SIMULATION OF PROBABILISTIC INVENTORY CONTROL MODEL

by

S. K. BATRA

B. E. (Electrical), University of Delhi, India, 1966

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

Approved by:

J. E. Grosh

Major Professor

TABLE OF CONTENTS

INTRODUCTION

The objective of this report is to write a simulation program in the FORTRAN language of a probabilistic inventory model and to investigate how the actual demand during a review period distribution and demand during a lead time distribution would behave when the variances of the demand and order lead time distributions are changed. This inventory problem is taken from Starr and Miller (12). It is assumed that customers follow an exponential arrival pattern with a normally distributed demand quantity. A delay has been incorporated between the warehouse and the posting department. The posting delay is assumed to follow a normal distribution.

In view of the compound distributions of the demand quantity (compound distribution of exponential and normal) and the demand during a lead time (compound distribution of exponential, normal and another normal), this model is very difficult to solve analytically, hence simulation was resorted to. The inventory policy used for this purpose was the S-s policy of inventory control. The S-s inventory policy is to order whenever the system stock falls below s, a quantity equal to S minus the system stock. System stock is the sum of the stock on hand and stock on order.

The reorder policy, that is the values of S and s, is not changed for all seven simulation runs made for investigating the distributions of the actual demand during lead time and of the actual demand during a review period. Also, the

values of the means of the known distributions are not changed, as it is obvious that changing the means will change the distributions to be investigated, but it is not known how the change in variances would effect the distributions. For runs 1 and 7 the distributions to be investigated are known and runs are made to check the model.

General Purpose Systems Simulator (GPSS) and SIMSCRIPT are the two languages used in industry for the purpose of solving the inventory problems of this type. For the IBM 1410, however, these languages can not be used. Hence the simulation program structure is written in the FORTRAN language similar to SIMSCRIPT, which is a better simulation language for this type of problem and is used in bigger computers like IBM 7090 etc..

We will first discuss why simulation has come into general use and the principal features required in a simulation program and then explain how the simulation made is similar to the SIMSCRIPT simulation language. The results obtained from the seven runs will then be discussed.

SIMULATION AND SIMULATION LANGUAGES

The term 'SIMULATION' is used to identify two distinct fields of research: (1) a "description" of the dynamic process underlying the behavior of an individual or system and there by understanding the present process. Examples of items are models of the ordering and pricing behavior of a department store buyer, the process of information flow and decision making in an organization and (2) a method of "evaluation" in which alternative courses of actions are examined in terms of their probable implications for the system under study for example a model of firm's distribution system capable of ascertaining the effects of various inventory policies upon lost sales.

In the most general sense, simulation means the representation of reality. Hence, verbal description or diagrammatic representation of some part of the real world constitutes simulation. Holstein and Soukup (7) observe, however, that these forms of simulations are not new. On the other hand

"If..... simulation necessarily involves the use of mathematical expressions and equations which closely approximate random fluctuations in the simulated system and which are so complex as to be impossible of solution without the aid of computers, then simulation is a very recent development".

In brief the purpose of this type (simulation) of computer models are:

1) To formulate theories which explain why organizations

run on the present system have behaved in particular ways.

2) To test these theories by comparing the observed past behavior with the simulated behavior generated by the model.

3) To predict how these organizations will behave in the future.

Simulation studies are thus carried out to explore reasonable assumptions in order that a system may be designed to function better under some stated conditions. In cases like these where models need to be very complex in order to be realistic, and defy solution in an analytical form, Monte Carlo simulations are resorted to.

The growth of Monte Carlo techniques as the growth of all computation dependent methods, was accelerated by the advance in digital computer technology. Monte Carlo computations had been made prior to the introduction of the digital computer, but it was the computer that provided the power for their expanded use.

The most important requirement in Monte Carlo simulations is the generation and use of random numbers. In an inventory system we generally know the probability distribution of the demand for a commodity or the arrival rate of customers etc.. Monte Carlo technique introduces this chance variation into the system by the generation of random numbers and using these random numbers to generate standard distributions (i.e. Normal,

exponential, Poisson etc.) based on our knowing the mean and/or standard deviation of these distributions. It is even possible to use a completely arbitrary or emprical distributions.

All simulations, whether they are written in assembly language, a general compiler FORTRAN or ALGOL or specialized simulation programming language have certain things in common.

Simulation models require a mechanism for the control and advancement of simulated time. Table 1 gives the important features of simulation programming languages. Some languages emphasize some features more than others, some provide greater capabilities in some areas than in others, but all provide some capability in each area.

An important factor in simulation is the flexibility of the program. The process of trying several different conditions or changes within an organization is inherent in simulation. These changes are typically within the decision process itself. Thus provision should be made to make these changes in the form of data at the object program level. This should reduce the need to recompile a program each time a change must be made.

Speed and efficiency are also a primary factors. Speed refers to the time required by a compiled decision system to make the correct decision. This time must be minimized without occupying excessive amounts of computer memory. Efficiency refers to the amount of computer memory used and speed of execution, since the memory must provide room for the compiled

TABLE 1

PRINCIPAL FEATURES OF SIMULATION LANGUAGES

GENERIC FEATURE	
1. Time Advance	Event, Activity or transaction flow organization. Coordination of parallel computations within a simulated system.
2. Date Generation	Generation of random numbers, generation of random variates from theoritical statistical distributions. Features for generating corelated data streams.
3. Data Structure	Flexible structuring of the data base of a system being modeled. Efficient storage of information. A readable description of the modeled system.
4. File Maintainance	Automatic filing of data into sets or queues according to the rules set forth. An efficient search mechanism for locating and removing filed information.
5. Data Output and Display	Convenient descriptive statistical and graphic reporting mechanism.
6. Specification of Initial System Conditions	Setting of all system elements to specify the initial state of a model and generation of initial random sequence to set the model in motion.

mainline program, and areas for working storage. The size of the main-line program, and the size of working areas are variable and depend upon the simulation being performed. Thus program space should be conserved by the proper use of subroutines and subprograms, which should be written in the most efficient manner and in a general enough form so that they may be used with other main line programs.

DISCRETE DIGITAL SIMULATION

In this report we deal with a particular type of simulation, dependent upon the use of numerical and logical models to represent discrete changes of state in a system as it moves through time. Simulation models of this type are usually formulated symbolically using flow diagrams which are then coded in some programming language and run on a digital computer. We shall briefly discuss here some discrete simulation languages and their important characteristics.

The major goal of any simulation is not simply to model the behavior of a system but also to obtain data and statistics on the performance of the model. It is therefore essential that the simulator supply the user with the type of data that he desires in a clearly understandable form. The method by which a simulator constructs and defines the model will have considerable impact upon its operational characteristics. Efficiency (speed of executation) is definitely of importance. However, the user also desires a simple procedures for:

1. Introducing changes into both the static and dynamic structure of a model.
2. Initializing the state of the model.
3. Altering the output to be generated.
4. Stacking a series of runs.

General Purpose Systems Simulator

The General Purpose Systems Simulator (GPSS) is one of the variety of discrete programming languages (6) designed

specifically for modeling queueing systems. GPSS bears some similarity to FORTRAN in that its statements include a number of English verbs such as GENERATE, HOLD, QUEUE, ADVANCE, PICK, TERMINATE etc..

GPSS develops simulation models in terms of BLOCK diagrams, which are graphical devices for portraying the physical and logical flow of transactions or information through a system. The blocks therefore represent certain activities of the system.

In the computer program which is composed from a GPSS block diagram, one block corresponds to one statement, and one statement is punched on one card.

Just as in FORTRAN, we call for subroutines to execute a certain activity, similarly in GPSS when for example control is sent to a QUEUE block, the GPSS compiler interprets the single queue statement to mean:

1. Compute and record the number and quantity of orders in the queue at this time.
2. Change the status of the queue to reflect the event which just occured.

In GPSS therefore the subroutines, the timing mechanism, the parts of the arrival pattern and service time generators (GENERATE) are built into the compiler as system functions.

In summary, the world-view employed by GPSS is shown in table 2. Each is referenced by one or more of the block types.

TABLE 2 World-View of GPSS

ENTITIES	
	Transactions
	Facilities
	Storages
ATTRIBUTES	
	Parameters
	System variables
ACTIVITIES	
	Blocks
CONDITIONS	
	Availability of equipment, Mathematical or logic expressions of system variables.
EVENT	
	Transfer of a transaction between blocks.

Practically all status changes in GPSS models occur as a result of a transaction entering one of the different block types, thereby causing the interpretive execution of the subroutine associated with the block type. New transactions are created and enter a GPSS model in GENERATE blocks. Transaction are destroyed when they enter a TERMINATE block. Transactions in one model can be written on a specified tape via the WRITE blocks.

'Facilities' are used in GPSS to represent unit capacity

equipment. When a transaction enters a SEIZE block with reference to a particular facility, the transaction denies entry into SEIZE blocks (associated with that facility) to all other transactions until the first transaction enters a RELEASE block. 'Storages' are used in GPSS model to represent equipment which can have a capacity between 1 and 32767 (on IBM 7090). Consequently, multiple transactions can move into ENTER blocks referencing a particular storage and thereby increase the current contents of the storage. Transactions entering LEAVE blocks reduce the contents of storage. When the contents of a storage are equal to the defined capacity, no further transaction can move into ENTER blocks associated with that storage.

Dynamic sequencing of events is provided by the overall GPSS scan which at each clock time continually recycles through the current events chain attempting to move transactions into some possible next block. When the overall GPSS scan succeeds in moving a transaction into some next block the scan attempts to keep the transaction moving through as many blocks as possible until one of three things happens:

1. The transaction encounters a positive time delay in a block, and is merged back into the future events chain.
2. The transaction is finally blocked from entering a next block and remains in the current events chain.
3. The transaction is destroyed in a TERMINATE or ASSEMBLE block.

GPSS concentrates almost exclusively on processing the single transaction which it is currently moving. Other transactions are only indirectly affected.

In the GPSS language, the creation of transactions and events and activities are completely programmed for the user hence the simplicity of using GPSS. Further the rules for the sequencing of events and the expressing of conditions are described in the block diagram description of the system being simulated and need no further programming. The restrictions of GPSS are due to the fact that flexibility in describing entities, activities and conditions has been lost by limiting the choice to fixed concepts.

SIMSCRIPT

Another simulation language is SIMSCRIPT. It has a much more extensive capability for constructing models than GPSS. It needs however a greater level of programming skill for its use.

SIMSCRIPT is based upon the notion that the state of a system can be described in terms of entities (the things or objects of which the system is composed), attributes (properties associated with entities) and sets (groups of entities). SIMSCRIPT uses the specific term ENTITY and distinguishes between temporary and permanent entities as being those that are created and destroyed during the simulation run and those that remain throughout the run. Temporary entities might correspond to "customers". Permanent entities might correspond

to "service facilities".

Thus in a machine shop, a machine group or a department through which work orders flow might be a "permanent entity" and orders themselves might be "temporary entities". In the machine shop setting an "attribute" of a department might be service time in that department; an order might be a temporary entity and attributes of an order might include the date of the order or its route through the shop. A work order in a machine shop would be part of a "set" of work orders.

SIMSCRIPT provides for defining "entities" (temporary and permanent) by both their "attributes" and the "sets" to which they belong or which they "own". That is, it would tell us that a certain customer number, owns such and such work orders. Similarly a department has so many work orders in a queue. In addition to the definition a segment of a SIMSCRIPT program, one also employs a variety of routines. Thus we may have a routine for describing the effect on the system of the receipt of customer's order; another routine may be for describing the arrival of a work order at a particular department.

The SIMSCRIPT language is a pure Event-centered simulation. That is it allows the dynamic creation of entities in the system and to study the events that these entities engage in. Some simulation languages like CSL (Control and Simulation Language) which we shall not discuss here are a

pure "activity language. The "activity" languages consider fixed amounts of equipment as basic elements of the system and study the cycle of activities in which these equipments engage in. In this program clocks are set that indicate the times at which the various elements involved in the activity will change state. There is no explicit scheduling done in an activity-centered system, a simulation executive program scans all activities each time one is completed to determine what to do next. All activities are processed, element clocks are set and the simulated time is advanced.

In contrast the 'event-centered' programs describe how the system status changes when these events occur. For example in a typical waiting line model there are arrival events, end of service events etc.. It is up to the programmer to provide event logic that schedules events to occur in the future whenever the system status allows or calls for it. Generally, future times are obtained by random sampling from statistical distributions.

Hence in SIMSCRIPT the state is modified by the occurrence of an event, which is a user defined subroutine written with SIMSCRIPT and FORTRAN statements. The state of the system may be changed in the following ways:

1. Alter a set membership of an individual entity.
2. Create and destroy an individual temporary entity.
3. Change the numerical values of an attribute.

The SIMSCRIPT world-view can hence be represented as shown below:

TABLE 3 World-View of SIMSCRIPT

ENTITIES	
	Temporary
	Permanent
ATTRIBUTES	
	Temporary
	Permanent
	Set relations
ACTIVITIES	
	Programs ⎫
EVENTS	⎬ Assisted by set of
	Programs ⎭ simulation-oriented routines.

Both the activities and events as noted in SIMSCRIPT must be programmed by the user, he can call upon FORTRAN in this programming.

Here we have considered two of the main simulation languages with little reference to another simulation language CSL in which the basic unit of structure is ACTIVITY in contrast to EVENT in SIMSCRIPT and BLOCK in case of GPSS.

These languages have been designed specially to minimize the programming time for simulations and can be used in a large scale computer. However in the IBM 1410 computer languages such as FORTRAN, AUTOCODER or COBOL must be used

in solving such problems. With a general purpose language such as FORTRAN one has the freedom of altering the simulations to suit another condition if it arises in the problem, whereas the special simulation languages force programmers to choose whatever method the particular language happens to use. For example SIMSCRIPT makes use of a variable time increament methods while GPSS makes use of fixed time increament methods. With FORTRAN one has freedom to choose either of these two methods, but pays the price for this flexibility by increased complexity of the program.

The special simulation languages are designed for the purpose of evaluating a special kind of model peculiar to each one. Their services depend upon the user's ability to understand the language faster than some other language and to get results in less computer time then by some other means. They are being extensively used on large scale computers by companies whose specific problems match those for which the language has been designed.

The main features of comparison between SIMSCRIPT, CSL and GPSS simulation languages are presented in table number 4.

TABLE 4

Comparison of Features Concerned with Dynamic Control of a Simulation Program

	SIMSCRIPT	CSL	GPSS
Structuring a Simulation Program.			
Basic unit of Structure.	Event	Activity	Block
Must basic unit be defined by user?	Yes	No	No
How are basic units tied together?	"Cause Event" statements.	Cycle through activities.	Next Block sections.
Can multiple occurrences of basic unit exist simultaneously?	Yes	Not automatically (at user's request)	Yes
Can basic unit take time?	No	Yes	Yes
Can basic unit be triggered external to model?	Yes	No	Yes

	SIMSCRIPT	CSL	GPSS
Sequence control of Simulation			
Timing Scan occurs	Next imminent event. After each event.	Next imminent event. Cyclially over all activities.	Next imminent event. After specified change in state.

CONTINUOUS DIGITAL SIMULATION

We have dealt with simulation languages for descrete systems. The problem that we have considered in this report belongs essentially to the discrete type. However in a manufacturing situation the aggregate inventory level may quite reasonably be treated as a continuous variable even though its detailed composition is discrete. At any instant in time the level of a variable is a single numeric value. DYNAMO is a simulation language which deals with these continuous closed information feed back systems. An information feedback system is one in which information concerning the state of the system at the present time is used in determining the future state of the system. Variables may be introduced in determining the future state of the system, and the flow of information. Time delays are important in establishing the dynamic performance of a system. They determine the lag between the time at which a change in one variable occurs and the time at which this change is reflected in one or more other variables. For example in a manufacturing system an increase in one level of the inventory may lead (through decision process) to a reduction in the level of orders for the system. In brief DYNAMO is directed at studying the stability (over time) of closed systems of continuous variables in which the broad characteristics of information feedback within the system are significant to its dynamic characteristics.

The DYNAMO program operates in the following six phases:

1. Input: The cards are read and internal tables are constructed for each equation. Each card includes an equation type specification which is used as a check against the equation actually programmed. Output specifications are established from PRINT and PLOT cards.

2. Generation: Machine coding is generated from skeleton instructions for each equation type. Additional checks are performed to avoid simultaneous equations and to establish a correct sequence of solution.

3. Running: At the outset of a run, the equation types concerned with initialization are solved to provide starting values for all variables requiring them.

4. Printing/Plotting: These phases Print/Plot the output data.

5. Rerun: Additional cards, modifying the values of defined constants or respecifying output requirements,

are read in. The program then returns to the running phase.

DYNAMO, lacks a flexible data structure and is limited to simulator defined equation types. Thus it appears to be the least powerful of the languages discussed to handle a problem of the type considered in this report.

PURPOSE AND PLAN

The purpose of this report is to investigate the problem of simulating an inventory control procedure with the aid of the IBM 1410 computor. None of the simulation languages previously discussed are available for the 1410, thus the central problem is how to formulate the desired simulation in the FORTRAN language. It is obvious from the preceeding pages that some sort of time advancing mechanism will be needed in addition to a random process generator to simulate the uncertainty of the inventory process. The random process generators use in this report will be discussed first. This will be followed by a discussion of possible alternate inventory control systems and there the central structure of this simulation plan will be developed.

Random Number Generator

The IBM 1410 computer systems on which this problem was done, has an autocoder routine written as a fortran function to generate five digit integer random numbers, (2) the calling sequence is IRANDM (ABCDE), where ABCDE is any five digit integer. The random numbers that are generated are uniformly distributed over the interval 0 to 99999. However in probabalistic inventory problems the demand distribution or arrival rate of customers is not a uniform distribution, therefore it is essential to generate-arithematically-a sequence of random numbers with some other specified distribution.

Exponential Variable Generator

If the arrival pattern is say a Poisson distribution (4) and the mean arrival rate has a specified value then we Know:

$$P_n = \frac{\lambda^n e^{-\lambda}}{n!} \quad , \; n = 0,1,2,....$$

This is a p.d.f. of a discrete type of random variable. In this distribution (Poisson) we limit the consideration to a random number of arrivals within an arbitrary but fixed period of duration 't'. $P_n(t)$ gives the probability of the number of arrivals in a unit time. The distribution depends on the continuous parameter (t).

The Poisson distribution giving the probability of the number of occurrences per unit time, Δt, is an excellant distribution for use with a simulation maodel formulated with a uniform time increament. Another option is the continuous case in which the probability of the time between arrivals is evaluated (Exponential).

The Discrete and Continuous time base for simulation can be best explained by taking an example and illustrating it by both distributions.

EXAMPLE 1. Discrete Time Base: As explained this can be best illustrated by the Poisson distribution:

$$P_n(t) = \frac{e^{-\lambda t} (\lambda t)^n}{n} \qquad (1)$$

Considering a stochastic process represented by an integral-valued random variable $X(t) \geqslant 0$. X(t) is the number of

arrivals say, a simple mathematical model can be arrived at by introducing two postulates. The increament X(t + s) - X(0) during the time interval from 0 to t + s is the sum of the increaments X(s) - X(0) and X(t +s) - X(s).

The two postulates are:

1. The increaments X(s) - X(0) and X(t +s) - X(s) are stochastically independent.
2. The distribution of X(t + s) - X(s) depends only on 't' (i.e. only on the length of the interval not on its position. This is a property of the homogenity of time).

Analytically, the independence of X(t + s) - X(s) and X(s) - X(0) is expressed by:

$$h_n(t + s) = \sum_{j=0}^{n} h_j(s) \cdot h_{n-j}(t) \qquad (2)$$

where $h_n(t)$ is the probability that X(t + s) - X(s) assumes the value n (where n = 0,1,2,3,.....).

This distribution with the property of equation 2 is the compound Poisson distribution, that is X(t) has the distribution of a random variable,

$$S_N \text{ with } P(N = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!} \qquad (3)$$

where $S_n = Y_1 + Y_2 + Y_3 + \ldots\ldots\ldots + Y_n$ is the sum of n mutually independent variables with the common distribution f_i. In our example f_i represents the probability distribution of demand for an individual customer's arrival in a time interval of length t.

Thus equation 3 has the same probability distribution as the demand $X(t + s) - X(s)$ during an arbitrary interval of length t, and the total demand is the sum of a random number N of individual demands. The number N of total arrivals has a Poisson distribution corresponding to equation 1 and the individual demand has a probability distribution f_i.

Equation 1, the Poisson distribution (known as Simple Poisson distribution) itself represents the special case where all demands are of unit quantity that is:

$$f_1 = 1, \; f_0 = f_2 = \ldots\ldots\ldots = 0.$$

EXAMPLE 2. Continuous Time Base: $p_n(t)$ in a Discrete time base was defined as a family of discrete probability distributions depending on t.

