A SIMULATION MODEL FOR MULTI-CHANNEL, TIME-DEPENDENT QUEUEING SYSTEMS AND AN APPLICATION TO TEST AND EVALUATE AN ANALYTICAL MODEL OF THE U. S. ARMY ACUTE MINOR ILLNESS CLINICS.

Bruce Byron Culmer

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Simulation Model for
Multi-Channel, Time-Dependent Queueing Systems
and an Application to Test and Evaluate
an Analytical Model of the U.S. Army
Acute Minor Illness Clinics

by

Bruce Byron Culmer

September 1975

Thesis Advisor: R. W. Butterworth

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Simulation Model for Multi-Channel, Time-Dependent Queueing Systems and an Application to Test and Evaluate an Analytical Model of the U.S. Army Acute Minor Illness Clinics		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bruce Byron Culmer		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1975
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Many organizations exist within the Department of Defense which exhibit the properties of multi-channel, time-dependent queueing systems. One such system is the Army's Acute Minor Illness Clinic (AMIC). Models of this system can be developed to determine optimum staffing levels and to upgrade the quality of service provided.

This study developed a time-dependent simulation program which was applied to a two queue, multi-channel queueing system. In particular, this model was designed to test and evaluate the results of an analytical model of the AMIC. The results of both of these models was compard to a second simulation program which more closely models the AMIC in order to measure the significant differences between it and the two other models.

A Simulation Model for Multi-Channel, Time-Dependent Queueing Systems and an Application to Test and Evaluate an Analytical Model of the U. S. Army Acute Minor Illness Clinics

by

Bruce Byron Culmer
Lieutenant, United States Navy
B.A., Eastern Nazarene College, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1974

ABSTRACT

Many organizations exist within the Department of Defense which exhibit the properties of multi-channel, time-dependent queueing systems. One such system is the Army's Acute Minor Illness Clinic (AMIC). Models of this system can be developed to determine optimum staffing levels and to upgrade the quality of service provided.

This study developed a time-dependent simulation program which was applied to a two queue, multi-channel queueing system. In particular, this model was designed to test and evaluate the results of an analytical model of the AMIC. The results of both of these models are compared to a second simulation program which more closely models the AMIC in order to measure the significant differences between it and the two other models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. BACKGROUND

A. AMIC PROGRAM DESCRIPTION

An analytical model has been developed utilizing an in-homogeneous Markov chain and transition matrices to approximate solutions of a multi-channel, time-dependent queueing system. The system that was modeled is the Acute Minor Illness Clinic (AMIC) which is currently in use by the U. S. Army at about twenty of their hospitals. The system integrates doctors and specially trained physician assistants called Amosists into a dual service process.

Patients are screened as they arrive by a group of Amosists at a reception desk called the Triage. The nature of the complaint is determined and the patients are sent to a doctor or Amosist for treatment. A patient may request to be seen by a doctor in any event. Patients are treated on a first come, first served basis as an Amosist or doctor becomes available.

Amosists treat patients according to set procedures detailed in technical manuals and occasionally on computer programs. These procedures, set up by the staff of physicians, indicate the actions an Amosist must take in treating a patient, and require the Amosist to seek consultations with a physician when the problem is beyond his abilities. This procedure helps to eliminate errors in diagnosis and treatment. When a consultation is indicated, the Amosist locates the

first available physician and together they treat the physician. Figure 1 depicts the patient flow of this system.

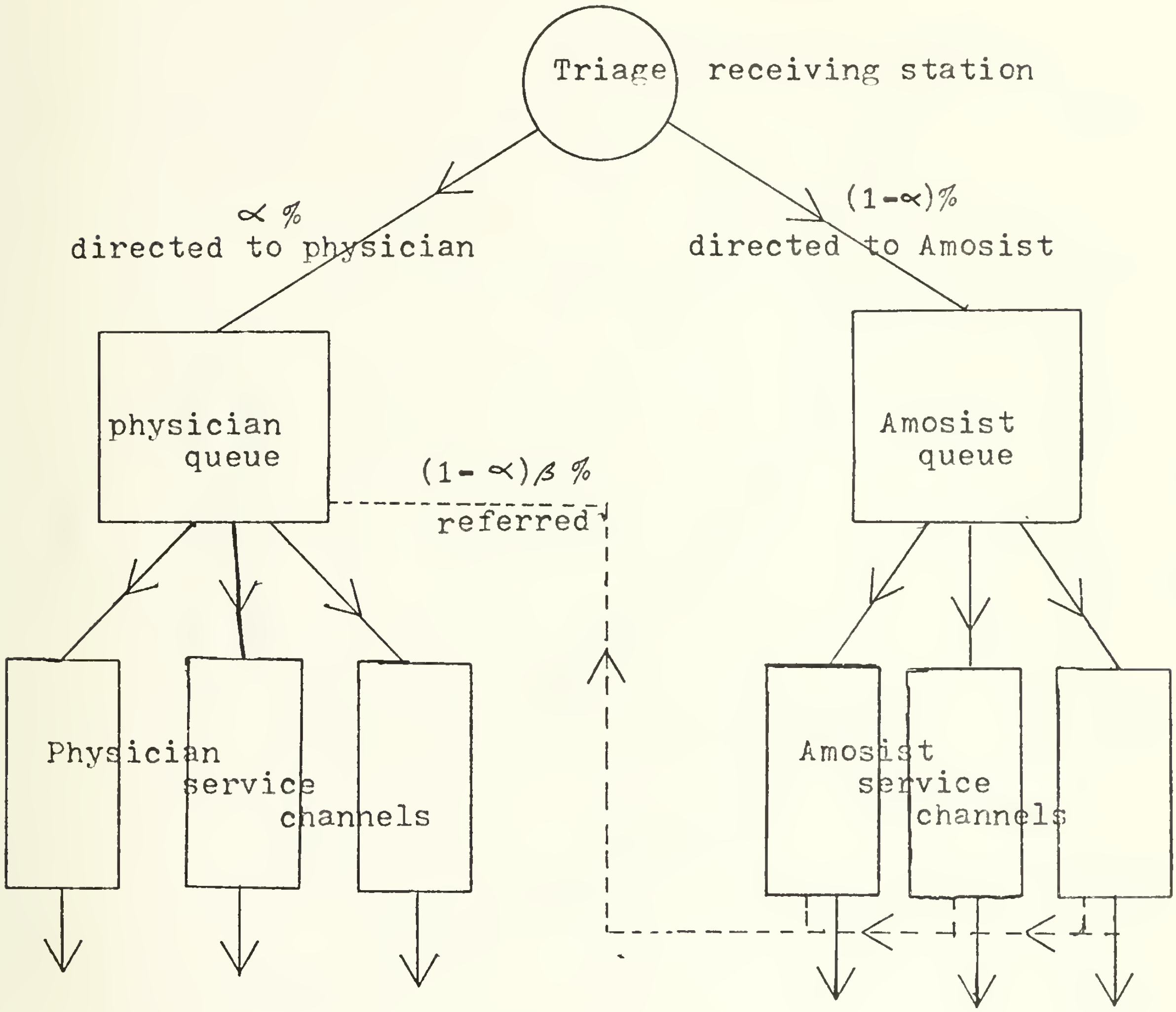
B. ANALYTICAL MODEL DEVELOPMENT

Under the direction of Professor Richard W. Butterworth, an analytical model of the stochastic process was developed with certain simplifying assumptions of the AMIC. Statistics were gathered by David L. VanAsdlen and Leonard O. Wahlig, and an initial model was implemented. The assumptions of this model are listed below.

First, the arrival stream is an inhomogeneous poisson process. Log sheets for 75 days were used to determine arrival rate function, $\lambda(t)$, for each day of the week. Upon examination, it was determined that the arrival function is the same for each day of the week with the exception that the expected number of arrivals varies. The function is divided into 32 time periods with a mean arrival rate for each period. Figure 2 is a graph of the input arrival function for the model.

Second, the distribution of services is exponential. The Amosist system has a distribution of service times for patients that do not see a doctor and a distribution of times for patients that are referred. The difference in mean service times is a result of the amount of time that an Amosist waits with the patient for a doctor to become available and to treat the patient. The doctors also have two distributions

DIAGRAM OF PATIENT FLOW IN THE REAL CLINIC



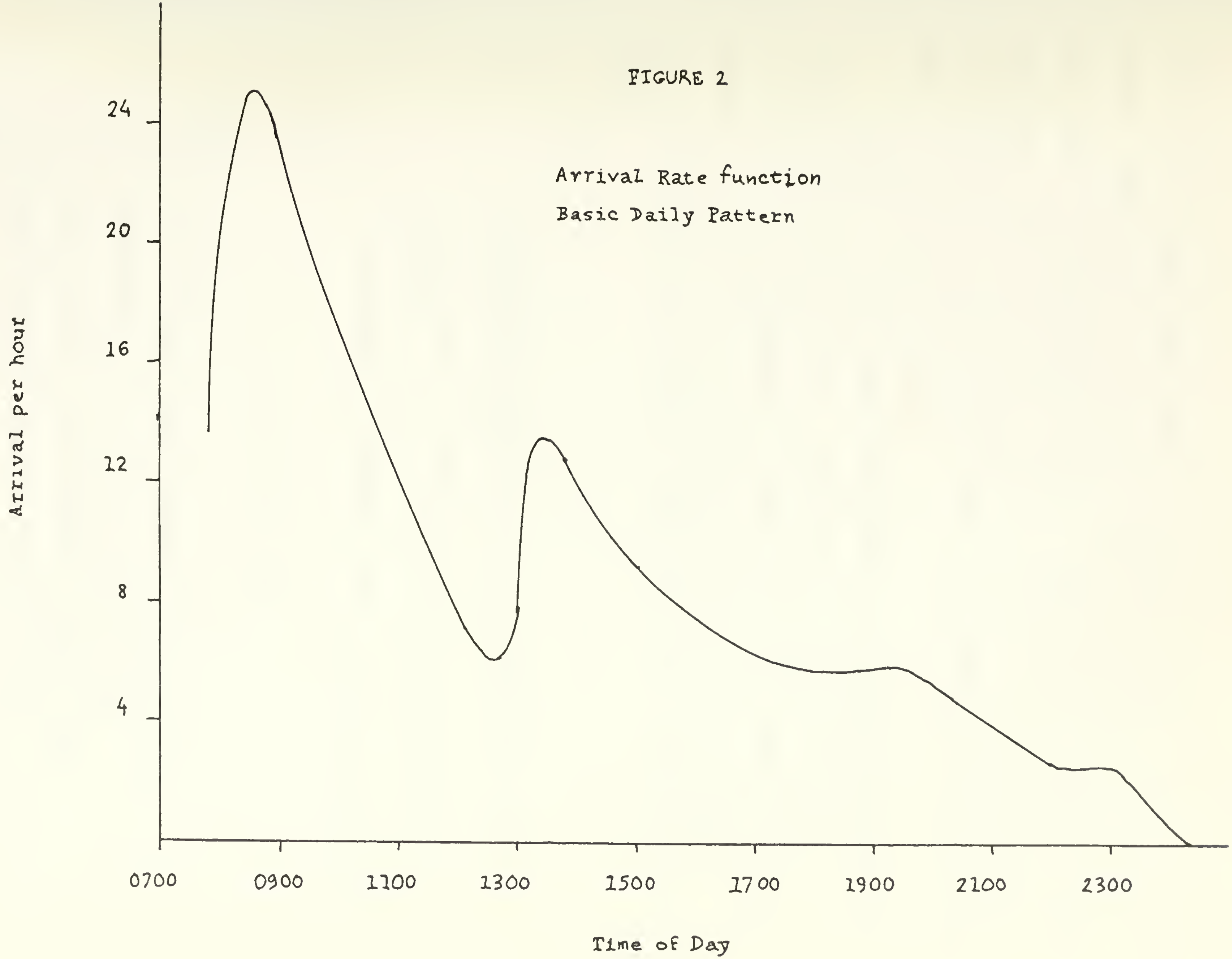
_ _ _ _ indicates path of referral patients which preempt patients in the physician queue.

Figure 1.

FIGURE 2

Arrival Rate function

Basic Daily Pattern



for service, one for referral patients and one for non-referrals. The mean of the distribution for physician referral patients is considerably shorter than that for patients sent directly from the Triage. This is due to the nature of the consultation. Mean service times are based on 75 observations over a three week period.

Thirdly, in the model the Amosist and the patient do not consult with the doctor. Instead, the queues are kept separate and independent by adding the percentage of referral patients directly to the physician queue and the Amosist queue and computing service times as if the consultation took place.

Let:

α = percentage of patients sent directly to the physician queue.

β = percentage of Amosist patients referred to a doctor.

γ = percentage of patients that are seen by a doctor.

Then:

$$\gamma = \alpha + (1 - \alpha)\beta$$

Let:

E(1) = mean Amosist non-referral service time

E(2) = mean Amosist referral service time

E(3) = mean doctor non-referral service time

E(4) = mean doctor referral service time

Then:

$(1 - \alpha)\beta$ = percentage of referral patients

Mean Amosist encounter time = $(1 - \beta)\ E(1) + (\beta)\ E(2)$

Mean doctor encounter time: $\frac{(\alpha)\ E(3)}{\alpha + (1 - \alpha)\beta} + \frac{(1 - \alpha)\beta\ (E(4))}{\alpha + (1 - \alpha)\beta}$

In each case the mean encounter time is the weighted sum of the average service times.

Finally, patients may arrive at the clinic prior to the time that the clinic opens. This was modeled under the assumption that the number of arrivals is distributed poisson, for each of the two queues.

Other parameters for the model include the schedules of the Amosists and the physicians, and the volume [italics added] or total expected number of patients for the particular day under study. For detailed studies of the analytical model, see References [1] and [5].

II. SIMULATION MODEL CASE 1

A. ASSUMPTIONS

This study developed a simulation program to validate and test the significance of the analytical model previously discussed. A comparison of results gives an indication of the correctness of the model's logic and precision by using the same assumptions.

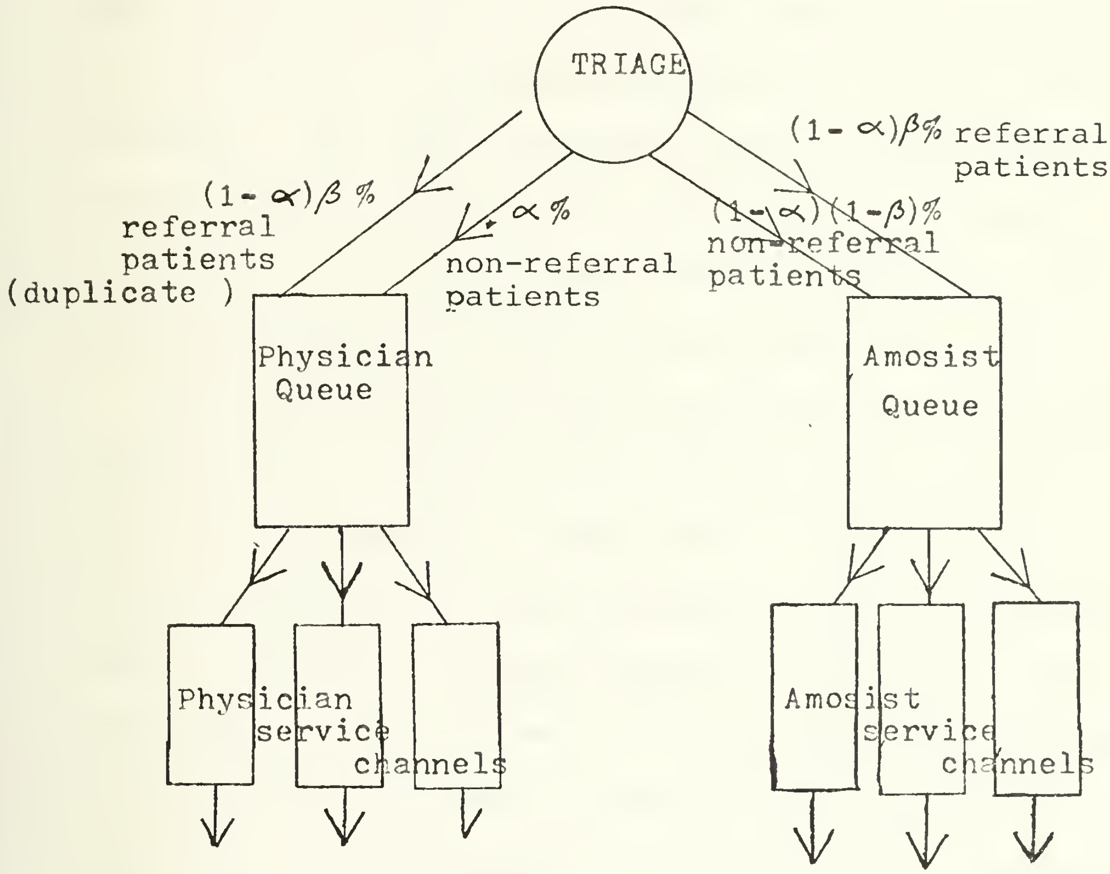
The simulation uses the same parameters as the analytical model. It assumes an inhomogeneous Poisson arrival process with three types of patients. These types are used to distinguish the number of patients arriving at each location. Other than for counting and filing in appropriate queues, each patient is considered the same by a server.

Secondly, patients are served using a mean service rate and an exponential distribution for each queue. Patients are filed in queues upon their arrival and there is no interaction between the queues or the servers. Additional variables are added to the physician queue for each referral-type patient placed in the Amosist queue.

Thirdly, for purposes of the model, patients who arrive prior to the opening of the clinic begin experiencing waiting time when the clinic opens. Figure 3 is a flow diagram which depicts patient flow for both the analytical model and the simulation model.

DIAGRAM OF PATIENT FLOW IN THE

CASE 1 SIMULATION AND ANALYTICAL MODELS



α = percentage of patients directed to a physician.

β = percentage of patients referred by an Amosist.

$(1-\alpha)\beta$ = percentage of total patients referred.

$(1-\alpha)(1-\beta)$ = percentage of patients seen by an Amosist only.

$(1-\alpha)\beta+\alpha$ = percentage of patients seen by a physician.

Figure 3.

B. MODEL IMPLEMENTATION

The model is written in Simscript II.5, a proprietary language supported by Consolidated Analysis Centers, Inc., 12011 San Vicente Blvd., Los Angeles, California, release 8D, for implementation on the IBM 360 batch processing system. It is an event increment program with first in, first out routines. The model generates a daily opening and closing of the clinic, beginning and ending empty. As such, each day is independent of the others. The simulation is based on a particular day of the week for as many iterations as requested.

Each day the model begins by generating two poisson random deviates. These are the number of patients which arrive prior to the opening of the clinic. They are drawn from poisson distributions whose means are input parameters to the model. A binomial random deviate is then drawn from a distribution with the number of trials equal to the number of arrivals for the Amosist queue and the probability of success equal to β, the percentage of referrals by an Amosist. This is the number of Amosist arrivals which are referred to a physician.

Staffing levels change throughout the day as set by input parameters to deal with periods of congestion and to allow the staff to eat meals. Throughout the day arrivals are generated randomly and added to the appropriate queues.

The service times are drawn from exponential distributions with one mean for each of the queues as discussed in the explanation of the analytical model.

Mean Amosist service time = $(1 - \beta)E(1) + (E(2))\beta$

Mean doctor service time: $\frac{(\alpha)E(3)}{\alpha + (1 - \alpha)\beta} + \frac{(1 - \alpha)\,\beta\,(E(4))}{\alpha + (1 - \alpha)\beta}$

C. ARRIVAL PROCESS

The arrival stream is generated using the arrival rate function supplied to the model. Figure 2 is a graph of the function developed from statistics of the Amic. The arrival rates for a particular day are computed using the input parameter, volume [italics added]. The volume [italics added] is the expected number of patients for a particular day. Since part of the daily volume [italics added] arrives prior to the opening of the clinic, the arrival rate function does not account for all arrivals. Therefore, the arrival rate is computed as follows:

Let:

v = average number of arrivals per day.

ave(1) = mean number of arrivals for Amosist queue prior to opening time.

ave(2) = mean number of arrivals for the physician queue prior to opening time.

$\Lambda(t)$ = expected number of arrivals in time [0,t].

or $\Lambda(t) = \int_0^t \lambda(x)dx$

where

$\lambda(t)$ = the arrival rate function shown in Fig. 2.

Scaling the arrival rate function so as to simulate a given average volume of arrivals, v, while keeping the same relative shape of the arrival function, gives

$$\lambda'(t) = \frac{v - ave(1) - ave(2)}{T} \quad x \quad \lambda(t)$$

where T is the finite horizon (length of the day). The scaled arrival rate function $\lambda'(t)$ is then actually used.

The actual times of arrival are determined using the cumulative arrival function. If a set of (x_j) of independent exponentially distributed variables are drawn with unit mean,

Let:

$$S_i = \sum_{j=1}^{i} x_j$$

Then S_i's form a homogeneous poisson process. Using the cumulative arrival function $\Lambda(t)$, a relationship exists between this homogeneous process and the inhomogeneous process desired.

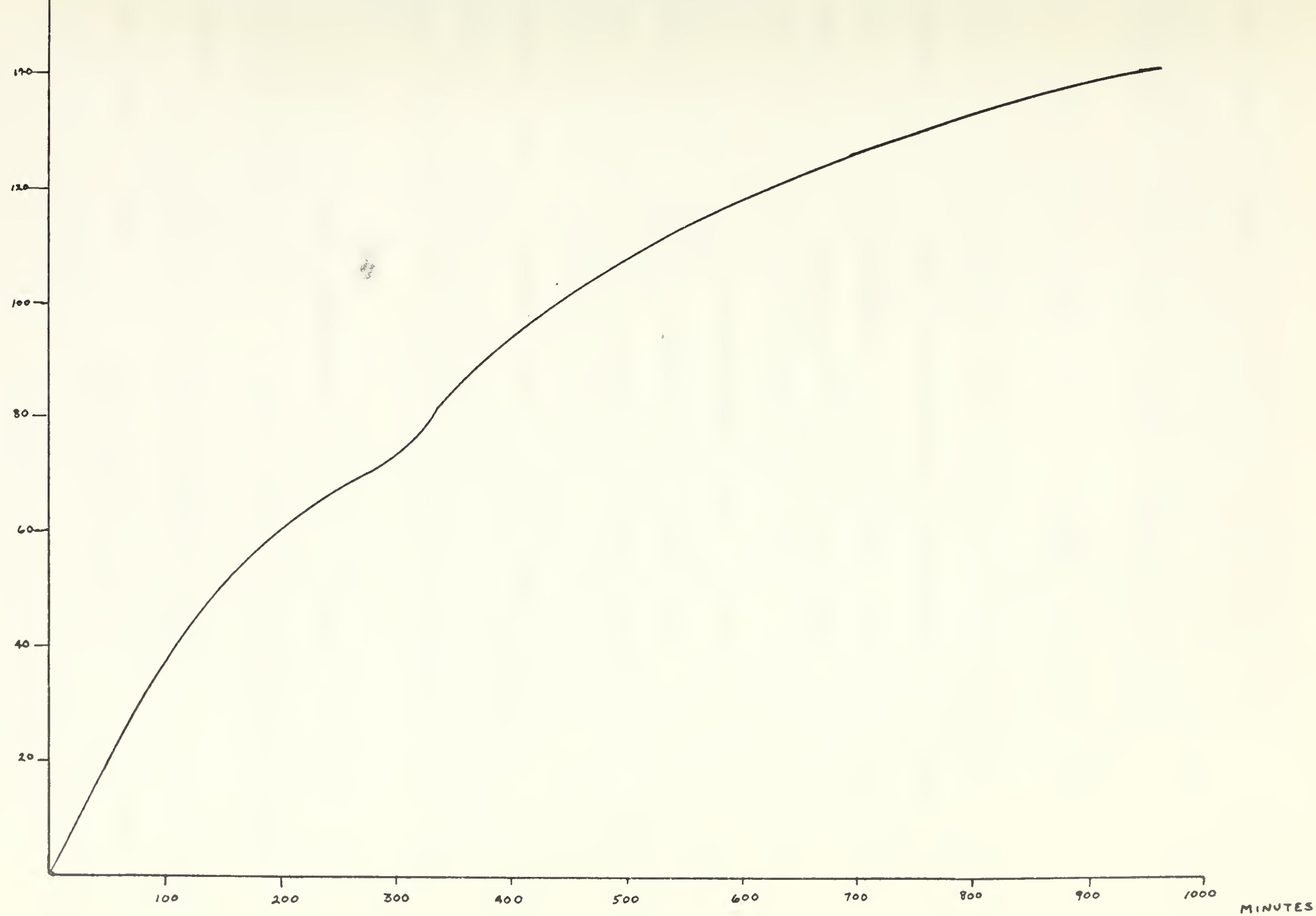
Letting $R_i = \Lambda^{-1}(S_i)$, the $\{R_i\}$ is one realization of the inhomogeneous poisson process and each R_i is the time of an arrival. Figure 4 is a graph of the cumulative arrival function which is computed by the simulation.

