings of all Model Outputs. This option will cause MRPM to operate continuously during the operation phase of the model. The only interaction between MRPM and the user will occur if end items are procured during the model run and the end items must be divided among different environments. Printer listings will be produced of all Output Screens and Output Summaries.

2.....Printer Listings of Summaries Only - Viewing of Screen Output Allowed. This option will produce printer listings of only the charts shown in the Output Summary section of this operator's guide. The user of the model must be interactive with MRPM throughout the operation phase of the model, as MRPM will halt execution when each Output Screen has been displayed.

3.....Printer Listings of Summaries Only - Viewing of Screen Output Not Allowed. This option will produce printer listings of only the charts shown in the Output Summary section of this operator's guide. The model will run continuously during the operation phase with no pauses during Output Screen display. The only interaction between MRPM and the user will occur if end items are procured during the model run and the end items must be divided among different environments.

Explanation of Choice of Model Mode on Input Screen 5:

A....The model will be used to build and evaluate a proposed population. The user should choose this option if there are presently no end items in operation that are to be considered by MRPM.

B....The model will be used to evaluate a population in which some units are already in operation. If MRPM will be used to evaluate future maintenance requirements for a population in which some of the end items are already in operation, then this option should be chosen. When this option is chosen, Input Screens 12.1 - 12.3 will be displayed for the user. These screens will allow the user to divide existing end items into streams.

Explanation of Spare Parts and Warehouse options on Input Screen 5:

1.....Compute spare parts and warehouse requirements. If this option is chosen, the user will be asked to input stockable item data through Input Screen 10.1. MRPM will display stream spare parts and warehouse requirements through Output Screen 2 and period spare parts and warehouse requirements through Output Screen 3.1. The printer listing provided by this option will be Summary 1.1.

2.....Compute spare parts and warehouse requirements and replenish inventory using EOQ. If this option is chosen, the user will be asked to input stockable item data through Input Screen 10.2. MRPM will display stream spare parts and warehouse requirements through Output Screen 2 and period spare parts and warehouse requirements through Output Screen 3.2. The printer listing provided by this option will be Summary 1.2.

3.....Do not compute spare parts and warehouse requirements. If this option is chosen, MRPM will not display Input Screens 10.1 or 10.2, Output Screens 2, 3.1, or 3.2, and Summaries 1.1 or 1.2.

Section 7: OUTPUT SUMMARIES

Output summaries are produced by printer.

Summary 1.1 (or 1.2) is produced during the operation of the model, so a printer must be ready to print during the operation phase of MRPM. The other Summaries are printed at the end of a model run.

After Summary 5 has been printed, MRPM returns the user to Input Screen 1.

```
PERIOD n  SPARE PARTS AND WAREHOUSE REQUIREMENT SUMMARY

                                        Spare Parts     Warehouse      Repair
Node        Stockable Item    Part#     Requirements    Requirements   Level

 n          nnnnnnnn          n         n               n              n
 n          nnnnnnnn          n         n               n              n
 .            .               .         .               .              .
 .            .               .         .               .              .
 .            .               .         .               .              .


 Total warehouse requirement repair level n = nn.nn
 Total warehouse requirement repair level n = nn.nn
            .
            .
            .

                                   press any key to continue
```

SUMMARY 1.1

This Summary is printed if the model has been instructed to output only spare parts and warehouse demands.

```
PERIOD n SPARE PARTS AND WAREHOUSE REQUIREMENT SUMMARY

Node      Part#     Demand     W'house     EOQ     Total      Repair
                               (cuft)             Cost       Level
____      _____     ______     ______      ___     _____      _____

nnnnn     n         n          n           n       nn.nn      n
nnnnn     n         n          n           n       nn.nn      n
 .         .         .          .           .        .          .
 .         .         .          .           .        .          .
 .         .         .          .           .        .          .

Total warehouse requirement repair level n =    n
Total warehouse requirement repair level n =    n
                    .
                    .
                    .
```

SUMMARY 1.2

This Summary is printed if the model has been instructed to output spare parts and warehouse requirements using EOQ theory.

PERIOD POINT PROBABILITIES OF FAILURE FOR NODE 1

Environment	nnnnnn	nnnnnn	nnnnnn	nnnnnn
PF (period)				
n	n	n	n	n
n	n	n	n	n
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

press any key to continue

SUMMARY 2

AVAILABILITY OF END ITEM

Period	nnnnn	nnnnn	nnnnn	nnnnn	Total System
n	n	n	n	n	n
n	n	n	n	n	n
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

press any key to continue

SUMMARY 3

PERIOD BY PERIOD MANPOWER REQUIREMENTS SUMMARY

Period	Repair Level 1 (man-days)	Repair Level 2 (man-days)	Repair Level 3 (man-days)	Repair Level 4 (man-days)	Repair Level 5 (man-days)
n	n	n	n	n	n
n	n	n	n	n	n
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

press any key to continue

SUMMARY 4

nnnnnnn COST SUMMARY

Period	Capital Cost $	Repair Cost $	Shortage Cost $	Inventory Cost $	Total Cost $
n	n	n	n	n	n
n	n	n	n	n	n
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Period	Procurement Cost $	# Units Start	# Units End	Operating Cost/Unit $
n	n	n	n	n
n	n	n	n	n
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

SUMMARY 5

The vita has been removed from the scanned document