Consider the zero term of equation 1

$$P_0(t) = e^{-\lambda t}$$

signifies the probability that no arrival occurs within a period of duration t. This formulation suggests that $P_0(t)$ can be interpreted as the probability that the waiting time (starting at an arbitrary moment) up to the first arrival exceeds t. It can be shown that this interpretation is correct, and it is also seen that this would involve probabilities in a continuum.

The operational meaning of a discrete time base is thus: "Make a series of 'identical observations' with a fixed observational period t. Each trial results in either no arrival, or one or more arrivals".

With the continuous time base interpretation we are to wait until a call arrives. Every positive number is a possible waiting time. Therefore the formula,

$F^*(t) = 1 - F(t) = e^{-\lambda t}$ = probability of no arrivals by time t. (4)

represents a continuous probability distribution (Exponential distribution).

In designing an EXPONENTIAL generator (calculation of time between arrivals) which has been used in the computer program, we are interested in equation (4) which tells us about the distribution of time between arrivals. Therefore to generate exponentially distributed values for the time between arrivals, we need to read a random fraction and determine values of t from (4) which corresponds to it. That is solving 4 for 't',

$$t = -\frac{1}{\lambda} \cdot \ln F^*(t)$$

For generating purposes F(t) is a uniformly distributed random variable, lying between 0 and 1 (it will never be 1 as it is a decimal fraction). Hence we can generate expenential-distributed time between arrivals by putting a random fraction in place of $F^*(t)$.

A Normal Distributed Variable Generator

Many processes, especially in business and economic systems, may be approximated by the normal distribution and to study the behavior of these processes we find ourselves in frequent need of a normal generator. The probability distribution function for the normal process is,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \qquad (5)$$

where μ is mean and σ is the standard deviation and x is the random variable. We can form a cumulative desity function from equation 5 just as we did for the Exponential generator, but solving for 'x' in terms of a cumulative probability is not possible.

A number of schemes have been employed to design an arithematic function which would (approximately) generate normally distributed variables. A rather widely used generator of this sort is used here (1). This generator requires two different uniformly distributed random fractions to generate two random variables from the same rectangular density function on interval (0,1). Consider the random variables:

$$X_1 = (- 2 \ln R1)^{\frac{1}{2}} \text{ Cos } 2\pi R2$$

$$X_2 = (- 2 \ln R1)^{\frac{1}{2}} \text{ Sin } 2\pi R2$$

Then (X_1, X_2) will be a pair of independent random variables from the same normal distribution with mean zero and unit variance.

JUSTIFICATION: From equation 6 we get the inverse relationships:

$$R1 = e^{\frac{-(X_1^2 + X_2^2)}{2}}$$

$$R2 = -\frac{1}{2\pi} \times \text{arc tan } \frac{X_2}{X_1}$$

It follows that the joint density of X_1, X_2 is:

$$f(X_1, X_2) = \frac{1}{2\pi} e^{\frac{-(X_1 + X_2)}{2}} = \frac{1}{\sqrt{2\pi}} e^{\frac{-X_1}{2}} \cdot \frac{1}{\sqrt{2\pi}} e^{\frac{-X_2}{2}}$$

$$= f(X_1) \cdot f(X_2)$$

thus the desired conclusions including the independence of X_1 and X_2 is obtained. (1)

The above approach is based on the considerations: the probability density of $f(X_1, X_2)$ is a constant on circles thus it is uniformly distributed on $(0, 2\pi)$. Further the square of the length of the radius vector $r^2 = X_1^2 + X_2^2$ has a chi-square distribution with two degrees of freedom. If R (random number) has a rectangular density (0,1) then - 2 ln R has a chi-squared distribution with two degrees of freedom. Proceeding in the reverse order we arrive at equation number 6.

Variable X_1 is normally distributed with mean zero and standard deviation1, thus to convert to another variable with a given mean and a given standard deviation, we simply use:

New variable = (X_1) * Given std. dev. + Given mean.

Thus we can generate a normally distributed number, given the mean and standard deviation of the distribution. The above techniques are applied in this report for the generation of random variables.

INVENTORY CONTROL

Inventory theory deals with the determination of optimal procedures for procuring stocks of commodities to meet future demand. Inventory control is necessary as "An inventory is an idle resource of any kind, provided that such a resource has an economic value" (11). Hence by inventory control we mean the determination of the optimal level of such an idle resource. The two most important inventory problems are procurement of the commodity in question and the future demand.

The possibilities with regard to our knowledge of future demand can be categorized under three headings:

1. Deterministic-Knowing future demand exactly.
2. Probabilistic-Knowing the probability distribution of demand.
3. Uncertainty-Do not know anything about the future level of demands.

In this report we have taken the probabilistic case, i.e. we know the probability distribution of demand. In general, utter ignorance-the case of uncertainty-is just as rare as complete certainty of knowing the demand (11).

The procurement process has to be taken into consideration in inventory control, in view of the fact that there is a time lag between the placement of an order and when it is actually received into inventory. This time lag or lead time as it is called can be either a constant or a probability distribution of possible lead times.

Our decision of how much to order depends upon the distribution of demand during lead times. The demand distribution is usually a compound distribution of (i) the time between customer's arrival and (ii) customer's order size once he has arrived. This compounding of probability distributions is difficult to handle mathematically but it is easy to approximate with simulation. The compounding of probability distributory will appear in our model in the following manners:

1. Two Fold: (a) the time between customer's arrival (exponential distribution) and
   (b) the customer's order size once he has arrived (normal distribution) compound to form the customers demand distribution per unit time.
2. Three Fold: A compound distribution of the demand distribution in (1) and the posting delay distribution gives three fold distribution.
3. Four Fold: A compound distribution of:
   (a) demand distribution (which itself is two distributions).
   (b) Posting delay distribution.
   (c) Order lead time distribution.

Types of Inventory Control

There are usually three systems followed in Industry to decide when to place an order:

P-System

In the P-system of inventory control there is a fixed order period and a varying order size. The procedure is: at periodic intervals-the period being analytically determined-the amount of inventory is reviewed and an order is placed to replentish the stock. The total amount which should be on hand and on order is determined from the analysis thus the amount of the order is determined by direct subtraction of the amount on hand from this predetermined total. However when the order period turns out to be shorter than the lead time period the inventory must be counted as including units on hand plus units on order but not yet delivered. Thus the P-system is characterized by having a fixed order period and variable order quantity.

Q-System

Q-system has a fixed order size and a varying order period. The procedure is: whenever the stock on hand falls to a certain minimal level based on the order lead time of the item, an order is automatically placed for a predetermined amount. Such a system is completely determined by the order size and the minimal stock level which represents the signal to place an order. That is an order in the Q-system is placed whenever stock on hand plus stock on order falls below the reorder level.

The major difference between the P-system and the Q-system is in determining the reserve stock. Reserve stock

is the stock maintained over and above the stock for the average demand to protect ourselves against the fluctuations in demand and fluctuations in lead time. Since there is no flexibility in the time at which orders are placed in the P-system, any fluctuations in demand between orders must be met from the stock carried for that purpose. It is necessary to maintain a reserve stock as a protection against demand fluctuations both during the lead time and the review period. On the other hand, in the Q-system a reserve stock is maintained as a protection against demand fluctuations during the lead time only, the fluctuations between orders being absorbed by the variation of the order period.

s-S System

The s-S system is completely characterized by two parameters s and S. Here s is the stock level or reorder point at which we place an order. That is whenever (stock on hand + stock on order) s(reorder level), an order is placed equal to:

Q = S - (stock on hand - stock on order).

Here S is a maximum quantity that has been decided upon by management. Thus the s-S system is a blending of the P-and the Q-systems.

In this report the simulated system is based on the s-S policy. However in case it is needed to convert this to the P-system or to the Q-system, all that would be necessary is to change a few cards in two of the subroutines (subroutines Post and Book).

PROBABILISTIC INVENTORY CONTROL

As indicated earlier, after we choose a probabilistic inventory model, our next objective is to build a structure on which the decision rules for this model may be tested.

A generalized of the system we are interested is shown in the figure below:

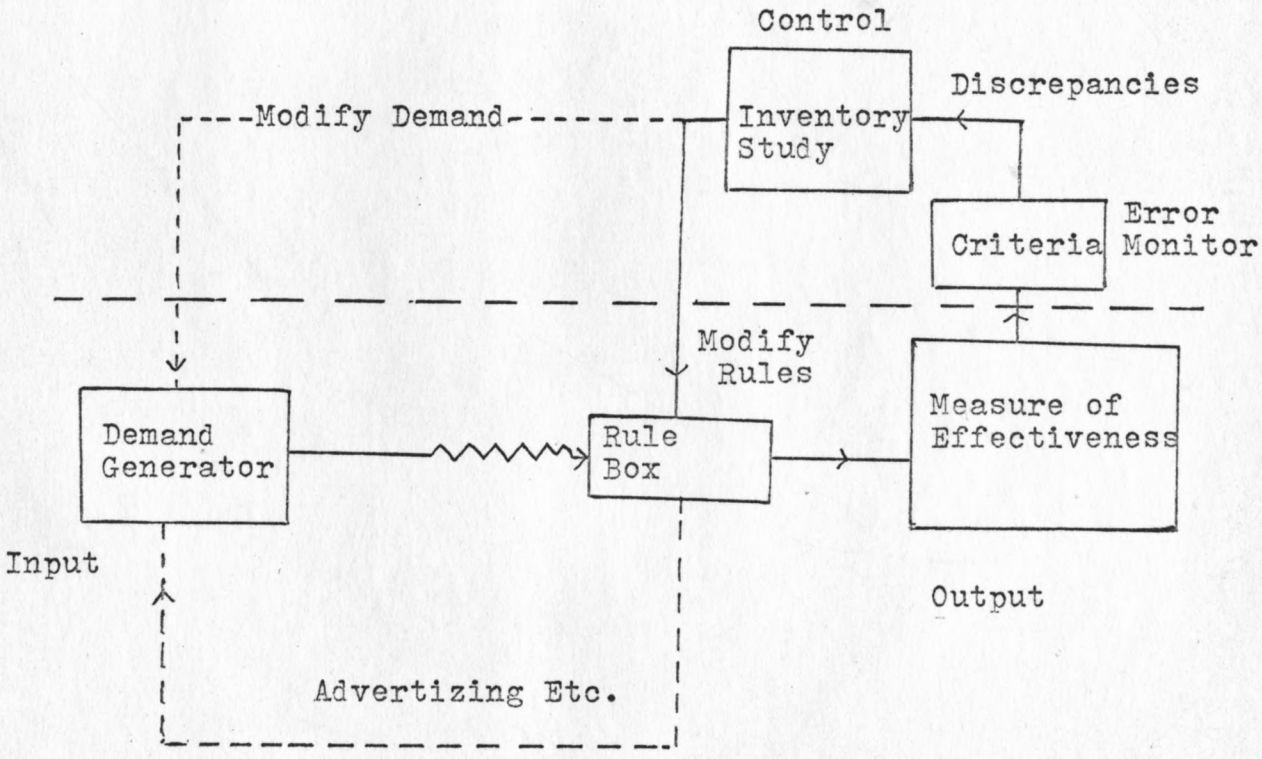


Figure 1

The demand generator is the input to the system. The demand generator is an analog of the market place and the purchase pattern (i.e. arrival rate of customers and their demand) over time. Information from the actual market place

comes to us in real time. That is, it takes a year to collect a year's data. With the demand generator we can effectively collapse time and produce a large amount of data in a few minutes.

The demand is fed into the rule box. The rule box contains the logic which states the company's inventory policy. The inventory policy in our problem would be what reorder policy to adopt (i.e. when and how much to order).

From the rule box, we obtain the desired output. The 'Measure of Effectiveness' in this problem is based on our observing the following simulated results after a fixed period of simulated time:

a) Back orders lost between the two checking periods (if we are out of stock at the warehouse, we lose the customers and hence the orders are lost).

b) Number of times we run out of stock.

c) Lead time.

d) Demand during lead time.

e) Amount of order.

In this report we have not designed for the error sensor monitor which tests the measure of effectiveness and reports on all discrepancies between the actual results and the criteria in standards which are management's objectives. Thus when testing two systems (two rule boxes), the same demand pattern is fed and respective measure of effectiveness reported for each. The error sensor, or monitor compares the

two systems using the output of one as criteria or standard against which the second system can be evaluated. If an error detected is favorable to the proposed system, then the purposed rule box will be substituted for the present rule box. On the contrary, if the proposed system is an impairmant, then control box should call for further study and redesign.

It may be noted that we did not consider costs such as carrying cost, set up costs etc. in our problem. This was done to simplify the problem and to focus our attention upon the essential simulation details. A study of decision rule modifications, for which the cost would be pertinent, will not be considered here. In view of this the computer program is made based on the boxes below the dotted line in figure 1.

QUEUEING MODEL

The model for this simulation program is a queueing model. The flow diagram for data and materials flow is shown in figure 2.

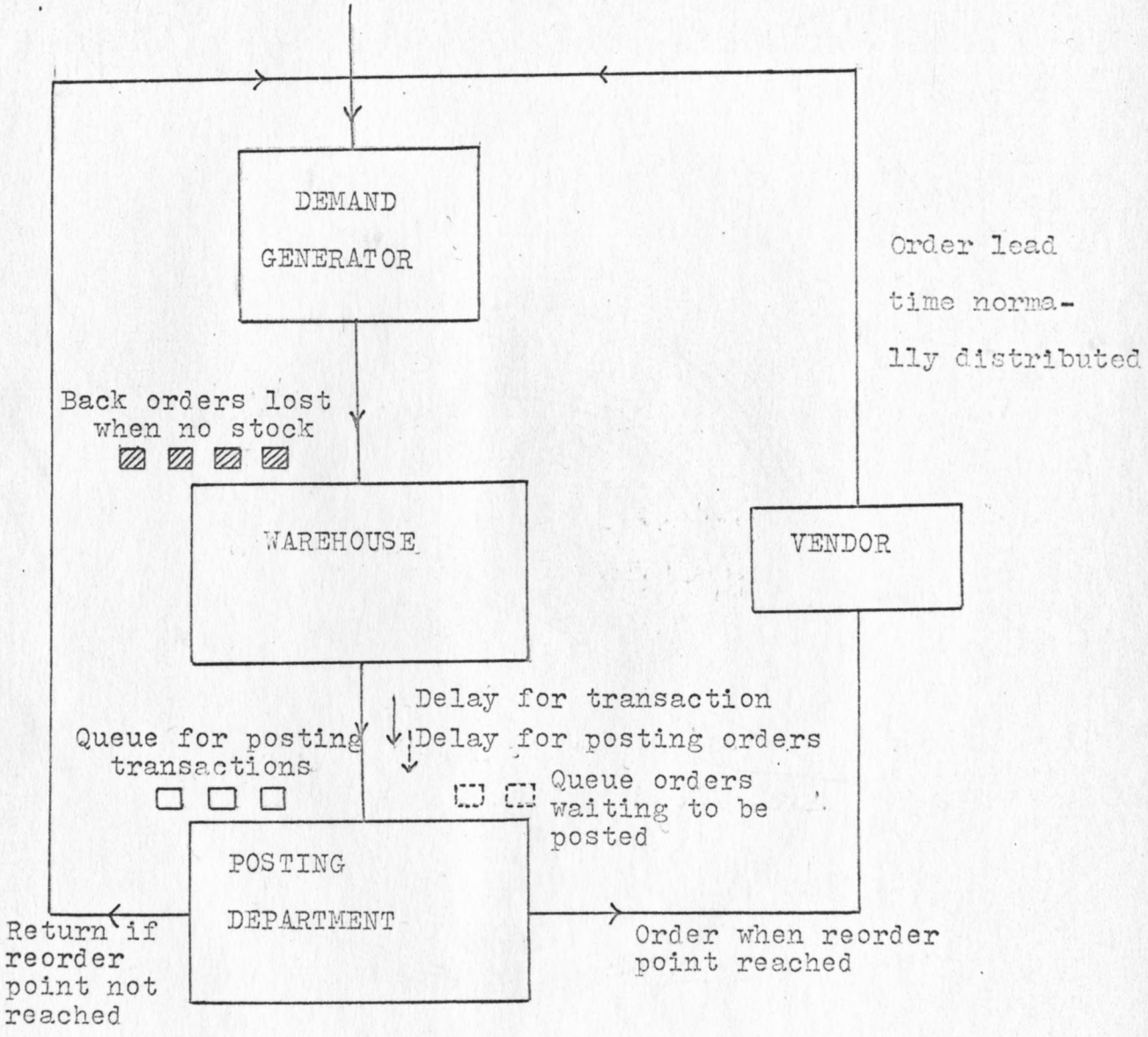
This system is encountered generally in practice as invariably in a big company, there is one posting department responsible for the replenishment of many warehouses in the vicinity. Since the warehouse and posting department are seldom located at the same place there is delay in notification of the transactions that have taken place (posting delay).

In this system, there is a delay between the actual sale and the posting of the transaction. Hence, we encounter two types of record keeping:

1. Physical stock record keeping: This is done at the warehouse, the physical stock is depleted by the amount of the sale.
2. Book stock record keeping: This is done in the posting department, the book stock is depleted by the amount of the sale, when notified to do so by the warehouse. The difference in time between those two postings is the posting delay.

The simulation for the above system has been performed on the IBM 1410 computer.

In this system it has been assumed the customers arrive at the warehouse to buy the commodity available in an exponential fashion, that is the customers interarrival time follows



QUEUEING MODEL

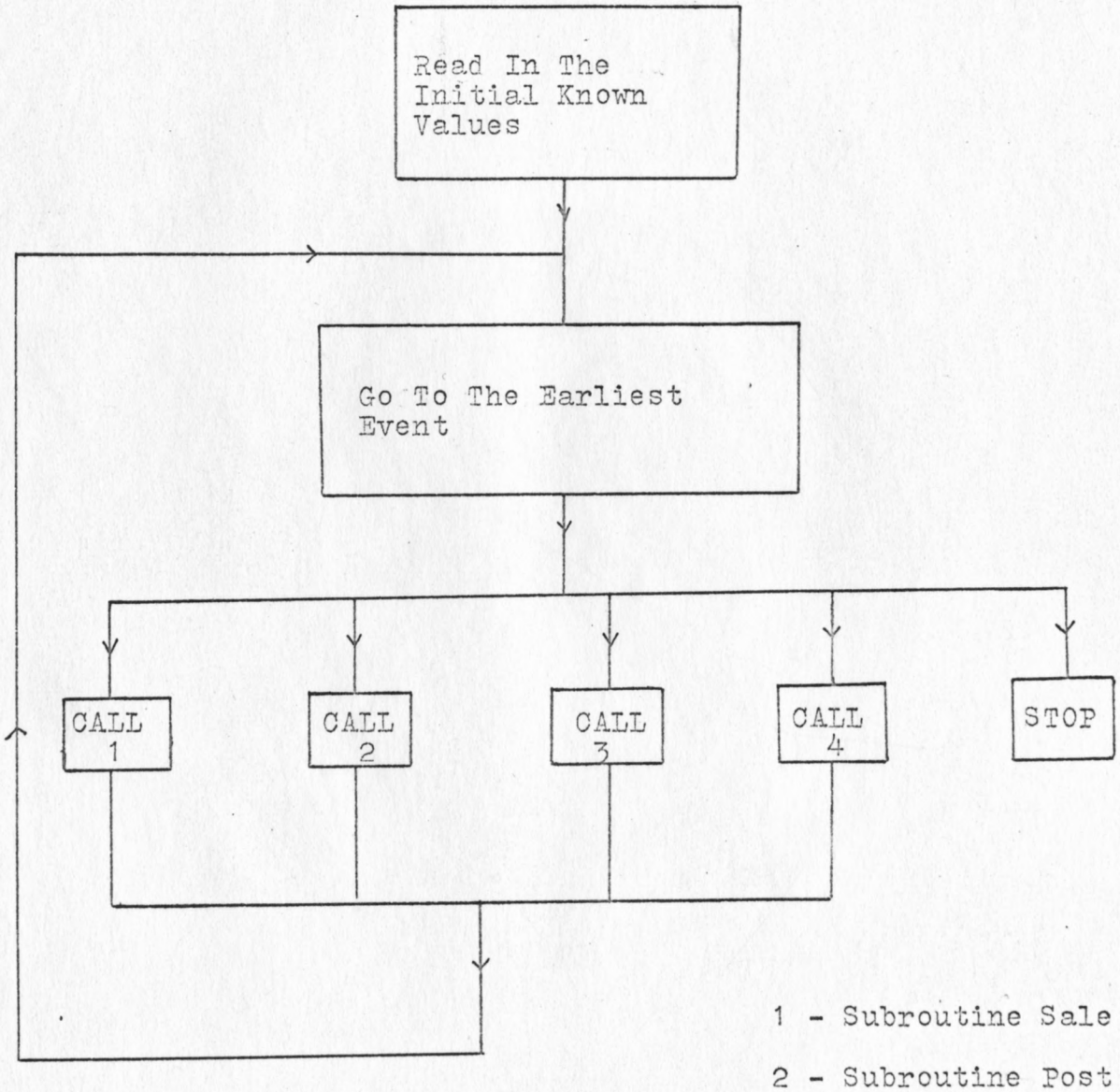
FIGURE 2

an exponential distribution with known (average number of arrivals per unit time). The demand for the commodity by each customer is assumed to have a normal probability distribution with a known mean and standard deviation.