D. STATISTICS

Statistics are accumulated for any specific time of the day. Because of the nature of the inhomogeneous process, the

Figure 4

CUMULATIVE ARRIVAL FUNCTION



average for entire days of system statistics, such as system size, do not have a useful meaning. The statistics for each parameter are the mean and the standard deviation.

$$\bar{X} = \sum_{i=1}^{n} X_i / N \qquad S = \sqrt{\left(\sum_{i=1}^{n} X_i^2 / N - \bar{X}^2 \right)}$$

The sum and sum of the squares are collected during the simulation and the means and standard deviations are computed at the end. Virtual delay [italics added] is the actual delay experienced by each patient during specified time periods. The objective is to determine the delay a patient arriving at a specific time would experience. In accomplishing this, patients who arrive five minutes prior until five minutes after a specific time are tagged and their delays are computed and averaged. Another statistic, probability that delay is greater than an input value called "delay criterion," [italics added] is computed by determining the number of patients in the time period that experience delay in excess of that value.

A list of the output statistics follows with definitions:

Average System Size	=	Total number of patients in service or the queue at a specific time.
Average Queue Size	=	Total number of patients in the queue at a specific time.
Utilization Rate	=	1 - number of idle servers/number of servers on duty.
(prob that delay=0)	=	1.0 if a server is available; 0.0 if a server is not available.

Virtual Delay	=	delay experienced by patients at a specific time.
(prob that delay > delay.criterion)	=	1.0 if delay is greater than delay. criterion; 0.0 otherwise.

E. INPUT TO THE MODEL

The first part of the simulation includes a routine to call input parameters. An example of input parameters is appended to this study. Each group of parameters has an option word which defines the parameters to be input. The routine defaults all parameters to Base Day Monday if input is not provided. Option words and their definitions follow.

Option Word	Definition	Parameter
ENCOUNTER	Mean service time	1-dimensional array > = 0 Encounter (1)=Amosist nonreferral mean service time Encounter (2)=Amosist referral mean service time Encounter (3)=doctor nonreferral mean service time Encounter (4)=doctor referral mean service time Default values: Encounter (1) = 17.96 (2) = 23.83 (3) = 13.55 (4) = 4.34
VOLUME. TOTAL	Average number of patients per day	Integer > 0 Default value: Volume = 156
OPENING. TIME	24-hour clock time that clinic opens	real > 0 Default value: Opening.time = 0800.

CLOSING. TIME	24-hour clock time that clinic closes	real > 0 Default value: Closing.time = 2400
PERCENTAGE	Percentage of patients tri- aged directly to a doctor	real > 0 Default value: alpha = .55
REFERRALS	Percentage of patients re- ferred by an Amosist	real ≥ 0 Default value: Beta = .31
ITERATIONS	Number of days of simulation	integer > 0 Default value: Last.day = 1
PRELOADING	Average number of patients arriving prior to opening	1-dimensional array Ave(1) = number of Amosist arrivals Ave(2) = number of doctor arrivals Default Values: Ave.arrival(1) = 7. Ave.arrival(2) = 4.
AMOSIST.SCH	Schedule for Amosist staff	integer > 0 No.duty.chgs = number of staff changes Hour(i) = time shift starts No.amosists(i) = no of staff for that shift Hour(1) must = opening.time Default values: No.duty.chgs = 5 Hour(1) = 0800 No.amosists(1)=5 (2) = 1100 (2)=3 (3) = 1300 (3)=5 (4) = 1530 (4)=7 (5) = 1630 (5)=2
PHYSICIANS	Schedule for Physician staff	integer > 0 No.rosterchgs = number of staff changes Clock(i) = time shift starts No.doctors(i) = number of staff for that shift Clock(1) must = opening.time Default values: Clock(1)=0800 No.doctors(1)=2 (2)=1100 (2)=1

		(3)=1200 (3)=2 (4)=1530 (4)=3 (5)=1630 (5)=1
ARRIVAL.FCN	Arrival function	integer > 0 pairs = no of periods defined 2,1-dimensional arrays Time(i) = end of period Customer(i) = rate of arrival alternate time and customer Default values: Pairs = 32 time(1) = 810 Customer(1)=22.5 (2) = 820 (2)=23.5 (3) = 840 (3)=24.5 (4) = 850 (4)=23.5 (5) = 900 (5)=22.5 (6) = 910 (6)=21.5 (7) = 920 (7)=20.5 (8) = 930 (8)=19.6 (9) = 945 (9)=18.6 (10) =1000 (10)=17.6 (11) =1015 (11)=16.6 (12) =1030 (12)=14.7 (13) =1045 (13)=13.35 (14) =1100 (14)=12.05 (15) =1115 (15)=10.95 (16) =1130 (16)=9.9 (17) =1145 (17)=8.9 (18) =1200 (18)=7.9 (19) =1215 (19)=7.0 (20) =1245 (20)=6.2 (21) =1300 (21)=7.2 (22) =1330 (22)=13.6 (23) =1400 (23)=12.5 (24) =1430 (24)=11.1 (25) =1500 (25)=9.85 (26) =1530 (26)=8.8 (27) =1630 (27)=7.55 (28) =1730 (28)=6.3 (29) =1930 (29)=5.8 (30) =2130 (30)=4.4 (31) =2330 (31)=2.8 (32) =2400 (32)=.01
DELAY.VALUE	Time in minutes to compute prob delay is greater than delay criterion	real > 0 Default value: Delay.criterion = 15.0

END.OF.DATA	End of input.	Nothing should follow this card.

The input routine is designed to be as user oriented as possible. All input is in free format with a blank space being the deliminater for each input word. The input option cards and their associated values may be read in whatever order desired as long as the end.of.data card is last.

F. VERIFICATION OF THE PROGRAM

In the early stages of formulation extensive output was used to follow, exactly, the patient flow through the model. In its final form the model gives a daily output of the number of patients treated in each system by type, the number of arrivals prior to opening time for each queue, and the time the clinic closed on that day. Provision has been made, if the congestion continues throughout the day, to count the number of patients that are waiting for service when the clinic closes. In the actual clinic these people would be sent to the emergency room for treatment.

The number of referred patients served in both queues was always equal and the sum of the referral patients and the non-referral patients from both queues always equalled the total number of arrivals for the day.

The means of the poisson deviates and the binomial deviate from the early arrivals were determined. For patients seen only by a doctor the mean was 7.2 with an input mean of 7 for a 200 day simulation. For the amosist queue the mean

was 3.92 with a standard deviation of 1.95 and an input parameter of 4. The expected number of referral patients was 2.17 and the actual average was 2.24 with a standard deviation of 1.47. There is no significant difference in the means.

The total number of arrivals for each queue was also summed by type over the simulation to compare to the expected values, to insure the correct percentage of patients was directed to each queue.

Ave. num. of doctor's patients = 84.16; expected num = 85.8

Ave. num. of amosist's patients = 48.94; expected num = 48.44

Ave. num of referral patients = 22.36; expected num = 21.76

A test of hypothesis was made on the arrival distribution to determine if the number of arrivals for each period was from the same distribution. The statistic is Q and is distributed $\chi^2(n-1)$

$$A = \sum_{i=1}^{32} \frac{(x_i - nr_i)^2}{nr_i}$$

where r_i is the mean number of arrivals in the ith period and x_i is the observed number of arrivals over all iterations. At the .025 level of significance the χ^2 is 47.0 with 31 degrees of freedom. The test statistic was 37.64, and the hypothesis was accepted.

The mean service times were computed using a language generated routine, and compared to the expected values. Since

the program uses internal exponential generating routines, it was only necessary to check the means.

Amosist mean service time = 19.583; expected value 19.78

Doctor mean service time = 11.559; expected value 11.687

The results of the model are tabulated in a later section of this study. Source listings and flow charts of this program are appended to the study.

III. CASE 2 SIMULATION MODEL

A. ASSUMPTIONS AND CHANGES TO THE MODEL

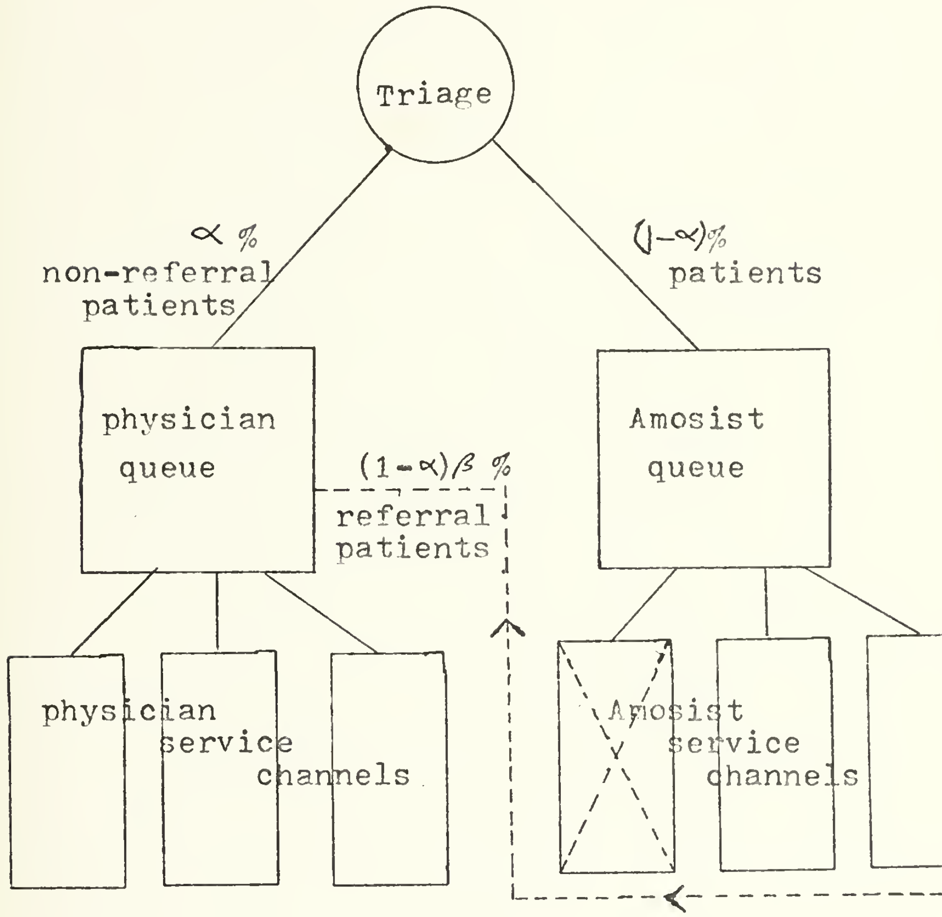
After developing a model of the system using the assumptions of the analytical model, the next step in the study was to produce a model whose assumptions more closely duplicated those of the clinic. The change required was in the handling of referral patients. In this model as in the actual system, all patients going to the Amosist have one distribution of service times. In the previous model, the extra time for a referral patient was time used waiting to see a physician and consulting with him. The Amosist is held out of service in this model until the consultation is complete and then the Amosist can continue treating other patients. In the physician queue, referral patients preempt non-referral patients and have a separate distribution of service times from the patients who are sent directly from the Triage to a doctor. All other assumptions of the model remain the same as the Case 1 model. Figure 5 is a flow diagram for patients in this model.

B. VERIFICATION

The input and arrival process for this model was unchanged and, therefore, was not retested. The service times averaged 18.119 minutes for the Amosist queue and 11.419 minutes for the physician queue. The expected values were 17.96 minutes

DIAGRAM OF PATIENT FLOW IN THE

CASE 2 SIMULATION MODEL



_ _ _ _ indicates path of referral patients and the Amosists treating referral patients. _ _ _ also indicate blocked Amosist service channels during a consoltation.

Figure 5.

for the Amosist queue and 11.687 minutes for the physician queue. A source listing for this model is appended to the study.

IV. SIMULATION RESULTS

A. COMPARISON OF ANALYTICAL AND CASE 1 MODELS

In order to test the results of the two simulations against the analytical model of the AMIC, the means and standard deviations of each of the output parameters were tabulated. The results of the analytical model are found in Table I. Table II is the results of the Case 1 simulation model.

Since the simulations and the anlytical model have separate distributions and alternate methods of obtaining values which are independent, a test can be made that the average values are now significantly different.

Ho: $\bar{X} = A$ where $\bar{X}$ = simulation statistic
A = analytical deterministic value

For purposes of the test the output parameter of the analytical model is assumed to be the true parameter of the system.

Let:

N = number of samples in the distribution (2))

$\bar{X}$ = mean value of simulation parameter

A = mean value of analytical model

$\hat{s}$ = standard deviation of X

T = test statistic

Then:

$$T = \frac{(\bar{X} - A)}{\hat{s}} \times \sqrt{N}$$

TABLE I. RESULTS OF THE ANALYTICAL MODEL
AMOSIST SYSTEM

AV SYS	SD	AV QUEUE	SD	UTIL	P(D=0)	P(D>D0)	AV DELAY	TIME
6.4348	3.0425	2.0108	2.4220	0.8848	0.2971	0.2932	10.7354	810.
6.0708	3.2597	1.8471	2.5147	0.8447	0.3707	0.2644	9.7967	820.
5.7123	3.4731	1.6953	2.6422	0.8034	0.4427	0.2349	8.9111	840.
5.5268	3.5082	1.6008	2.6462	0.7852	0.4734	0.2195	8.4158	850.
5.3121	3.5034	1.4844	2.6100	0.7655	0.5058	0.2021	7.8273	900.
5.0722	3.4633	1.3517	2.5387	0.7441	0.5401	0.1832	7.1667	910.
4.8120	3.3917	1.2083	2.4366	0.7207	0.5762	0.1635	6.4563	920.
4.5430	3.2936	1.0625	2.3108	0.6961	0.6130	0.1438	5.7341	930.
4.1430	3.1089	0.8528	2.0917	0.6580	0.6672	0.1158	4.6901	945.
3.7704	2.8972	0.6672	1.8554	0.6206	0.7174	0.0911	3.7575	1000.
3.4241	2.6720	0.5078	1.6126	0.5833	0.7640	0.0698	2.9427	1015.
3.0247	2.4197	0.3566	1.3428	0.5336	0.8170	0.0494	2.1348	1030.
2.6571	2.1632	0.2368	1.0814	0.4841	0.8637	0.0332	1.4759	1045.
2.3252	1.9187	0.1482	0.8413	0.4354	0.9028	0.0210	0.9707	1100.
2.2170	1.9553	0.4442	1.2613	0.5910	0.6631	0.1237	5.1495	1115.
2.0716	1.9190	0.4014	1.2020	0.5567	0.6976	0.1117	4.6405	1130.
1.8923	1.8354	0.3370	1.1046	0.5184	0.7371	0.0949	3.9553	1145.
1.6898	1.7176	0.2648	0.9800	0.4750	0.7802	0.0761	3.1955	1200.
1.4843	1.5791	0.1966	0.8423	0.4292	0.8226	0.0582	2.4659	1215.
1.1746	1.3163	0.1022	0.5923	0.3575	0.8836	0.0331	1.4409	1245.
1.1913	1.2719	0.0907	0.5330	0.3669	0.8809	0.0315	1.3834	1300.
1.9111	1.4351	0.0301	0.2780	0.3762	0.9517	0.0046	0.3100	1330.
1.9415	1.4424	0.0314	0.2687	0.3820	0.9490	0.0049	0.3258	1400.
1.7750	1.3713	0.0212	0.2153	0.3508	0.9621	0.0033	0.2340	1430.
1.5812	1.2828	0.0120	0.1562	0.3138	0.9748	0.0019	0.1472	1500.
1.4078	1.2015	0.0065	0.1103	0.2803	0.9836	0.0011	0.0906	1530.
1.1712	1.0829	0.0000	0.0080	0.1673	0.9990	0.0000	0.0030	1630.
0.9372	1.2899	0.2044	0.6882	0.3664	0.7609	0.1066	4.3857	1730.
0.8235	1.2156	0.1646	0.6365	0.3295	0.7960	0.0878	3.6457	1930.
0.5345	0.9102	0.0656	0.3733	0.2344	0.8795	0.0447	1.8407	2130.
0.2846	0.6034	0.0138	0.1498	0.1354	0.9509	0.0148	0.6227	2330.
0.0275	0.2063	0.0014	0.0488	0.0131	0.9944	0.0014	0.0688	2400.

TABLE I (CONTINUED)
PHYSICAL SYSTEM

AV SYS	SD	AV QUEUE	SD	UTIL	P(D=0)	P(D>D0)	AV DELAY	TIME
3.9986	2.3964	2.5175	2.1769	0.9284	0.1032	0.6044	23.6817	810.
4.9032	2.8787	3.3987	2.6943	0.9430	0.0815	0.6966	30.2639	820.
6.8646	3.7050	5.3169	3.5846	0.9701	0.0429	0.8265	44.5413	840.
7.7387	4.0538	6.1812	3.9502	0.9762	0.0339	0.8620	50.9250	850.
8.5144	4.3666	6.9511	4.2739	0.9799	0.0285	0.8852	56.5963	900.
9.1930	4.6480	7.6262	4.5624	0.9821	0.0252	0.9006	61.5608	910.
9.7748	4.9006	8.2062	4.8194	0.9833	0.0234	0.9106	65.8200	920.
10.2694	5.1272	8.7000	5.0485	0.9838	0.0225	0.9169	69.4423	930.
10.8652	5.4233	9.2959	5.3458	0.9837	0.0224	0.9220	73.8083	945.
11.3147	5.6738	9.7464	5.5960	0.9831	0.0231	0.9236	77.1040	1000.
11.6189	5.8854	10.0529	5.8047	0.9816	0.0249	0.9222	79.3389	1015.
11.6575	6.0566	10.0978	5.9667	0.9776	0.0299	0.9145	79.6385	1030.
11.5139	6.1969	9.9634	6.0939	0.9719	0.0373	0.9024	78.6111	1045.
11.2006	6.3073	9.6636	6.1862	0.9634	0.0479	0.8851	76.3527	1100.
11.7187	6.3950	10.9336	6.3602	0.9842	0.0154	0.9566	171.6929	1115.
12.0849	6.4685	11.2982	6.4364	0.9863	0.0133	0.9618	177.0582	1130.
12.3081	6.5310	11.5212	6.4991	0.9866	0.0130	0.9633	180.3281	1145.
12.3921	6.5851	11.6060	6.5518	0.9854	0.0142	0.9621	181.5586	1200.
11.3735	6.6293	9.8542	6.4841	0.9524	0.0598	0.8726	77.6799	1215.
9.2728	6.5369	7.8519	6.2839	0.8906	0.1331	0.7757	62.5838	1245.
8.4380	6.4291	7.0584	6.1440	0.8647	0.1660	0.7296	56.5794	1300.
8.5012	6.4480	7.0634	6.2349	0.9012	0.1301	0.7361	56.8251	1330.
8.2294	6.4651	6.8064	6.2418	0.8919	0.1412	0.7195	54.8783	1400.
7.6276	6.4000	6.2477	6.1434	0.8650	0.1738	0.6792	50.5943	1430.
6.7913	6.2248	5.4772	5.9238	0.8237	0.2234	0.6204	44.6607	1500.
5.8239	5.9255	4.5965	5.5770	0.7694	0.2878	0.5482	37.8327	1530.
2.5006	3.6043	1.2585	2.9952	0.5191	0.6806	0.1661	7.3903	1630.
2.9577	3.6131	2.3255	3.4717	0.7925	0.2066	0.5899	43.3424	1730.
3.0837	3.5937	2.4476	3.4484	0.7974	0.2017	0.6084	45.1892	1930.
2.1988	3.1019	1.6527	2.9228	0.6847	0.3143	0.4741	32.2263	2130.
1.0336	2.0815	0.6626	1.8799	0.4651	0.5339	0.2609	15.1556	2330.
0.4668	1.6319	0.3398	1.4480	0.1593	0.8397	0.1034	6.8520	2400.

TABLE II. RESULTS OF THE CASE 1 SIMULATION MODEL

AMOSIST SYSTEM

SYS AV	S.D.	AV QUE	S.D.	UTIL	SD	P(D=0)	SD	P(D> *)	AV DELAY	S.D.	NUM	TIME
6.475	3.000	2.030	2.349	.889	.222	.265	.441	.306	10.674	11.566	360	810
6.140	3.276	1.890	2.461	.850	.271	.305	.460	.271	10.787	12.876	373	820
5.800	3.575	1.765	2.720	.807	.281	.400	.490	.247	9.233	12.566	385	840
5.630	3.215	1.565	2.363	.813	.270	.420	.494	.183	6.888	10.472	345	850
5.385	3.289	1.480	2.341	.781	.290	.450	.497	.269	9.554	12.635	312	900
4.960	3.438	1.350	2.391	.722	.306	.565	.496	.192	7.034	12.484	323	910
4.725	3.226	1.165	2.193	.712	.300	.600	.490	.144	6.008	11.173	305	920
4.435	3.094	1.020	1.970	.683	.314	.615	.487	.147	5.594	10.587	292	930
4.300	2.956	0.890	1.868	.682	.311	.615	.487	.115	4.763	8.795	287	945
3.765	2.674	0.595	1.546	.634	.321	.690	.462	.066	2.709	6.395	289	1000
3.390	2.406	0.420	1.294	.594	.310	.755	.430	.058	2.157	6.069	240	1015
2.885	2.126	0.240	1.069	.529	.300	.830	.376	.040	1.455	4.759	224	1030
2.395	1.941	0.150	0.733	.449	.310	.855	.352	.039	1.868	6.598	204	1045
1.385	1.287	0.110	0.646	.425	.309	.885	.319	.021	0.599	3.345	194	1100
1.650	1.363	0.160	0.543	.497	.346	.785	.411	.079	2.486	6.751	165	1115
1.675	1.356	0.150	0.527	.508	.353	.755	.430	.103	3.259	8.227	145	1130
1.560	1.388	0.165	0.564	.465	.346	.810	.392	.042	2.242	7.640	118	1145
1.405	1.379	0.125	0.583	.427	.355	.815	.388	.018	1.192	4.419	111	1200
1.350	1.314	0.110	0.546	.413	.340	.860	.347	.018	1.248	5.785	109	1215
1.205	1.189	0.085	0.410	.373	.321	.885	.319	.0	0.773	2.331	106	1245
1.220	1.171	0.0	0.0	.244	.234	.985	.122	.006	0.252	1.460	179	1300
1.830	1.487	0.050	0.328	.356	.267	.960	.196	.0	0.384	1.619	214	1330
1.940	1.529	0.045	0.251	.379	.282	.945	.228	.010	0.414	2.141	203	1400
1.625	1.302	0.010	0.099	.323	.254	.975	.156	.0	0.043	0.369	147	1430
1.485	1.136	0.0	0.0	.297	.227	.995	.071	.0	0.008	0.093	143	1500
1.410	1.242	0.0	0.0	.201	.177	1.0	.0	.0	0.0	0.0	109	1530
0.305	0.522	0.0	0.0	.152	.261	.970	.171	.008	0.324	1.950	119	1630
1.130	1.159	0.175	0.552	.477	.410	.685	.465	.099	4.247	10.743	81	1730
1.145	1.243	0.215	0.631	.465	.414	.690	.462	.123	5.186	10.818	81	1930
0.785	0.979	0.095	0.431	.345	.362	.845	.362	.078	3.628	10.686	51	2130
0.470	0.699	0.015	0.122	.227	.327	.910	.286	.042	1.305	4.576	24	2330
0.105	0.322	0.0	0.0	.052	.161	.995	.071	.0	0.0	0.0	1	2400

TABLE II.(CONT.) RESULTS OF THE CASE 1 SIMULATION MODEL

PHYSICIAN SYSTEM

SYS AV	S.D.	AV QJE	S.D.	UTIL	SD	P(D=0)	SD	P(D> *)	AV DELAY	S.D.	NUM	TIME
7.035	3.055	5.055	3.017	.990	.086	.015	.122	.776	34.876	24.666	531	810
8.250	3.542	6.275	3.494	.987	.078	.025	.156	.879	42.635	25.442	587	820
10.790	4.726	8.810	4.687	.990	.070	.020	.140	.919	59.923	35.026	583	840
12.105	5.103	10.110	5.093	.997	.035	.005	.071	.923	64.496	38.271	558	850
13.220	5.606	11.225	5.596	.997	.035	.005	.071	.951	74.815	43.886	530	900
14.110	5.980	12.125	5.948	.992	.061	.015	.122	.947	80.913	48.607	514	910
15.070	6.223	13.080	6.200	.995	.071	.005	.071	.971	84.624	49.464	512	920
15.765	6.362	13.770	6.350	.997	.035	.005	.071	.959	90.992	54.172	489	930
16.390	6.784	14.405	6.751	.992	.061	.015	.122	.970	101.786	58.290	397	945
16.975	7.312	14.980	7.301	.997	.035	.005	.071	.959	109.074	56.529	417	1000
17.255	7.512	15.275	7.469	.990	.099	.010	.099	.954	112.892	60.021	388	1015
17.575	7.953	15.615	7.869	.980	.131	.025	.156	.958	128.732	60.955	313	1030
17.460	8.317	15.485	8.268	.987	.093	.020	.140	.912	116.890	62.645	341	1045
16.425	8.873	15.445	8.837	.980	.140	.020	.140	.975	124.088	62.026	281	1100
17.035	8.940	16.050	8.912	.985	.122	.015	.122	.975	113.897	52.319	236	1115
17.630	9.275	16.645	9.248	.985	.122	.015	.122	.963	116.197	60.593	218	1130
17.810	9.448	16.835	9.402	.975	.156	.025	.156	.969	105.558	56.395	195	1145
18.070	9.490	16.110	9.418	.980	.131	.025	.156	.873	92.182	59.890	181	1200
16.795	9.461	14.875	9.328	.960	.169	.060	.237	.922	100.663	60.761	141	1215
14.205	9.582	12.330	9.410	.937	.218	.085	.279	.778	70.914	56.287	153	1245
13.270	9.564	11.440	9.347	.915	.235	.130	.336	.806	70.153	55.002	273	1300
13.140	9.925	11.270	9.767	.935	.220	.090	.286	.766	68.846	59.930	325	1330
12.825	10.284	10.980	10.106	.922	.230	.115	.319	.783	70.794	67.142	267	1400
12.050	10.244	10.205	10.079	.922	.219	.125	.331	.729	60.159	63.296	236	1430
10.855	10.127	9.065	9.924	.895	.253	.165	.371	.656	51.055	72.559	215	1500
9.515	9.522	7.160	8.966	.785	.340	.325	.468	.567	48.385	78.197	187	1530
3.160	5.919	2.630	5.684	.530	.499	.470	.499	.351	41.522	81.430	168	1630
3.810	5.334	3.070	5.164	.740	.439	.260	.439	.547	42.209	61.090	137	1730
3.960	5.130	3.210	4.952	.750	.433	.250	.433	.603	37.076	44.066	136	1930
2.805	4.259	2.155	4.050	.650	.477	.350	.477	.467	28.228	42.653	75	2130
1.335	2.734	0.900	2.492	.435	.496	.565	.496	.345	11.206	12.787	29	2330
0.615	2.144	0.470	1.915	.145	.352	.855	.352	.0	0.0	0.0	0	2400

Since there are 200 iterations in the simulation, the T statistic approaches its limiting distribution which is normal (0,1). Tables III and IV are the summary of this comparison. All "Undef" entries indicate a standard deviation of 0.0 indicating that all values of X were equal at that time.