It is assumed that there is no queue of customers waiting (i.e. as and when the customer comes, his demand is met if the stock is available). The receipt of the transaction just taken place is then sent to the posting department, where it is recorded (record keeping). The delay involved in the posting of the transaction is assumed to have a normal distribution with a known mean and standard deviation. Because of this delay between the actual sale and actual posting of the transaction, there is a queue formed of the sale quantities not recorded. The quantity in "Book" is depleted as and when the notice of a sale reaches the posting department. On the basis of Book stock at that time, the reorder level is checked after each such transaction, the s-S ordering policy as explained before is applied here to determine of an order should be placed at this time.

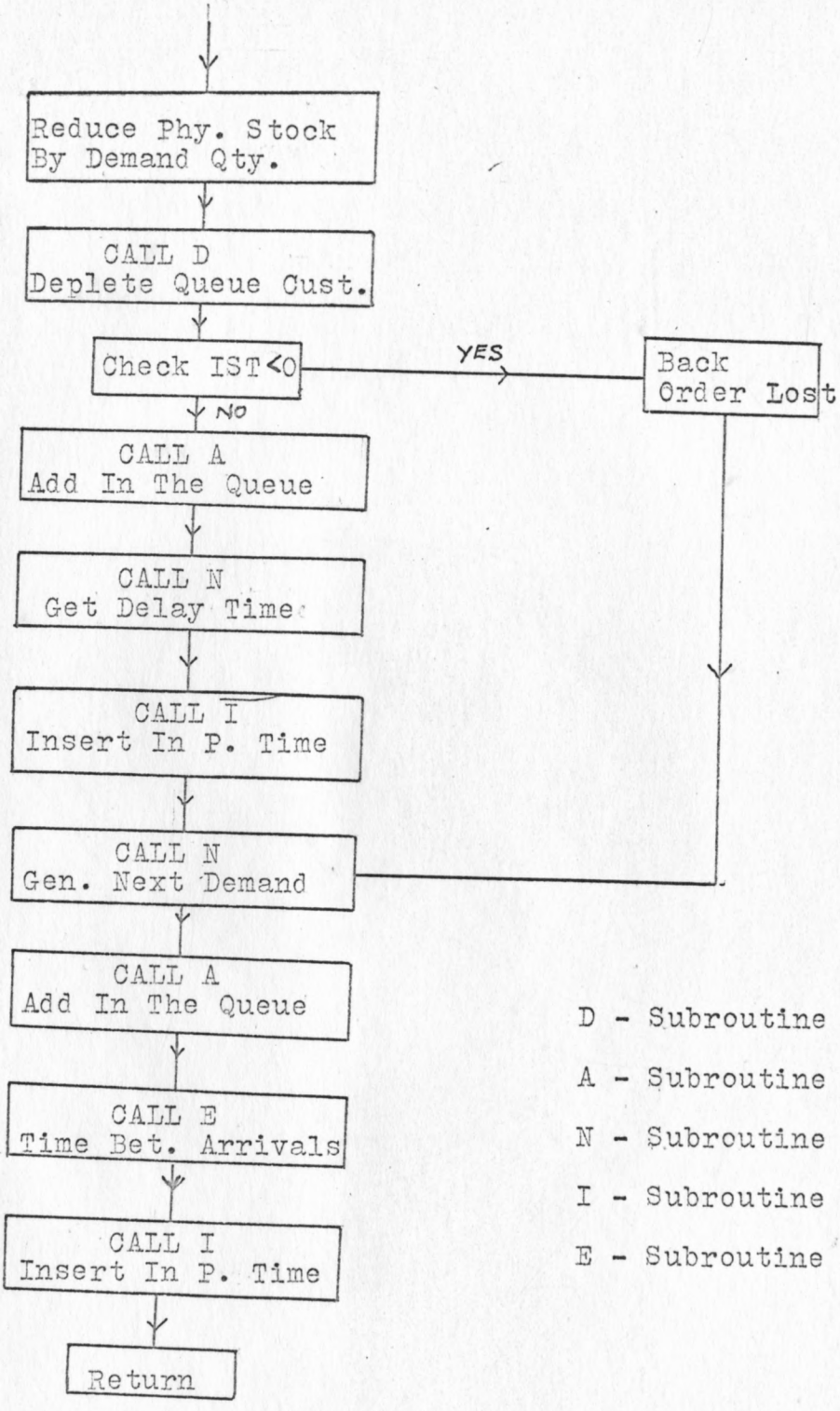
The lead time for the order to arrive at the warehouse is assumed to have a normal probability distribution, the delay time of the information being recorded on the Book is also assumed to follow a normal distribution.

The computer program written in FORTRAN uses a technique similar to that applied in SIMSCRIPT programs. The flow diagrams are shown in figure number 3 to 10. As in SIMSCRIPT the



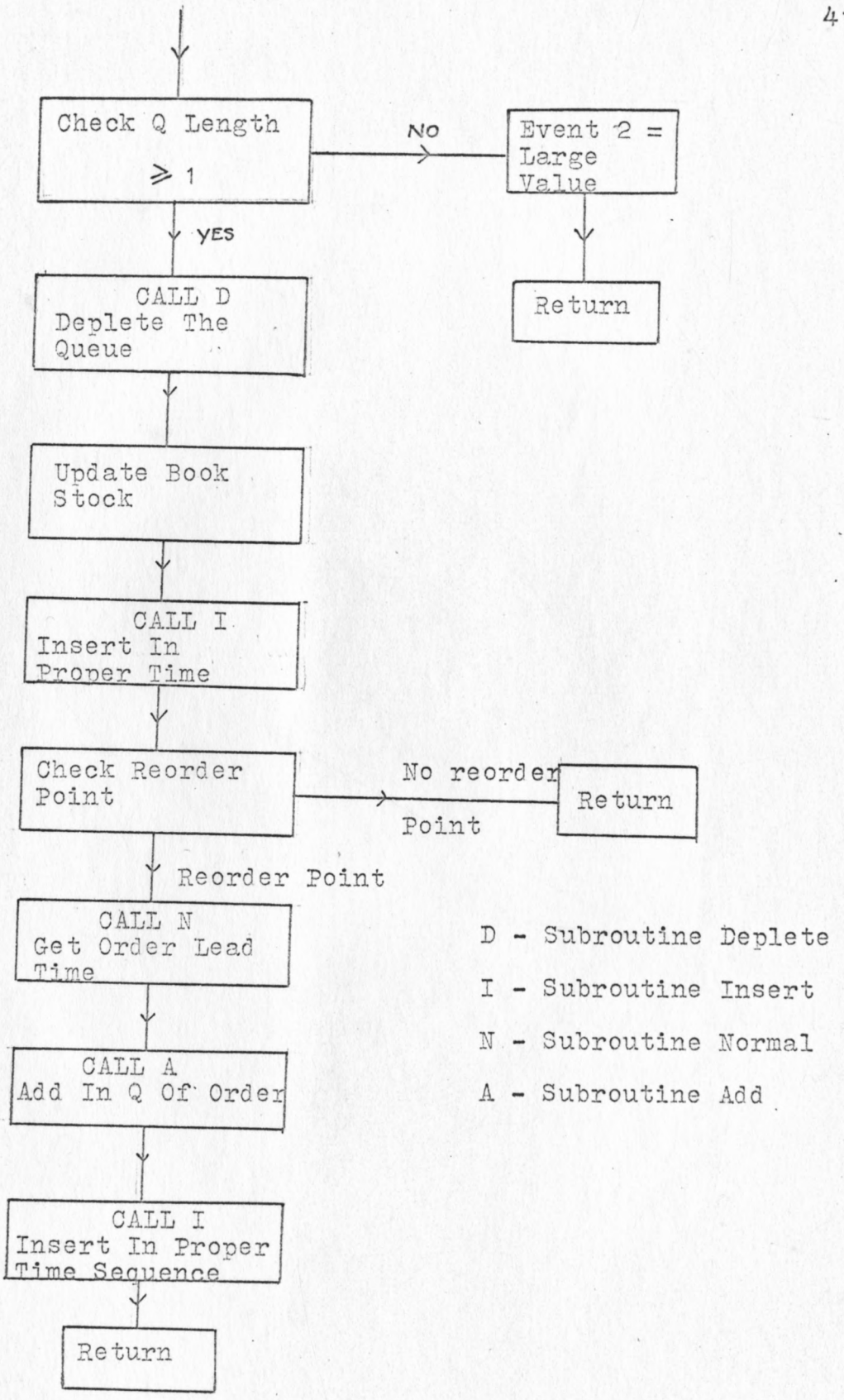
MAIN PROGRAM

FIGURE 3



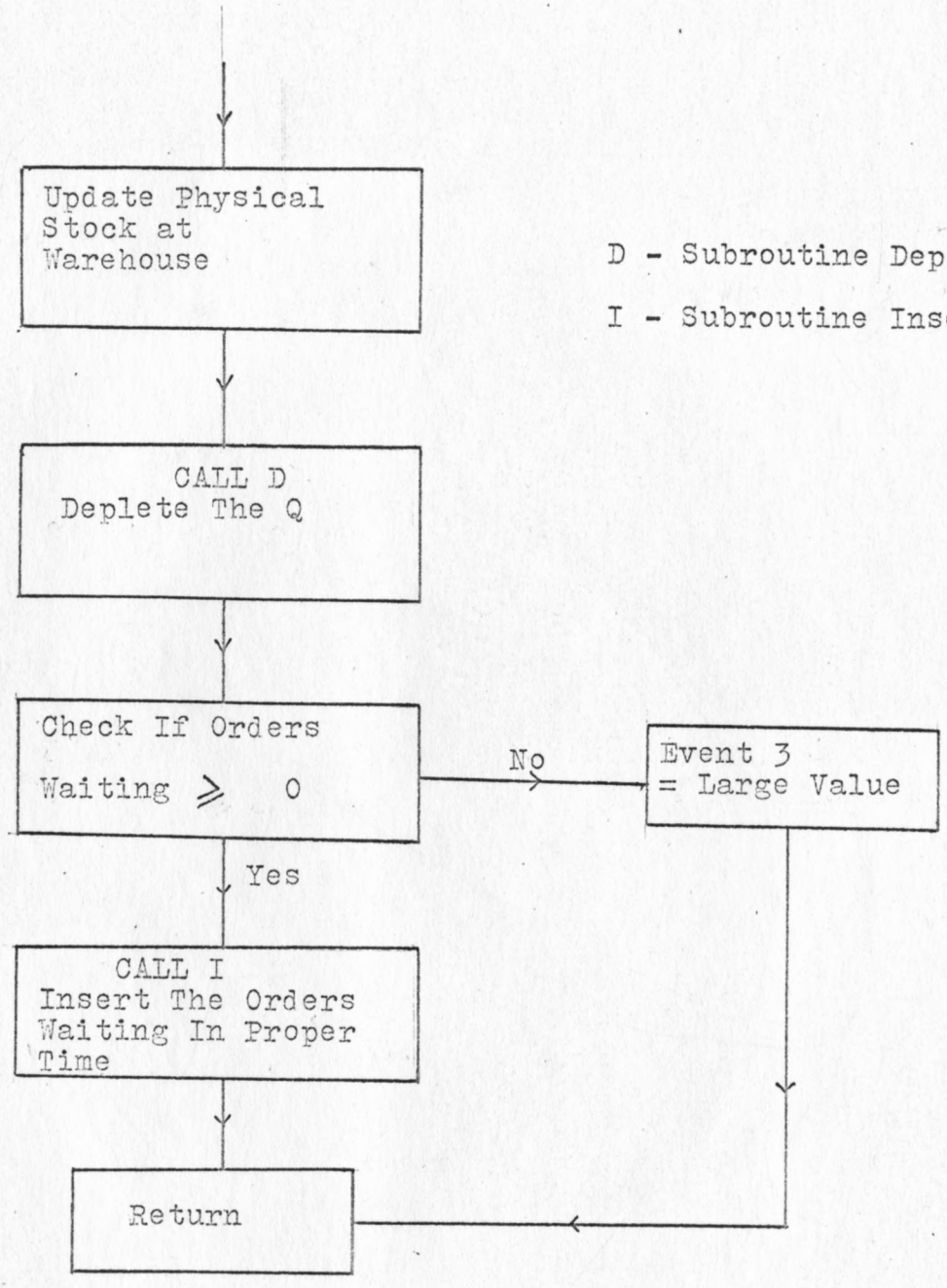
SUBROUTINE SALE

FIGURE 4



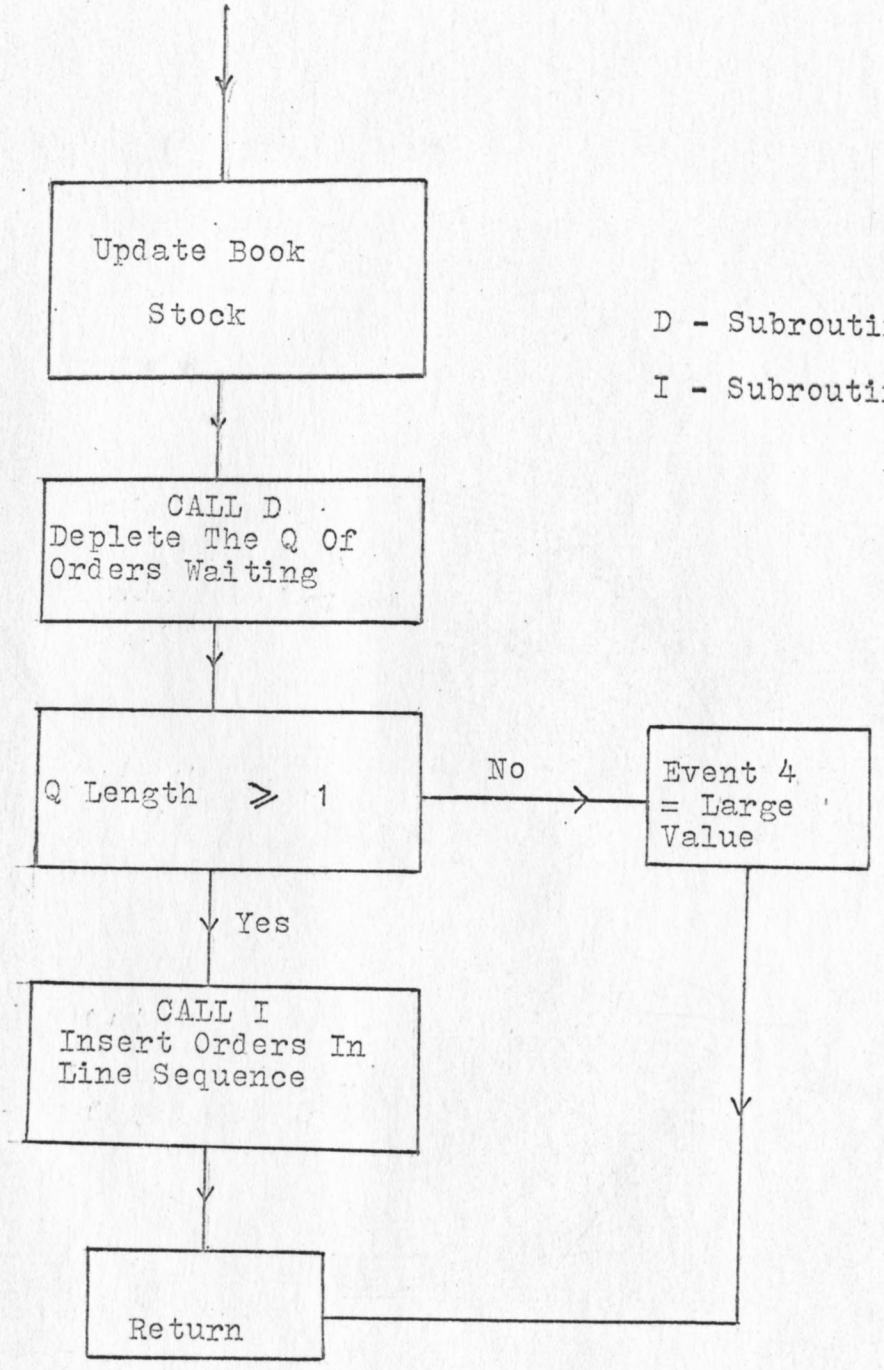
SUBROUTINE POST

FIGURE 5



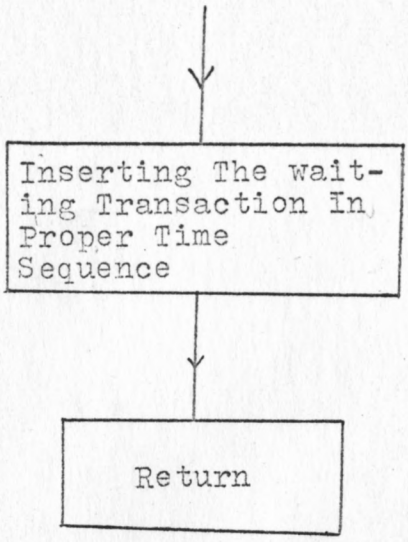
SUBROUTINE RECEIVE

FIGURE 6

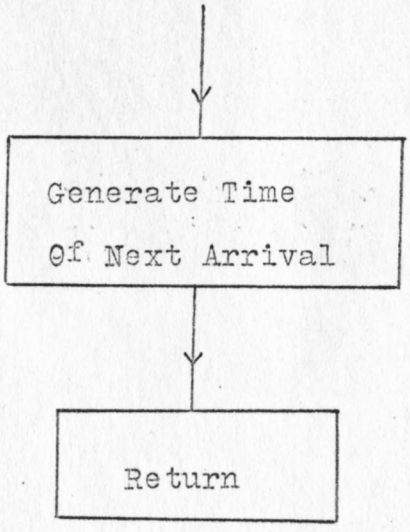


SUBROUTINE BOOK

FIGURE 7

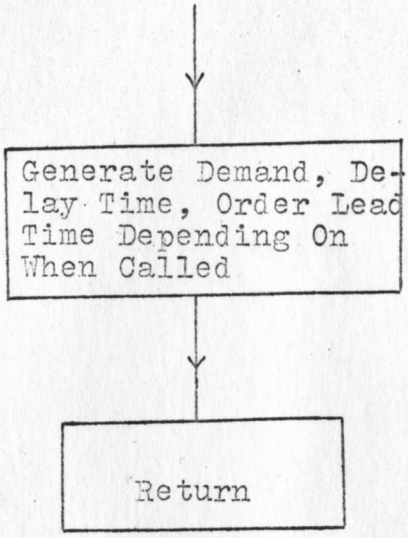


SUBROUTINE INSERT



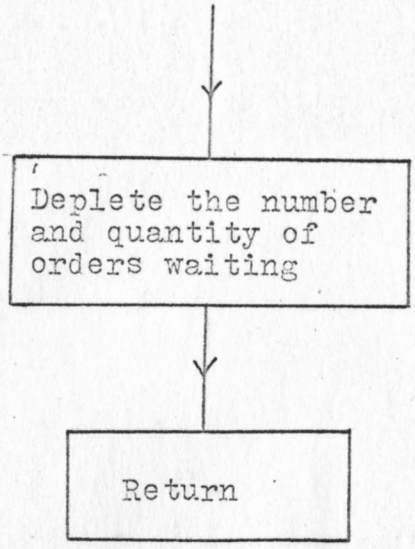
SUBROUTINE EXPO

FIGURE 8

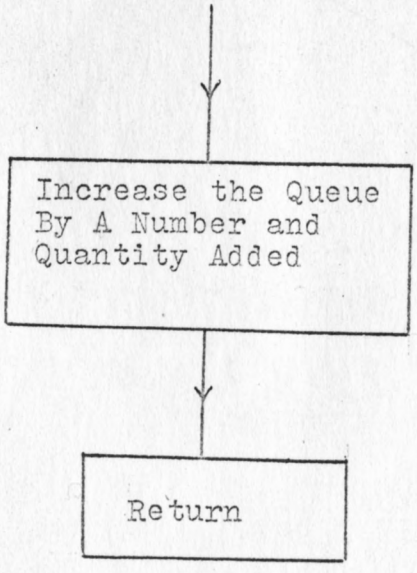


SUBROUTINE NORMAL

FIGURE 9



SUBROUTINE DEPLETE



SUBROUTINE ADD

FIGURE 10

basic unit of the structure is an EVENT, the basic unit has been defined.

Based on our inventory model, only five events can occur. They are:

1. SALE: The commodity is sold to the customer.
2. POSTING: The commodity sold is posted in 'Book".
3. RECEIVE: The order placed is received at the warehouse.
4. RECEIVE AT BOOK: An order which has been received at the warehouse is recorded in the 'Book' inventory.
5. STOP: A mean of terminating the simulations when a stated amount of simulation time has elapsed.

Basic units are units tied together by a "cause event" statements as in SIMSCRIPT. A multiple occurrences of a basic unit themselves do not take time. For example it is possible for the four basic events (1,2,3,4) to occur simultaneously. The multiple events will be recorded as taken place at the same simulated time. The control sequence of the simulation here is performed as in SIMSCRIPT, after each event has been executed.

The system here is changed by:

Creation and Removal of an Individual Temporary Entity

In our model the creation of temporary entity is done by:

1. Generation of time of arrival of the next customer.
2. Creation of another transaction waiting to be posted

in the queue before the posting department.

3. Creation of the quantity and delivery time of an order placed.
4. Creation of the time when an order received will be posted in the stock record book).

Alter Set Membership of an Individual Entity

In this model, this is done by inserting, each new order into the orders waiting queue (these are all endogenous events here) to be posted in the proper time sequence, thus the generating sequence will in general differ from the posting sequence, i.e. the ordering of the set elements has been changed. Similarly in the case of orders placed, if the second order placed has a lead time much shorter than the first, it may be delivered before the first order placed, hence again inserting of the orders placed in the proper time sequence for receipt.

Change Numerical Values of Attributes

The attributes in the problem are the actual stock (physical) available in the warehouse and stock available as per the "Book" figures. The numerical values of the attribute are changed, hence whenever a customer's demand is met, the physical stock is depleted by that amount, and whenever the notification of the transaction that has already taken place reaches the posting department, the book stock is updated. Similarly, when a replenishment order is physically received at the warehouse, a receipt transaction is created

and after the generated delay it reaches the posting department and the stock available is increased by adding the stock received to the "Book" stock on hand at that time. All these actions require the ability to locate an individual entity and/or attribute upon which to operate.

In the program, nine subroutines are used. The flow diagrams of each are shown in figure 4 to 10. The purpose of each subroutine can be summarized as follows:

1. Subroutine Sale:
   a) Creates the posting delay for the current sale transaction and inserts the sale quantity and the actual posting date (time) into the posting queue.
   b) Remove the current sale from the sale queue and generate the next interarrival time and demand quantity with the insertions of the next sale into the event queue with the proper date, i.e. current time plus interarrival time (Note- in this simulations the sale queue has a constant length of one sale).
2. Subroutine Post:
   a) Destroys the endogeneous entity of "sale transactions", by removal from the posting queue as the "Book Quantity" is updated.
   b) After the sale is posted the "book quantity" is tested for the reorder point, an order entity is created and placed into the order queue with the proper delivery date if necessary.

3. Subroutine Receive:

a) Destroys the temporary entity of the order and its quantity as it is received into the warehouse physical stock and alters the set membership of orders placed.

b) Changes the numerical value of the attribute "Physical" as stock is received at the warehouse.

c) Creates the temporary entity of the time at which this order received will be posted in Book and changes the set membership of orders waiting to be posted.

4. Subroutine Book:

a) Destroys the temporary entity of the order and quantity that was placed and now is posted in Book.

b) Changes the numerical value of the attribute "Book" when the stock received is recorded in Book.

5. Subroutine Add:

This subroutine adds the quantity and the number of transactions that have taken place and are waiting to be posted in Book. Thus giving the total quantity and total number that are in the queue. Similarly for the order queue and the customers queue, whenever a temporary entity is created it is put into the proper queue.

6. Subroutine Insert:

This subroutine alters the set membership of an individual queue by inserting the new entity in the proper time sequence.

7. Subroutine Deplete:

   Depletes the queue length and the quantity in the proper queue whenever we destroy a temporary entity.

8. Subroutine Expo:

   Generates an interarrival time from a specified exponential distribution which is then treated as a temporary entity when coupled with the sale quantity.

9. Subroutine Normal:

   Generates a normal deviate with a specified mean and standard deviation. The normal deviate may then be considered as:

   a) A customer's demand.
   b) A sale posting delay.
   c) An open order lead time.
   d) An order received posting delay.

10. Stop:

    When the desired amount of simulation time has expired, then event "Stop" ends the simulation.

The results of this simulation program were presented in two different forms:

A) In the proper time sequence of each event as it occurs. This is useful to determine the accuracy of the system.

B) A periodic summary is made to simulate the common practice of weekly or monthly performance reports.

These summaries have two forms:

1. A summary of actual demands, sales, and lost sales.
2. A summary of demand during each replenishment order lead time.

The format of these output forms are discussed in the analysis of results.

TESTING THE DISTRIBUTION GENERATORS

The two subroutines EXPO and NORMAL were tested by the Kolmogrov-Smirnov one sample test (10). The purpose of this test was to determine if values obtained from the two subroutines do actually represent their respective theoretical distributions. The Kolmogrov-Smirnov one sample test is a test of goodness of fit. That is, it is concerned with the degree of agreement between the distributions of a set of sample values (observed scores) and some theoretical distribution.

Briefly, the test involves specifying the cumulative frequency distribution which would occur under the theoretical distribution and comparing this with the observed cumulative frequency distribution. The theoretical distribution represents what would be expected under ideal conditions. The point at which these two distributions, theoretical and observed show the greatest divergence is determined. Reference to the sampling distribution indicates whether such a large divergence is likely to occur on the basis of chance. That is, the sampling distribution indicates whether a divergence of the observed magnitude would probably occur if the observations were really a random sample from the theoretical distribution.

With the Normal generator that we made, we generated 1000 readings with a mean of 7.50 and standard deviation equal to 1.50. The Normal generator is used to generate demand and delay in the program. Demand is essentially an integer quantity,

so in generating demand and rounding it off to the nearest integer, we add 0.5 to the value generated (real) and then take the largest integer contained in that value, thus keeping our mean the same as that we assumed. For the 1000 readings generated (integer), the frequency distribution is shown in table 5.

The maximum absolute difference is 0.022. Now from Kolmogrov-Smirnov one sample test for 'N' over 35, the critical values at the 5% level is $1.36/\sqrt{N}$, where N is the number of observations. That is any observed difference which is equal to or greater than $1.36/\sqrt{N}$ will be significant at the 0.05 level.

Here we have a sample size of N = 1000, therefore the maximum difference allowed is $\frac{1.36}{\sqrt{1000}} = .04$. Our difference of 0.022 is much below the limit, thus we may conclude that the values generated by the Normal generator do follow the Normal distribution.

This test was also applied to the Exponential generator. The value of (average arrival rate) was taken as 1.0, and 1000 interarrival times were calculated by the generator and grouped in the fashion shown in table 6. Table 6 also shows the frequency distribution of the generated and theoretical distribution.

The maximum absolute difference, we note is 0.023. By the Kolmogrov-Smirnov one sample test for N over 35 the critical value at the 5% level is $1.36/\sqrt{N}$, where N is the number

TABLE 5

Testing of Normal Generator

Value of standard deviation taken = 7.50

Value of mean taken = 7.50

Value	Number of times it occurred	Cumulative Distribution of observed value	Theoretical Cumulative Distribution	Absolute difference
3	4	.004	.004	0
4	13	.017	.018	.001
5	75	.092	.086	.006
6	177	.269	.247	.022
7	238	.507	.500	.007
8	258	.765	.753	.012
9	150	.915	.914	.001
10	59	.974	.982	.008
11	25	.999	.996	.003
12	0	.999	1.000	.001
13	1	1.000	1.000	0
14	0	1.000	1.000	0

Value of mean from generated values = 7.459

Value of standard deviation = 1.507

TABLE 6

Testing of Exponential Generator

Value of arrival rate taken = 1.0

Falling in interval	Frequency of occurrence	Cumulative distribution of the generated values	Theoretical cumulative distribution	Absolute difference
0-.25	226	.226	.224	.002
.26-.50	180	.406	.394	.012
.51-.75	128	.534	.541	.007
.76-1.00	104	.638	.632	.006
1.01-1.25	100	.738	.715	.023
1.26-1.50	57	.795	.777	.018
1.51-1.75	45	.840	.826	.014
1.76-2.00	42	.882	.865	.017
2.01-2.50	57	.939	.919	.020
2.51-3.00	21	.960	.951	.009
3.01-3.50	11	.971	.967	.004
3.51-4.00	10	.981	.982	.001
4.01-5.00	12	.993	.994	.001
5.01-6.00	4	.997	.998	.001
6.01-7.00	2	.999	.999	.001
7.01-8.00	1	1.0	1.0	0

Value of arrival rate calculated from generated time between arrivals = 1.045

of observations which is 1000 here. Hence,

$$\text{critical value} = \frac{1.36}{\sqrt{1000}} = 0.04$$

Since the difference 0.023 is much below the critical value, we conclude that values generated by the Exponential generator follow the expoenetial distribution.

The chi-square test was also applied to the two generators. Table 7 gives the frequency distribution of the generated and theoretical values for the two generators.

The value of chi-square for the normal generator is 7.80 with 7 degrees of freedom. From tables at 5% level with 5 d.f. $\chi^2 = 14.06$. Thus the value calculated lies very much below the critical value, there is no reason to suppose that the generated numbers do not follow a normal distribution. Similarly for the Exponential generator, the value of χ^2 calculated is 17.79 with 13 degrees of freedom. At the 5% level with 13 d.f., the critical value is:

$$\chi^2 = 22.362.$$

Again the calculated value lies well within the critical value, hence the values generated by the Exponential generator may be assumed to follow the Exponential distribution.

Output Presentation

The output is presented in two ways as shown in the sheets A and B attached in the appendix. The output represented in sheet A shows the status of various activities at each time any event occurs. This representation helps us to determine if the system is functioning correctly as per the intended

TABLE 7

Testing of Generators with Chi-Square Test

(i) Normal Generator

Value	Observed frequency	Theoretical frequency	χ^2
4 or less	17	18	.055
5	75	68	.645
6	177	161	1.44
7	238	253	.900
8	258	253	.100
9	150	161	.750
10	59	68	1.19
11	25	18	2.72
			7.800

TABLE 7

Testing Generators with Chi-Square Test

(ii) Exponential Generator

Value	Observed frequency	Theoretical frequency	χ^2
0-.25	226	224	.02
.20-.50	150	170	2.35
.51-.75	128	147	3.00
.76-1.00	104	91	1.85
1.00-1.25	100	83	3.38
1.26-1.50	57	62	.30
1.51-1.75	45	49	.33
1.76-2.00	42	39	.21
2.01-2.50	57	54	.17
2.51-3.00	21	32	3.78
3.01-3.50	11	16	1.57
3.51-4.00	10	15	1.67
4.01-5.00	12	12	0
over 5.00	7	6	.16
			17.79

$\chi^2 = 17.79$ with 13 degrees of freedom

program. The meaning of the various headings is noted below the dotted line, and the initial known values on which the program was run are shown above the dotted line.

Thus at the simulated time 0.00, we have an initial physical stock (IST) of 300 and an initial Book stock (IBS) of 300. The time below T, which is 0.24, indicates the time at which the next customer arrival will be. Hence in the second line at 0.24 simulated time, the next customer arrives. There is a sale of 14 (below SL), hence physical stock reduces by that amount and becomes = 300 - 14 = 286. Below 'T' we get 1.08, giving the time between arrivals values showing that next customer will arrive at 1.08 + 0.24 = 1.32, which can be verified from the next line. Below 'D' the value 8.53 gives the delay encountered before this sale would be posted in Book. That is at time 0.24 + 8.53 = 8.77, this sale would be posted in 'Book'. We can verify this by noting that at 8.78 (as 8.77 was the sum of two truncated values, the actual truncated total was 8.78) this sale is registered. We note below QB (quantity of Book sale) at simulated time 8,78, the sale posted is 14 units which agrees with the amount under SL at simulated time 0.24. Below QS we find the total quantity of sales that have been made up to this time, which is 14 here as there has been no sale before this. NOS gives us the number of sales (here = 1) that have been made up to this time.

Below IBS the 'Book stock' at that time is noted, which is 300 at time 0.24 since this event has only occurred at the

warehouse, and the posting department has not yet been notified. The values below 'ISS' give us the stock in the system at any time. System stock is the sum of the stock recorded in Book and the stock on order. Quantities under 'BQ' give the total number of transactions (sales) already taken place at the warehouse, but not recorded in the 'Book'. 'BQQ' gives the total quantity of these transactions waiting to be posted (queue before the posting department). 'QB', 'BS' and 'NOBS' just as in actual sales at the warehouse, are quantity of sale recorded in 'Book', cumulative quantity of sales recorded in the 'Book' thus far.

Our reorder point is 200 units, that is whenever the stock on hand (in 'Book') plus stock on order falls below the 200 point, an order is placed. Hence we note that at time 15.58 here the 'Book stock' is 188, less than 200, hence an order is placed and below 'LT1', 8.46 is written indicating the order lead time, that is at simulated time 15.58 + 8.46 = 24.04 the order will be received at the warehouse. The value below 'SO' gives the quantity of order placed. We note that at 24.05 (24.04 was the truncated value), the order is received at the warehouse, and below 'LT2', 9.00 is written at the same simulated time. 'LT2' is the lead time before the order received is posted. (delay in intimation between the two departments) That is at the simulated time 24.05 + 9.00 = 33.05, the order will be noted as received in 'Book'.

Thus sheet 'A' shows the detailed sequence of occurrence

of the various events. The total time taken by the computer to perform this simulations is noted on the last line.

Sheet 'B' gives a summary of all transactions that have taken place during a specific interval of time (10 units of time is taken here). In actual practice, an inventory system is usually checked after a week/month and weekly/monthly reports are made of the total number of sales, total number of orders lost etc. during that period. We have done the same thing here. Comparing sheets 'A' and 'B' we note that after 10 units of simulated time, the total number of sales in both cases are 10, physical sale and actual demand is 140, the number of Book sales during this period are 3, the quantity of the Book sales is 44, the number and the quantity in Book queue are 7 and 96 respectively. All of the figures on sheet 'A' agree with those shown in summary on sheet'B'. In addition to this, sheet 'B' adds another feature, that of finding demand during a lead time. Actual sales during a lead time and the number of sales recorded during this period are shown (lead time here is the time between order placing and replenishment of Book stock when the order received is posted). Here in our case for the first time it would be 33.05 - 15.58 = 17.47 as noted. The demand during the lead time and the actual sales during this time form the basis on which management can evaluate its reorder policy.

PLAN OF SIMULATIONS AND DISCUSSION OF RESULTS

The purpose of the simulations is to investigate the curve shapes of the demand distributions and the demand during a lead time when the variances of the demand and lead time distributions are altered. Seven runs were made with the parameter values shown in Table 8. The output generated was taken in the form of sheet 'B'. Each run was made for 2500 units of simulated time so that 250 check points were recorded for 10 units of time each.

The purpose of run numbers 1 and 7 was to investigate how the exponential and normal generators work in the system.

For run number 1, we have the mean of the demand quantity 15.0 and its standard deviation zero, and time between arrivals follows an exponential distribution with $\lambda = 1.0$. Hence in our results for run number 1, where the system is checked after every 10 units of simulated time, we should expect the number of demands (number of customers in every 10 units of times) to follow a Poisson distribution with mean of 1 X 10 = 10 customers. Since the standard deviation of the demand is zero (Normal distribution) and the mean is 15 units, the quantity of the actual demand should have a mean of 150 units.

To find the mean and standard deviation of the physical demand and the demand during a lead time, the output generated was also collected in the form of punched cards. These cards were separately fed into another program to find the mean and standard deviation required.

TABLE 8

Initial Known Values of 7 Runs

Run	λ	Demand		Lead Time		Posting Delay		Order Receipt Delay	
		mean	σ_D	mean	σ_{LT}	mean	σ_d	mean	std. dev.
1	1.0	15	0	8.0	1.5	7.5	1.5	7.5	1.5
2	1.0	15	1.5	8.0	1.5	7.5	1.5	7.5	1.5
3	1.0	15	3.0	8.0	1.5	7.5	1.5	7.5	1.5
4	1.0	15	4.5	8.0	1.5	7.5	1.5	7.5	1.5
5	1.0	15	3.0	8.0	0.5	7.5	1.5	7.5	1.5
6	1.0	15	3.0	8.0	2.5	7.5	1.5	7.5	1.5
7	*	15	3.0	8.0	1.5	7.5	1.5	7.5	1.5

Reorder level = 200 units in each case

Initial physical stock = Initial book stock = 300 units in each case

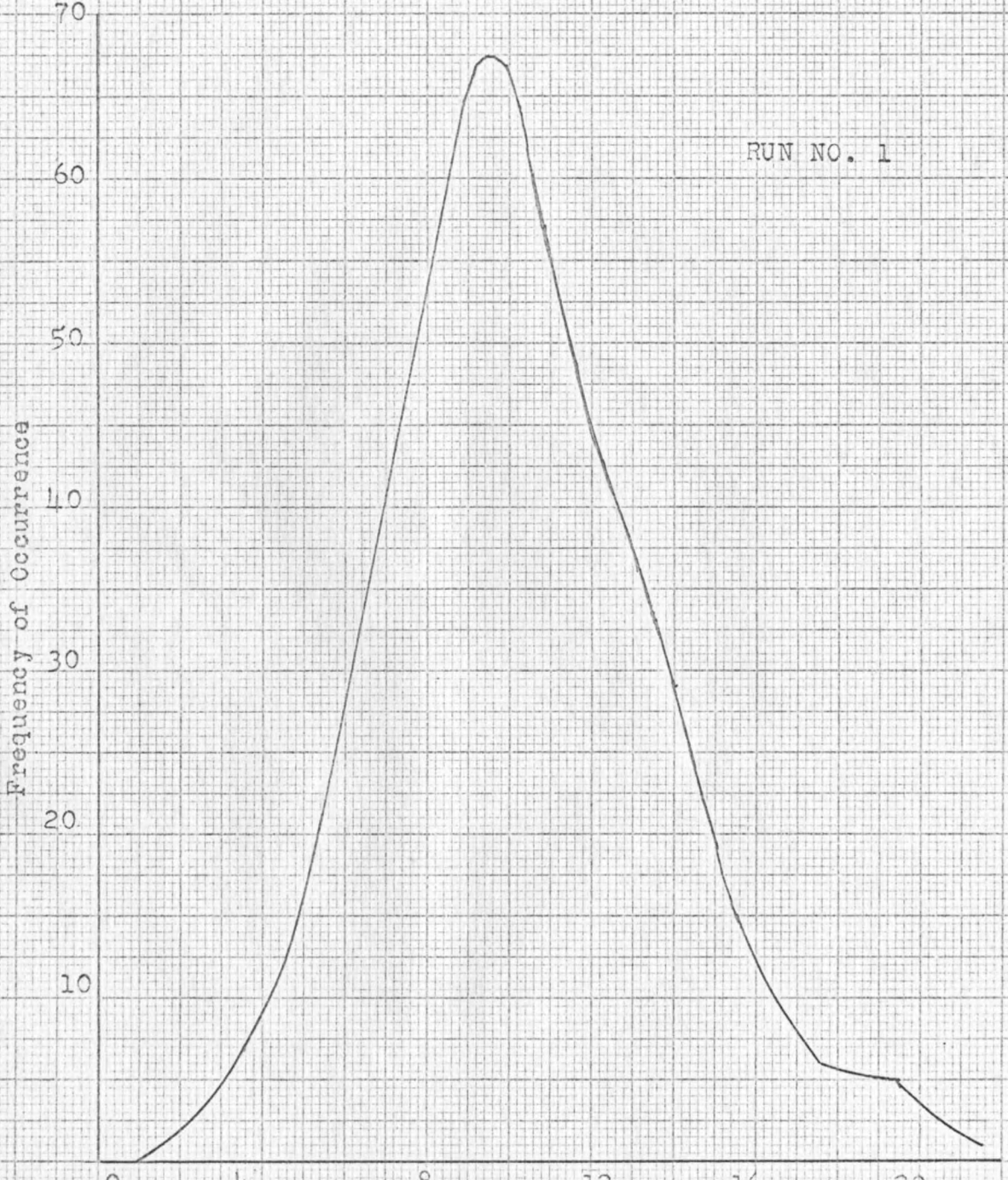
* Uniform time between arrivals = 1.0

For run number 1 the mean is found to be 154.30, very close to 150 units expected. To verify if the number of customers (at each 10 units of simulated time) follows the Poisson distribution, a Kolmogrov-Smirnov one sample test was applied as before to check the goodness of fit. The frequency distribution of the number of customers and its cumulative distribution and finally a Kolmogrov-Smirnov one sample test is applied as shown in table 9.