The two models were compared for the parameters of base day Monday. A critical region for accepting or rejecting the hypothesis was not chosen since entries are not independent of time. Although each day's statistics are independent, the statistics taken at different times of the day are serially correlated, and cannot be pooled. Therefore, subjective judgment based on the normal statistics computed was used to judge the fit of the two models.

Using the above criteria, the analytical model is a very good fit in the Amosist system. However, when congestion overloads the queues, the analytical model does not conserve all of the probability used in computing the output parameters. As a result, the analytical model underestimates the number of patients in the system. Because the physician system in this particular set of parameters was highly congested, the physician system of the analytical model indicated fewer patients being serviced. The values of both models were very close after the congestion diminished.

TABLE III. COMPARISON OF THE ANALYTICAL AND SIMULATION MODELS

NORMAL STATISTICS FOR THE CASE 1 AMOSIST SYSTEM

System size	Queue size	Utilization	Prob(Del=0)	Average Delay	Time
-.953	-.708	-1.080	1.107	.110	810
-2.295	-1.788	-2.287	.988	-1.970	820
-2.202	-1.039	-3.056	2.045	-.910	840
-1.653	-1.050	-2.098	1.318	-3.166	850
-1.320	-1.162	-1.108	.828	-.378	900
-1.646	-1.151	-1.836	1.288	-1.208	910
.997	-.527	-1.405	2.031	.920	920
-.612	-.202	-1.018	1.097	.222	930
.300	.427	.044	-.066	.699	945
.839	.737	.691	-1.096	.003	1000
.490	.448	.376	-.773	.356	1015
-.184	.088	-.339	-.429	-1.582	1030
1.167	-.324	1.750	-1.609	1.406	1045
-7.575	.033	1.37	-2.805	2.049	1100
-5.112	-6.367	-3.096	3.786	-2.830	1115
-4.913	-6.013	-3.066	2.067	-5.443	1130
-5.167	-5.375	-3.740	4.472	-7.619	1145
-3.651	-5.957	-2.020	1.873	-5.100	1200
-1.639	-4.455	-.259	.670	-7.166	1215
-2.019	-1.421	-1.801	1.033	-.700	1245
-1.384	Undef	-9.400	15.585	-9.096	1300
-.645	.274	-.747	.225	-.075	1339
.425	-.429	.512	-.537	-.655	1400
-.816	-3.230	-.649	-.152	.443	1430
-1.251	Undef	-1.100	Undef	Undef	1500
.583	Undef	-6.273	Undef	Undef	1530
-24.392	Undef	-1.130	-2.176	1.148	1630
.920	-2.340	2.746	.137	.272	1730
2.205	-.973	3.701	-1.034	.963	1930
3.168	.993	3.710	-1.696	-.186	2130
2.910	-.381	3.185	-1.159	.457	2330
3.417	Undef	3.479	Undef	Undef	2400

TABLE IV. COMPARISON OF THE ANALYTICAL AND SIMULATION MODELS

NORMAL STATISTICS FOR THE CASE 1 PHYSICIAN SYSTEM

System size	Queue size	Utilization	Prob(Del=0)	Average Delay	Time
13.708	12.006	5.069	- 10.201	7.295	810
12.263	10.527	11.360	- 5.122	5.930	820
10.574	9.541	1.273	- 2.313	6.273	840
10.673	9.672	1.455	- 5.756	4.549	850
10.098	11.041	.944	- 4.681	4.751	900
9.872	10.497	2.295	- 1.182	5.259	910
9.669	8.923	.243	- 3.665	5.861	920
10.225	9.474	.487	- 3.486	4.870	930
9.744	8.999	1.035	- .858	5.688	945
9.877	9.079	Undef	- 3.605	6.116	1000
9.852	9.156	1.200	- 2.128	7.019	1015
9.681	8.955	3.309	- 1.444	9.278	1030
9.432	8.699	2.251	- 1.748	7.861	1045
7.198	8.113	.886	- 2.818	9.489	1100
7.486	7.202	- .424	- .046	-15.471	1115
7.495	7.189	.529	.197	-16.362	1130
7.522	7.597	- .667	1.088	-19.723	1145
7.491	5.794	- 1.325	.979	-19.315	1200
7.155	6.672	.220	.011	3.082	1215
6.624	6.013	3.512	- 2.438	- .137	1245
6.078	5.530	- 2.474	- 1.515	2.926	1300
6.030	5.449	2.124	- 1.962	1.078	1330
6.343	5.715	3.503	- 1.162	.019	1400
5.565	4.972	2.730	- 2.085	.757	1430
5.598	4.903	4.232	- 2.226	- .149	1500
4.522	2.835	.523	1.124	- .487	1530
.096	2.227	- .257	- 5.969	3.942	1630
1.062	2.326	- 1.844	1.720	- 2.343	1730
2.245	.788	.271	1.578	- .417	1930
.727	.557	- 2.245	1.058	- 2.079	2130
1.349	.909	.139	.887	- 5.231	2330
.905	.874	- .575	.615	Undef	2400

B. COMPARISON OF THE CASE 2 SIMULATION AND ANALYTICAL MODELS

The same procedure outlined for Case 1 was used in comparing the results of the Case 2 model. Table V is the results of the Case 2 model and Tables VI and VII are the summary of the comparison tests for the models.

In this case other factors caused differences between the two models. In the Case 2 model, Amosists spend more time than expected waiting for consultations. Therefore, the two models have different distributions for referral service. The result was that the Amosist queue was longer than the original model and the analytical model results were not as close as previously seen. In the other system, the physician queue was smaller than in the Case 1 model. This was the result of the phenomenon of shortest processing time. Since all of the patients with the shorter processing distribution were served first, the size of the queues are expected to be shorter. The average time in the system for a patient does not change, but because of the preemption fewer people are waiting for service at any one time.

C. COMPARISON OF CASE 1 AND CASE 2 MODELS

As discussed in the previous section, the average service time for an Amosist referral patient was longer than expected in the Case 1 model or the analytical model. The physician system had shorter queues and fewer patients in the system at any one time due to the fact that the distribution of service times for referral patients was much shorter than the

TABLE V. RESULTS OF THE CASE 2 SIMULATION MODEL

AMOSIST SYSTEM

SYS AV	S.D.	AV QUE	S.D.	UTIL	SD	P(D=0)	SD	P(D> *)	AV DELAY	S.D.	NUM	TIME
6.935	2.995	2.360	2.496	.915	.171	.250	.433	.337	12.770	14.256	400	810
6.780	3.291	2.330	2.693	.890	.202	.285	.451	.371	12.939	14.160	350	820
6.635	3.664	2.325	2.983	.862	.229	.330	.470	.317	12.654	15.025	344	840
6.210	3.563	2.065	2.764	.829	.247	.385	.487	.315	11.152	13.933	314	850
6.035	3.628	1.960	2.818	.815	.253	.440	.496	.298	10.772	14.589	322	900
5.895	3.602	1.865	2.762	.806	.263	.445	.497	.250	8.758	13.624	340	910
5.400	3.367	1.500	2.530	.780	.256	.520	.500	.155	6.373	11.728	316	920
5.145	3.295	1.345	2.412	.760	.269	.545	.498	.171	6.419	11.297	315	930
4.640	3.180	1.070	2.206	.714	.296	.575	.494	.165	5.681	10.107	272	945
4.275	2.960	0.830	1.913	.689	.317	.600	.490	.160	6.398	11.609	262	1000
4.085	3.092	0.840	1.984	.649	.327	.645	.479	.187	8.511	18.566	251	1015
3.810	3.040	0.705	2.009	.621	.318	.710	.454	.074	5.830	18.867	229	1030
3.350	2.817	0.540	1.752	.562	.316	.790	.407	.070	4.267	14.395	186	1045
1.890	2.149	0.405	1.514	.495	.350	.770	.421	.092	4.478	13.930	173	1100
2.130	2.392	0.555	1.722	.525	.361	.740	.439	.151	5.808	12.898	159	1115
2.170	2.114	0.470	1.449	.567	.357	.680	.466	.154	4.678	9.049	136	1130
2.110	2.009	0.445	1.252	.555	.378	.675	.468	.122	4.871	11.177	131	1145
2.085	2.088	0.480	1.338	.535	.370	.710	.454	.094	4.585	11.591	128	1200
1.625	1.782	0.275	1.081	.450	.349	.815	.388	.030	1.685	6.012	99	1215
1.330	1.422	0.135	0.690	.398	.343	.855	.352	.028	1.230	4.526	106	1245
1.170	1.269	0.030	0.359	.228	.224	.980	.140	.0	0.111	0.654	158	1300
2.210	1.596	0.045	0.230	.433	.299	.895	.307	.005	0.549	2.157	192	1330
1.995	1.427	0.025	0.211	.394	.271	.955	.207	.0	0.224	1.088	170	1400
2.075	1.469	0.030	0.198	.409	.279	.940	.237	.0	0.247	1.474	198	1430
1.715	1.461	0.025	0.156	.338	.279	.940	.237	.0	0.241	1.442	135	1500
1.575	1.317	0.0	0.0	.225	.188	1.0	.0	.0	0.044	0.422	121	1530
0.325	0.529	0.0	0.0	.162	.264	.970	.171	.0	0.003	0.029	112	1630
1.090	1.201	0.205	0.559	.442	.410	.715	.451	.132	5.190	10.856	106	1730
1.200	1.367	0.245	0.851	.477	.398	.705	.456	.111	5.448	15.178	81	1930
0.905	1.156	0.150	0.606	.377	.386	.795	.404	.074	3.373	9.602	54	2130
0.555	0.792	0.045	0.207	.255	.335	.900	.300	.045	0.794	3.409	22	2330
0.145	0.392	0.005	0.071	.070	.181	.995	.071	.0	0.0	0.0	0	2400

TABLE V.(CONT.) RESULTS OF THE CASE 2 SIMULATION MODEL

PHYSICIAN SYSTEM

SYS AV	S.D.	AV QUE	S.D.	UTIL	SD	P(D=0)	SD	P(D> *)	AV DELAY	S.D.	NUM	TIME
5.360	2.885	3.455	2.749	.952	.163	.085	.279	.712	33.564	28.400	455	810
6.425	3.521	4.490	3.425	.967	.142	.055	.228	.767	40.904	32.498	451	820
8.610	4.505	6.655	4.431	.977	.125	.035	.184	.836	55.190	40.118	483	840
9.505	4.855	7.530	4.811	.987	.105	.015	.122	.883	65.284	44.890	428	850
10.630	5.212	8.655	5.167	.987	.093	.020	.140	.886	71.381	50.915	446	900
11.390	5.539	9.395	5.530	.997	.035	.005	.071	.918	76.379	52.582	379	910
12.065	5.683	10.070	5.674	.997	.035	.005	.071	.927	85.870	54.675	384	920
12.525	6.178	10.540	6.151	.992	.061	.015	.122	.929	90.601	60.654	350	930
13.040	6.114	11.055	6.085	.992	.079	.010	.099	.908	92.594	59.671	314	945
13.660	6.367	11.680	6.328	.990	.086	.015	.122	.921	100.475	61.842	353	1000
14.325	6.766	12.340	6.737	.992	.079	.010	.099	.963	106.304	58.456	328	1015
14.260	6.810	12.265	6.801	.997	.035	.005	.071	.907	110.444	57.862	258	1030
13.965	7.038	11.990	6.993	.987	.093	.020	.140	.943	111.470	56.297	282	1045
12.790	7.189	11.815	7.146	.975	.156	.025	.156	.956	109.781	55.542	203	1100
13.325	7.487	12.345	7.453	.980	.140	.020	.140	.943	108.198	59.715	175	1115
13.810	7.372	12.825	7.345	.985	.122	.015	.122	.965	104.316	52.836	201	1130
14.070	7.552	13.075	7.543	.995	.071	.005	.071	.932	91.435	54.606	161	1145
14.115	7.586	12.175	7.483	.970	.155	.040	.196	.886	86.554	56.136	140	1200
12.930	7.608	11.055	7.410	.937	.229	.075	.263	.927	92.066	56.684	124	1215
10.880	7.757	9.060	7.525	.910	.268	.115	.319	.748	66.472	53.289	119	1245
9.840	7.608	8.050	7.364	.895	.258	.160	.367	.742	59.970	50.739	217	1300
10.090	7.264	8.245	7.070	.922	.235	.115	.319	.764	60.182	49.474	225	1330
9.900	7.176	8.035	7.009	.932	.222	.095	.293	.723	60.839	56.316	238	1400
9.435	7.508	7.605	7.318	.915	.240	.125	.331	.695	50.841	45.094	213	1430
8.485	7.308	6.805	6.973	.840	.323	.220	.414	.633	39.958	47.246	166	1500
7.470	7.001	5.240	6.294	.743	.384	.345	.475	.551	40.186	58.239	147	1530
2.190	3.995	1.620	3.785	.570	.495	.430	.495	.284	25.777	51.165	148	1630
2.990	4.007	2.270	3.819	.720	.449	.280	.449	.500	32.173	44.241	112	1730
3.555	3.731	2.770	3.544	.785	.411	.215	.411	.637	44.581	50.038	102	1930
2.335	3.010	1.670	2.779	.665	.472	.335	.472	.400	24.303	34.719	60	2130
1.150	1.910	0.680	1.636	.470	.499	.530	.499	.158	5.640	10.482	19	2330
0.515	1.470	0.325	1.216	.190	.392	.810	.392	.0	0.0	0.0	0	2400

TABLE VI. COMPARISON OF THE ANALYTICAL AND SIMULATION MODELS

NORMAL STATISTICS FOR THE CASE 2 AMOSIST SYSTEM

System size	Queue size	Utilization	Prob(Del=0)	Average Delay	Time
2.362	1.979	2.498	- 1.538	2.856	810
3.048	2.536	3.171	- 2.687	4.152	820
3.561	2.985	3.619	- 3.391	4.620	840
2.712	2.375	2.508	- 2.567	3.480	850
2.818	2.387	2.767	- 1.876	3.622	900
3.230	2.627	3.329	- 2.706	2.153	910
2.470	1.631	3.276	- 1.590	- .126	920
2.631	1.656	3.359	- 1.931	1.078	930
2.207	1.392	2.676	- 2.639	1.617	945
2.411	1.204	3.051	- 3.388	3.682	1000
3.023	2.368	2.841	- 3.513	4.752	1015
3.651	2.453	3.887	- 3.333	2.964	1030
3.479	2.447	3.486	- 2.561	2.644	1045
- 2.864	2.399	2.408	- 4.461	3.312	1100
- .514	.910	- 2.586	2.477	.644	1115
.658	.670	.408	.534	.048	1130
1.532	1.220	1.369	- 1.876	.938	1145
2.677	2.275	2.293	- 2.187	1.356	1200
1.117	1.030	.843	- .277	- 1.292	1215
1.545	.672	1.670	- 1.149	- .480	1245
- .237	- 2.391	- 8.769	10.011	-24.455	1300
2.649	.916	2.687	- 2.612	1.535	1330
.530	- .429	.626	.410	- 1.220	1400
2.888	.629	2.950	- 1.319	.129	1430
1.295	1.179	1.227	- 2.077	.758	1500
1.795	Undef	- 4.160	Undef	- 1.160	1530
-22.622	Undef	- .284	- 2.398	0.0	1630
1.995	.015	2.608	- 1.439	.763	1730
3.895	1.336	5.241	- 2.822	1.069	1930
4.533	1.970	5.225	- 2.958	1.173	2130
4.828	2.132	5.049	- 2.399	.236	2330
4.239	.717	4.446	.120	Undef	2400

TABLE VII. COMPARISON OF THE ANALYTICAL AND SIMULATION MODELS

NORMAL STATISTICS FOR THE CASE 2 PHYSICIAN SYSTEM

System size	Queue size	Utilization	Prob(Del=0)	Average Delay	Time
6.674	4.823	2.048	- .922	7.422	810
6.112	4.506	2.048	1.644	6.953	820
5.479	4.271	- .781	- .607	5.834	840
5.145	3.965	1.455	- 2.191	6.620	850
5.740	4.664	1.080	- 1.404	6.132	900
5.609	4.523	6.021	- 4.084	5.486	910
5.699	4.645	5.536	- 3.665	7.186	920
5.163	4.230	1.901	- .869	6.526	930
5.031	4.088	1.486	- 1.771	5.579	945
5.209	4.321	1.135	- .939	7.100	1000
5.656	4.801	1.862	- 2.128	8.575	1015
5.405	4.507	7.839	- 4.980	14.987	1030
4.925	4.158	2.296	- 1.748	9.801	1045
3.127	4.258	1.052	- 2.076	8.575	1100
3.034	2.678	- .424	.465	-14.066	1115
3.309	2.939	- .151	.197	-19.519	1130
3.299	3.331	1.673	- 1.593	-20.656	1145
3.212	1.075	- 1.405	1.862	-19.765	1200
2.893	2.292	- .951	.817	2.826	1215
2.930	2.270	1.024	- .802	.796	1245
2.606	1.904	1.661	- .231	.984	1300
3.093	2.364	1.252	- .669	1.018	1330
3.292	2.479	2.555	- 2.230	1.633	1400
3.404	2.623	2.946	- 2.085	.327	1430
3.278	2.693	.714	- .116	- 1.282	1500
3.325	1.446	- .972	1.703	.490	1530
-1.100	1.351	1.454	- 7.160	4.372	1630
.114	- .206	- 2.284	2.314	- 2.672	1730
1.786	1.287	- .427	.458	- .123	1930
.640	.088	.590	.620	- 1.768	2130
.862	.150	.139	- .111	- 3.957	2330
.464	- .172	.536	- 1.071	Undef	2400

distribution for other patients. The result of servicing these shorter processing times first, causes the shorter queues. However, the output statistics of the two models were quite close, indicating that the Case 1 model is a good model of the real system. The fact that the output deterministic values for the analytical model were so close to the other two models is an indication that this analytical model is a good approach to solving the problem.

V. CASE 3 SIMULATION MODEL

A. ASSUMPTIONS

The weakest of the assumptions in all of the models was the assumption of exponential service times distributions. A third simulation was conducted to determine the sensitivity of the model to the service distributions. A degenerate distribution was used for these service times. The constant value was equal to the means of the exponential service distributions used in the other models. All other assumptions remained the same as in the Case 1 model. Table VIII is a summary of the results of the simulation.

B. RESULTS

Based on this one variation to the base case, the exponential assumption is apparently robust. Due to the randomness inherent in simulations the queues were larger than the previous models in the earlier part of the day and smaller in the later part of the day. There is not sufficient difference to indicate that the exponential distribution assumption is invalid. The means of the service times is critical, however, as was indicated in the previous section.