The maximum absolute difference is 0.039. By the Kolmogrov-Smirnov one sample test, a difference greater or equal to $1.36/\sqrt{N}$ will be significant at the 0.05 level. Here $N = 250$, and $\frac{1.36}{\sqrt{250}} = 0.086$, the absolute difference being much less than the critical value signifies that the number of customers appears to follow the Poisson distribution with mean = 10.00. The graph for this is drawn on graph number 1.

In run number 7, we had assumed that the customers arrive after a fixed time interval of 1 unit and the demand quantity follows a normal distribution with a mean of 15.00 and standard deviation equal to 3.00. In view of this should expect the actual demand quantity to follow a normal distribution with mean equal to 150 and standard deviation equal to 9.47 (in an interval of 10 units the demand occurs 10 times, each independent of each other with standard deviations of 3, hence $\sigma^2 = 3^2 . 10$, or $\sigma = \sqrt{10} . (3) = 9.47$). The mean and standard deviation was calculated to be equal to 149.07 and 9.47

Frequency Distribution Of Number Of Demand



GRAPH NO. 1

TABLE 9

Kolmogrov-Smirnov One Sample Test on Run 1

No. of Customers	Frequency Distribution (output)	Cumulative Frequency Distribution	Theoretical Cumulative Distribution (From Tables)	Absolute Difference
3	4	.016	.009	.007
4	3	.028	.024	.004
5	12	.076	.056	.020
6	10	.116	.112	.004
7	19	.192	.168	.024
8	27	.300	.300	0
9	40	.460	.44	.039
10	25	.560	.545	.015
11	27	.668	.662	.006
12	22	.756	.761	.005
13	21	.804	.841	.037
14	13	.892	.899	.007
15	8	.924	.939	.015
16	7	.952	.965	.013
17	4	.968	.980	.012
18	2	.976	.989	.013
19	4	.992	.994	.002
20	1	.996	.997	.001
21	0	.996	.999	.003
22	1	1.00	1.00	0

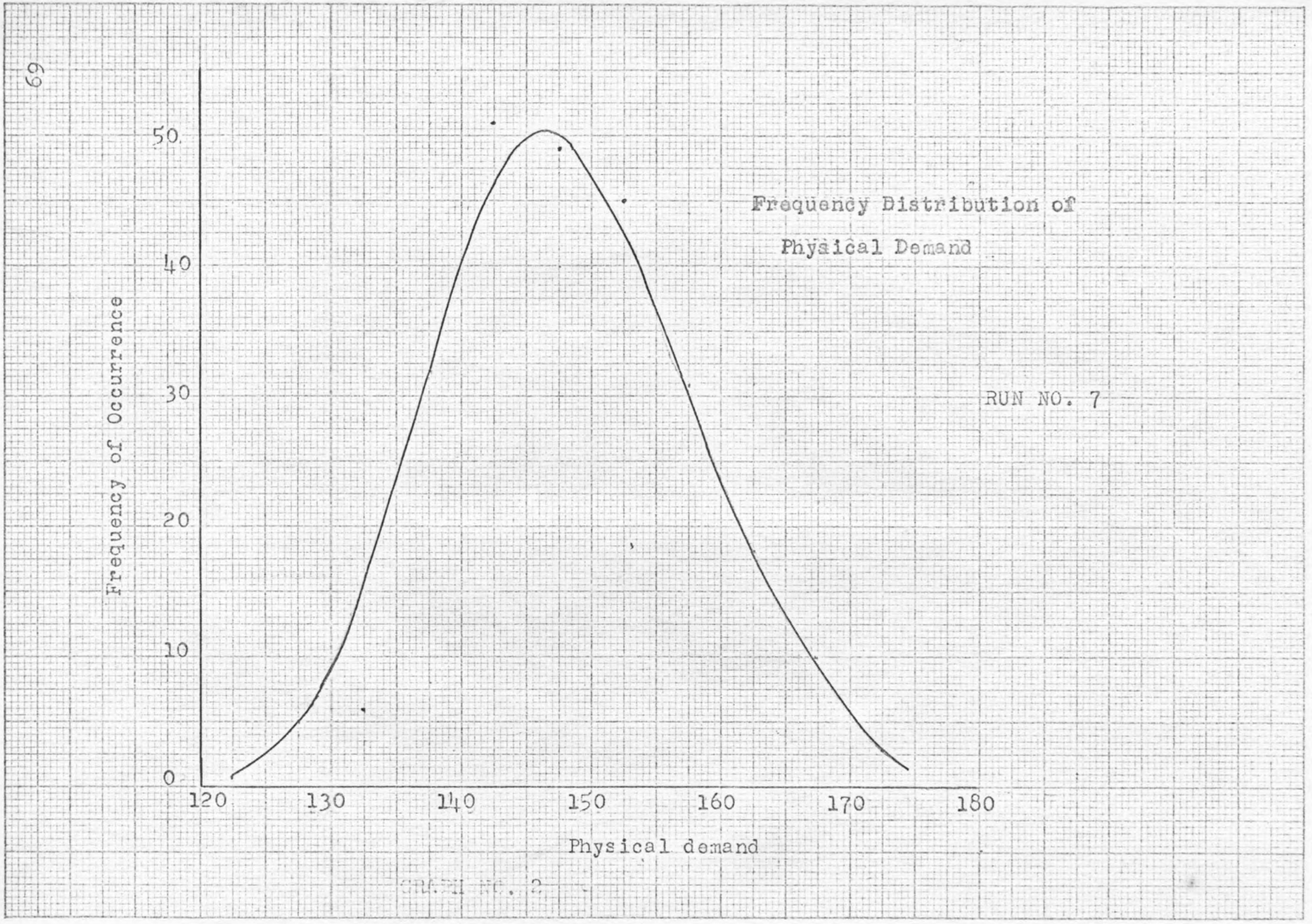
respectively. Again to verify the goodness of fit with the theoretical normal distribution, with this mean Kolmogrov-Smirnov one sample test was applied and tabulated result is shown in table 10.

The maximum absolute difference here is 0.068. By Kolmogrov-Smirnov one sample test the difference allowed at 5% level is 0.086. The value being less than the critical value shows that physical demand generated in run number follows a normal distribution as was expected. The graph for this is shown in graph number 2.

For run numbers 2,3 and 4 the mean and standard deviation of the demand quantity shown in table 11 the demand during a lead time is tabulated in table 12. The mean and standard deviation of the demand during a lead time for run numbers 5, 3 and 6 are also tabulated in table 12.

For run numbers 2,3 and 4 wherein we had changed the standard deviation of the demand quantity with the same mean, we note that the demand quantity of the customers, which is a compound distribution of a Normal distribution and an Exponential distribution, does not show much difference in the mean and standard deviation of the total demand during every 10 units of time. The two extreme variances were tested for significance by an F test. Here the two extreme values have S.D. of 50.35 and 47.25 each with 249 d.f.

hence $F = \frac{50.35}{47.25} = 1.12$


Frequency Distribution of
Physical Demand
RUN NO. 7
Frequency of Occurrence
50
40
30
20
10
0
120
130
140
150
160
170
180
Physical demand

Graph No. 2

TABLE 10

Kolmogrov-Smirnov One Sample Test on Run 7

Physical Demand lying between	No. of Times it Occured	Cumulative Frequency (Output)	Theoretical Cumulative Frequency Distribution	Absolute Difference
121-125	1	.004	.005	.001
126-130	5	.024	.022	.002
131-135	6	.048	.0694	.0214
136-140	32	.176	.171	.005
141-145	51	.416	.4015	.0145
146-149	38	.568	.500	.068
150-155	55	.788	.7360	.052
156-160	31	.912	.8770	.035
161-165	18	.984	.9430	.041
166-170	10	.986	.987	.001
171-175	3	1.00	1.00	0

TABLE 11

Physical Demand

Variation of Demand Variance

Run No.	Sample Size	σ_D	σ_{LT}	Mean of Physical Demand	Standard Deviation of Physical Demand
2	250	1.5	1.5	147.47	47.25
3	250	3.0	1.5	149.70	50.35
4	250	4.5	1.5	154.70	50.28

TABLE 12

Demand during lead time

(i) Variation of Demand Variance

Run No.	Sample Size (No. of replenish-ments made)	σ_D	σ_{LT}	Mean of Demand dur-ing Lead Time	Standard Deviation of Demand during Lead Time
2	273	1.5	1.5	224.86	66.46
3	269	3.0	1.5	235.94	71.13
4	279	4.5	1.5	238.62	69.88

(ii) Variation of Lead Time Variance

Run No.	Sample Size (No. of replenishments made)	σ_D	σ_{LT}	Mean of Demand during Lead Time	Standard Deviation of Demand during Lead Time
5	278	3.0	0.5	241.09	66.77
3	269	3.0	1.5	235.94	71.13
6	266	3.0	2.5	232.09	76.87

At the 5% level with 249 D.f. the value of F should be greater than or equal to 1.25 for significance. The value is 1.12, hence we conclude that the variances are not significantly different in these cases.

To test for the significance of means, 'student's' t test is applied to the two extreme values of 't' (5).

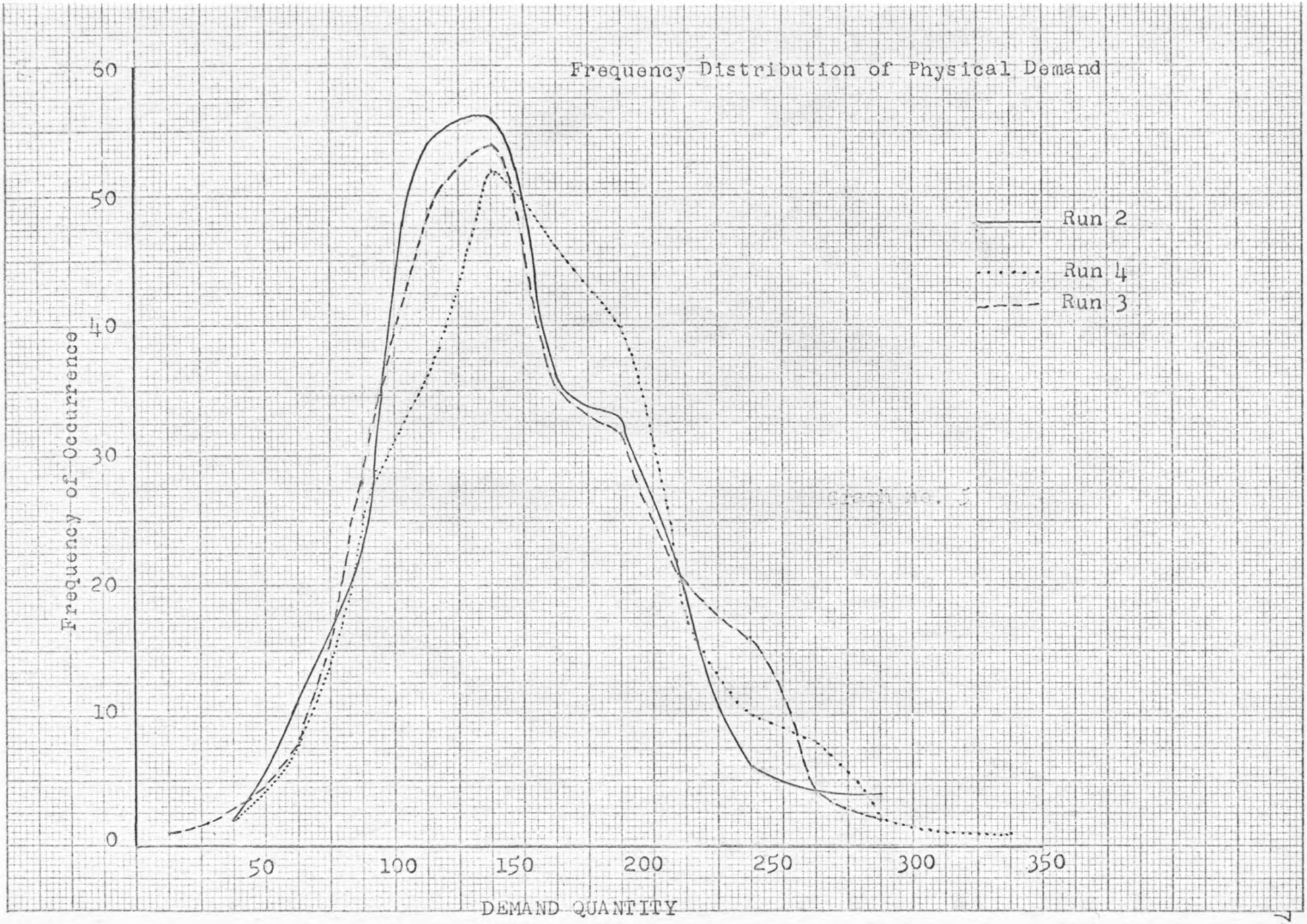
$$t = \frac{\bar{X} - \bar{Y}}{\sigma\sqrt{\frac{1}{n_x} + \frac{1}{n_Y}}} = \frac{154.70 - 147.47}{50\sqrt{\frac{1}{250} + \frac{1}{250}}}$$

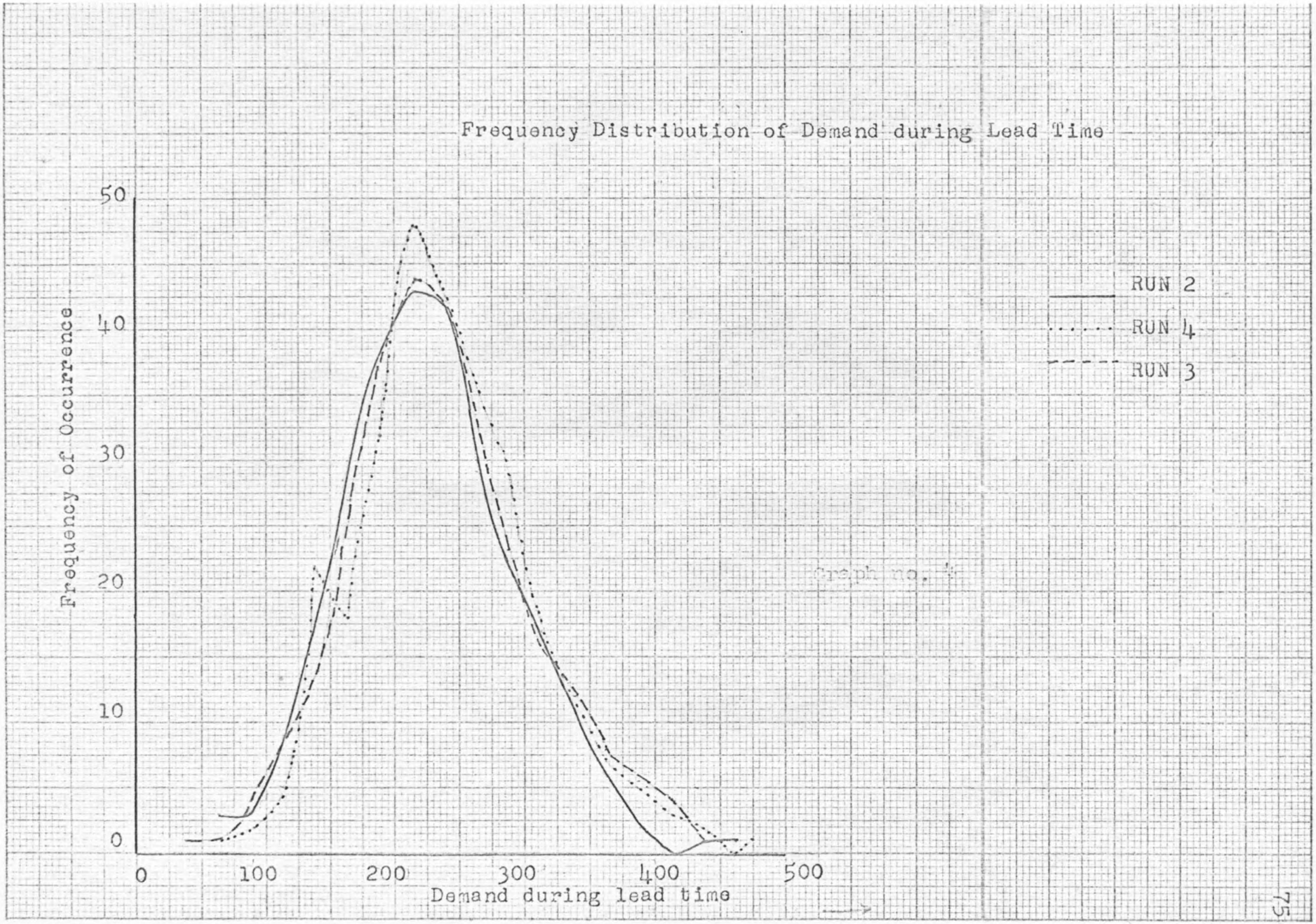
$$= 0.624 \text{ with } 249 \text{ d.f.}$$

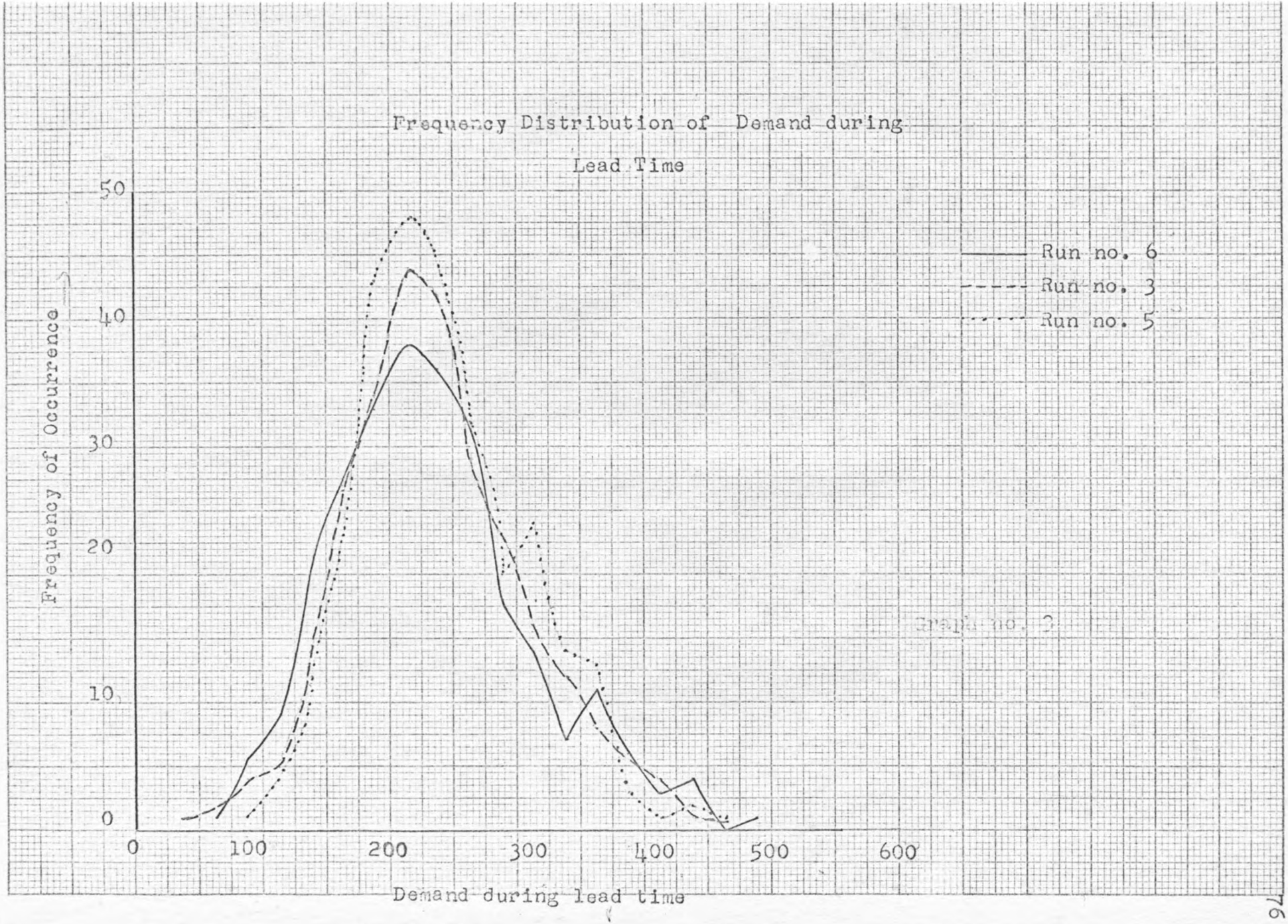
for significance at the 5% level t should be greater or equal to 1.96. Hence, the means are not significantly different.

For the demand during lead time for these runs (i.e. 2, 3 and 4) the value of F for the extreme variances was 1.18. Indicating again that the variances are not significantly different. The value of 't' was 1.4 showing that the means are not significantly different either.

For run numbers 5,3 and 6 where we kept the demand constant in each case and the order lead time was varied by changing the standard deviation. It was noted that as the standard deviation of order lead time increases, the standard deviation of demand during a lead time gradually increases. Also the mean of the demand during a lead time decreases. Here the value of F for extreme variances is 1.32. For significance at the 5% level the value of F should be equal to or greater than 1.25 (from tables). As the value obtained is greater than


Frequency Distribution of Physical Demand
Run 2
Run 4
Run 3
Frequency of Occurrence
0
10
20
30
40
50
60
50
100
150
200
250
300
350
DEMAND QUANTITY


Frequency Distribution of Demand during Lead Time
Frequency of Occurrence
50
40
30
20
10
0
0
100
200
300
400
500
Demand during lead time
RUN 2
RUN 4
RUN 3
Graph no. 4


Frequency Distribution of Demand during Lead Time
Run no. 6
Run no. 3
Run no. 5
Frequency of Occurrence
50
40
30
20
10
0
0
100
200
300
400
500
600
Demand during lead time

1.25, we conclude that there is significance between number 5 and number 6 runs. That is the variances are significantly different. When testing run 5 and 6 with run 3 for significance the value of F obtained were 1.17 and 1.18, which are below the critical value 1.25 indicating no significance of run 3 with either run 5 or 6.

The curves of demand during a lead time for all the three cases considered are drawn on graphs 3,4 and 5.

CONCLUSION

From the simulation run we conclude:

(i) The Exponential and Normal Generators when run in the system as a whole operate satisfactorily.

(ii) The Computer program based on the proposed inventory model works as planned.

(iii) When the demand standard deviation is changed from 1.5 to 4.5 (a 3 fold change) and the mean is constant, the variances and mean of the actual demand do not significantly change when observed after every 10 units of time.

(iv) When the demand standard deviation is varied from 1.5 to 4.5 with a constant mean, the variances and mean of the demand during a lead time do not change significantly.

(v) When the order lead time standard deviation is changed from 0.5 to 2.5 (a 5 fold change), there is a significant difference in the variance of demand during an order lead time. However, the change is small a three fold (0.5 to 1.5) change shows no significant difference in the variance and mean (same population) of the demand during an order lead time.

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his Major Professor, Dr. L. E. Grosh, for his guidance and helpful assistance during the preparation of this report and for the time and energy he devoted for counsel.

BIBLIOGRAPHY

1) Box G. E. P. and Muller Marvin. E., Princeton University, "A Note on the Generation of Random Normal Variates", Annals of Mathematical Statistics, vol. 29, 1958.

2) Brown R. G., "Statistical Forecasting for Inventory Control", McGraw-Hill, 1959, p. 163.

3) Chu Kong and Naylor, Thomas H., "Two Alternative Methods for Simulating Waiting Line Models", Journal of Industrial Engg., Nov.-December, 1965, Vol. No. 6 pp. 390-394.

4) Feller W., "An Introduction to the Theory of Probability and its Applications", Vol. 1,2, edition, Wiley (1957).

5) Freeman H. A., "Industrial Statistics", John Wiley & Sons Inc., New York.

6) Gordon Geoffery, IBM Corp., "Simulation Languages for Discrete Systems", Proceeding IBM Scientific Computing Symposium Simulation Models and Gaming, pp. 101-118.

7) Holstein W. K. and Soukup William R., "Monte Carlo Simulation", Institute Paper No. 23, Institute for Quantitative Research and Economics and Management, Graduate School of Industrial Administration, Purdue University.

8) Kiviat Philip J., "Development of Discrete Digital Simulation Languages", Sept. 22, 1966, The Rand Corporation, Santa Monica, California.

9) Krasnow Howard S. and Merikallio Reino A., IBM Corp., "The Past, Present and Future of General Simulation Languages", Management Science Journal, Vol. 11, no. 2, November 1964, pp. 236-261.

10) McMillan Claude and Gonzalez Richard F., "System Analysis", A Computer Approach to Decision Models, 1965, Richard D. Irwin Inc., Homewood, Illinois.

11) Seigel Sidney, "Nonparametric Statistics for the Behavioral Sciences", McGraw-Hill Book Company Inc., New York, 1956.

12) Starr Martin K. and Miller David W., "Inventory Control, Theory and Practice", Prentice Hall International Inc., Cliffs, N. J.

APPENDIX 1

PROGRAM TO GET THE OUTPUT SHOWN

IN FORM 'A'(APPENDIX 3)

MAIN PROGRAM

```
 50 FORMAT(2I6,9F6.1,2I6)
908 FORMAT(39H INITIAL KNOWN VALUES OF RUN NO.... ARE)
 550FORMAT(30X,22H IST=INT. PHY. STOCK =I4,40X,16H IBS=BOOK STOCK=I4
   1/1HK,10X,42H SD=STD. DEV. OF DELAY IN ORDER LEAD TIME=F4.1,10X,
   237H TM=MEAN OF DELAY IN ORDER LEAD TIME=F5.1/1HK,50H SD1=STD. DEV.
   3OF DELAY IN ORDER RECEIPT POSTING  =F4.1,10X,45H TM1=MEAN OF DELAY
   4 IN ORDER RECEIPT POSTING =F4.1/1HK,50H SD2=STD. DEV. OF DELAY IN
   5SALES ORDER POSTING   =F4.1,10X,44H TM2=MEAN OFDELAY IN SALES ORDE
   6R POSTING   =F4.1/1HK,18X,37H SD3=STD. DEV. OF DEMAND QUANTITY   =
   7F4.1,18X,32H TM3=MEAN OF DEMAND QUANTITY   =F4.1/1HK,10X,31H ART=A
   8RRIVAL RATE OF CUSTOMERS=F4.1,12X,18H IR=REORDER LEVEL=I5,12X,21H
   1 QS     NOS   LT1    LT2     SO     IBS    ISS     QB     BS    NO
   2BS    BQ   BQQ)
909 FORMAT(120H ----------------------------------------------------
   1------+-------------------------------------------------------)
906 FORMAT                    (1HK,110H     QOL=QTY. OF ORDERS LOST
   1                                       QB=TOTAL QTY. OF BOOK SALES
   2            /1HK,100H    BS=QTY. OF BOOK SALES
   3                          NOBS=NO. OF BOOK SALES       /1HK,100H
   4  NOL=NO. OF ORDERS LOST
   5 BQ=NO. IN QUEUE BEFORE POSTED /1HK,100H      QS=TOTAL QTY. OF SA
   6LES                                    BQQ=QTY. IN BOOK QUEUE
   7         )
905 FORMAT(1HK,129H   EVENT=TIME AT WHICH AN EVENT OCCURS
   1                  NOS=NO. OF SALES
   2            /1HK,124H     SL=QUANTITY OF SALE
   3                         LT1=LEAD TIME BEFORE ORDER IS RECEIVED
   4 AT WAREHOUSE      /1HK,129H    T=ARRIVAL TIME BETWEEN CUSTOMERS
   5                              LT2=LEAD TIME BEFORE ORDER RECE
   6IVED IS POSTED           /1HK,123H     D=DELAY BEFORE IT IS POSTED
   7                               SO=STOCKS ON ORDER
   8                         )
   9IM=MAX. STOCK LEVEL=I5)
200 FORMAT(1HK,131H EVENT    IST      SL      T      D      QOL    NOL
201 FORMAT(/18H TOTAL TIME TAKEN=F8.2,8H MINUTES)
    ITIME=ICLOCK(ITIME)
    Z5=ITIME
    READ(1,50)IST,IBS,SD,TM,SD1,TM1,SD2,TM2,SD3,TM3,ART,IR,IM
    WRITE(3,908)
    WRITE(3,55)IST,IBS,SD,TM,SD1,TM1,SD2,TM2,SD3,TM3,ART,IR,IM
    WRITE(3,909)
    WRITE(3,905)
    WRITE(3,906)
    WRITE(3,200)
    ISS=IBS
    EVENT(1,2)=0.0
    EVENT(1,1)=0.0
    IK=1
    DO 90 I=2,5
    EVENT(I,1)=125.
 90 CONTINUE
 91 I=1
    L=2
  9 IF(EVENT(I,1)-EVENT(L,1))7,8,8
```

MAIN PROGRAM (CONTD.)

```
 7 L=L+1
   IF(L-5)9,9,10
 8 I=L
   L=L+1
   IF(L-5)9,9,10
10 GO TO(15,16,17,18,19),I
150CALL SALE(IST,EVENT,QS,QC,TM2,SD2,TM3,SD3,ART,B,QSC2,N,M,IC1,IC2,
  1IS,I1,ID,IBS,ISS,T)
   GO TO 91
16 CALL POST (IBS,ISS,EVENT,SD,TM,IR,IM,J,QR,QC,M,IB,QB,IK,I3,IST)
   GO TO 91
17 CALL RECEIV(EVENT,IST,QB,SD1,TM1,QR,J,IK)
   GO TO 91
18 CALL BOOK(IBS,EVENT,IK,QB)
   GO TO 91
19 ITIME=ICLOCK(ITIME)
   Z21=ITIME
   Z3=(Z21-Z5)*0.60
   WRITE(3,201)Z3
   STOP
   END
```

SUBROUTINE SALE

```
      SUBROUTINE SALE (IST,EVENT,QS,QC,TM2,SD2,TM3,SD3,ART,B,QSC2,N,M,
     1IC1,IC2,IS,I1,ID,IBS,ISS,T)
      DIMENSION EVENT(6,2),QS(20,2),QC(20,2)
   51 FORMAT(F7.2,2I7,2F7.2,4I7,21X,2I7,21X,2I7)
      ID1=EVENT(1,2)
      IST=IST-ID1
      IF(I1-1)33,81,81
   81 K=1
C     DEPLETE THE QUEUE
      CALL DEPLET(QS,N,EVENT,K)
      IF(IST)30,31,31
   30 IC1=IC1+1
      IC2=IC2+ID1
      IC3=ID1
      I1=I1-1
      IST=IST+ID1
      ID1=0
      M1=QC(1,1)
      M2=QC(1,2)
      V1=0.
      GO TO 33
   31 IC1=0
      IC2=0
      IC3=0
      IS=IS+ID1
      QCC2=QSC2
      CALL ADD(QC,QCC2)
C     ADD IN THE QUEUE OF ORDERS BEFORE POSTING DEPT.
      M1=QC(1,1)
      M2=QC(1,2)
      M=QC(1,1)+1.
      CALL NORMAL(TM2,SD2,V1)
C     ASSUMED DELAY FOLLOWS A NORMAL DISTRIBUTION
      QC(M,1)=B+V1
      QC(M,2)=QCC2
      CALL INSERT(QC,M)
C     INSERT IN THE QUEUE IN PROPER TIME SEQUENCE
      EVENT(2,1)=QC(2,1)
      EVENT(2,2)=QC(2,2)
   33 CALL NORMAL(TM3,SD3,V2)
C     ASSUMED THE DEMAND FOLLOWS A NORMAL DISTRIBUTION
      IV2=V2+0.5
      QSC2=IV2
      CALL ADD(QS,QSC2)
C     ADD IN THE QUEUE OFCUSTOMERS
      N=QS(1,1)+1.
      CALL EXPO(ART,T)
C     ASSUMED RATE OF ARRIVAL FOLLOWS AN EXPONETIAL DISTRIBUTION
      WRITE(3,51)EVENT(1,1),IST,ID1,T,V1,IC2,IC1,IS,I1,IBS,ISS,M1,M2
      QS(N,1)=B+T
```

SUBROUTINE SALE(CONTD.)

```
B=QS(N,1)
QS(N,2)=QSC2
CALL INSERT (QS,N)
EVENT(1,1)=QS(2,1)
EVENT(1,2)=QS(2,2)
I1=I1+1
ID=ID+ID1+IC3
RETURN
END
```

SUBROUTINE POST

```
      SUBROUTINE POST(IBS,ISS,EVENT,SD,TM,IR,IM,J,QR,QC,M,IB,QB,IK,I3,
     1IST)
      DIMENSION EVENT(6,2),QR(20,2),QC(20,2),QB(20,2)
  104 FORMAT(63X,F7.2,7X,I7)
   52 FORMAT(F7.2,I7,70X,7I7)
      IF(M-2)450,450,451
  450 EVENT(2,1)=3000.
      GO TO 42
  451 ISC=QB(1,2)
      IB2=EVENT(2,2)
      T7=EVENT(2,1)
      IB=IB+IB2
      IBS=IBS-IB2
      ISS=IBS+ISC
      K=2
      CALL DEPLET(QC,M,EVENT,K)
C     DEPLETE THE QUEUE OF ORDERS WAITING
      IQ1=QC(1,1)
      IQ2=QC(1,2)
      I3=I3+1
      WRITE(3,52)EVENT(2,1),IST,IBS,ISS,IB2,IB,I3,IQ1,IQ2
      CALL INSERT (QC,M)
      EVENT(2,1)=QC(2,1)
      EVENT(2,2)=QC(2,2)
      IF(IBS+ISC-IR)40,40,42
C     CHECK TO SEE IF ORDER FOR STOCK REPLENISHMENT SHOULD BE PLACED
   40 QBC2=IM-IBS-ISC
      QRC2=QBC2
      CALL ADD(QR,QRC2)
      J=QR(1,1)+1.
      CALL NORMAL(TM,SD,V1)
C     ASSUMED DELAY IN ORDER ARRIVING FOLLOWS A NORMAL DISTRIBUTION
      QR(J,1)=V1+T7
      QR(J,2)=QRC2
      CALL INSERT(QR,J)
      EVENT(3,2)=QR(2,2)
      EVENT(3,1)=QR(2,1)
      I32=EVENT(3,2)
      WRITE(3,104)V1,I32
      CALL ADD(QB,QBC2)
   42 RETURN
      END
```

SUBROUTINE BOOK

```
      SUBROUTINE BOOK(IBS,EVENT,IK,QB)
      DIMENSION EVENT(6,2),QB(20,2)
   54 FORMAT(F7.2,77X,I7,30H ORDER RECEIVED POSTED IN BOOK)
      IBS1=EVENT(4,2)
      IBS=IBS+IBS1
      WRITE(3,54)EVENT(4,1),IBS
      K=4
      CALL DEPLET (QB,IK,EVENT,K)
      IF(IK-1)70,70,71
   70 EVENT(4,1)=3000.
      EVENT(4,2)=0.0
      GO TO 72
   71 CALL INSERT (QB,IK)
      EVENT(4,1)=QB(2,1)
      EVENT(4,2)=QB(2,2)
   72 RETURN
      END
```

SUBROUTINE DEPLETE

```
      SUBROUTINE DEPLET(QX,M1,EVENT,K)
      DIMENSION QX(20,2),EVENT(6,2)
      F=QX(1,2)
      M2=QX(1,1)
      DO 50 I=2,M1
      QX(I-1,1)=QX(I,1)
   50 QX(I-1,2)=QX(I,2)
      M1=M1-1
      QX(1,1)=M2-1
      QX(1,2)=F-EVENT(K,2)
      RETURN
      END
```

SUBROUTINE NORMAL

```
      SUBROUTINE NORMAL(T2,S2,V1)
      Z2=IRANDM(56910)
      Z1=IRANDM(56910)
      Q1=Z1/100000.
      Q2=Z2/100000.
      V1=(-2.0*ALOG(Q1))**0.5*COS(6.283*Q2)*S2+T2
      RETURN
      END
```

SUBROUTINE RECEIVE

```
      SUBROUTINE RECEIV(EVENT,IST,QB,SD1,TM1,QR,J,IK)
      DIMENSION EVENT(6,2),QR(20,2),QB(20,2)
   53 FORMAT(F7.2,I7,28H ORDER RECEIVED AT WAREHOUSE)
  109 FORMAT(70X,F7.2,I7)
      IST1=EVENT(3,2)
      IST=IST+IST1
      WRITE(3,53)EVENT(3,1),IST
      QBO2=EVENT(3,2)
      B1=EVENT(3,1)
      K=3
      CALL DEPLET(QR,J,EVENT,K)
      IF(J-1)60,60,61
   60 EVENT(3,1)=3000.
      EVENT(3,2)=0.0
      GO TO 62
   61 CALL INSERT(QR,J)
      EVENT(3,1)=QR(2,1)
      EVENT(3,2)=QR(2,2)
   62 CALL NORMAL(TM1,SD1,V4)
      IK=IK+1
      QB(IK,1)=B1+V4
      QB(IK,2)=QBO2
      CALL INSERT(QB,IK)
      EVENT(4,1)=QB(2,1)
      EVENT(4,2)=QB(2,2)
      I42=EVENT(4,2)
      WRITE(3,109)V4,I42
      RETURN
      END
```

SUBROUTINE ADD

```
      SUBROUTINE ADD(QX,QXO2)
      DIMENSION QX(20,2)
      QX(1,1)=QX(1,1)+1.
      QX(1,2)=QX(1,2)+QXO2
      RETURN
      END
```

SUBROUTINE INSERT

```
      SUBROUTINE INSERT(QX,MM)
      DIMENSION QX(20,2)
      QXC1=QX(2,1)
      QXC2=QX(2,2)
      NN=MM-1
      DO 12 I=1,NN
      KK=MM-I+1
      IF(QXC1-QX(KK,1))12,12,13
   13 X=QXC1
      X1=QXC2
      QXC1=QX(KK,1)
      QXC2=QX(KK,2)
      QX(KK,1)=X
      QX(KK,2)=X1
   12 CONTINUE
      QX(2,2)=QXC2
      QX(2,1)=QXC1
      RETURN
      END
```

SUBROUTINE EXPONENTIAL

```
      SUBROUTINE EXPO(ART,T)
      Y=IRANDM(56910)
      Y1=Y/100000.
      T=(-1.0/ART)*ALOG(Y1)
      RETURN
      END
```

FUNCTION RANDOM GENERATOR

```
 1AMON$$    EXEQ AUTOCODER,,,NOPCH
 2A         HEADR*****RANDOM NUMBER GENERATOR SUBROUTINE*****
 3A         TITLEIRANDM
 4A         SBR  X13
 5A         MLNA 4+X13,*+6
 6A         MLNB 0,RECEIVE=3
 7A         C    RECEIVE,STORE=3
 8A         BE   CALC
 9A         MLNA RECEIVE,STORE
10A         MLNA -0000001-,RNUM=21-11
11A         MLNA STORE,RNUM-18
12ACALC     M    -1977326743-,RNUM
13A         MLCWARNUM-5,299
14A         SW   295
15A         MLNB RNUM,RNUM-11
16A         B    5+X13
17A         LTORG*
18A         DCW  -)-
19A         END
```

FUNCTION CLOCK

```
MON$$       EXEQ AUTOCODER,,,NOPCH
            TITLEICLOCK
            SBR  X13
            MLCWABLANKS,299
            STC  299
            BCE  *-18,295,9
            B    5+X13
BLANKS      DCW  =5
            DCW  =1
            END
```