TABLE VIII. RESULTS OF THE CASE 3 SIMULATION MODEL

AMOSIST SYSTEM

SYS AV	S.D.	AV QJE	S.D.	UTIL	SD	P(D=0)	SD	P(D> *)	AV DELAY	S.D.	NUM	TIME
8.855	2.901	3.995	2.649	.972	.106	.070	.255	.409	17.653	11.542	381	810
6.090	2.928	1.740	2.265	.870	.217	.320	.466	.623	16.310	13.439	387	820
5.670	3.126	1.545	2.271	.825	.276	.380	.485	.456	12.803	13.274	388	840
6.910	3.657	2.510	3.022	.880	.222	.285	.451	.241	11.518	12.248	353	850
5.005	3.012	1.165	2.095	.768	.282	.505	.500	.312	8.874	12.066	369	900
6.105	3.486	1.915	2.740	.838	.249	.365	.481	.148	8.405	10.752	337	910
4.550	2.767	0.880	1.813	.734	.285	.560	.496	.261	8.030	11.913	310	920
4.895	3.396	1.260	2.369	.727	.315	.515	.500	.130	6.586	10.090	299	930
3.915	2.746	0.615	1.748	.660	.309	.690	.462	.101	4.370	9.405	267	945
3.160	2.331	0.350	1.195	.562	.315	.785	.411	.089	3.369	8.150	258	1000
3.470	2.802	0.535	1.766	.587	.318	.750	.433	.058	2.391	6.378	241	1015
2.665	2.131	0.250	1.004	.483	.300	.870	.336	.009	0.880	3.239	220	1030
2.405	1.772	0.115	0.567	.458	.296	.895	.307	.015	0.667	2.759	197	1045
1.265	1.102	0.040	0.242	.408	.336	.870	.336	.023	1.115	3.632	176	1100
1.705	1.381	0.160	0.561	.515	.354	.765	.424	.040	1.602	4.166	176	1115
1.790	1.499	0.210	0.629	.527	.373	.710	.454	.012	2.023	4.517	163	1130
1.690	1.301	0.135	0.432	.518	.352	.755	.430	.007	1.786	3.605	143	1145
1.400	1.166	0.065	0.301	.445	.342	.835	.371	.0	0.537	1.734	108	1200
1.070	1.037	0.040	0.196	.343	.313	.915	.279	.0	0.443	1.907	106	1215
1.035	1.074	0.020	0.140	.338	.342	.880	.325	.0	0.512	1.911	107	1245
1.145	0.997	0.0	0.0	.229	.199	1.0	.0	.0	0.010	0.093	150	1300
2.110	1.410	0.015	0.122	.419	.275	.950	.218	.0	0.105	0.693	232	1330
1.885	1.217	0.005	0.071	.376	.240	.965	.184	.0	0.061	0.317	151	1400
1.660	1.282	0.0	0.0	.332	.256	.985	.122	.0	0.0	0.0	155	1430
1.395	1.170	0.0	0.0	.279	.234	.995	.071	.0	0.142	1.043	133	1500
1.290	1.112	0.0	0.0	.184	.159	1.0	.0	.0	0.0	0.0	118	1530
0.315	0.535	0.0	0.0	.157	.267	.965	.184	.0	0.086	0.766	94	1630
1.170	1.078	0.150	0.477	.510	.400	.670	.470	.064	3.119	6.140	109	1730
1.040	0.974	0.095	0.341	.472	.397	.710	.454	.011	1.627	4.022	87	1930
0.730	0.853	0.045	0.207	.342	.376	.825	.380	.030	1.414	3.816	67	2130
0.415	0.594	0.0	0.0	.207	.297	.945	.228	.0	0.0	0.0	19	2330
0.0	0.0	0.0	0.0	.0	.0	1.0	.0	.0	0.0	0.0	0	2400

TABLE VIII.(CONT.) RESULTS OF THE CASE 3 SIMULATION MODEL

PHYSICIAN SYSTEM

SYS AV	S.D.	AV QUE	S.D.	UTIL	SD	P(D=0)	SD	P(D> *)	AV DELAY	S.D.	NUM	TIME
8.940	3.003	6.940	3.003	1.0	.0	.0	.0	.935	39.495	17.428	581	810
9.795	3.386	7.795	3.386	1.0	.0	.0	.0	.939	45.064	20.019	559	820
11.415	3.968	9.415	3.968	1.0	.0	.0	.0	.973	58.212	23.008	551	840
12.250	4.341	10.250	4.341	1.0	.0	.0	.0	.983	64.854	26.849	577	850
12.970	4.492	10.970	4.492	1.0	.0	.0	.0	.987	70.977	26.474	543	900
15.760	4.746	13.760	4.746	1.0	.0	.0	.0	.979	77.930	29.702	515	910
16.200	4.944	14.200	4.944	1.0	.0	.0	.0	.992	84.678	35.244	484	920
16.440	5.243	14.440	5.243	1.0	.0	.0	.0	.984	91.737	39.137	429	930
17.665	5.442	15.665	5.442	1.0	.0	.0	.0	.988	100.809	41.698	428	945
16.705	5.657	14.705	5.657	1.0	.0	.0	.0	.995	109.613	43.436	382	1000
17.605	5.947	15.605	5.947	1.0	.0	.0	.0	.997	112.901	39.874	358	1015
18.120	5.938	16.125	5.924	.997	.035	.005	.071	.985	115.896	38.946	331	1030
16.510	5.844	14.510	5.844	1.0	.0	.0	.0	.974	115.422	38.504	309	1045
15.780	6.109	14.780	6.109	1.0	.0	.0	.0	.993	117.743	37.333	295	1100
16.730	6.254	15.735	6.241	.995	.071	.005	.071	.0	112.050	38.139	251	1115
17.455	6.491	16.460	6.478	.995	.071	.005	.071	.992	108.440	39.695	238	1130
17.155	6.574	16.155	6.574	1.0	.0	.0	.0	.990	102.702	36.153	203	1145
17.555	6.619	15.560	6.607	.997	.035	.005	.071	.987	94.197	36.163	153	1200
16.680	6.687	14.700	6.640	.990	.099	.010	.099	.971	90.565	41.591	173	1215
13.950	6.816	11.980	6.761	.985	.085	.030	.171	.895	69.334	39.710	172	1245
12.355	6.823	10.425	6.708	.965	.162	.050	.218	.884	68.835	39.379	249	1300
12.605	7.074	10.655	6.993	.975	.130	.040	.196	.890	65.954	37.647	310	1330
12.165	6.938	10.230	6.866	.977	.125	.035	.184	.828	53.933	35.439	267	1400
11.270	6.929	9.340	6.827	.965	.154	.055	.228	.833	52.401	31.021	258	1430
9.930	6.852	8.030	6.724	.950	.180	.080	.271	.671	35.977	31.865	228	1500
8.395	6.480	5.805	6.040	.863	.289	.205	.404	.540	28.128	33.603	202	1530
1.710	2.906	1.145	2.684	.565	.496	.435	.496	.239	14.823	30.662	134	1630
2.625	2.906	1.790	2.778	.835	.371	.165	.371	.437	21.345	29.660	135	1730
2.465	2.396	1.640	2.241	.825	.380	.175	.380	.508	20.948	22.550	126	1930
1.435	1.888	0.795	1.662	.640	.480	.360	.480	.222	10.897	21.367	81	2130
0.630	0.976	0.200	0.693	.430	.495	.570	.495	.108	5.181	7.651	37	2330
0.045	0.416	0.030	0.315	.015	.122	.985	.122	.0	0.0	0.0	0	2400

VI. CONCLUSIONS

A. COMPARISON OF THE ANALYTICAL MODEL TO THE SIMULATION MODELS

The purpose of this study was to devise a simulation for multi-purpose queueing systems which could be applied to many varied systems. Second, the purpose was to determine the precision and correctness of an analytical model which had been applied to such a system. A first case study, duplicating the assumptions of the model indicated that, although some weaknesses existed in the analytical model, the application was a good approximation of the AMIC Clinic. The model has a finite queue and, therefore, underestimates the system at very high levels of congestion. Also, in the real system, the preemption of patients by patients whose average service times is so much less causes smaller queues than a first in, first out system.

The second simulation program revealed the difference between the real system and the analytical model in that the real world referral patients preempt patients waiting to see physicians, and the Amosists waiting with the patient cannot service other patients. The distribution of service times for these Amosist referral patients has a larger mean than originally believed, causing Amosist statistics for the second program to be larger. This program also indicated smaller physician statistics due to the preemption.

B. SENSITIVITY OF THE MODEL

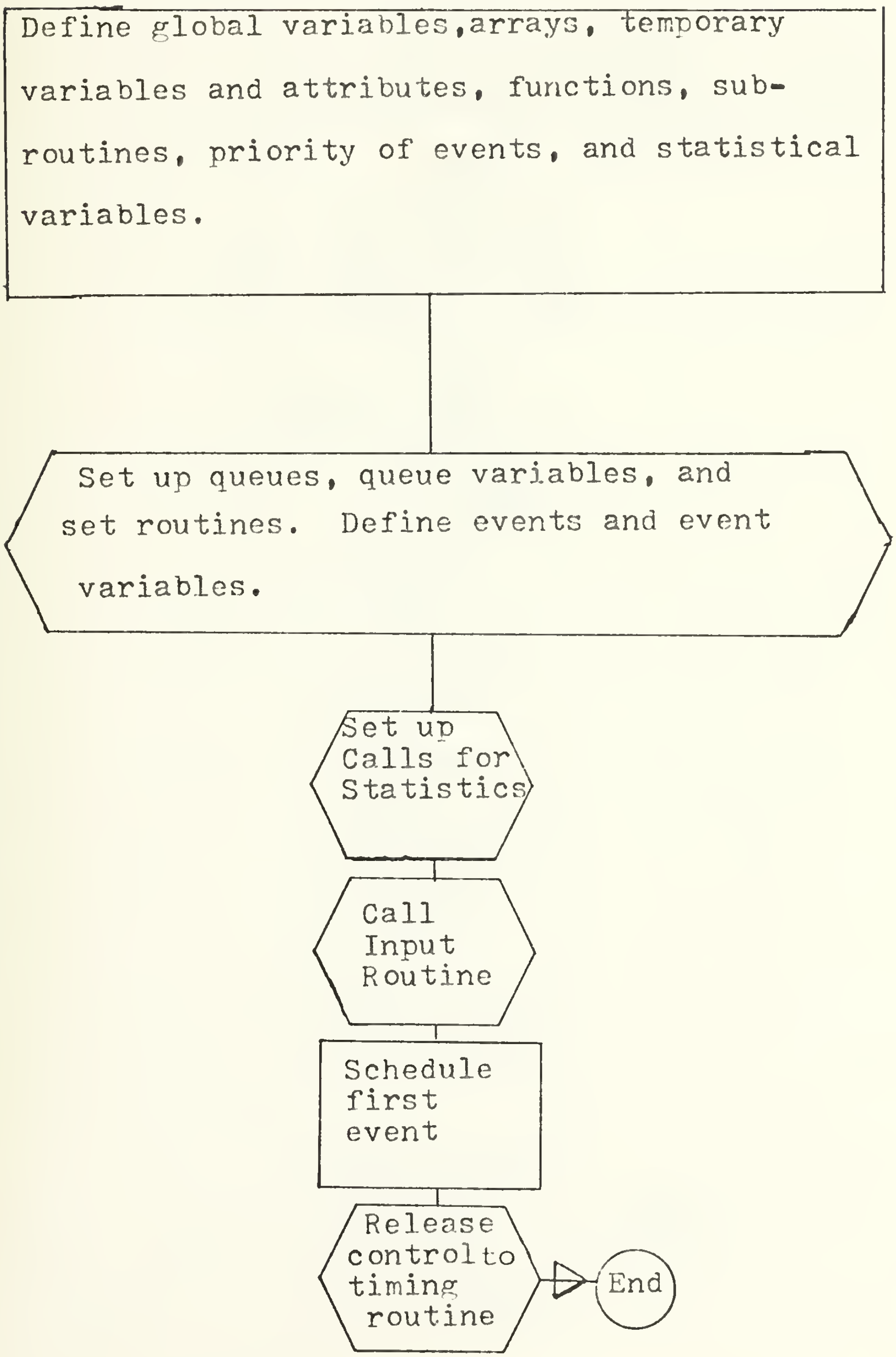
A third simulation model indicated that a service distribution assumption of exponentially distributed service times was apparently robust. However, the system is very sensitive to the means of the service distributions. In all the analytical model appears to be a good fit, and useful in providing the user with information to upgrade service and optimize staffing levels.

VI. GENERAL NATURE OF THE SIMULATION MODEL

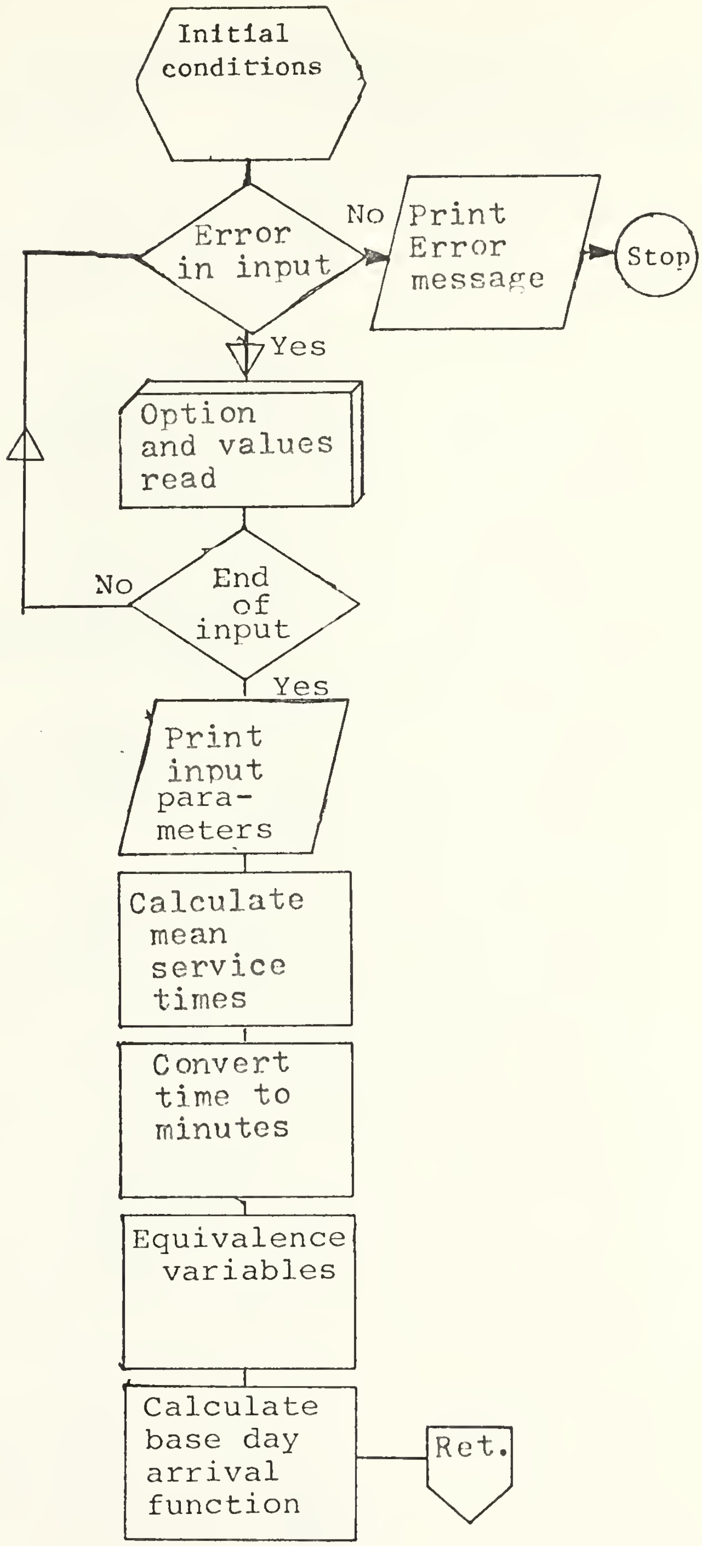
The simulation programs were developed in as general a way as possible to make them applicable to many varied systems. By manipulating the input parameters, any queueing system with up to two queues and some limitations on interaction is capable of being modeled by these programs. For example, a system with only one queue is possible by setting the percentage input parameter to 1.0, thus sending all arrivals to one queue. Several assumptions remain. Services are distributed exponentially, and the arrivals make up a poisson process. The preloading or allowing of arrivals prior to opening may be surpressed by setting the average arrival prior to opening to 0.0. The arrival process may be homogeneous or inhomogeneous depending on the functional values which are input. Supplying one arrival rate for the entire day produces a homogeneous poisson process, with a mean arrival rate equal to the input parameter.

CASE 1 SIMULATION FLOW CHART

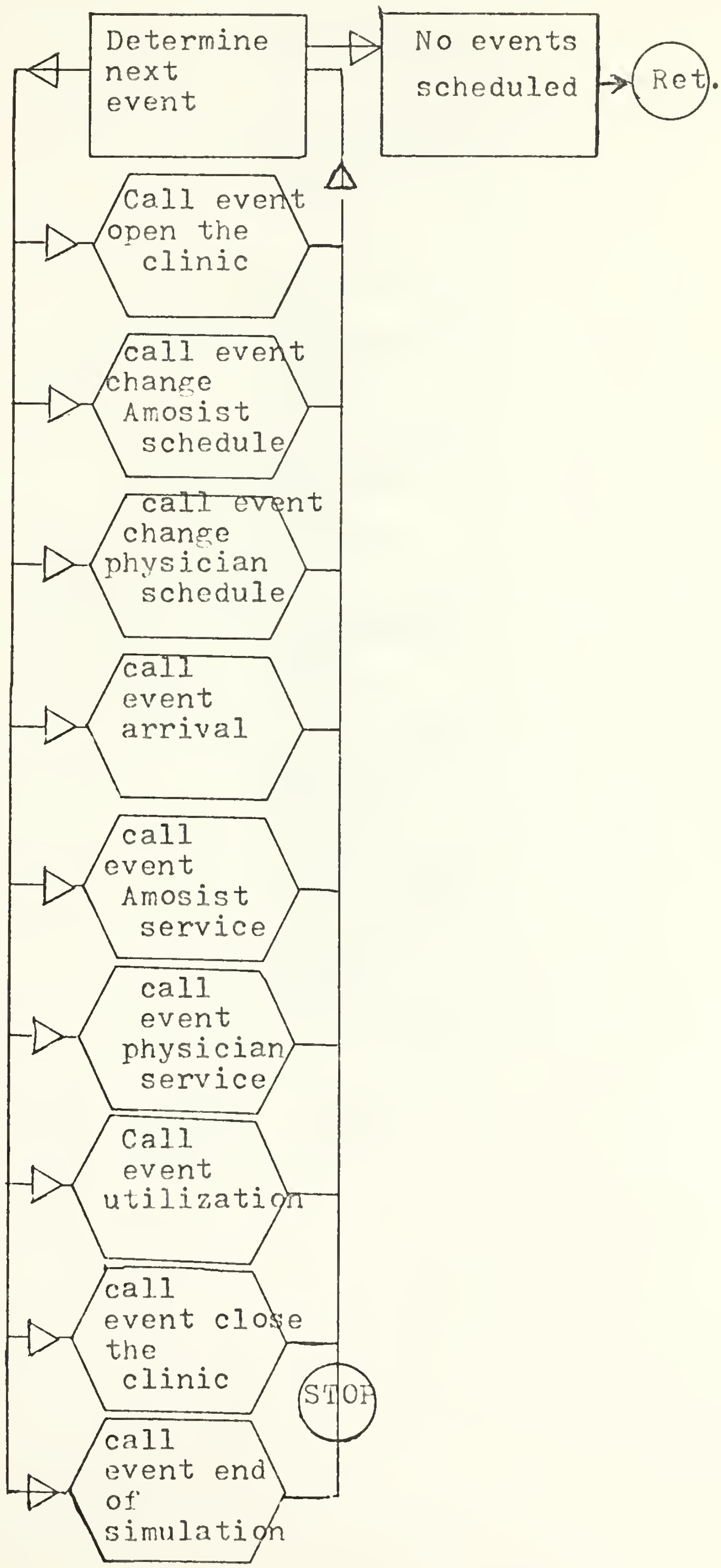
MAIN PROGRAM



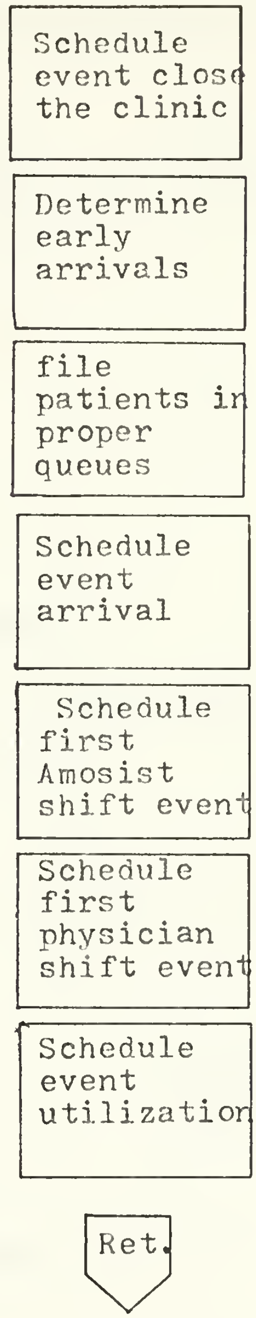
INPUT ROUTINE



SIMULATION TIMING ROUTINE

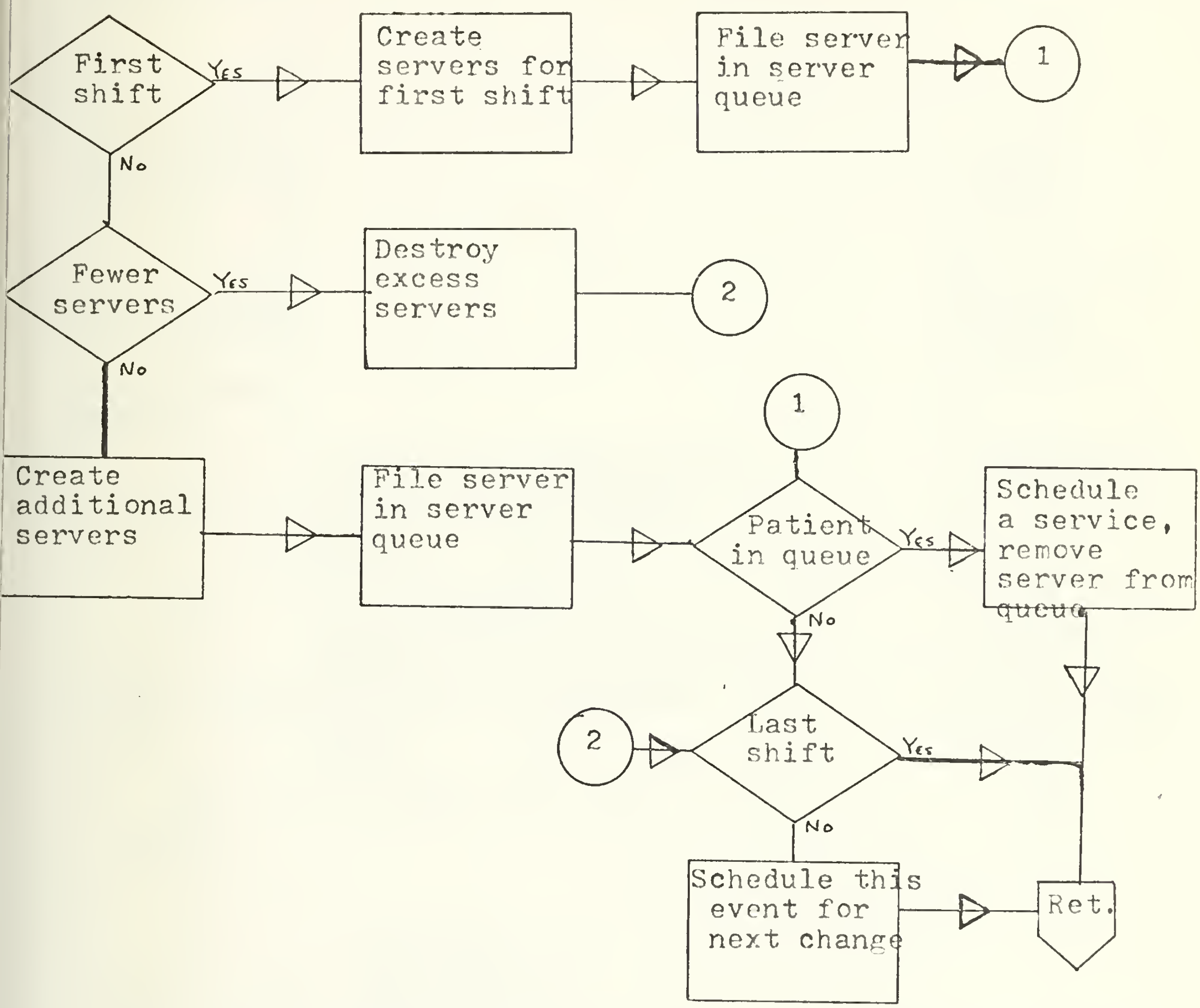


EVENT OPEN THE CLINIC



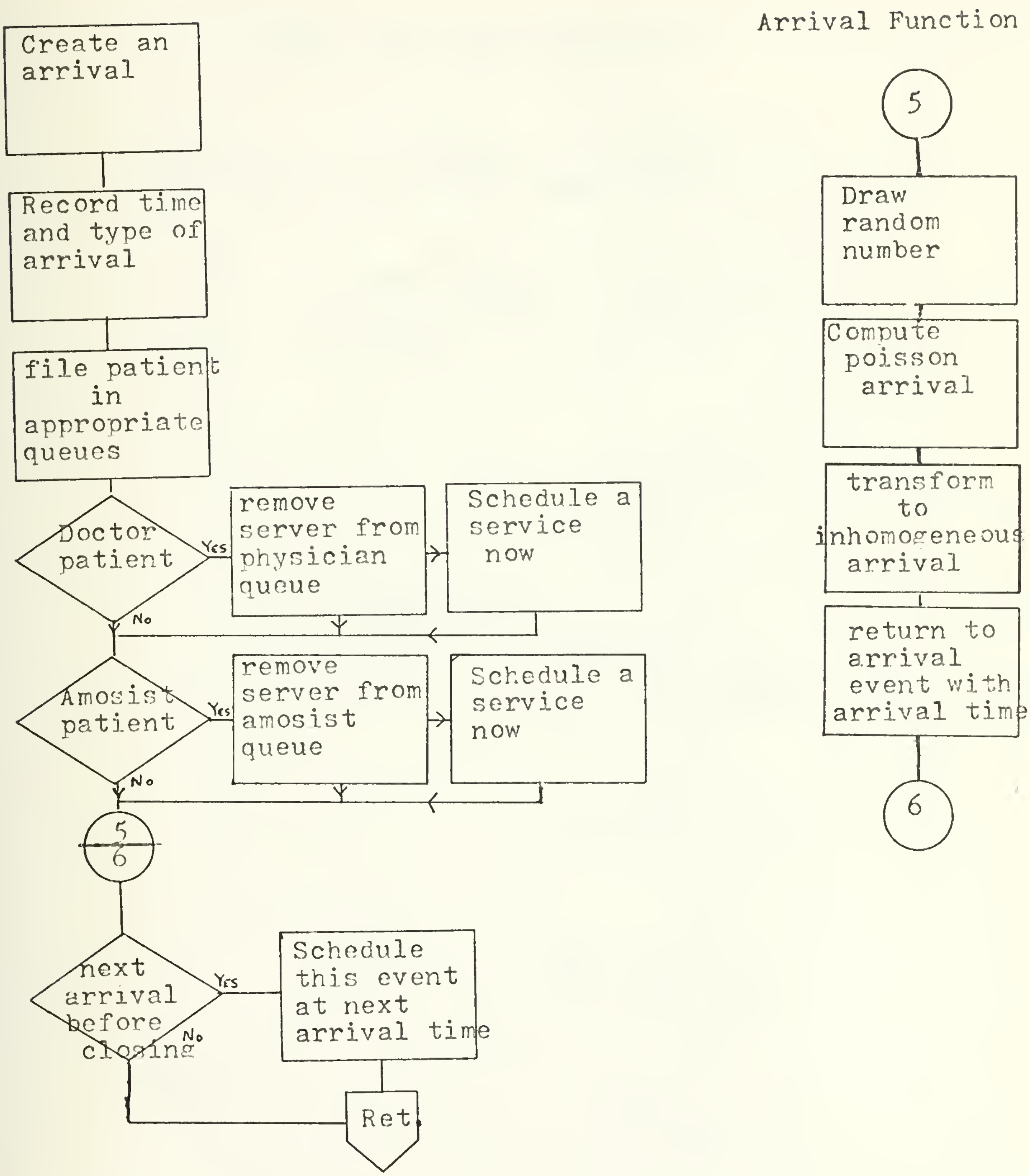
CHANGE AMOSIST SCHEDULE EVENT

CHANGE PHYSICIAN SCHEDULE EVENT



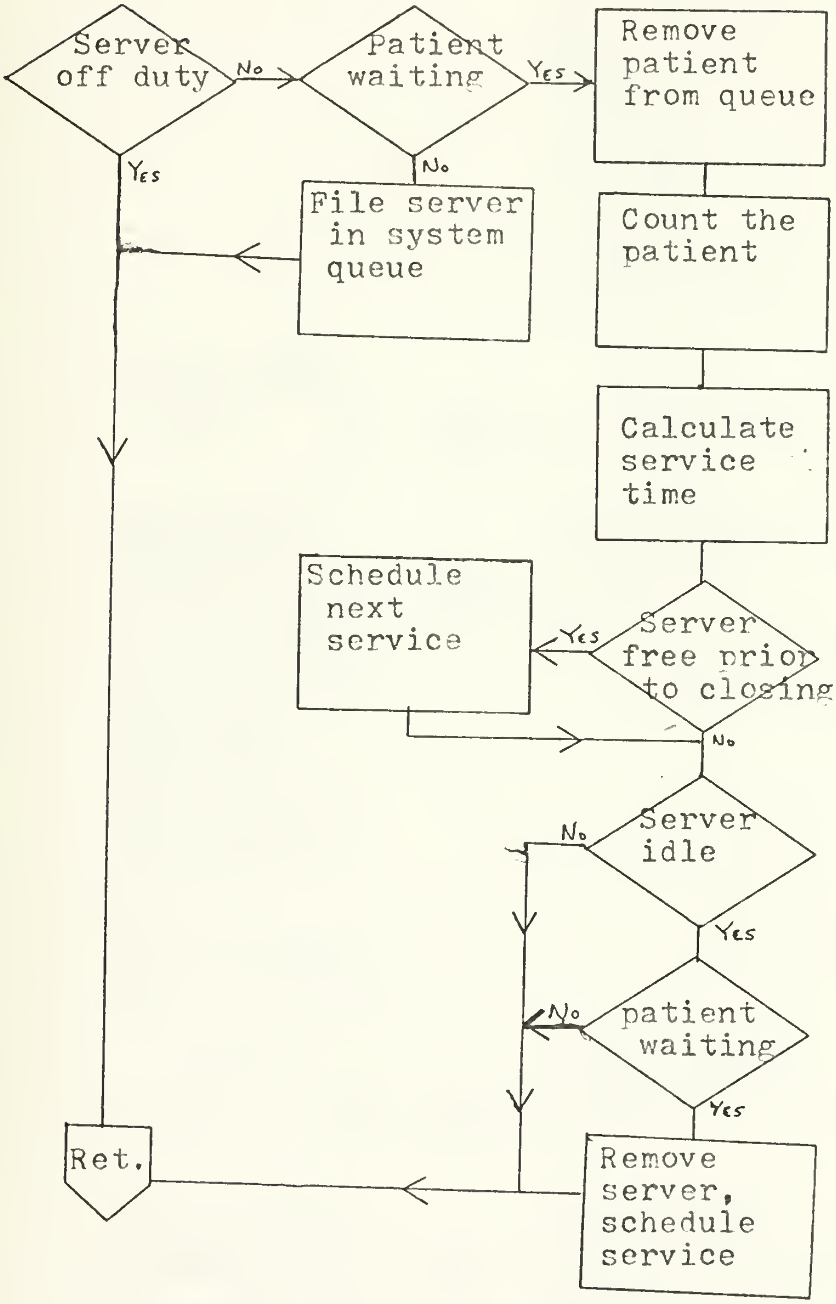
The flow charts for thses two events are exactly the same.

EVENT ARRIVAL AND ARRIVAL FUNCTION



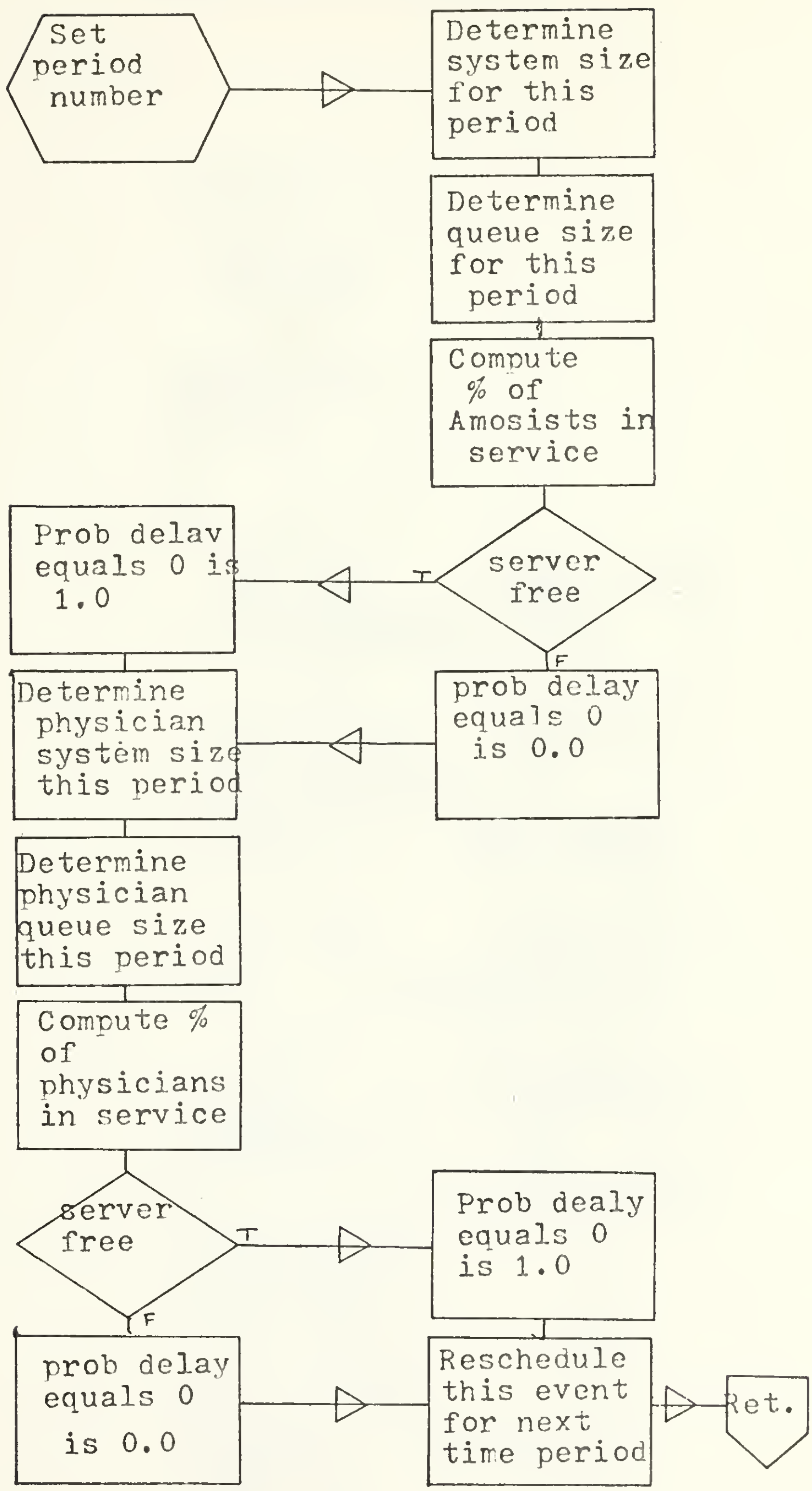
EVENT AMOSIST APPOINTMENT

EVENT PHYSICIAN APPOINTMENT

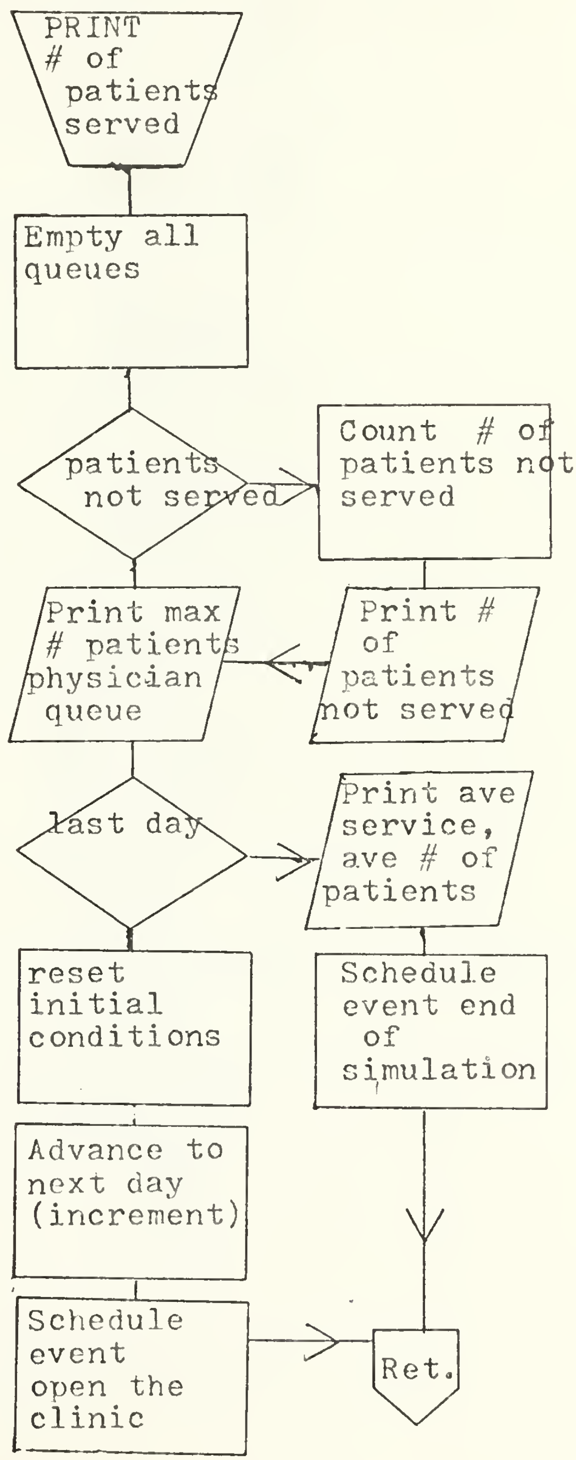


The Amosist and physician service events are identical.

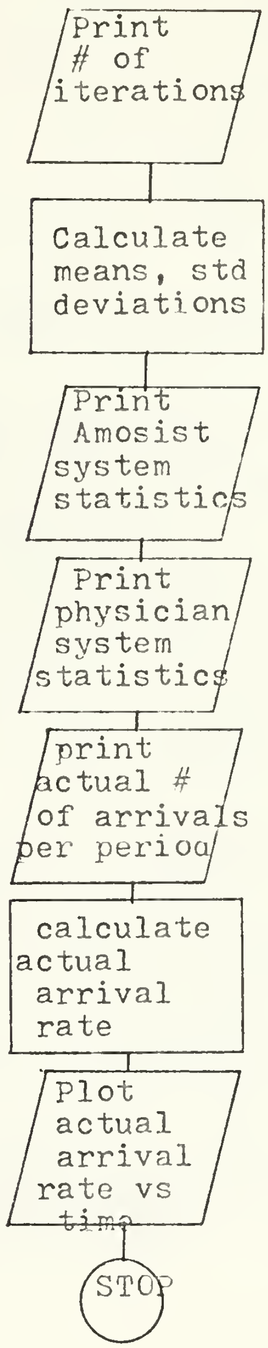
EVENT UTILIZATION



EVENT CLOSE THE CLINIC



EVENT END OF SIMULATION



```
PREAMBLE
LAST COLUMN IS 60

TEMPORARY ENTITIES
  EVERY AMOSIST MAY BELONG TO AN AVAIL.NEXT.AMOS AND HAS
          AN OFFICE
  EVERY PHYSICIAN MAY BELONG TO A NEXT.AVAIL.DOCTOR AND HAS
          A CUBICAL
  EVERY PATIENT MAY BELONG TO AN AMOS.QUEUE,A PHYS.QUEUE,
          HAS A TYPE AND A TIME.OF.ARRIVAL

EVENT NOTICES INCLUDE END.OF.SIMULATION,CLOSE.CLINIC,
          OPEN.CLINIC,TRIAGE AND UTILIZATION
  EVERY AMOS.APPOINT  HAS A NUMBER
  EVERY PHYS.APPOINT  HAS A LISTING
  EVERY CHG.IN.MEDICS HAS A SUBSCRIPT
  EVERY CHG.NUM.PHYS  HAS AN INDEX

THE SYSTEM OWNS A PHYS.QUEUE, AN AMOS.QUEUE, A
          NEXT.AVAIL.DOCTOR AND AN AVAIL.NEXT.AMOS
  DEFINE AVAIL.NEXT.AMOS, NEXT.AVAIL.DOCTOR, PHYS.QUEUE AND
          AMOS.QUEUE AS FIFO SETS WITHOUT FF,FB,FA AND RL
          ROUTINES
PRIORITY ORDER IS END.OF.SIMULATION,CHG.IN.MEDICS,
          CHG.NUM.PHYS,AMOS.APPOINT,PHYS.APPOINT,UTILIZATION,
          CLOSE.CLINIC AND TRIAGE
  DEFINE NUMBER,LISTING,OFFICE,CUBICAL,PAIRS,INDEX,REFERAL,
          SUBSCRIPT,NONREF,CONSOL,NONCONSOL,DAY,NO,LAST.DAY,
          TYPE,VOLUME,RATE.NO,NO.DUTY.CHGS AND NO.ROSTER.CHGS
          AS INTEGER VARIABLES
  DEFINE NUM,NO.DOCTORS,NO.AMOSISTS AS INTEGER,1-DIM ARRAYS
  DEFINE ALPHA,BETA,CLOSING.TIME,TIME.OF.ARRIVAL,RECORD,
          MEAN.AMOS.SERVICE,MEAN.PHYS.SERVICE,INTEGRAL AND
          DELAY.CRITERION AS REAL VARIABLES
  DEFINE S,U,TIME.TC,DOCTOR.TIME.TC,FUNCTION,CUSTOMER,TIME,
          AVE.ARRIVAL,PLOT AND RATE.FUNCTION AS 1-DIM ARRAYS
  DEFINE PERCENT AS A REAL FUNCTION
  DEFINE ARRIVAL AS A REAL FUNCTION
  DEFINE DPLTP AS A FORTRAN ROUTINE

TALLY AVE.SERVICE.A AS THE MEAN OF A.SERVICE
  TALLY AV.SERVICE.P AS THE MEAN OF P.SERVICE
    DEFINE A.SERVICE AND P.SERVICE AS REAL VARIABLES
  TALLY ACASE AS THE MEAN, SCASE AS THE STD.DEV AND MCASE AS
          THE MAXIMUM OF CASES
    DEFINE CASES AS AN INTEGER VARIABLE
  TALLY MAX1 AS THE MAXIMUM OF N.PHYS.QUEUE
  TALLY MEAN1 AS THE MEAN AND STD AS THE STD.DEV OF MAXD
    DEFINE MAXD AS A REAL VARIABLE
    DEFINE A.NO.DELAY,P.NO.DELAY,A.DELAY,P.DELAY,A.UTIL,
          A.MEAN.DELAY,P.MEAN.DELAY,P.UTIL,A.SYS.SIZE,
          P.SYS.SIZE,A.AV.QUEUE AND P.AV.QUEUE AS 1-DIM
          ARRAYS
    DEFINE MEAN AND STAT AS 2-DIM ARRAYS
    DEFINE COUNT1,COUNT2 AS INTEGER,1-DIM ARRAYS

BEFORE REMOVING FROM AMOS.QUEUE, CALL ACALC
  BEFORE REMOVING FROM PHYS.QUEUE, CALL PCALC
END
```