APPENDIX 2

PROGRAM TO GET THE OUTPUT SHOWN
IN FORM 'B' (APPENDIX 4)

```
    DIMENSION QS(15,2),QC(15,2),QR(15,2),EVENT(5,2),QB(15,2)
    DIMENSION AD1(4,2),IB1(4,2),IS1(4,2)
 50 FORMAT(2I6,9F6.1,2I6)
200 FORMAT(/18H TOTAL TIME TAKEN=F8.2,8H MINUTES)
908 FORMAT(39H INITIAL KNOWN VALUES OF RUN NO.... ARE)
 550FORMAT(30X,22H IST=INT. PHY. STOCK =I4,40X,16H IBS=BOOK STOCK=I4
   1/1HK,10X,42H SD=STD. DEV. OF DELAY IN ORDER LEAD TIME=F4.1,10X,
   237H TM=MEAN OF DELAY IN ORDER LEAD TIME=F5.1/1HK,50H SD1=STD. DEV.
   3OF DELAY IN ORDER RECEIPT POSTING  =F4.1,10X,45H TM1=MEAN OF DELAY
   4 IN ORDER RECEIPT POSTING =F4.1/1HK,50H SD2=STD. DEV. OF DELAY IN
   5SALES ORDER POSTING   =F4.1,10X,44H TM2=MEAN OFDELAY IN SALES ORDE
   6R POSTING   =F4.1/1HK,18X,37H SD3=STD. DEV. OF DEMAND QUANTITY   =
   7F4.1,18X,32H TM3=MEAN OF DEMAND QUANTITY   =F4.1/1HK,10X,31H ART=A
   8RRIVAL RATE OF CUSTOMERS=F4.1,12X,18H IR=REORDER LEVEL=I5,12X,21H
   9IM=MAX. STOCK LEVEL=I5)
909 FORMAT(120H ------------------------------------------------------
   1------+-------------------------------------------------------)
855 FORMAT(1HK,128H      T        NS      PS       PD        NOL      QOL
   1  NBS    QBSS    NBQ      QBQ       LT      DLT      ASLT    NSLT   BS
   2LT    NBSLT)
777 FORMAT(1HK,10X,45H T=TIME AT THE SYSTEM IS CHECKED                ,20
   1X,45H NS=NO. OF SALES DURING THIS PERIOD          /1HK,10X,45H PS=
   2ACTUAL SALE(QTY.) DURING THIS PERIOD      ,20X,45H PD=ACTUAL DEMAND
   3(QTY.) DURING THIS PERIOD   /1HK,10X,45H NOL=NO. OF ORDERS LOST IN
   4 THIS PERIOD        ,20X,45H QOL=QTY. OF ORDERS LOST IN THIS PERIOD
   5      /1HK,10X,45H NBS=NO. OF BOOK SALES DURING THIS PERIOD    ,20
   6X,45H QBS=QTY. OF BOOK SALES IN THIS PERIOD        /1HK,10X,45H NBQ
   7=NO. IN BOOK QUEUE IN THIS PERIOD         ,20X,45H QBQ=QTY. IN BOOK
   8 QUEUE IN THIS PERIOD       /1HK,10X,45H LT=LEAD TIME IN ORDER REC
   9EIVING              ,20X,38H DLT=DEMAND DURING LEAD TIME          )
778 FORMAT(1HK,10X,45H ASLT=ACTUAL SALES DURUNG LEAD TIME          ,20
   1X,45H NSLT=NO. OF SALES DURING LEAD TIME          /1HK,10X,45H BSL
   2T=QTY. OF SALES POSTED INBOOK  IN L.T.   ,20X,45H NBSLT=NO. OF SAL
   3ES RECORDED IN THIS PERIOD  )
500 FORMAT(I6)
501 FORMAT(1HK,16H CHECKING POINT=I6,8H MINUTES)
809 FORMAT(10I4)
800 FORMAT(10I8)
    ITIME=ICLOCK(ITIME)
    Z5=ITIME
    READ(1,500)KP
    READ(1,50)IST,IBS,SD,TM,SD1,TM1,SD2,TM2,SD3,TM3,ART,IR,IM
    WRITE(3,908)
    WRITE(3,55)IST,IBS,SD,TM,SD1,TM1,SD2,TM2,SD3,TM3,ART,IR,IM
    WRITE(3,501)KP
    WRITE(3,909)
    WRITE(3,777)
    WRITE(3,778)
    WRITE(3,855)
    EVENT(1,1)=0.0
    K1=KP
    EVENT(1,2)=0.0
    IK=1
```

MAIN PROGRAM (CONTD.)

```
      ISS=IBS
      A1=K1
      DO 90 I=2,5
      EVENT(I,1)=125.
   90 CONTINUE
   91 I=1
      L=2
    9 IF(EVENT(I,1)-EVENT(L,1))7,8,8
    7 L=L+1
      IF(L-5)9,9,10
    8 I=L
      L=L+1
      IF(L-5)9,9,10
   10 GO TO(15,16,17,18,19),I
   15 IF(EVENT(1,1)-A1)150,151,151
 1500CALL SALE(IST,EVENT,QS,QC,TM2,SD2,TM3,SD3,ART,B,QSC2,N,M,IC1,IC2,
     1IS,I1,ID,IBS,ISS,T,I4,I5)
      GO TO 91
   16 IF(EVENT(2,1)-A1)152,151,151
  152 CALL POST (IBS,ISS,EVENT,SD,TM,IR,IM,J,QR,QC,M,IB,QB,I3,IST,ID,AD1
     1,LL,I1,IS,IS1,IB1)
      GO TO 91
   17 CALL RECEIV(EVENT,IST,QB,SD1,TM1,QR,J,IK)
      GO TO 91
   18 CALL BOOK(IBS,EVENT,IK,QB,AD1,I1,ID,IS,IS1,IB,IB1,I3,LL)
      GO TO 91
  151 J1=I1
      J2=IS
      J3=ID
      J6=I3
      J7=IB
      I1=I1-K4
      ID=ID-JD2
      IS=IS-JS2
      I3=I3-J5
      IB=IB-JB2
      M8=QC(1,1)
      M9=QC(1,2)
      WRITE(3,800)K1,I1,IS,ID,I4,I5,I3,IB,M8,M9
      WRITE(2,809)K1,I1,IS,ID,I4,I5,I3,IB,M8,M9
      K1=K1+KP
      A1=K1
      K4=J1
      JS2=J2
      JD2=J3
      J5=J6
      JB2=J7
      I1=J1
      IS=J2
      ID=J3
      I3=J6
      IB=J7
```

MAIN PROGRAM (CONTD.)

```
      I4=0
      I5=0
      GO TO 91
   19 ITIME=ICLOCK(ITIME)
      Z21=ITIME
      Z3=(Z21-Z5)*0.60
      WRITE(3,200)Z3
      STOP
      END
```

SUBROUTINE SALE

```
      SUBROUTINE SALE (IST,EVENT,QS,QC,TM2,SD2,TM3,SD3,ART,B,QSC2,N,M,
     1IC1,IC2,IS,I1,ID,IBS,ISS,T,I4,I5)
      DIMENSION EVENT(5,2),QS(15,2),QC(15,2)
      ID1=EVENT(1,2)
      IST=IST-ID1
      IF(N1-1)33,81,81
   81 K=1
C     DEPLETE THE QUEUE
      CALL DEPLET(QS,N,EVENT,K)
      IF(IST)30,31,31
   30 IC1=IC1+1
      IC3=ID1
      IC2=IC2+ID1
      IST=IST+ID1
      I4=I4+1
      ID1=0
      GO TO 33
   31 IC1=0
      I1=I1+1
      IC2=0
      IC3=0
      IS=IS+ID1
      QCC2=QSC2
      CALL ADD(QC,QCC2)
C     ADD IN THE QUEUE OF ORDERS BEFORE POSTING DEPT.
      M=QC(1,1)+1.
      CALL NORMAL(TM2,SD2,V1)
C     ASSUMED DELAY FOLLOWS A NORMAL DISTRIBUTION
      QC(M,1)=B+V1
      QC(M,2)=QCC2
      CALL INSERT(QC,M)
C     INSERT IN THE QUEUE IN PROPER TIME SEQUENCE
      EVENT(2,1)=QC(2,1)
      EVENT(2,2)=QC(2,2)
   33 CALL NORMAL(TM3,SD3,V2)
C     ASSUMED THE DEMAND FOLLOWS A NORMAL DISTRIBUTION
      IV2=V2+0.5
      QSC2=IV2
      CALL ADD(QS,QSC2)
C     ADD IN THE QUEUE OFCUSTOMERS
      N=QS(1,1)+1.
      CALL EXPO(ART,T)
C     ASSUMED RATE OF ARRIVAL FOLLOWS AN EXPONETIAL DISTRIBUTION
      QS(N,1)=B+T
      B=QS(N,1)
      QS(N,2)=QSC2
      CALL INSERT (QS,N)
      EVENT(1,1)=QS(2,1)
      EVENT(1,2)=QS(2,2)
      N1=N1+1
      I5=I5+IC3
      ID=ID+ID1+IC3
      RETURN
      END
```

SUBROUTINE POST

```
      SUBROUTINE POST(IBS,ISS,EVENT,SD,TM,IR,IM,J,QR,QC,M,IB,QB,I3,IST,
     1ID,AD1,LL,I1,IS,IS1,IB1)
      DIMENSION EVENT(5,2),QR(15,2),QC(15,2),QB(15,2)
      DIMENSION AD1(4,2),IB1(4,2),IS1(4,2)
      IF(M-2)450,450,451
  450 EVENT(2,1)=3000.
      GO TO 42
  451 ISC=QB(1,2)
      IB2=EVENT(2,2)
      T7=EVENT(2,1)
      IR=IR+IB2
      IBS=IBS-IB2
      ISS=IBS+ISC
      K=2
      CALL DEPLET(QC,M,EVENT,K)
C     DEPLETE THE QUEUE OF ORDERS WAITING
      CALL INSERT (QC,M)
      EVENT(2,1)=QC(2,1)
      EVENT(2,2)=QC(2,2)
      I3=I3+1
      IF(IBS+ISC-IR)40,40,42
C     CHECK TO SEE IF ORDER FOR STOCK REPLENISHMENT SHOULD BE PLACED
   40 QBC2=IM-IBS-ISC
      QRC2=QBC2
      CALL ADD(QR,QRC2)
      J=QR(1,1)+1.
      CALL NORMAL(TM,SD,V1)
C     ASSUMED DELAY IN ORDER ARRIVING FOLLOWS A NORMAL DISTRIBUTION
      QR(J,1)=V1+T7
      QR(J,2)=QRC2
      CALL INSERT(QR,J)
      EVENT(3,2)=QR(2,2)
      EVENT(3,1)=QR(2,1)
      CALL ADD(QB,QBC2)
      LL=LL+1
      IS1(LL,2)=I1
      IS1(LL,1)=IS
      IB1(LL,1)=IB
      AD1(LL,1)=ID
      IB1(LL,2)=I3
      AD1(LL,2)=T7
   42 RETURN
      END
```

SUBROUTINE RECEIVE

```
      SUBROUTINE RECEIV(EVENT,IST,QB,SD1,TM1,QR,J,IK)
      DIMENSION EVENT(5,2),QR(15,2),QB(15,2)
      IST1=EVENT(3,2)
      IST=IST+IST1
      QBC2=EVENT(3,2)
      B1=EVENT(3,1)
      K=3
      CALL DEPLET(QR,J,EVENT,K)
      IF(J-1)60,60,61
   60 EVENT(3,1)=3000.
      EVENT(3,2)=0.0
      GO TO 62
   61 CALL INSERT(QR,J)
      EVENT(3,1)=QR(2,1)
      EVENT(3,2)=QR(2,2)
   62 CALL NORMAL(TM1,SD1,V4)
      IK=IK+1
      QB(IK,1)=B1+V4
      QB(IK,2)=QBC2
      CALL INSERT(QB,IK)
      EVENT(4,1)=QB(2,1)
      EVENT(4,2)=QB(2,2)
      RETURN
      END
```

SUBROUTINE DEPLETE

```
      SUBROUTINE DEPLET(QX,M1,EVENT,K)
      DIMENSION QX(15,2),EVENT(5,2)
      M2=QX(1,1)
      F=QX(1,2)
      DO 50 I=2,M1
      QX(I-1,1)=QX(I,1)
   50 QX(I-1,2)=QX(I,2)
      M1=M1-1
      QX(1,1)=M2-1
      QX(1,2)=F-EVENT(K,2)
      RETURN
      END
```

SUBROUTINE EXPONENTIAL

```
      SUBROUTINE EXPO(ART,T)
      Y=IRANDM(56910)
      Y1=Y/100000.
      T=(-1.0/ART)*ALOG(Y1)
      RETURN
      END
```

SUBROUTINE BOOK

```
      SUBROUTINE BOOK(IBS,EVENT,IK,QB,AD1,I1,ID,IS,IS1,IB,IB1,I3,LL)
      DIMENSION EVENT(5,2),QB(15,2)
      DIMENSION AD1(4,2),IB1(4,2),IS1(4,2)
  600 FORMAT(80X,F8.2,5I8)
  609 FORMAT(40X,F5.1,5I5)
      I4=AD1(1,1)
      T9=EVENT(4,1)-AD1(1,2)
      I10=I1-IS1(1,2)
      ID5=ID-I4
      IS5=IS-IS1(1,1)
      IB5=IB-IB1(1,1)
      I6=I3-IB1(1,2)
      WRITE(3,600)T9,ID5,IS5,I10,IB5,I6
      WRITE(2,609)T9,ID5,IS5,I10,IB5,I6
      IBS1=EVENT(4,2)
      IBS=IBS+IBS1
      K=4
      CALL DEPLET (QB,IK,EVENT,K)
      IF(IK-1)70,70,71
   70 EVENT(4,1)=3000.
      EVENT(4,2)=0.0
      IF(LL-2)72,73,73
   71 CALL INSERT (QB,IK)
      EVENT(4,1)=QB(2,1)
      EVENT(4,2)=QB(2,2)
   73 AD1(1,1)=AD1(2,1)
      IS1(1,1)=IS1(2,1)
      IS1(1,2)=IS1(2,2)
      IB1(1,2)=IB1(2,2)
      AD1(1,2)=AD1(2,2)
      IB1(1,1)=IB1(2,1)
      IF(LL-3)72,74,74
   74 IS1(2,1)=IS1(3,1)
      IB1(2,1)=IB1(3,1)
      AD1(2,2)=AD1(3,2)
      AD1(2,1)=AD1(3,1)
      IB1(2,2)=IB1(3,2)
      IS1(2,2)=IS1(3,2)
      IF(LL-4)72,75,75
   75 IS1(3,1)=IS1(4,1)
      IB1(3,1)=IB1(4,1)
      AD1(3,1)=AD1(4,1)
      AD1(3,2)=AD1(4,2)
      IB1(3,2)=IB1(4,2)
      IS1(3,2)=IS1(4,2)
   72 LL=LL-1
      RETURN
      END
```

SUBROUTINE INSERT

```
      SUBROUTINE INSERT(QX,MM)
      DIMENSION QX(15,2)
      QXC1=QX(2,1)
      QXC2=QX(2,2)
      NN=MM-1
      DO 12 I=1,NN
      KK=MM-I+1
      IF(QXC1-QX(KK,1))12,12,13
   13 X=QXC1
      X1=QXC2
      QXC1=QX(KK,1)
      QXC2=QX(KK,2)
      QX(KK,1)=X
      QX(KK,2)=X1
   12 CONTINUE
      QX(2,2)=QXC2
      QX(2,1)=QXC1
      RETURN
      END
```

SUBROUTINE ADD

```
      SUBROUTINE ADD(QX,QXC2)
      DIMENSION QX(15,2)
      QX(1,1)=QX(1,1)+1.
      QX(1,2)=QX(1,2)+QXC2
      RETURN
      END
```

SUBROUTINE NORMAL

```
      SUBROUTINE NORMAL(T2,S2,V1)
      Z2=IRANDM(56910)
      Z1=IRANDM(56910)
      Q1=Z1/100000.
      Q2=Z2/100000.
      V1=(-2.0*ALOG(Q1))**0.5*COS(6.283*Q2)*S2+T2
      RETURN
      END
```

FUNCTION RANDOM GENERATOR

```
 1AMON$$    EXEQ AUTOCODER,,,NOPCH
 2A         HEADR*****RANDOM NUMBER GENERATOR SUBROUTINE*****
 3A         TITLEIRANDM
 4A         SBR  X13
 5A         MLNA 4+X13,*+6
 6A         MLNB 0,RECEIVE=3
 7A         C    RECEIVE,STORE=3
 8A         BE   CALC
 9A         MLNA RECEIVE,STORE
10A         MLNA -0000001-,RNUM=21-11
11A         MLNA STORE,RNUM-18
12ACALC     M    -1977326743-,RNUM
13A         MLCWARNUM-5,299
14A         SW   295
15A         MLNB RNUM,RNUM-11
16A         B    5+X13
17A         LTORG*
18A         DCW  -)-
19A         END
```

FUNCTION CLOCK

```
MON$$       EXEQ AUTOCODER,,,NOPCH
            TITLEICLOCK
            SBR  X13
            MLCWABLANKS,299
            STC  299
            BCE  *-18,295,9
            B    5+X13
BLANKS      DCW  =5
            DCW  =1
            END
```

APPENDIX 3

FORM 'A'

INITIAL KNOWN VALUES OF RUN NO.... ARE

IST=INT. PHY. STOCK = 300 IBS=BOOK STOCK= 300

SD=STD. DEV. OF DELAY IN ORDER LEAD TIME= 1.5 TM=MEAN OF DELAY IN ORDER LEAD TIME= 8.0

SD1=STD. DEV.OF DELAY IN ORDER RECEIPT POSTING = 1.5 TM1=MEAN OF DELAY IN ORDER RECEIPT POSTING = 7.5

SD2=STD. DEV. OF DELAY IN SALES ORDER POSTING = 1.5 TM2=MEAN OFDELAY IN SALES ORDER POSTING = 7.5

SD3=STD. DEV. OF DEMAND QUANTITY = 1.5 TM3=MEAN OF DEMAND QUANTITY =15.0

ART=ARRIVAL RATE OF CUSTOMERS= 1.0 IR=REORDER LEVEL= 200 IM=MAX. STOCK LEVEL= 300

EVENT=TIME AT WHICH AN EVENT OCCURS

SL=QUANTITY OF SALE

T=ARRIVAL TIME BETWEEN CUSTOMERS

D=DELAY BEFORE IT IS POSTED

QOL=QTY. OF ORDERS LOST

BS=QTY. OF BOOK SALES

NOL=NO. OF ORDERS LOST

QS=TOTAL QTY. OF SALES

NOS=NO. OF SALES

LT1=LEAD TIME BEFORE ORDER IS RECEIVED AT WAREHOUSE

LT2=LEAD TIME BEFORE ORDER RECEIVED IS POSTED

SO=STOCKS ON ORDER

QB=TOTAL QTY. OF BOOK SALES

NOBS=NO. OF BOOK SALES

BQ=NO. IN QUEUE BEFORE POSTED

BQQ=QTY. IN BOOK QUEUE

EVENT	IST	SL	T	D	QOL	NOL	QS	NOS	LT1	LT2	SO	IBS	ISS	QB	BS	NOBS	BQ	BQQ
.00	300	0	.24	.00	0	0	0	0				300	300				0	0
.24	286	14	1.08	8.53	0	0	14	1				300	300				1	14
1.32	270	16	.46	7.32	0	0	30	2				300	300				2	30
1.78	256	14	.57	6.51	0	0	44	3				300	300				3	44
2.35	241	15	.79	8.41	0	0	59	4				300	300				4	59
3.15	227	14	1.86	8.07	0	0	73	5				300	300				5	73
5.02	213	14	1.09	7.82	0	0	87	6				300	300				6	87
6.12	200	13	.09	9.46	0	0	100	7				300	300				7	100
6.21	188	12	1.88	8.95	0	0	112	8				300	300				8	112
8.09	174	14	.74	7.55	0	0	126	9				300	300				9	126
8.30	174											286	286	14	14	1	8	112
8.64	174											270	270	16	30	2	7	96
8.78	174											256	256	14	44	3	6	82
8.84	160	14	1.61	7.41	0	0	140	10				256	256				7	96
10.45	148	12	1.04	6.49	0	0	152	11				256	256				8	108
10.77	148											241	241	15	59	4	7	93
11.23	148											227	227	14	73	5	6	79
11.50	136	12	.71	7.27	0	0	164	12				227	227				7	91
12.21	122	14	2.35	7.06	0	0	178	13				227	227				8	105
12.85	122											213	213	14	87	6	7	91
14.56	105	17	.46	7.90	0	0	195	14				213	213				8	108
15.02	90	15	2.38	9.17	0	0	210	15				213	213				9	123
15.17	90											201	201	12	99	7	8	111
15.58	90											188	188	13	112	8	7	98
									8.46		112							
15.65	90											174	286	14	126	9	6	84
16.26	90											160	272	14	140	10	5	70
16.95	90											148	260	12	152	11	4	58
17.41	77	13	1.28	9.79	0	0	223	16				148	260				5	71
18.69	60	17	2.40	7.53	0	0	240	17				148	260				6	88
18.77	60											136	248	12	164	12	5	76
19.27	60											122	234	14	178	13	4	62
21.10	44	16	1.39	7.24	0	0	256	18				122	234				5	78
22.46	44											105	217	17	195	14	4	61
22.49	28	16	1.10	6.09	0	0	272	19				105	217				5	77
23.60	14	14	.00	6.74	0	0	286	20				105	217				6	91
23.60	0	14	1.81	6.90	0	0	300	21				105	217				7	105
24.05	112	ORDER RECEIVED AT WAREHOUSE																
										9.00	112							
24.19	112											90	202	15	210	15	6	90
25.41	99	13	.05	7.79	0	0	313	22				90	202				7	103
25.47	84	15	.62	7.35	0	0	328	23				90	202				8	118
26.09	67	17	1.95	6.97	0	0	345	24				90	202				9	135
26.23	67											73	185	17	227	16	8	118
									8.35		115							