CASE 1 SIMULATION MODEL

```
MAIN
 CALL INPUT
 RELEASE INPUT
 ADD 1 TO DAY
 SKIP 1 OUTPUT LINE
PRINT 1 LINE THUS
                        SIMULATION BEGINS
 SCHEDULE AN OPEN.CLINIC NOW
 START NEW PAGE
 START SIMULATION
END



EVENT OPEN.CLINIC SAVING THE EVENT NOTICE
  DEFINE I AS AN INTEGER VARIABLE
  DEFINE X AS A 1-DIM ARRAY
  RESERVE X(*) AS 3
  SCHEDULE A CLOSE.CLINIC AT CLOSING.TIME
IF AVE.ARRIVAL(1)=0  JUMP AHEAD ELSE
  LET NUM(1) =POISSON.F(AVE.ARRIVAL(1),1)
IF NUM(1) <= 0  JUMP AHEAD ELSE
  LET NUM(3)=BINOMIAL.F(NUM(1),BETA,1)
  LET NUM(1)=(-NUM(3))+NUM(1)
HERE
IF AVE.ARRIVAL(2)=0  JUMP AHEAD ELSE
  LET NUM(2) = POISSON.F(AVE.ARRIVAL(2),1)
HERE
FOR I=1 TO 3 DO
              LET X(I)=REAL.F(NUM(I))
              LOOP
FOR I=1 TO NUM(1)+NUM(2)+NUM(3),DO
                                   LET LOAD=1.0/(X(1)+X(2)+X(3))
     CREATE A PATIENT
                                    LET T=PERCENT
     IF T<=X(1)*LOAD
                                   LET TYPE(PATIENT)=1
                                   LET X(1)=X(1)-1
                                   FILE PATIENT IN AMOS.QUEUE
                                   CYCLE
     ELSE
     IF T<=(X(1)+X(2))*LOAD
                                   LET TYPE(PATIENT)=2
                                   LET X(2)=X(2)-1
                                   FILE PATIENT IN PHYS.QUEUE
                                   CYCLE
     ELSE
                                   LET TYPE(PATIENT)=3
                                   LET X(3) = X(3)-1
                                   FILE PATIENT IN AMOS.QUEUE
                                   FILE PATIENT IN PHYS.QUEUE
                                   LOOP
SCHEDULE A TRIAGE AT ARRIVAL
SCHEDULE A UTILIZATION AT FUNCTION(1)
SCHEDULE A CHG.IN.MEDICS  NOW
SCHEDULE A CHG.NUM.PHYS NOW
LET INDEX(CHG.NUM.PHYS) = 1
LET SUBSCRIPT(CHG.IN.MEDICS) = 1
DESTROY OPEN.CLINIC
RETURN
END
```

```
EVENT CHG.IN.MEDICS SAVING THE EVENT NOTICE
  NORMALLY MODE IS INTEGER
  LET I = SUBSCRIPT(CHG.IN.MEDICS)

IF TIME.V=0.00
        FOR J = 1 TO NO.AMOSISTS(1),DO
                                     CREATE AN AMOSIST
                                     LET CUBICAL (AMOSIST)=J
                                     FILE AMOSIST IN AVAIL.NEXT.AMOS
                                     LOOP
        GO TO APPOINTMENT
ELSE
  LET MEDICS=NO.AMOSISTS(I)-NO.AMOSISTS(I-1)

IF MEDICS< 0
        FOR EACH AMOSIST OF AVAIL.NEXT.AMOS,DO
                IF CUBICAL (AMOSIST) >NO.AMOSISTS(I)
                         REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                         DESTROY AMOSIST
                ALWAYS
                LOOP
        JUMP AHEAD
OTHERWISE
        FOR J = NO.AMOSISTS(I-1)+1 TO NO.AMOSISTS(I),DO
                                     CREATE AN AMOSIST
                                     LET CUBICAL(AMOSIST) = J
                                     FILE AMOSIST IN AVAIL.NEXT.AMOS
                                     LOOP

'APPOINTMENT'
IF THE AMOS.QUEUE IS NOT EMPTY
                       LET AMOSIST = F.AVAIL.NEXT.AMOS
                       REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                       SCHEDULE AN AMOS.APPOINT NOW
                       LET NUMBER(AMOS.APPOINT)=CUBICAL(AMOSIST)
                       DESTROY AMOSIST
ALWAYS

HERE
    ADD 1 TO I
    LET SUBSCRIPT(CHG.IN.MEDICS) = I
IF I <=NO.DUTY.CHGS
        THEN IF TIME.TC(I)<CLOSING.TIME
                       SCHEDULE THIS CHG.IN.MEDICS AT TIME.TC(I)
ALWAYS
RETURN
END
```

```
EVENT CHG.NUM.PHYS SAVING THE EVENT NOTICE
   NORMALLY MODE IS INTEGER
   LET I = INDEX(CHG.NUM.PHYS)

IF TIME.V=0.00
        FOR J = 1 TO NO.DOCTORS(1),DO
                          CREATE A PHYSICIAN
                          LET OFFICE (PHYSICIAN) = J
                          FILE PHYSICIAN IN NEXT.AVAIL.DOCTOR
                          LOOP
        GO TO APPOINTMENT
ELSE
   LET DOC=NO.DOCTORS(I)-NO.DOCTORS(I-1)

IF DOC < 0
        FOR EACH PHYSICIAN OF NEXT.AVAIL.DOCTOR , DO
                IF OFFICE(PHYSICIAN) > NO.DOCTORS(I)
                     REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
                     DESTROY PHYSICIAN
                ALWAYS
                LOOP
        JUMP AHEAD
OTHERWISE
        FOR J = NO.DOCTORS(I-1)+1 TO NO.DOCTORS(I),DO
                          CREATE A PHYSICIAN
                          LET OFFICE(PHYSICIAN) = J
                          FILE PHYSICIAN IN NEXT.AVAIL.DOCTOR
                          LOOP

'APPOINTMENT'
IF THE PHYS.QUEUE IS NOT EMPTY
                 LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
                 REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
                 SCHEDULE A PHYS.APPOINT NOW
                 LET LISTING(PHYS.APPOINT)=OFFICE(PHYSICIAN)
                 DESTROY PHYSICIAN
ALWAYS

HERE
   ADD 1 TO I
   LET INDEX(CHG.NUM.PHYS)= I
IF I <= NO.ROSTER.CHGS
        THEN IF DOCTOR.TIME.TC(I)<CLOSING.TIME
             SCHEDULE THIS CHG.NUM.PHYS AT DOCTOR.TIME.TC(I)
ALWAYS
RETURN
END
```

```
EVENT TRIAGE SAVING THE EVENT NOTICE
CREATE A PATIENT
LET TIME.OF.ARRIVAL(PATIENT) = TIME.V
LET T=PERCENT
IF T <=ALPHA
  LET TYPE(PATIENT) =2
  FILE PATIENT IN PHYS.QUEUE
  JUMP AHEAD
OTHERWISE
IF T <= ALPHA + (1.0-ALPHA)*(1.0-BETA)
  LET TYPE(PATIENT)=1
  FILE PATIENT IN AMOS.QUEUE
  JUMP AHEAD
ELSE
  LET TYPE(PATIENT)=3
  FILE PATIENT IN AMOS.QUEUE
  FILE PATIENT IN PHYS.QUEUE
HERE
IF TYPE(PATIENT)=2 OR TYPE(PATIENT)=3
  THEN IF NEXT.AVAIL.DOCTOR IS NOT EMPTY
  LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
  REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
  SCHEDULE A PHYS.APPOINT NOW
  LET LISTING(PHYS.APPOINT)=OFFICE(PHYSICIAN)
  DESTROY PHYSICIAN
ALWAYS
IF TYPE(PATIENT)=1 OR TYPE(PATIENT)=3
  THEN IF THE AVAIL.NEXT.AMOS IS NOT EMPTY
  LET AMOSIST = F.AVAIL.NEXT.AMOS
  REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
  SCHEDULE AN AMOS.APPOINT NOW
  LET NUMBER(AMOS.APPOINT)=CUBICAL(AMOSIST)
  DESTROY AMOSIST
ALWAYS
LET NEXT = ARRIVAL
IF NEXT < CLOSING.TIME
  SCHEDULE THIS TRIAGE AT NEXT
  RETURN
OTHERWISE
DESTROY TRIAGE
RETURN
END
```

```
EVENT AMOS.APPOINT SAVING THE EVENT NOTICE
  DEFINE I AS AN INTEGER VARIABLE
  LET I=NUMBER(AMOS.APPOINT)

IF I > NO.AMOSISTS(SUBSCRIPT(CHG.IN.MEDICS)-1)
         DESTROY AMOS.APPOINT
         JUMP AHEAD
ELSE

IF THE AMOS.QUEUE IS EMPTY
                 CREATE AN AMOSIST
                 LET CUBICAL(AMOSIST)=NUMBER(AMOS.APPOINT)
                 FILE AMOSIST IN AVAIL.NEXT.AMOS
                 DESTROY AMOS.APPOINT
                 RETURN
ELSE
  LET PATIENT = F.AMOS.QUEUE
  LET A.SERVICE  = -MEAN.AMOS.SERVICE*LOG.E.F(PERCENT)

IF TYPE(PATIENT)=1
                ADD 1 TO NONREF
ALWAYS

IF TYPE (PATIENT)=3
                 ADD 1 TO REFERAL
ALWAYS
  REMOVE PATIENT FROM AMOS.QUEUE

IF M.PHYS.QUEUE(PATIENT) = 0
                         DESTROY PATIENT
ALWAYS

IF A.SERVICE +TIME.V < CLOSING.TIME
              SCHEDULE THIS AMOS.APPOINT AT A.SERVICE+TIME.V
REGARDLESS

HERE
IF THE AMOS.QUEUE IS NOT EMPTY
        FOR I=1 TO N.AMOS.QUEUE
                 WHILE AVAIL.NEXT.AMOS IS NOT EMPTY, DO
                 LET AMOSIST = F.AVAIL.NEXT.AMOS
                 REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                 SCHEDULE AN AMOS.APPOINT NOW
                 LET NUMBER(AMOS.APPOINT)=CUBICAL(AMOSIST)
                 DESTROY AMOSIST
                 LOOP
ALWAYS
RETURN END
```

```
EVENT PHYS.APPOINT SAVING THE EVENT NOTICE
  DEFINE J AS AN INTEGER VARIABLE
  LET J = LISTING(PHYS.APPOINT)

IF J > NO.DOCTORS(INDEX(CHG.NUM.PHYS)-1)
                  DESTROY PHYS.APPOINT
                  JUMP AHEAD
ELSE

IF THE PHYS.QUEUE IS EMPTY
                  CREATE A PHYSICIAN
                  LET OFFICE(PHYSICIAN)=LISTING(PHYS.APPOINT)
                  FILE PHYSICIAN IN NEXT.AVAIL.DOCTOR
                  DESTROY PHYS.APPOINT
                  RETURN
ELSE
  LET PATIENT = F.PHYS.QUEUE
  LET P.SERVICE  = -MEAN.PHYS.SERVICE*LOG.E.F(PERCENT)

IF TYPE(PATIENT)=2
         ADD 1 TO NONCONSOL
ALWAYS

IF TYPE(PATIENT)= 3
         ADD 1 TO CONSOL
ALWAYS
  REMOVE PATIENT FROM PHYS.QUEUE

IF M.AMUS.QUEUE(PATIENT) = 0
           DESTROY PATIENT
REGARDLESS

IF P.SERVICE + TIME.V < CLOSING.TIME
               SCHEDULE THIS PHYS.APPOINT AT P.SERVICE+TIME.V
ALWAYS

HERE
IF THE PHYS.QUEUE IS NOT EMPTY
      FOR I=1 TO N.PHYS.QUEUE
                  WHILE NEXT.AVAIL.DOCTOR IS NOT EMPTY, DO
               LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
               REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
               SCHEDULE A PHYS.APPOINT NOW
               LET LISTING(PHYS.APPOINT)=OFFICE(PHYSICIAN)
               DESTROY PHYSICIAN
               LOOP
ALWAYS
RETURN END
```

```
EVENT UTILIZATION SAVING THE EVENT NOTICE
  DEFINE I AS INTEGER,SAVED VARIABLE
IF TIME.V=FUNCTION(1) LET I =1 REGARDLESS
LET Y = NO.AMOSISTS(SUBSCRIPT(CHG.IN.MEDICS)-1)
LET DUMMY = N.AMOS.QUEUE + Y - N.AVAIL.NEXT.AMOS
ADD DUMMY TO A.SYS.SIZE(I)
ADD DUMMY * DUMMY TO STAT(1,I)
ADD N.AMOS.QUEUE TO A.AV.QUEUE(I)
ADD N.AMOS.QUEUE * N.AMOS.QUEUE TO STAT(3,I)
LET DUMMY = 1.0- N.AVAIL.NEXT.AMOS/Y
ADD DUMMY TO A.UTIL(I)
ADD DUMMY * DUMMY TO STAT(5,I)
IF AVAIL.NEXT.AMOS IS NOT EMPTY
    ADD 1.0 TO A.NO.DELAY(I)
REGARDLESS
LET Y = NO.DOCTORS(INDEX(CHG.NUM.PHYS)-1)
LET DUMMY = N.PHYS.QUEUE + Y - N.NEXT.AVAIL.DOCTOR
ADD DUMMY TO P.SYS.SIZE(I)
ADD DUMMY * DUMMY TO STAT(2,I)
ADD N.PHYS.QUEUE TO P.AV.QUEUE(I)
ADD N.PHYS.QUEUE * N.PHYS.QUEUE TO STAT(4,I)
LET DUMMY = 1.0- N.NEXT.AVAIL.DOCTOR/Y
ADD DUMMY TO P.UTIL(I)
ADD DUMMY * DUMMY TO STAT(6,I)
IF   NEXT.AVAIL.DOCTOR IS NOT EMPTY
    ADD 1.0 TO P.NO.DELAY(I)
REGARDLESS
LET Y = FUNCTION(I)
IF Y <= CLOSING.TIME
ADD 1 TO I
SCHEDULE THIS UTILIZATION AT FUNCTION(I)
RETURN
ELSE
DESTROY UTILIZATION
RETURN
END
```

```
EVENT CLOSE.CLINIC SAVING THE EVENT NOTICE
  NORMALLY MODE IS INTEGER
  PRINT 1 LINE WITH NUM(1),NUM(2),NUM(3) THUS
PRELOADING FOR QUEUES 1 TO 3 IS    *    *    *
  PRINT 1 DOUBLE LINE WITH REFERAL,NONREF,CONSOL,NONCONSOL
        AND TIME.V THUS
REFERALS    *  NONREF    *  CONSOLS    * NONCON    * CLOSING
 TIME   *.*
FOR EACH AMOSIST OF AVAIL.NEXT.AMOS,DO
                        REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                        DESTROY AMOSIST
                        LOOP
FOR EACH PHYSICIAN OF NEXT.AVAIL.DOCTOR,DO
                         REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
                         DESTROY PHYSICIAN
                         LOOP
IF THE AMOS.QUEUE IS NOT EMPTY
       FOR EACH PATIENT IN AMOS.QUEUE, DO
                            REMOVE PATIENT FROM AMOS.QUEUE
                            IF M.PHYS.QUEUE(PATIENT) = 0
                            DESTROY PATIENT
                            ALWAYS
                            ADD 1 TO A.PATIENTS.NOT.SEEN
                            LOOP
ALWAYS
IF THE PHYS.QUEUE IS NOT EMPTY
       FOR EACH PATIENT IN PHYS.QUEUE, DO
                            REMOVE PATIENT FROM PHYS.QUEUE
                            DESTROY PATIENT
                            ADD 1 TO P.PATIENTS.NOT.SEEN
                            LOOP
ALWAYS
  PRINT 1 LINE WITH A.PATIENTS.NOT.SEEN AND
         P.PATIENTS.NOT.SEEN THUS
LEFT IN AMOSIST CLINIC ***   LEFT IN DOCTORS OFFICE ***
  PRINT 1 LINE WITH MAX1 THUS
THE MAX NUMBER OF PATIENTS IN THE PHYSICIAN QUEUE IS      *
  SKIP 1 OUTPUT LINE
  LET MAXD = MAX1
IF DAY =LAST.DAY
  PRINT 2 LINES WITH AV.SERVICE.P AND AVE.SERVICE.A THUS
   AVERAGE PHYSICIAN SERVICE        AVERAGE AMOSIST SERVICE
         *.***                                *.***
  PRINT 1 LINE WITH MEAN1 AND STD THUS
THE AVE MAX FOR PHYS.QUEUE IS ***.***  STD DEV    ***.***
  SKIP 1 OUTPUT LINE
  PRINT 1 DOUBLE LINE WITH ACASE,SCASE AND MCASE THUS
AN NO OF PATIENTS    *.**  STD DEV    *.**    MAXIMUM
* PER DAY
  SCHEDULE AN END.OF.SIMULATION NOW
  RETURN
ELSE
  LET TIME.V =0.00  LET DAY=DAY+1  LET NO=0  LET RATE.NO=1
  LET REFERAL=0  LET CONSOL=0  LET NONCONSOL=0  LET NONREF=0
  LET RECORD=0  RESET TOTALS OF N.PHYS.QUEUE
  DESTROY CHG.IN.MEDICS  DESTROY CHG.NUM.PHYS
  DESTROY CLOSE.CLINIC          SCHEDULE AN OPEN.CLINIC NOW
  RETURN
END
```

```
EVENT END.OF.SIMULATION
  DEFINE MODCUR AS AN INTEGER VARIABLE
  PRINT 1 LINE WITH DAY THUS
END.OF.SIMULATION DAY IS ***
FOR J = 1 TO 8  ALSO FOR I = 1 TO PAIRS, DO
      IF J = 7 OR J = 8  LET STAT(J,I) = MEAN(J,I) REGARDLESS
      LET MEAN(J,I) = MEAN (J,I)/LAST.DAY
       LET STAT(J,I) = STAT(J,I)/LAST.DAY-MEAN(J,I)*MEAN(J,I)
      LET STAT(J,I) = SQRT.F(STAT(J,I))
      LOOP
FOR I = 1 TO PAIRS, DO
      IF COUNT1(I) = 0.0 JUMP AHEAD ELSE
      LET STAT(9,I) = MEAN(9,I)
      LET MEAN(9,I) = MEAN(9,I)/COUNT1(I)
      LET STAT(9,I)=SQRT.F(STAT(9,I)/COUNT1(I)- MEAN(9,I)*
          MEAN(9,I))
      LET MEAN(11,I) = MEAN(11,I)/COUNT1(I)
      LET STAT(11,I)=SQRT.F(STAT(11,I)/COUNT1(I)-MEAN(11,I)*
          MEAN(11,I))
HERE
      IF COUNT2(I) = 0.0 CYCLE ELSE
      LET STAT(10,I) = MEAN(10,I)
        LET MEAN(10,I) = MEAN(10,I) /COUNT2(I)
      LET STAT(10,I)=SQRT.F(STAT(10,I)/COUNT2(I)-MEAN(10,I) *
          MEAN(10,I))
      LET MEAN(12,I) = MEAN(12,I)/COUNT2(I)
      LET STAT(12,I)=SQRT.F(STAT(12,I)/COUNT2(I)-MEAN(12,I)*
          MEAN(12,I))
      LOOP
START NEW PAGE
PRINT 1 LINE THUS
                                              AMOSIST SYSTEM
SKIP 1 OUTPUT LINE
PRINT 1 DOUBLE LINE WITH DELAY.CRITERION THUS
  SYS SIZE  STD DEV   AVE QUEUE  STD DEV   UTILIZ  STD DEV
  P(D=0)  STD DEV  P(D> *.)  AVE DELAY  STD DEV  NO  TIME
SKIP 1 OUTPUT LINE
FOR I = 1 TO PAIRS, DO
      PRINT 1 DOUBLE LINE WITH MEAN(1,I),STAT(1,I),MEAN(3,I),
          STAT(3,I),MEAN(5,I),STAT(5,I),MEAN(7,I),STAT(7,I),
          MEAN(9,I),MEAN(11,I),STAT(11,I),COUNT1(I) AND
          TIME(I) THUS
   *.***      *.***      *.***      *.***      *.***      *.***
  *.***     *.***     *.***     *.***     *.***     *      *
      IF FRAC.F(I/5.0) =0.0 SKIP 1 OUTPUT LINE ALWAYS
      LOOP
START NEW PAGE
PRINT 1 LINE THUS
                                             PHYSICIAN SYSTEM
SKIP 1 OUTPUT LINE
PRINT 1 DOUBLE LINE WITH DELAY.CRITERION THUS
  SYS SIZE  STD DEV   AVE QUEUE  STD DEV   UTILIZ  STD DEV
  P(D=0)  STD DEV  P(D> *.)  AVE DELAY  STD DEV  NO  TIME
SKIP 1 OUTPUT LINE
FOR I =1 TO PAIRS,DO
      PRINT 1 DOUBLE LINE WITH MEAN(2,I),STAT(2,I),MEAN(4,I),
          STAT(4,I),MEAN(6,I),STAT(6,I),MEAN(8,I),STAT(8,I),
          MEAN(10,I),MEAN(12,I),STAT(12,9),COUNT2(I) AND
          TIME(I) THUS
 ***.***    ***.***      *.***      *.***      *.***      *.***
  *.***     *.***     *.***     *.***     *.***     *      *
      IF FRAC.F(I/5.0) =0.0 SKIP 1 OUTPUT LINE ALWAYS
      LOOP
START NEW PAGE
PRINT 1 LINE THUS
              ACTUAL NUMBER OF ARRIVALS PER PERIOD
SKIP 1 OUTPUT LINE
FOR I= 1 TO PAIRS, WRITE PLOT(I) AS D (15,4)
FOR I= 1 TO PAIRS-1, DO
     IF I = 1 LET PLOT(I)=PLOT(1)*60.0/(FUNCTION(1)*LAST.DAY)
     ALWAYS
     LET PLOT(I+1)=PLOT(I+1)*60.0/((FUNCTION(I+1)-
```

```
          FUNCTION(I))*LAST.DAY)
     LOOP
   CALL DPLTP(TIME(*),PLOT(*),PAIRS,MODCUR)
STOP END
```

```
ROUTINE PERCENT
LET X=RANDOM.F(2)
RETURN WITH X
END
```

```
ROUTINE INPUT
  DEFINE I AS AN INTEGER VARIABLE
  DEFINE ENCOUNTER,HOUR AND CLOCK AS 1-DIM ARRAYS
  DEFINE OPTION AS AN ALPHA,1-DIM ARRAY
  RESERVE ENCOUNTER(*),AVE.ARRIVAL(*),NUM(*) AND OPTION(*)
         AS 4
  LET ENCOUNTER(1) = 17.96 LET ENCOUNTER(2) = 23.83
  LET ENCOUNTER(3) = 13.55 LET ENCOUNTER(4) = 4.34
  LET VOLUME = 156
  LET OPENING.TIME = 300. LET CLOSING.TIME = 2400.
  LET ALPHA = .55 LET BETA = .31 LET LAST.DAY = 1
  LET AVE.ARRIVAL(1) = 7. LET AVE.ARRIVAL(2) = 4.
  LET DELAY.CRITERION=15.0
HERE
IF MODE IS NOT ALPHA
     PRINT 1 LINE WITH READ.V THUS
     EXPECTED OPTION WORD WHILE READING CARD NUMBER    *
     STOP
OTHERWISE
IF EFIELD.F-SFIELD.F > 12
     PRINT 1 LINE WITH READ.V THUS
WHILE READING CARD NUMBER    *  OPTION WORD WAS TOO LONG
     STOP
OTHERWISE
  READ OPTION(1),OPTION(2),OPTION(3)
IF OPTION(1) = "ENCO"
                FOR I = 1 TO 4, DO
                                READ ENCOUNTER(I)
                                LOOP
                JUMP BACK
ELSE
IF OPTION(1)= "VOLU"
                READ Y
                LET VOLUME= Y
                JUMP BACK
ELSE
IF OPTION(1)="OPEN"
                READ OPENING.TIME
                JUMP BACK
ELSE
IF OPTION(1)="CLOS"
                READ CLOSING.TIME
                JUMP BACK
ELSE
IF OPTION(1)="PERC"
                READ ALPHA
                JUMP BACK
ELSE
IF OPTION(1)="REFE"
                READ BETA
```

```
                JUMP BACK
ELSE
IF OPTION(1)="ITER"
                READ Y
                LET LAST.DAY = Y
                JUMP BACK
ELSE
IF OPTION(1)="PREL"
                READ AVE.ARRIVAL(1),AVE.ARRIVAL(2)
                JUMP BACK
ELSE
IF OPTION(1)="AMOS"
                READ Y
                LET NO.DUTY.CHGS = Y
           RESERVE TIME.TC(*),NO.AMOSISTS(*) AND CLOCK(*) AS
           NO.DUTY.CHGS
                FOR I = 1 TO NO.DUTY.CHGS ,DO
                        READ CLOCK(I),NO.AMOSISTS(I)
                        LOOP
                JUMP BACK
ELSE
IF OPTION(1)="PHYS"
                READ Y
                LET NO.ROSTER.CHGS = Y
           RESERVE DOCTOR.TIME.TC(*),NO.DOCTORS(*),HOUR(*) AS
           NO.ROSTER.CHGS
                FOR I = 1 TO NO.ROSTER.CHGS, DO
                          READ HOUR(I),NO.DOCTORS(I)
                        LOOP
                JUMP BACK
ELSE
IF OPTION(1)="ARRI"
                READ Y
                LET PAIRS = Y
                RESERVE FUNCTION(*),RATE.FUNCTION(*),
                CUSTOMER(*),COUNT1(*) AND COUNT2(*) AS PAIRS
                RESERVE TIME AS PAIRS
           RESERVE STAT(*,*) AND MEAN(*,*) AS 12 BY PAIRS
                FOR I = 1 TO PAIRS,DO
                        READ TIME(I),CUSTOMER(I)
                        LOOP
                JUMP BACK
ELSE
IF OPTION(1)="DELA"
                READ DELAY.CRITERION
                JUMP BACK
ELSE
IF OPTION(1) ¬= "END."
                PRINT 1 DOUBLE LINE WITH OPTION(I) THUS
**** IS NOT THE PROPER BEGINNING TO AN OPTION WORD.  SIMULAT
ION ENDED
                STOP
OTHERWISE
PRINT 1 DOUBLE LINE WITH ENCOUNTER(1), ENCOUNTER(2),
        ENCOUNTER(3) AND ENCOUNTER(4) THUS
ENCOUNTER(1) ***.**   ENCOUNTER(2) ***.**  ENCOUNTER(3) ***.
**    ENCOUNTER(4)     *.**
PRINT 1 DOUBLE LINE WITH VOLUME,OPENING.TIME,CLOSING.TIME,
        ALPHA,BETA AND LAST.DAY THUS
VOLUME     * OPENING TIME      * CLOSING TIME      * PERCENT TO
 DOCTOR .*** PERCENT REFERRED .*** NO OF DAYS      *
LET MEAN.AMOS.SERVICE =(1.0-BETA)*ENCOUNTER(1)+BETA
        *ENCOUNTER(2)
LET MEAN.PHYS.SERVICE= (ALPHA*ENCOUNTER(3)+(1.0-ALPHA)*
        ENCOUNTER(4)*BETA)/(ALPHA+(1.0-ALPHA)*BETA)
PRINT 1 LINE WITH MEAN.AMOS.SERVICE,MEAN.PHYS.SERVICE THUS
MEAN SERVICE TIME FOR AMOSIST ***.***      FOR DOCTOR ***.***
SKIP 1 OUTPUT LINE
FOR I = 1 TO 2 , DO
        PRINT 1 LINE WITH I,AVE.ARRIVAL(I) THUS
AVE PRELOADING OF QUEUE    * IS    *
        LOOP
```

```
SKIP 1 OUTPUT LINE
IF NO.DUTY.CHGS <=0
     LET NO.DUTY.CHGS = 5
     RESERVE TIME.TC(*),NO.AMOSISTS(*) AND CLOCK(*) AS 5
     LET NO.AMOSISTS(1) =5 LET NO.AMOSISTS(2) =3 LET
     NO.AMOSISTS(3) =5 LET NO.AMOSISTS(4) =7 LET
     NO.AMOSISTS(5) =2 LET CLOCK(1) =800. LET CLOCK(2)=1100.
    LET CLOCK(3)=1300. LET CLOCK(4)=1530. LET CLOCK(5)=1630.
REGARDLESS
IF NO.ROSTER.CHGS <=0
     LET NO.ROSTER.CHGS=5
     RESERVE DOCTOR.TIME.TC(*),NO.DOCTORS(*),HOUR(*) AS 5
     LET NO.DOCTORS(1)=2 LET NO.DOCTORS(2)=1 LET
     NO.DOCTORS(3)=2 LET NO.DOCTORS(4)=3 LET NO.DOCTORS(5)=1
     LET HOUR(1)=800. LET HOUR(2)=1100. LET HOUR(3)=1200.
     LET HOUR(4)=1530. LET HOUR(5)=1630.
REGARDLESS
FOR I = 1 TO NO.DUTY.CHGS,DO
         LET Y = CLOCK(I)-OPENING.TIME
         LET Y=Y/100.0
         LET TIME.TC(I) = TRUNC.F(Y)*60.0+FRAC.F(Y)*100.0
         PRINT 1 LINE WITH NO.AMOSISTS(I),CLOCK(I) THUS
NUMBER OF AMOSISTS ON DUTY ***       TIME SHIFT STARTS      *
         LOOP
SKIP 1 OUTPUT LINE
FOR I = 1 TO NO.ROSTER.CHGS,DO
         LET Y = HOUR(I)-OPENING.TIME
         LET Y= Y/100.0
         LET DOCTOR.TIME.TC(I)=TRUNC.F(Y)*60.0 +FRAC.F(Y)*
         100.0
         PRINT 1 LINE WITH NO.DOCTORS(I),HOUR(I) THUS
NUMBER OF DOCTORS ON DUTY     *      TIME SHIFT STARTS      *
         LOOP
SKIP 1 OUTPUT LINE
IF PAIRS <=0
     LET PAIRS = 32
     RESERVE FUNCTION(*),RATE.FUNCTION(*),CUSTOMER(*),
    COUNT1(*),COUNT2(*) AND TIME(*) AS 32
    RESERVE STAT(*,*) AND MEAN(*,*) AS 12 BY 32
     LET TIME(1)=810. LET TIME(2)=820. LET TIME(3)=840. LET
     TIME(4)=850. LET TIME(5)=900. LET TIME(6)=910. LET
     TIME(7)=920. LET TIME(8)=930. LET TIME(9)=945. LET
     TIME(10)=1000. LET TIME(11)=1015. LET TIME(12)=1030.
     LET TIME(13)=1045. LET TIME(14)=1100. LET TIME(15)=
     1115. LET TIME(16)=1130. LET TIME(17)=1145. LET
     TIME(18)=1200. LET TIME(19)=1215. LET TIME(20)=1245.
     LET TIME(21)=1300. LET TIME(22)=1330. LET TIME(23)=
     1400. LET TIME(24)=1430. LET TIME(25)=1500. LET
     TIME(26)=1530. LET TIME(27)=1630. LET TIME(28)=1730.
     LET TIME(29)=1930. LET TIME(30)=2130. LET TIME(31)=
     2330. LET TIME(32)=2400.
LET CUSTOMER(1)=22.5 LET CUSTOMER(2)=23.5 LET CUSTOMER(3)=
24.5 LET CUSTOMER(4)=23.5 LET CUSTOMER(5)=22.5 LET
CUSTOMER(6)=21.5 LET CUSTOMER(7)=20.5 LET CUSTOMER(8)=19.6
LET CUSTOMER(9)=18.6 LET CUSTOMER(10)=17.6 LET CUSTOMER(11)
=16.6 LET CUSTOMER(12)=14.7 LET CUSTOMER(13)=13.35 LET
CUSTOMER(14)=12.05 LET CUSTOMER(15)=10.95 LET CUSTOMER(16)=
9.9 LET CUSTOMER(17)=8.9 LET CUSTOMER(18)=7.9 LET
CUSTOMER(19)=7. LET CUSTOMER(20)=6.2 LET CUSTOMER(21)=7.2
LET CUSTOMER(22)=13.6 LET CUSTOMER(23)=12.5 LET CUSTOMER(24)
=11.1 LET CUSTOMER(25)=9.85 LET CUSTOMER(26)=8.8 LET
CUSTOMER(27)=7.55 LET CUSTOMER(28)=6.3 LET CUSTOMER(29)=5.8
LET CUSTOMER(30)=4.4 LET CUSTOMER(31)=2.8 LET CUSTOMER(32)=
.01
REGARDLESS
FOR I= 1 TO PAIRS,DO
         LET Y =(TIME(I)-OPENING.TIME)/100.0
         LET FUNCTION(I)=TRUNC.F(Y)*60.0 + FRAC.F(Y) *100.0
         PRINT 1 LINE WITH I,CUSTOMER(I),TIME(I) THUS
PERIOD NUMBER  * AVE ARRIVAL RATE    *.*** AT TIME    *
         LOOP
LET Y = (CLOSING.TIME-OPENING.TIME)/100.0
```

```
LET CLOSING.TIME=TRUNC.F(Y)*60.0 + FRAC.F(Y)*100.0
LET A.SYS.SIZE(*) = MEAN(1,*)  LET P.SYS.SIZE(*) = MEAN(2,*)
LET A.AV.QUEUE(*) = MEAN(3,*)  LET P.AV.QUEUE(*) = MEAN(4,*)
LET A.UTIL(*) = MEAN(5,*)  LET P.UTIL(*) = MEAN(6,*)
LET A.NO.DELAY(*) = MEAN(7,*)  LET P.NO.DELAY(*) = MEAN(8,*)
LET A.DELAY(*) = MEAN(9,*)  LET P.DELAY(*) = MEAN(10,*)
LET A.MEAN.DELAY(*) = MEAN(11,*)  LET P.MEAN.DELAY(*) =
         MEAN(12,*)
IF PAIRS=1
         LET INTEGRAL=CLOSING.TIME*CUSTOMER(I)
         JUMP AHEAD
ELSE
LET INTEGRAL = FUNCTION(1)*CUSTOMER(1)
FOR I = 2 TO PAIRS WHILE FUNCTION(I) < CLOSING.TIME, DO
         LET INTEGRAL =INTEGRAL+(FUNCTION(I)-FUNCTION(I-1))
         *CUSTOMER(I)
         LOOP
LET INTEGRAL = INTEGRAL+(CLOSING.TIME-FUNCTION(I-1))
         *CUSTOMER(I)
HERE
LET REMAINDER=VOLUME-AVE.ARRIVAL(1)-AVE.ARRIVAL(2)
  LET NORM.CONSTANT = REMAINDER/INTEGRAL
FOR I=1 TO PAIRS,DO
            LET RATE.FUNCTION (I)= CUSTOMER(I)*NORM.CONSTANT
            LOOP
LET 'I = INTEGRAL/CLOSING.TIME
RESERVE S(*) AND U(*) AS VOLUME +I*10
RESERVE PLOT(*) AS PAIRS
LET RATE.NO = 1
RETURN
END


ROUTINE ARRIVAL
  ADD 1 TO NO
IF NO = 1
            LET S(1)=-LOG.E.F(PERCENT)
           LET U(1) =S(1)/RATE.FUNCTION(1)
           IF U(1) >FUNCTION(1)
                   LET RECORD = FUNCTION(1)*RATE.FUNCTION(1)
                       GO TO TRANSFORM
                   OTHERWISE
           ADD 1 TO PLOT(RATE.NO)
           RETURN WITH U (1)
           ELSE
  LET S(NO)= -LOG.E.F(PERCENT)
  LET U(NO) = S(NO)/RATE.FUNCTION(RATE.NO) +U(NO-1)
  LET S(NO)=S(NO)+S(NO-1)
HERE
IF U(NO)>FUNCTION(RATE.NO)
         LET RECORD=(FUNCTION(RATE.NO)-FUNCTION(RATE.NO-1))*
         RATE.FUNCTION(RATE.NO) + RECORD
'TRANSFORM'
         ADD 1 TO RATE.NO
         IF RATE.NO > PAIRS
                   LET CASES = NO-1+NUM(1)+NUM(2)+NUM(3)
                   PRINT 1 LINE WITH CASES AND DAY THUS
    * ARRIVALS FOR DAY    *
                   RETURN WITH U(NO)
         ELSE
         LET U(NO)=(S(NO)-RECORD)/RATE.FUNCTION(RATE.NO)+
         FUNCTION(RATE.NO-1)
         JUMP BACK
ELSE
IF U(NO) >=CLOSING.TIME
         LET CASES = NO-1+NUM(1)+NUM(2)+NUM(3)
         PRINT 1 LINE WITH CASES THUS
ARRIVAL NO ****
ALWAYS
  ADD 1 TO PLOT(RATE.NO)
  RETURN WITH U(NO)
END
```

```
ROUTINE ACALC
  DEFINE I AS AN INTEGER,SAVED VARIABLE
IF TIME.V <= FUNCTION(1)      LET I = 1  REGARDLESS
HERE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)-5.0
                                                  RETURN
ELSE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)+5.0
         ADD 1 TO COUNT1(I)
         LET WAITING.TIME = TIME.V -TIME.OF.ARRIVAL(PATIENT)
         ADD WAITING.TIME TO A.MEAN.DELAY(I)
         ADD WAITING.TIME*WAITING.TIME TO STAT(11,I)
         IF WAITING.TIME > DELAY.CRITERION
                                    ADD 1.0 TO A.DELAY(I)
         ALWAYS
         RETURN
ELSE
  ADD 1 TO I
  JUMP BACK
END
```

```
ROUTINE PCALC
  DEFINE I AS AN INTEGER,SAVED VARIABLE
IF TIME.V <= FUNCTION(1)      LET I = 1  REGARDLESS
HERE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)-5.0
                                                  RETURN
ELSE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)+5.0
         ADD 1 TO COUNT2(I)
         LET WAITING.TIME = TIME.V -TIME.OF.ARRIVAL(PATIENT)
         ADD WAITING.TIME TO P.MEAN.DELAY(I)
         ADD WAITING.TIME*WAITING.TIME TO STAT(12,I)
         IF WAITING.TIME > DELAY.CRITERION
                                ADD 1.0 TO P.DELAY(I)
         ALWAYS
         RETURN
ELSE
  ADD 1 TO I
  JUMP BACK
END
```

```
PREAMBLE
LAST COLUMN IS 60

TEMPORARY ENTITIES
   EVERY AMOSIST MAY BELONG TO AN AVAIL.NEXT.AMOS AND HAS
         AN OFFICE
   EVERY PHYSICIAN MAY BELONG TO A NEXT.AVAIL.DOCTOR AND HAS
         A CUBICAL
   EVERY PATIENT MAY BELONG TO AN AMOS.QUEUE,A PHYS.QUEUE,
         A REF.QUEUE, HAS A TYPE,A TIME.OF.ARRIVAL AND A
         PH.ASSIST

EVENT NOTICES INCLUDE END.OF.SIMULATION,CLOSE.CLINIC,
         OPEN.CLINIC,TRIAGE,UTILIZATION AND CONSULT
   EVERY AMOS.APPOINT  HAS A NUMBER
   EVERY PHYS.APPOINT  HAS A LISTING
   EVERY CHG.IN.MEDICS HAS A SUBSCRIPT
   EVERY CHG.NUM.PHYS  HAS AN INDEX

THE SYSTEM OWNS A PHYS.QUEUE, AN AMOS.QUEUE, A
         NEXT.AVAIL.DOCTOR, AN AVAIL.NEXT.AMOS AND A
         REF.QUEUE
   DEFINE AVAIL.NEXT.AMOS, NEXT.AVAIL.DOCTOR AND AMOS.QUEUE
         AS FIFO SETS WITHOUT FF,FB,FA AND RL ROUTINES
   DEFINE PHYS.QUEUE AS A FIFO SET WITHOUT FF,FA AND RL
         ROUTINES
   DEFINE REF.QUEUE AS A SET RANKED BY LOW TIME.OF.ARRIVAL
         WITHOUT RF AND RL ROUTINES

PRIORITY ORDER IS END.OF.SIMULATION,CHG.IN.MEDICS,
         CHG.NUM.PHYS,AMOS.APPOINT,PHYS.APPOINT,CONSULT,
         UTILIZATION,CLOSE.CLINIC AND TRIAGE
  -DEFINE NUMBER,LISTING,OFFICE,CUBICAL,PAIRS,INDEX,REFERAL,
         SUBSCRIPT,NONREF,CONSOL,NONCONSOL,DAY,NO,LAST.DAY,
         TYPE,VOLUME,RATE.NO,NO.DUTY.CHGS,PH.ASSIST AND
         NO.ROSTER.CHGS AS INTEGER VARIABLES
   DEFINE NUM,NO.DOCTORS,NO.AMOSISTS AS INTEGER,1-DIM ARRAYS
   DEFINE ALPHA,BETA,CLOSING.TIME,TIME.OF.ARRIVAL,RECORD,
         INTEGRAL AND DELAY.CRITERION AS REAL VARIABLES
   DEFINE S,U,TIME.TC,DOCTOR.TIME.TC,FUNCTION,CUSTOMER,TIME,
         AVE.ARRIVAL,PLOT,ENCOUNTER AND RATE.FUNCTION AS
         1-DIM ARRAYS
   DEFINE PERCENT AS A REAL FUNCTION
   DEFINE ARRIVAL AS A REAL FUNCTION
   DEFINE DPLTP AS A FORTRAN ROUTINE

TALLY AVE.SERVICE.A AS THE MEAN OF A.SERVICE
   TALLY AV.SERVICE.P AS THE MEAN OF P.SERVICE
      DEFINE A.SERVICE AND P.SERVICE AS REAL VARIABLES
   TALLY ACASE AS THE MEAN, SCASE AS THE STD.DEV AND MCASE AS
         THE MAXIMUM OF CASES
      DEFINE CASES AS AN INTEGER VARIABLE
   TALLY MAX1 AS THE MAXIMUM OF N.PHYS.QUEUE
   TALLY MEAN1 AS THE MEAN AND STD AS THE STD.DEV OF MAXD
      DEFINE MAXD AS A REAL VARIABLE
      DEFINE A.NO.DELAY,P.NO.DELAY,A.DELAY,P.DELAY,A.UTIL,
         A.MEAN.DELAY,P.MEAN.DELAY,P.UTIL,A.SYS.SIZE,
         P.SYS.SIZE,A.AV.QUEUE AND P.AV.QUEUE AS 1-DIM
         ARRAYS
      DEFINE MEAN AND STAT AS 2-DIM ARRAYS
      DEFINE COUNT1,COUNT2 AS INTEGER,1-DIM ARRAYS

BEFORE REMOVING FROM AMOS.QUEUE, CALL ACALC
   BEFORE REMOVING FROM PHYS.QUEUE, CALL PCALC
END
```