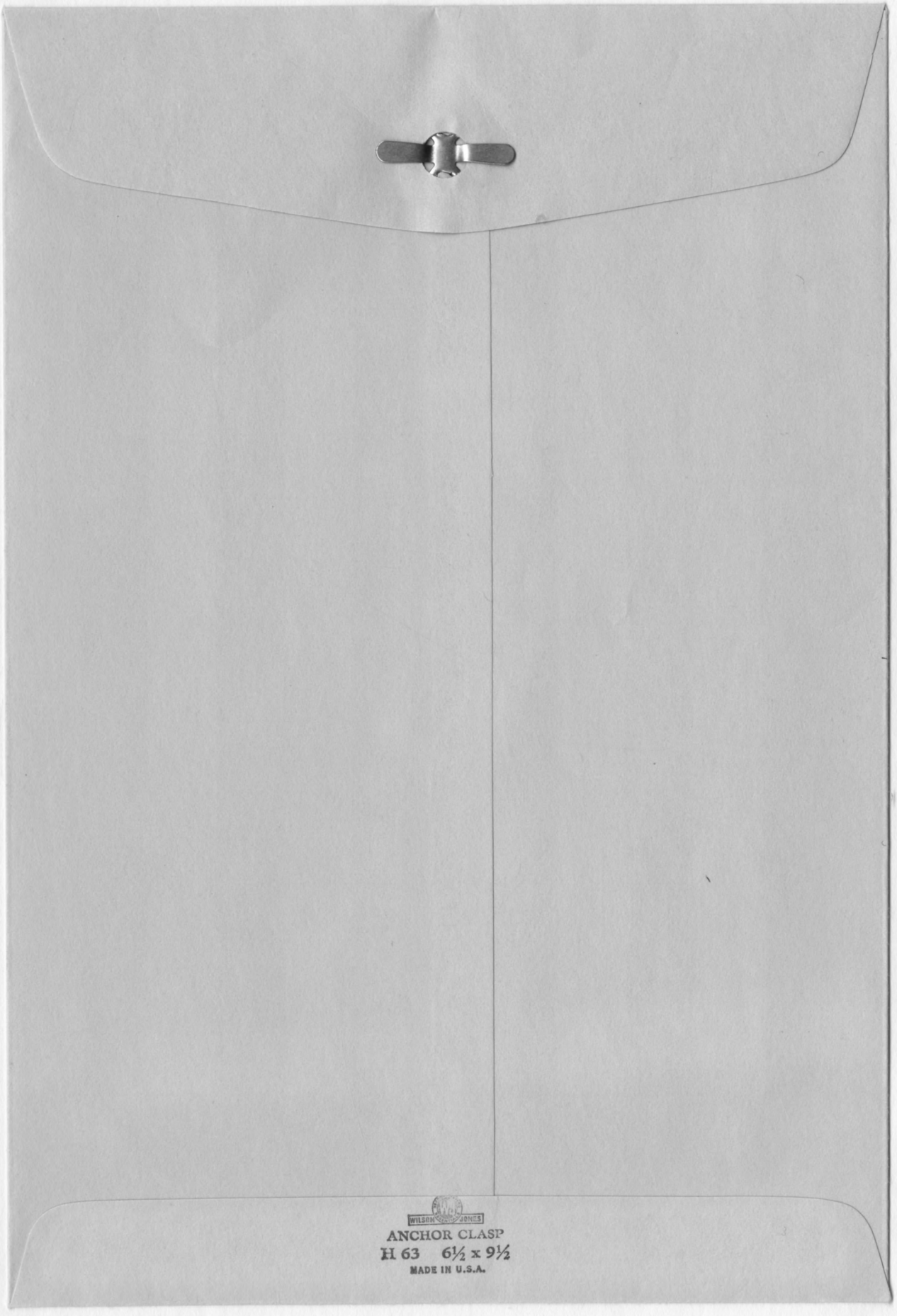
27.20	67										60	287	13	240	17	7	105
28.05	54	13	.60	8.56	0	0	358	25			60	287				8	118
28.34	54										44	271	16	256	18	7	102
28.59	54										28	255	16	272	19	6	86
28.65	41	13	1.30	8.29	0	0	371	26			28	255				7	99
29.96	29	12	.28	7.73	0	0	383	27			28	255				8	111
30.25	12	17	.08	7.43	0	0	400	28			28	255				9	128
30.33	12	0	1.17	.00	15	1	400	28			28	255				9	128
30.34	12										14	241	14	286	20	8	114
30.50	12										0	227	14	300	21	7	100
31.51	12	0	.07	.00	30	2	400	28			0	227				7	100
31.58	12	0	3.54	.00	50	3	400	28			0	227				7	100
32.83	12										-15	212	15	315	22	6	85
33.06											97	ORDER RECEIVED POSTED IN BOOK					
33.06	12										80	195	17	332	23	5	68
									8.33	115							
33.21	12										67	287	13	345	24	4	55
34.58	127	ORDER RECEIVED AT WAREHOUSE															
									7.77	115							
35.13	111	16	.67	7.41	0	0	416	29			67	287				5	71
35.81	97	14	.08	7.38	0	0	430	30			67	287				6	85
35.89	79	18	.04	7.95	0	0	448	31			67	287				7	103
35.94	61	18	1.96	7.43	0	0	466	32			67	287				8	121
36.62	61										54	274	13	358	25	7	108
36.95	61										41	261	13	371	26	6	95
37.68	61										24	244	17	388	27	5	78
37.70	61										12	232	12	400	28	4	66
37.91	48	13	3.54	5.32	0	0	479	33			12	232				5	79
41.39	153	ORDER RECEIVED AT WAREHOUSE															
									8.24	115							
41.45	136	17	2.47	6.11	0	0	496	34			12	232				6	96
42.36											127	ORDER RECEIVED POSTED IN BOOK					
42.55	136										111	216	16	416	29	5	80
43.19	136										97	202	14	430	30	4	66
43.23	136										84	189	13	443	31	3	53
									10.92	111							
43.37	136										66	282	18	461	32	2	35
43.85	136										48	264	18	479	33	1	17
43.92	119	17	.80	7.32	0	0	513	35			48	264				2	34
44.73	106	13	3.01	5.60	0	0	526	36			48	264				3	47
47.56	106										31	247	17	496	34	2	30
47.74	92	14	.39	10.10	0	0	540	37			31	247				3	44
48.14	76	16	.29	5.39	0	0	556	38			31	247				4	60
48.44	59	17	.17	8.39	0	0	573	39			31	247				5	77
48.62	41	18	.43	9.90	0	0	591	40			31	247				6	95
49.05	24	17	.89	7.46	0	0	608	41			31	247				7	112
49.64											136	ORDER RECEIVED POSTED IN BOOK					
49.95	10	14	.26	7.10	0	0	622	42			136	247				8	126
50.21	10	0	.97	.00	14	1	622	42			136	247				8	126
50.34	10										123	234	13	509	35	7	113
51.18	10	0	.01	.00	30	2	622	42			123	234				7	113
51.20	10	0	.05	.00	45	3	622	42			123	234				7	113
51.25	10										106	217	17	526	36	6	96
51.25	10	0	1.22	.00	60	4	622	42			106	217				6	96
52.48	10	0	1.50	.00	72	5	622	42			106	217				6	96
53.53	10										90	201	16	542	37	5	80
53.99	10	0	1.96	.00	84	6	622	42			90	201				5	80
54.16	121	ORDER RECEIVED AT WAREHOUSE															
									5.21	111							
55.95	105	16	1.68	5.45	0	0	638	43			90	201				6	96
56.52	105										73	184	17	559	38	5	79
									8.21	116							
56.83	105										56	283	17	576	39	4	62
57.06	105										42	269	14	590	40	3	48
57.64	90	15	.06	5.24	0	0	653	44			42	269				4	63
57.70	76	14	.10	6.67	0	0	667	45			42	269				5	77
57.80	61	15	.14	6.53	0	0	682	46			42	269				6	92
57.85	61										28	255	14	604	41	5	78
57.95	46	15	.15	8.98	0	0	697	47			28	255				6	93
58.11	29	17	.37	8.26	0	0	714	48			28	255				7	110
58.49	14	15	.07	7.48	0	0	729	49			28	255				8	125
58.52	14										10	237	18	622	42	7	107
58.56	1	13	2.58	7.85	0	0	742	50			10	237				8	120
59.37											121	ORDER RECEIVED POSTED IN BOOK					
61.15	1	0	.46	.00	14	1	742	50			121	237				8	120
61.40	1										105	221	16	638	43	7	104

61.61	1	0	.26	.00	29	2	742	50				105	221				7	104
61.88	1	0	.44	.00	43	3	742	50				105	221				7	104
62.33	1	0	.15	.00	57	4	742	50				105	221				7	104
62.49	1	0	1.14	.00	71	5	742	50				105	221				7	104
62.88	1											90	206	15	653	44	6	89
63.64	1	0	2.38	.00	86	6	742	50				90	206				6	89
64.33	1											75	191	15	668	45	5	74
									7.75		116							
64.38	1											61	286	14	682	46	4	60
64.73	117	ORDER RECEIVED AT WAREHOUSE																
										5.90	116							
65.98	117											46	271	15	697	47	3	45
66.02	99	18	.03	6.63	0	0	760	51				46	271				4	63
66.06	82	17	.32	6.68	0	0	777	52				46	271				5	80
66.37	82											29	254	17	714	48	4	63
66.39	68	14	.93	7.56	0	0	791	53				29	254				5	77
66.42	68											16	241	13	727	49	4	64
66.94	68											1	226	15	742	50	3	49
67.32	54	14	.05	7.73	0	0	805	54				1	226				4	63
67.38	38	16	.32	6.31	0	0	821	55				1	226				5	79
67.70	21	17	.90	9.41	0	0	838	56				1	226				6	96
68.61	9	12	1.14	5.04	0	0	850	57				1	226				7	108
69.76	9	0	.35	.00	15	1	850	57				1	226				7	108
70.11	9	0	1.55	.00	27	2	850	57				1	226				7	108
70.64												117	ORDER RECEIVED POSTED IN BOOK					
71.66	9	0	.22	.00	42	3	850	57				117	226				7	108
71.89	9	0	.27	.00	57	4	850	57				117	226				7	108
72.09	118	ORDER RECEIVED AT WAREHOUSE																
										7.06	109							
72.16	99	19	.89	9.30	0	0	869	58				117	226				8	127
72.65	99											99	208	18	760	51	7	109
72.74	99											82	191	17	777	52	6	92
									7.66		109							
73.06	85	14	.53	7.68	0	0	883	59				82	191				7	106
73.59	68	17	.46	4.77	0	0	900	60				82	191				8	123
73.66	68											70	288	12	789	53	7	111
73.70	68											54	272	16	805	54	6	95
73.96	68											40	258	14	819	55	5	81
74.06	52	16	.76	7.74	0	0	916	61				40	258				6	97
74.82	38	14	1.74	8.60	0	0	930	62				40	258				7	111
75.06	38											26	244	14	833	56	6	97
76.56	21	17	.12	8.79	0	0	947	63				26	244				7	114
76.68	6	15	3.68	10.78	0	0	962	64				26	244				8	129
77.11	6											9	227	17	850	57	7	112
78.37	6											-8	210	17	867	58	6	95
79.16												101	ORDER RECEIVED POSTED IN BOOK					
80.36	6	0	.44	.00	12	1	962	64				101	210				6	95
80.40	115	ORDER RECEIVED AT WAREHOUSE																
										6.36	109							
80.74	115											87	196	14	881	59	5	81
									9.00		104							
80.81	101	14	4.86	6.25	0	0	976	65				87	196				6	95
81.47	101											68	281	19	900	60	5	76
81.80	101											52	265	16	916	61	4	60
83.42	101											38	251	14	930	62	3	46
85.36	101											21	234	17	947	63	2	29
85.67	85	16	.71	9.05	0	0	992	66				21	234				3	45
86.38	71	14	.01	6.50	0	0	1006	67				21	234				4	59
86.40	57	14	.39	8.99	0	0	1020	68				21	234				5	73
86.77												130	ORDER RECEIVED POSTED IN BOOK					
86.79	39	18	1.18	7.26	0	0	1038	69				130	234				6	91
87.06	39											116	220	14	961	64	5	77
87.47	39											101	205	15	976	65	4	62
87.98	24	15	.07	7.01	0	0	1053	70				101	205				5	77
88.05	8	16	1.04	6.44	0	0	1069	71				101	205				6	93
89.10	8	0	.54	.00	16	1	1069	71				101	205				6	93
89.64	8	0	.30	.00	31	2	1069	71				101	205				6	93
89.75	112	ORDER RECEIVED AT WAREHOUSE																
										8.44	104							
89.95	96	16	1.14	8.68	0	0	1085	72				101	205				7	109
91.09	80	16	.12	3.40	0	0	1101	73				101	205				8	125
91.21	65	15	.17	7.89	0	0	1116	74				101	205				9	140
91.39	50	15	1.52	5.81	0	0	1131	75				101	205				10	155
92.89	50											87	191	14	990	66	9	141
									9.04		109							
92.92	35	15	1.14	9.31	0	0	1146	76				87	191				10	156

94.05	35											69	282	18	1008	67	9	138
94.07	20	15	.23	5.30	0	0	1161	77				69	282				10	153
94.30	6	14	1.86	7.58	0	0	1175	78				69	282				11	167
94.49	6											53	266	16	1024	68	10	151
94.50	6											37	250	16	1040	69	9	135
94.73	6											21	234	16	1056	70	8	119
95.00	6											6	219	15	1071	71	7	104
95.39	6											-8	205	14	1085	72	6	90
96.16	6	0	.62	.00	14	1	1175	78				-8	205				6	90
96.79	6	0	.32	.00	28	2	1175	78				-8	205				6	90
97.11	6	0	7.68	.00	44	3	1175	78				-8	205				6	90
97.21	6											-23	190	15	1100	73	5	75
									7.17		109							
98.19												81	ORDER RECEIVED POSTED IN BOOK					
98.63	6											65	284	16	1116	74	4	59
99.11	6											50	269	15	1131	75	3	44
99.37	6											35	254	15	1146	76	2	29
101.89	6											21	240	14	1160	77	1	15
101.94	115	ORDER RECEIVED AT WAREHOUSE																
										8.41	109							
104.39	225	ORDER RECEIVED AT WAREHOUSE																
										8.72	109							
104.80	212	13	1.74	5.52	0	0	1188	79				21	240				2	28
102.23	212											6	225	15	1175	78	1	13
106.54	197	15	.75	4.98	0	0	1203	80				6	225				2	28
107.30	184	13	2.96	8.21	0	0	1216	81				6	225				3	41
110.26	170	14	.16	6.33	0	0	1230	82				6	225				4	55
110.32	170											-7	212	13	1188	79	3	42
110.35												102	ORDER RECEIVED POSTED IN BOOK					
110.43	152	18	.04	7.83	0	0	1248	83				102	212				4	60
110.47	138	14	1.37	6.00	0	0	1262	84				102	212				5	74
111.52	138											87	197	15	1203	80	4	59
									8.37		103							
111.85	122	16	.66	6.51	0	0	1278	85				87	197				5	75
112.51	108	14	1.57	6.77	0	0	1292	86				87	197				6	89
113.11												197	ORDER RECEIVED POSTED IN BOOK					
114.09	93	15	.94	6.08	0	0	1307	87				197	197				7	104
115.04	76	17	1.61	6.74	0	0	1324	88				197	197				8	121
115.51	76											184	287	13	1216	81	7	108
116.48	76											170	273	14	1230	82	6	94
116.60	76											156	259	14	1244	83	5	80
116.65	60	16	.09	6.09	0	0	1340	89				156	259				6	96
116.75	42	18	2.48	7.95	0	0	1358	90				156	259				7	114
118.27	42											138	241	18	1262	84	6	96
118.36	42											122	225	16	1278	85	5	80
119.23	27	15	.74	7.04	0	0	1373	91				122	225				6	95
119.29	27											108	211	14	1292	86	5	81
119.90	130	ORDER RECEIVED AT WAREHOUSE																
										7.94	103							
119.97	117	13	.75	6.38	0	0	1386	92				108	211				6	94
120.18	117											93	196	15	1307	87	5	79
									6.80		104							
120.72	105	12	.17	6.93	0	0	1398	93				93	196				6	91
120.90	92	13	.53	8.67	0	0	1411	94				93	196				7	104
121.43	78	14	1.06	6.27	0	0	1425	95				93	196				8	118
121.78	78											76	283	17	1324	88	7	101
122.50	63	15	.30	8.21	0	0	1440	96				76	283				8	116
122.75	63											60	267	16	1340	89	7	100
122.80	49	14	.49	8.34	0	0	1454	97				60	267				8	114
123.29	35	14	.26	7.43	0	0	1468	98				60	267				9	128
123.56	19	16	.87	6.42	0	0	1484	99				60	267				10	144
124.43	2	17	1.95	7.76	0	0	1501	100				60	267				11	161
124.70	2											42	249	18	1358	90	10	143

TOTAL TIME TAKEN= 3.00 MINUTES

APPENDIX 4



FORM 'B'

INITIAL KNOWN VALUES OF RUN NO.... ARE

IST=INT. PHY. STOCK = 300 IBS=BOOK STOCK= 300

SD=STD. DEV. OF DELAY IN ORDER LEAD TIME= 1.5 TM=MEAN OF DELAY IN ORDER LEAD TIME= 8.0

SD1=STD. DEV.OF DELAY IN ORDER RECEIPT POSTING = 1.5 TM1=MEAN OF DELAY IN ORDER RECEIPT POSTING = 7.5

SD2=STD. DEV. OF DELAY IN SALES ORDER POSTING = 1.5 TM2=MEAN OFDELAY IN SALES ORDER POSTING = 7.5

SD3=STD. DEV. OF DEMAND QUANTITY = 1.5 TM3=MEAN OF DEMAND QUANTITY =15.0

ART=ARRIVAL RATE OF CUSTOMERS= 1.0 IR=REORDER LEVEL= 200 IM=MAX. STOCK LEVEL= 300

CHECKING POINT= 10 MINUTES

T=TIME AT THE SYSTEM IS CHECKED

PS=ACTUAL SALE(QTY.) DURING THIS PERIOD

NOL=NO. OF ORDERS LOST IN THIS PERIOD

NBS=NO. OF BOOK SALES DURING THIS PERIOD

NBQ=NO. IN BOOK QUEUE IN THIS PERIOD

LT=LEAD TIME IN ORDER RECEIVING

ASLT=ACTUAL SALES DURUNG LEAD TIME

BSLT=QTY. OF SALES POSTED INBOOK IN L.T.

NS=NO. OF SALES DURING THIS PERIOD

PD=ACTUAL DEMAND(QTY.) DURING THIS PERIOD

QOL=QTY. OF ORDERS LOST IN THIS PERIOD

QBS=QTY. OF BOOK SALES IN THIS PERIOD

QBQ=QTY. IN BOOK QUEUE IN THIS PERIOD

DLT=DEMAND DURING LEAD TIME

NSLT=NO. OF SALES DURING LEAD TIME

NBSLT=NO. OF SALES RECORDED IN THIS PERIOD

T	NS	PS	PD	NOL	QOL	NBS	QBSS	NBQ	QBQ	LT	DLT	ASLT	NSLT	BSLT	NBSLT
10	10	140	140	0	0	3	44	7	96						
20	7	100	100	0	0	10	134	4	62						
30	10	143	143	0	0	6	94	8	111						
										17.47	240	190	13	203	14
40	6	96	146	3	50	9	128	5	79						
										16.13	201	151	10	173	12
										16.57	208	208	13	164	11
50	9	143	143	0	0	6	96	8	126						
										16.14	330	246	16	179	11
60	8	120	204	6	84	8	126	8	120						
70	7	108	209	7	101	8	120	7	108						
										14.12	325	212	14	183	12
										14.82	277	220	14	199	13
80	7	112	154	3	42	8	125	6	95						
										14.02	163	151	10	170	11
90	8	123	166	3	43	7	109	7	109						
										17.44	288	213	14	219	14
100	6	90	134	3	44	11	170	2	29						
110	3	41	41	0	0	2	29	3	41						
										17.46	143	99	7	198	13
										15.89	117	117	8	103	7
120	11	170	170	0	0	8	117	6	94						

TOTAL TIME TAKEN= 2.40 MINUTES

SIMULATION OF PROBABILISTIC INVENTORY CONTROL MODEL

by

S. K. BATRA

B. E. (Electrical), University of Delhi, India, 1966

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1967

ABSTRACT

The objective of this report is to write a simulation program of a probabilistic inventory model in the FORTRAN language for the IBM 1410 computer. The program structure is similar to the special discrete simulation language SIMSCRIPT, which is used on larger computers, but is not available for the IBM 1410.

The simulations are performed assuming an exponential arrival rate, with a normally distributed demand. A delay has been incorporated between the warehouse and the posting department. The posting delay is assumed to follow a normal distribution with known parameters. The order lead time is also assumed to follow a known normal distribution.

The simulations are performed by changing the variances of the demand and order lead time distribution. The effect of this variation upon the distribution of the customer physical demand during a review period and during a lead time were found to be quite small. Statistical tests were made to insure that the random generators used did adequately represent the desired population.