CASE 2 SIMULATION MODEL

```
MAIN
 CALL INPUT
 RELEASE INPUT
 ADD 1 TO DAY
SKIP 1 OUTPUT LINE
PRINT 1 LINE THUS
                    SIMULATION BEGINS
 SCHEDULE AN OPEN.CLINIC NOW
START NEW PAGE
START SIMULATION
END



EVENT OPEN.CLINIC SAVING THE EVENT NOTICE
  DEFINE I AS AN INTEGER VARIABLE
  DEFINE X AS A 1-DIM ARRAY
  RESERVE X(*) AS 3
  SCHEDULE A CLOSE.CLINIC AT CLOSING.TIME
IF AVE.ARRIVAL(1)=0  JUMP AHEAD ELSE
  LET NUM(1) =POISSON.F(AVE.ARRIVAL(1),1)
IF NUM(1) <= 0  JUMP AHEAD ELSE
  LET NUM(3)=BINOMIAL.F(NUM(1),BETA,1)
  LET NUM(1)=(-NUM(3))+NUM(1)
HERE
IF AVE.ARRIVAL(2)=0  JUMP AHEAD ELSE
  LET NUM(2) = POISSON.F(AVE.ARRIVAL(2),1)
HERE
FOR I=1 TO 3 DO
             LET X(I)=REAL.F(NUM(I))
             LOOP
FOR I=1 TO NUM(1)+NUM(2)+NUM(3),DO
                                LET LOAD=1.0/(X(1)+X(2)+X(3))
     CREATE A PATIENT
                                  LET T=PERCENT
     IF T<=X(1)*LOAD
                                LET TYPE(PATIENT)=1
                                 LET X(1)=X(1)-1
                                  FILE PATIENT IN AMOS.QUEUE
                                CYCLE
     ELSE
     IF T<=(X(1)+X(2))*LOAD
                                LET TYPE(PATIENT)=2
                                LET X(2)=X(2)-1
                                 FILE PATIENT IN PHYS.QUEUE
                                CYCLE
     ELSE
                                LET TYPE(PATIENT)=3
                                LET X(3) = X(3)-1
                                FILE PATIENT IN AMOS.QUEUE
                                 LOOP
SCHEDULE A TRIAGE AT ARRIVAL
SCHEDULE A UTILIZATION AT FUNCTION(1)
SCHEDULE A CHG.IN.MEDICS  NOW
SCHEDULE A CHG.NUM.PHYS NOW
LET INDEX(CHG.NUM.PHYS) = 1
LET SUBSCRIPT(CHG.IN.MEDICS) = 1
DESTROY OPEN.CLINIC
RETURN
END
```

```
EVENT CHG.IN.MEDICS SAVING THE EVENT NCTICE
  NORMALLY MODE IS INTEGER
   LET I = SUBSCRIPT(CHG.IN.MEDICS)

IF TIME.V=0.00
        FOR J = 1 TO NO.AMOSISTS(1),DO
                           CREATE AN AMOSIST
                           LET CUBICAL (AMOSIST)=J
                           FILE AMOSIST IN AVAIL.NEXT.AMOS
                           LOOP
        GO TO APPOINTMENT
ELSE
  LET MEDICS=NO.AMOSISTS(I)-NO.AMOSISTS(I-1)

IF MEDICS< 0
        FOR EACH AMOSIST OF AVAIL.NEXT.AMOS,DO
                IF CUBICAL (AMOSIST) >NO.AMOSISTS(I)
                        REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                        DESTROY AMOSIST
                ALWAYS
                LOOP
        JUMP AHEAD
OTHERWISE
        FOR J = NO.AMOSISTS(I-1)+1 TO NO.AMOSISTS(I),DO
                           CREATE AN AMOSIST
                           LET CUBICAL(AMOSIST) = J
                           FILE AMOSIST IN AVAIL.NEXT.AMOS
                LOOP

'APPCINTMENT'
IF THE AMOS.QUEUE IS NOT EMPTY
                   LET AMOSIST = F.AVAIL.NEXT.AMOS
                   REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                  SCHEDULE AN AMOS.APPOINT NOW
                   LET NUMBER(AMOS.APPOINT)=CUBICAL(AMOSIST)
                   DESTROY AMOSIST
ALWAYS

HERE
    ADD 1 TO I
    LET SUBSCRIPT(CHG.IN.MEDICS) = I
IF I <=NO.DUTY.CHGS
              THEN IF TIME.TC(I)<CLOSING.TIME
                   SCHEDULE THIS CHG.IN.MEDICS AT TIME.TC(I)
ALWAYS
RETURN
END
```

```
EVENT CHG.NUM.PHYS SAVING THE EVENT NOTICE
  NORMALLY MODE IS INTEGER
  LET I = INDEX(CHG.NUM.PHYS)

IF TIME.V=0.00
        FOR J = 1 TO NO.DOCTORS(1),DO
                         CREATE A PHYSICIAN
                         LET OFFICE (PHYSICIAN) = J
                         FILE PHYSICIAN IN NEXT.AVAIL.DOCTOR
                         LOOP
        GO TO APPOINTMENT
ELSE
  LET DOC=NO.DOCTORS(I)-NO.DOCTORS(I-1)

IF DOC < 0
        FOR EACH PHYSICIAN OF NEXT.AVAIL.DOCTOR , DO
                IF OFFICE(PHYSICIAN) > NO.DOCTORS(I)
                     REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
                     DESTROY PHYSICIAN
                ALWAYS
                LOOP
        JUMP AHEAD
OTHERWISE
        FOR J = NO.DOCTORS(I-1)+1 TO NO.DOCTORS(I),DO
                         CREATE A PHYSICIAN
                         LET OFFICE(PHYSICIAN) = J
                         FILE PHYSICIAN IN NEXT.AVAIL.DOCTOR
                         LOOP

'APPOINTMENT'
IF THE PHYS.QUEUE IS NOT EMPTY
                 LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
                 REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
                 SCHEDULE A PHYS.APPOINT NOW
                 LET LISTING(PHYS.APPOINT)=OFFICE(PHYSICIAN)
                 DESTROY PHYSICIAN
ALWAYS

HERE
  ADD 1 TO I
  LET INDEX(CHG.NUM.PHYS)= I
IF I <= NO.ROSTER.CHGS
        THEN IF DOCTOR.TIME.TC(I) < CLOSING.TIME
             SCHEDULE THIS CHG.NUM.PHYS AT DOCTOR.TIME.TC(I)
ALWAYS
RETURN
END
```

```
EVENT TRIAGE SAVING THE EVENT NOTICE
CREATE A PATIENT
LET TIME.OF.ARRIVAL(PATIENT) = TIME.V
LET T=PERCENT
IF T <=ALPHA
  LET TYPE(PATIENT) =2
  FILE PATIENT IN PHYS.QUEUE
  JUMP AHEAD
OTHERWISE
IF T <= ALPHA + (1.0-ALPHA)*(1.0-BETA)
  LET TYPE(PATIENT)=1
  FILE PATIENT IN AMOS.QUEUE
  JUMP AHEAD
ELSE
  LET TYPE(PATIENT)=3
  FILE PATIENT IN AMOS.QUEUE
HERE
IF TYPE(PATIENT) = 2
  THEN IF NEXT.AVAIL.DOCTOR IS NOT EMPTY
  LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
  REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
  SCHEDULE A PHYS.APPOINT NOW
  LET LISTING(PHYS.APPOINT)=OFFICE(PHYSICIAN)
  DESTROY PHYSICIAN
ALWAYS
IF TYPE(PATIENT) = 1 OR TYPE(PATIENT) = 3
  THEN IF THE AVAIL.NEXT.AMOS IS NOT EMPTY
  LET AMOSIST = F.AVAIL.NEXT.AMOS
  REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
  SCHEDULE AN AMOS.APPOINT NOW
  LET NUMBER(AMOS.APPOINT)=CUBICAL(AMOSIST)
  DESTROY AMOSIST
ALWAYS
LET NEXT = ARRIVAL
IF NEXT < CLOSING.TIME
  SCHEDULE THIS TRIAGE AT NEXT
  RETURN
OTHERWISE
DESTROY TRIAGE
RETURN
END
```

```
EVENT AMOS.APPOINT SAVING THE EVENT NOTICE
  DEFINE I AS AN INTEGER VARIABLE
  LET I = NUMBER(AMOS.APPOINT)
IF I > NO.AMOSISTS(SUBSCRIPT(CHG.IN.MEDICS)-1)
  DESTROY AMOS.APPOINT
  JUMP AHEAD
ELSE
IF THE AMOS.QUEUE IS EMPTY
                   CREATE AN AMOSIST
                   LET CUBICAL(AMOSIST)=NUMBER(AMOS.APPOINT)
                   FILE AMOSIST IN AVAIL.NEXT.AMOS
                   DESTROY AMOS.APPOINT
                   RETURN
ELSE
LET PATIENT = F.AMOS.QUEUE
LET A.SERVICE= -ENCOUNTER(1)*LOG.E.F(PERCENT)
LET NEXT = A.SERVICE+TIME.V
REMOVE PATIENT FROM AMOS.QUEUE
IF TYPE(PATIENT)=1
                  ADD 1 TO NONREF
                  DESTROY PATIENT
ALWAYS
IF TYPE (PATIENT)=3
                 ADD 1 TO REFERAL
                 SCHEDULE A CONSULT AT NEXT
                 LET TIME.OF.ARRIVAL(PATIENT)=NEXT
                 LET PH.ASSIST(PATIENT)=NUMBER(AMCS.APPOINT)
                 DESTROY AMOS.APPOINT
                 FILE PATIENT IN REF.QUEUE
                 JUMP AHEAD
OTHERWISE
IF NEXT < CLOSING.TIME
              SCHEDULE THIS AMOS.APPOINT AT  NEXT
ALWAYS
HERE
IF THE AMOS.QUEUE IS NOT EMPTY
                      FOR I=1 TO N.AMOS.QUEUE
                      WHILE AVAIL.NEXT.AMOS IS NOT EMPTY, DO
                   LET AMOSIST = F.AVAIL.NEXT.AMOS
                   REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                   SCHEDULE AN AMOS.APPOINT NOW
                   LET NUMBER(AMOS.APPOINT)=CUBICAL(AMOSIST)
                   DESTROY AMOSIST
                   LOOP
ALWAYS
RETURN END
```

```
EVENT PHYS.APPOINT SAVING THE EVENT NOTICE
  DEFINE I AS AN INTEGER VARIABLE
  LET I = LISTING(PHYS.APPOINT)
IF I > NO.DOCTORS(INDEX(CHG.NUM.PHYS)-1)
                     DESTROY PHYS.APPOINT
                     JUMP AHEAD
ELSE
IF THE PHYS.QUEUE IS EMPTY
                   CREATE A PHYSICIAN
                   LET OFFICE(PHYSICIAN)=LISTING(PHYS.APPOINT)
                   FILE PHYSICIAN IN NEXT.AVAIL.DOCTOR
                   DESTROY PHYS.APPOINT
                   RETURN
ELSE
LET PATIENT = F.PHYS.QUEUE
IF TYPE(PATIENT)=2
          LET P.SERVICE = -ENCOUNTER(3)*LOG.E.F(PERCENT)
         ADD 1 TO NONCONSOL
ALWAYS
IF TYPE(PATIENT)= 3
         LET P.SERVICE=-ENCOUNTER(4)*LOG.E.F(PERCENT)
         ADD 1 TO CONSOL
         SCHEDULE AN AMOS.APPOINT AT P.SERVICE+TIME.V
         LET NUMBER(AMOS.APPOINT)=PH.ASSIST(PATIENT)
ALWAYS
REMOVE PATIENT FROM PHYS.QUEUE
DESTROY PATIENT
IF P.SERVICE + TIME.V < CLOSING.TIME
              SCHEDULE THIS PHYS.APPOINT AT P.SERVICE+TIME.V
ALWAYS
HERE
IF THE PHYS.QUEUE IS NOT EMPTY
                     FOR I=1 TO N.PHYS.QUEUE
                     WHILE NEXT.AVAIL.DOCTOR IS NOT EMPTY, DO
              LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
              REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
              SCHEDULE A PHYS.APPOINT NOW
              LET LISTING(PHYS.APPOINT)=OFFICE(PHYSICIAN)
               DESTROY PHYSICIAN
              LOOP
ALWAYS
RETURN END
```

```
EVENT UTILIZATION SAVING THE EVENT NOTICE
  DEFINE I AS INTEGER,SAVED VARIABLE
IF TIME.V=FUNCTION(1) LET I =1 REGARDLESS
LET Y = NO.AMOSISTS(SUBSCRIPT(CHG.IN.MEDICS)-1)
LET DUMMY = N.AMOS.QUEUE + Y - N.AVAIL.NEXT.AMOS
ADD DUMMY TO A.SYS.SIZE(I)
ADD DUMMY * DUMMY TO STAT(1,I)
ADD N.AMOS.QUEUE TO A.AV.QUEUE(I)
ADD N.AMOS.QUEUE * N.AMOS.QUEUE TO STAT(3,I)
LET DUMMY = 1.0- N.AVAIL.NEXT.AMOS/Y
ADD DUMMY TO A.UTIL(I)
ADD DUMMY * DUMMY TO STAT(5,I)
IF AVAIL.NEXT.AMOS IS NOT EMPTY
    ADD 1.0 TO A.NO.DELAY(I)
REGARDLESS
LET Y = NO.DOCTORS(INDEX(CHG.NUM.PHYS)-1)
LET DUMMY = N.PHYS.QUEUE + Y - N.NEXT.AVAIL.DOCTOR
ADD DUMMY TO P.SYS.SIZE(I)
ADD DUMMY * DUMMY TO STAT(2,I)
ADD N.PHYS.QUEUE TO P.AV.QUEUE(I)
ADD N.PHYS.QUEUE * N.PHYS.QUEUE TO STAT(4,I)
LET DUMMY = 1.0- N.NEXT.AVAIL.DOCTOR/Y
ADD DUMMY TO P.UTIL(I)
ADD DUMMY * DUMMY TO STAT(6,I)
IF   NEXT.AVAIL.DOCTOR IS NOT EMPTY
    ADD 1.0 TO P.NO.DELAY(I)
REGARDLESS
LET Y = FUNCTION(I)
IF Y <= CLOSING.TIME
ADD 1 TO I
SCHEDULE THIS UTILIZATION AT FUNCTION(I)
RETURN
ELSE
DESTROY UTILIZATION
RETURN
END
```

```
EVENT CLOSE.CLINIC SAVING THE EVENT NOTICE
  NORMALLY MODE IS INTEGER
IF THE REF.QUEUE IS NOT EMPTY
PRINT 1 LINE WITH N.REF.QUEUE THUS
AT CLOSING.TIME  * PATIENTS ARE WAITING FOR CONSOLTATION
RETURN ELSE
PRINT 1 LINE WITH NUM(1),NUM(2),NUM(3) THUS
PRELOADING FOR QUEUES 1 TO 3 IS    *    *    *
PRINT 1 DOUBLE LINE WITH REFERAL,NONREF,CONSOL,NONCONSOL
        AND TIME.V THUS
REFERALS     *  NONREF      *  CONSOLS     * NONCON     * CLOSING
 TIME    *.*
FOR EACH AMOSIST OF AVAIL.NEXT.AMOS,DO
                          REMOVE AMOSIST FROM AVAIL.NEXT.AMOS
                          DESTROY AMOSIST
                          LOOP
FOR EACH PHYSICIAN OF NEXT.AVAIL.DOCTOR,DO
                           REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
                           DESTROY PHYSICIAN
                           LOOP
IF THE AMOS.QUEUE IS NOT EMPTY
       FOR EACH PATIENT IN AMOS.QUEUE, DO
                              REMOVE PATIENT FROM AMOS.QUEUE
                              DESTROY PATIENT
                              ADD 1 TO A.PATIENTS.NOT.SEEN
                              LOOP
ALWAYS
IF THE PHYS.QUEUE IS NOT EMPTY
       FOR EACH PATIENT IN PHYS.QUEUE, DO
                              REMOVE PATIENT FROM PHYS.QUEUE
                              DESTROY PATIENT
                              ADD 1 TO P.PATIENTS.NOT.SEEN
                              LOOP
ALWAYS
PRINT 1 LINE WITH A.PATIENTS.NOT.SEEN AND
        P.PATIENTS.NOT.SEEN THUS
LEFT IN AMOSIST CLINIC ***   LEFT IN DOCTORS OFFICE ***
PRINT 1 LINE WITH MAX1 THUS
THE MAX NUMBER OF PATIENTS IN THE PHYSICIAN QUEUE IS      *
SKIP 1 OUTPUT LINE
LET MAXD = MAX1
 IF DAY =LAST.DAY
  PRINT 2 LINES WITH AV.SERVICE.P AND AVE.SERVICE.A THUS
   AVERAGE PHYSICIAN SERVICE        AVERAGE AMOSIST SERVICE
        *.***                              *.***
  PRINT 1 LINE WITH MEAN1 AND STD THUS
THE AVE MAX FOR PHYS.QUEUE IS ***.***  STD DEV   ***.***
  SKIP 1 OUTPUT LINE
PRINT 1 DOUBLE LINE WITH ACASE,SCASE AND MCASE THUS
MEAN NO OF PATIENTS    *.**  STD DEV    *.**    MAXIMUM
* PER DAY
  SCHEDULE AN END.OF.SIMULATION NOW
  RETURN
ELSE
LET TIME.V =0.00  LET DAY=DAY+1  LET NO=0  LET RATE.NO=1
LET REFERAL=0  LET CONSOL=0  LET NONCONSOL=0  LET NONREF=0
LET RECORD=0  RESET TOTALS OF N.PHYS.QUEUE
DESTROY CHG.IN.MEDICS  DESTROY CHG.NUM.PHYS
DESTROY CLOSE.CLINIC             SCHEDULE AN OPEN.CLINIC NOW
RETURN
END
```

```
EVENT END.OF.SIMULATION
  DEFINE MODCUR AS AN INTEGER VARIABLE
  PRINT 1 LINE WITH DAY THUS
END.OF.SIMULATION DAY IS ***
FOR J = 1 TO 8  ALSO FOR I = 1 TO PAIRS, DO
      IF J = 7 OR J = 8  LET STAT(J,I) = MEAN(J,I) REGARDLESS
      LET MEAN(J,I) = MEAN (J,I)/LAST.DAY
       LET STAT(J,I) = STAT(J,I)/LAST.DAY-MEAN(J,I)*MEAN(J,I)
      LET STAT(J,I) = SQRT.F(STAT(J,I))
      LOOP
FOR I = 1 TO PAIRS, DO
      IF COUNT1(I) = 0.0 JUMP AHEAD ELSE
      LET STAT(9,I) = MEAN(9,I)
      LET MEAN(9,I) = MEAN(9,I)/COUNT1(I)
      LET STAT(9,I)=SQRT.F(STAT(9,I)/COUNT1(I)- MEAN(9,I)*
          MEAN(9,I))
      LET MEAN(11,I) = MEAN(11,I)/COUNT1(I)
      LET STAT(11,I)=SQRT.F(STAT(11,I)/COUNT1(I)-MEAN(11,I)*
          MEAN(11,I))
HERE
      IF COUNT2(I) = 0.0 CYCLE ELSE
      LET STAT(10,I) = MEAN(10,I)
        LET MEAN(10,I) = MEAN(10,I) /COUNT2(I)
      LET STAT(10,I)=SQRT.F(STAT(10,I)/COUNT2(I)-MEAN(10,I) *
          MEAN(10,I))
      LET MEAN(12,I) = MEAN(12,I)/COUNT2(I)
      LET STAT(12,I)=SQRT.F(STAT(12,I)/COUNT2(I)-MEAN(12,I)*
          MEAN(12,I))
      LOOP
START NEW PAGE
PRINT 1 LINE THUS
                                          AMOSIST SYSTEM
SKIP 1 OUTPUT LINE
PRINT 1 DOUBLE LINE WITH DELAY.CRITERION THUS
  SYS SIZE  STD DEV   AVE QUEUE  STD DEV   UTILIZ  STD DEV
  P(D=0)  STD DEV  P(D> *.)  AVE DELAY  STD DEV  NO  TIME
SKIP 1 OUTPUT LINE
FOR I = 1 TO PAIRS, DO
      PRINT 1 DOUBLE LINE WITH MEAN(1,I),STAT(1,I),MEAN(3,I),
          STAT(3,I),MEAN(5,I),STAT(5,I),MEAN(7,I),STAT(7,I),
          MEAN(9,I),MEAN(11,I),STAT(11,I),COUNT1(I) AND
          TIME(I) THUS
   *.***       *.***       *.***       *.***      *.***     *.***
  *.***     *.***       *.***       *.***      *.***     *      *
      IF FRAC.F(I/5.0) =0.0 SKIP 1 OUTPUT LINE ALWAYS
      LOOP
START NEW PAGE
PRINT 1 LINE THUS
                                         PHYSICIAN SYSTEM
SKIP 1 OUTPUT LINE
PRINT 1 DOUBLE LINE WITH DELAY.CRITERION THUS
  SYS SIZE  STD DEV   AVE QUEUE  STD DEV   UTILIZ  STD DEV
  P(D=0)  STD DEV  P(D> *.)  AVE DELAY  STD DEV  NO  TIME
SKIP 1 OUTPUT LINE
FOR I =1 TO PAIRS,DO
      PRINT 1 DOUBLE LINE WITH MEAN(2,I),STAT(2,I),MEAN(4,I),
          STAT(4,I),MEAN(6,I),STAT(6,I),MEAN(8,I),STAT(8,I),
          MEAN(10,I),MEAN(12,I),STAT(12,9),COUNT2(I) AND
          TIME(I) THUS
 ***.***    ***.***       *.***       *.***      *.***     *.***
  *.***     *.***       *.***       *.***      *.***     *      *
      IF FRAC.F(I/5.0) =0.0 SKIP 1 OUTPUT LINE ALWAYS
      LOOP
START NEW PAGE
PRINT 1 LINE THUS
              ACTUAL NUMBER OF ARRIVALS PER PERIOD
SKIP 1 OUTPUT LINE
FOR I= 1 TO PAIRS, WRITE PLOT(I) AS D (15,4)
FOR I= 1 TO PAIRS-1, DO
     IF I = 1 LET PLOT(I)=PLOT(1)*60.0/(FUNCTION(1)*LAST.DAY)
     ALWAYS
     LET PLOT(I+1)=PLOT(I+1)*60.0/((FUNCTION(I+1)-
```

```
          FUNCTION(I))*LAST.DAY)
     LOOP
CALL DPLTP(TIME(*),PLOT(*),PAIRS,MODCUR
STOP END
```

```
ROUTINE PERCENT
LET X=RANDOM.F(2)
RETURN WITH X
END
```

```
ROUTINE INPUT
  DEFINE I AS AN INTEGER VARIABLE
  DEFINE HOUR AND CLOCK AS 1-DIM ARRAYS
  DEFINE OPTION AS AN ALPHA,1-DIM ARRAY
  RESERVE ENCOUNTER(*),AVE.ARRIVAL(*),NUM(*) AND OPTION(*)
          AS 4
LET ENCOUNTER(1) = 17.96 LET ENCOUNTER(2) = 23.83
LET ENCOUNTER(3) = 13.55 LET ENCOUNTER(4) = 4.34
LET VOLUME = 156
LET OPENING.TIME = 800. LET CLOSING.TIME = 2400.
LET ALPHA = .55 LET BETA = .31 LET LAST.DAY = 1
LET AVE.ARRIVAL(1) = 7. LET AVE.ARRIVAL(2) = 4.
LET DELAY.CRITERION=15.0
HERE
IF MODE IS NOT ALPHA
     PRINT 1 LINE WITH READ.V THUS
     EXPECTED OPTION WORD WHILE READING CARD NUMBER    *
     STOP
OTHERWISE
IF EFIELD.F-SFIELD.F > 12
     PRINT 1 LINE WITH READ.V THUS
WHILE READING CARD NUMBER    *  OPTION WORD WAS TOO LONG
     STOP
OTHERWISE
READ OPTION(1),OPTION(2),OPTION(3)
IF OPTION(1) = "ENCO"
               FOR I = 1 TO 4, DO
                          READ ENCOUNTER(I)
                          LOOP
               JUMP BACK
ELSE
IF OPTION(1)= "VOLU"
               READ Y
               LET VOLUME= Y
               JUMP BACK
ELSE
IF OPTION(1)="OPEN"
               READ OPENING.TIME
               JUMP BACK
ELSE
IF OPTION(1)="CLOS"
               READ CLOSING.TIME
               JUMP BACK
ELSE
IF OPTION(1)="PERC"
               READ ALPHA
               JUMP BACK
ELSE
IF OPTION(1)="REFE"
               READ BETA
```

```
                    JUMP BACK
ELSE
IF OPTION(1)="ITER"
                    READ Y
                    LET LAST.DAY = Y
                    JUMP BACK
ELSE
IF OPTION(1)="PREL"
                    READ AVE.ARRIVAL(1),AVE.ARRIVAL(2)
                    JUMP BACK
ELSE
IF OPTION(1)="AMOS"
                    READ Y
                    LET NO.DUTY.CHGS = Y
              RESERVE TIME.TC(*),NO.AMOSISTS(*) AND CLOCK(*) AS
              NO.DUTY.CHGS
                    FOR I = 1 TO NO.DUTY.CHGS ,DO
                            READ CLOCK(I),NO.AMOSISTS(I)
                    LOOP
                    JUMP BACK
ELSE
IF OPTION(1)="PHYS"
                    READ Y
                    LET NO.ROSTER.CHGS = Y
              RESERVE DOCTOR.TIME.TC(*),NO.DOCTORS(*),HOUR(*) AS
              NO.ROSTER.CHGS
                    FOR I = 1 TO NO.ROSTER.CHGS, DO
                              READ HOUR(I),NO.DOCTORS(I)
                    LOOP
                    JUMP BACK
ELSE
IF OPTION(1)="ARRI"
                    READ Y
                    LET PAIRS = Y
                    RESERVE FUNCTION(*),RATE.FUNCTION(*),
                    CUSTOMER(*),COUNT1(*) AND COUNT2(*) AS PAIRS
                    RESERVE TIME AS PAIRS
              RESERVE STAT(*,*) AND MEAN(*,*) AS 12 BY PAIRS
                    FOR I = 1 TO PAIRS,DO
                            READ TIME(I),CUSTOMER(I)
                            LOOP
                    JUMP BACK
ELSE
IF OPTION(1)="DELA"
                    READ DELAY.CRITERION
                    JUMP BACK
ELSE
IF OPTION(1) ¬= "END."
                    PRINT 1 DOUBLE LINE WITH OPTION(1) THUS
**** IS NOT THE PROPER BEGINNING TO AN OPTION WORD.  SIMULAT
ION ENDED
                    STOP
OTHERWISE
PRINT 1 DOUBLE LINE WITH ENCOUNTER(1), ENCOUNTER(2),
         ENCOUNTER(3) AND ENCOUNTER(4) THUS
ENCOUNTER(1) ***.**   ENCOUNTER(2) ***.**  ENCOUNTER(3) ***.
**   ENCOUNTER(4)    *.**
PRINT 1 DOUBLE LINE WITH VOLUME,OPENING.TIME,CLOSING.TIME,
         ALPHA,BETA AND LAST.DAY THUS
VOLUME     * OPENING TIME      * CLOSING TIME      * PERCENT TO
 DOCTOR .*** PERCENT REFERRED .*** NO OF DAYS     *
SKIP 1 OUTPUT LINE
FOR I = 1 TO 2 , DO
         PRINT 1 LINE WITH I,AVE.ARRIVAL(I) THUS
AVE PRELOADING OF QUEUE    * IS    *
         LOOP
SKIP 1 OUTPUT LINE
IF NO.DUTY.CHGS <= 0
         LET NO.DUTY.CHGS = 5
     RESERVE TIME.TC(*),NO.AMOSISTS(*) AND CLOCK(*) AS 5
     LET NO.AMOSISTS(1) =5 LET NO.AMOSISTS(2) =3 LET
     NO.AMOSISTS(3) =5 LET NO.AMOSISTS(4) =7 LET
```

```
    NO.AMOSISTS(5) =2 LET CLOCK(1) =800. LET CLOCK(2)=1100.
   LET CLOCK(3)=1300. LET CLOCK(4)=1530. LET CLOCK(5)=1630.
REGARDLESS
IF NO.ROSTER.CHGS < = 0
    LET NO.ROSTER.CHGS=5
    RESERVE DOCTOR.TIME.TC(*),NO.DOCTORS(*),HOUR(*) AS 5
    LET NO.DOCTORS(1)=2 LET NO.DOCTORS(2)=1 LET
    NO.DOCTORS(3)=2 LET NO.DOCTORS(4)=3 LET NO.DOCTORS(5)=1
    LET HOUR(1)=800. LET HOUR(2)=1100. LET HOUR(3)=1200.
    LET HOUR(4)=1530. LET HOUR(5)=1630.
REGARDLESS
FOR I = 1 TO NO.DUTY.CHGS,DO
        LET Y = CLOCK(I)-OPENING.TIME
        LET Y=Y/100.0
        LET TIME.TC(I) = TRUNC.F(Y)*60.0+FRAC.F(Y)*100.0
        PRINT 1 LINE WITH NO.AMOSISTS(I),CLOCK(I) THUS
NUMBER OF AMOSISTS ON DUTY ***      TIME SHIFT STARTS      *
        LOOP
SKIP 1 OUTPUT LINE
FOR I = 1 TO NO.ROSTER.CHGS,DO
        LET Y = HOUR(I)-OPENING.TIME
        LET Y= Y/100.0
         LET DOCTOR.TIME.TC(I)=TRUNC.F(Y)*60.0 +FRAC.F(Y)*
        100.0
        PRINT 1 LINE WITH NO.DOCTORS(I),HOUR(I) THUS
NUMBER OF DOCTORS ON DUTY     *     TIME SHIFT STARTS      *
        LOOP
SKIP 1 OUTPUT LINE
IF PAIRS<= 0
        LET PAIRS=32
   RESERVE FUNCTION(*),RATE.FUNCTION(*),CUSTOMER(*),
        COUNT1(*),COUNT2(*) AND TIME(*) AS 32
        RESERVE STAT(*,*) AND MEAN(*,*) AS 12 BY 32
    LET TIME(1)=810. LET TIME(2)=820. LET TIME(3)=840. LET
    TIME(4)=850. LET TIME(5)=900. LET TIME(6)=910. LET
    TIME(7)=920. LET TIME(8)=930. LET TIME(9)=945. LET
    TIME(10)=1000. LET TIME(11)=1015. LET TIME(12)=1030.
    LET TIME(13)=1045. LET TIME(14)=1100. LET TIME(15)=
    1115. LET TIME(16)=1130. LET TIME(17)=1145. LET
    TIME(18)=1200. LET TIME(19)=1215. LET TIME(20)=1245.
    LET TIME(21)=1300. LET TIME(22)=1330. LET TIME(23)=
    1400. LET TIME(24)=1430. LET TIME(25)=1500. LET
    TIME(26)=1530. LET TIME(27)=1630. LET TIME(28)=1730.
    LET TIME(29)=1930. LET TIME(30)=2130. LET TIME(31)=
    2330. LET TIME(32)=2400.
LET CUSTOMER(1)=22.5 LET CUSTOMER(2)=23.5 LET CUSTOMER(3)=
24.5 LET CUSTOMER(4)=23.5 LET CUSTOMER(5)=22.5 LET
CUSTOMER(6)=21.5 LET CUSTOMER(7)=20.5 LET CUSTOMER(8)=19.6
LET CUSTOMER(9)=18.6 LET CUSTOMER(10)=17.6 LET CUSTOMER(11)
=16.6 LET CUSTOMER(12)=14.7 LET CUSTOMER(13)=13.35 LET
CUSTOMER(14)=12.05 LET CUSTOMER(15)=10.95 LET CUSTOMER(16)=
9.9 LET CUSTOMER(17)=8.9 LET CUSTOMER(18)=7.9 LET
CUSTOMER(19)=7. LET CUSTOMER(20)=6.2 LET CUSTOMER(21)=7.2
LET CUSTOMER(22)=13.6 LET CUSTOMER(23)=12.5 LET CUSTOMER(24)
=11.1 LET CUSTOMER(25)=9.85 LET CUSTOMER(26)=8.8 LET
CUSTOMER(27)=7.55 LET CUSTOMER(28)=6.3 LET CUSTOMER(29)=5.8
LET CUSTOMER(30)=4.4 LET CUSTOMER(31)=2.8 LET CUSTOMER(32)=
.01
REGARDLESS
FOR I= 1 TO PAIRS,DO
        LET Y =(TIME(I)-OPENING.TIME)/100.0
        LET FUNCTION(I)=TRUNC.F(Y)*60.0 + FRAC.F(Y) *100.0
        PRINT 1 LINE WITH I,CUSTOMER(I),TIME(I) THUS
PERIOD NUMBER  * AVE ARRIVAL RATE     *.*** AT TIME      *
        LOOP
LET Y = (CLOSING.TIME-OPENING.TIME)/100.0
LET CLOSING.TIME=TRUNC.F(Y)*60.0 + FRAC.F(Y)*100.0
LET A.SYS.SIZE(*) = MEAN(1,*)  LET P.SYS.SIZE(*) = MEAN(2,*)
LET A.AV.QUEUE(*) = MEAN(3,*)  LET P.AV.QUEUE(*) = MEAN(4,*)
LET A.UTIL(*) = MEAN(5,*)  LET P.UTIL(*) = MEAN(6,*)
LET A.NO.DELAY(*) = MEAN(7,*)  LET P.NO.DELAY(*) = MEAN(8,*)
LET A.DELAY(*) = MEAN(9,*)  LET P.DELAY(*) = MEAN(10,*)
```

```
LET A.MEAN.DELAY(*) = MEAN(11,*)  LET P.MEAN.DELAY(*) =
        MEAN(12,*)
IF PAIRS=1
        LET INTEGRAL=CLOSING.TIME*CUSTOMER(I)
        JUMP AHEAD
ELSE
LET INTEGRAL = FUNCTION(1)*CUSTOMER(1)
FOR I = 2 TO PAIRS WHILE FUNCTION(I) < CLOSING.TIME, DO
        LET INTEGRAL =INTEGRAL+(FUNCTION(I)-FUNCTION(I-1))
        *CUSTOMER(I)
        LOOP
LET INTEGRAL = INTEGRAL+(CLOSING.TIME-FUNCTION(I-1))
        *CUSTOMER(I)
HERE
LET REMAINDER=VOLUME-AVE.ARRIVAL(1)-AVE.ARRIVAL(2)
  LET NORM.CONSTANT = REMAINDER/INTEGRAL
FOR I=1 TO PAIRS,DO
           LET RATE.FUNCTION (I)= CUSTOMER(I)*NORM.CONSTANT
           LOOP
LET I = INTEGRAL/CLOSING.TIME
RESERVE S(*) AND U(*) AS VOLUME +I*10
RESERVE PLOT(*) AS PAIRS
LET RATE.NO = 1
RETURN
END




ROUTINE ARRIVAL
  ADD 1 TO NO
IF NC = 1
           LET S(1)=-LOG.E.F(PERCENT)
          LET U(1) =S(1)/RATE.FUNCTION(1)
          IF U(1) >FUNCTION(1)
                  LET RECORD = FUNCTION(1)*RATE.FUNCTION(1)
                      GO TO TRANSFORM
                  OTHERWISE
          ADD 1 TO PLOT(RATE.NO)
          RETURN WITH U (1)
          ELSE
  LET S(NO)= -LOG.E.F(PERCENT)
  LET U(NO) = S(NO)/RATE.FUNCTION(RATE.NO) +U(NO-1)
  LET S(NO)=S(NO)+S(NO-1)
HERE
IF U(NO)>FUNCTION(RATE.NO)
        LET RECORD=(FUNCTION(RATE.NO)-FUNCTION(RATE.NO-1))*
        RATE.FUNCTION(RATE.NO) + RECORD
'TRANSFORM'
        ADD 1 TO RATE.NO
        IF RATE.NO > PAIRS
                  LET CASES = NO-1+NUM(1)+NUM(2)+NUM(3)
        PRINT 1 LINE WITH CASES AND DAY THUS
    * ARRIVALS FOR DAY    *
                  RETURN WITH U(NO)
        ELSE
        LET U(NO)=(S(NO)-RECORD)/RATE.FUNCTION(RATE.NO)+
        FUNCTION(RATE.NO-1)
        JUMP BACK
ELSE
IF U(NO) >=CLOSING.TIME
        LET CASES = NO-1+NUM(1)+NUM(2)+NUM(3)
        PRINT 1 LINE WITH CASES THUS
ARRIVAL NO ****
ALWAYS
  ADD 1 TO PLOT(RATE.NO)
  RETURN WITH U(NO)
END
```

```
ROUTINE ACALC
  DEFINE I AS AN INTEGER,SAVED VARIABLE
IF TIME.V <= FUNCTION(1)      LET I = 1  REGARDLESS
HERE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)-5.0
                                                RETURN
ELSE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)+5.0
         ADD 1 TO COUNT1(I)
         LET WAITING.TIME = TIME.V -TIME.OF.ARRIVAL(PATIENT)
         ADD WAITING.TIME TO A.MEAN.DELAY(I)
         ADD WAITING.TIME*WAITING.TIME TO STAT(11,I)
         IF WAITING.TIME > DELAY.CRITERION
                                ADD 1.0 TO A.DELAY(I)
         ALWAYS
         RETURN
ELSE
  ADD 1 TO I
  JUMP BACK
END
```

```
ROUTINE PCALC
  DEFINE I AS AN INTEGER,SAVED VARIABLE
IF TYPE(PATIENT) = 3
         RETURN
ELSE
IF TIME.V <= FUNCTION(1)      LET I = 1  REGARDLESS
HERE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)-5.0
                                                RETURN
ELSE
IF TIME.OF.ARRIVAL(PATIENT) < FUNCTION(I)+5.0
         ADD 1 TO COUNT2(I)
         LET WAITING.TIME = TIME.V -TIME.OF.ARRIVAL(PATIENT)
         ADD WAITING.TIME TO P.MEAN.DELAY(I)
         ADD WAITING.TIME*WAITING.TIME TO STAT(12,I)
         IF WAITING.TIME > DELAY.CRITERION
                            ADD 1.0 TO P.DELAY(I)
         ALWAYS
         RETURN
ELSE
  ADD 1 TO I
  JUMP BACK
END
```

```
EVENT CONSULT SAVING THE EVENT NOTICE
  NORMALLY MODE IS INTEGER
  DESTROY CONSULT
IF TIME.V > CLOSING.TIME
                       SCHEDULE A CLOSE.CLINIC NOW
REGARDLESS
  LET PATIENT = F.REF.QUEUE
  REMOVE PATIENT FROM REF.QUEUE
IF THE NEXT.AVAIL.DOCTOR IS NOT EMPTY
            FILE PATIENT IN PHYS.QUEUE
            SCHEDULE A PHYS.APPOINT NOW
            LET PHYSICIAN = F.NEXT.AVAIL.DOCTOR
            REMOVE PHYSICIAN FROM NEXT.AVAIL.DOCTOR
            LET LISTING(PHYS.APPOINT) = OFFICE(PHYSICIAN)
            RETURN
```

```
OTHERWISE
  LET KIND = F.PHYS.QUEUE
HERE
IF KIND ¬= 0
            IF TYPE(KIND) ¬=3
                   FILE PATIENT BEFORE KIND IN PHYS.QUEUE
                   RETURN
            OTHERWISE
            LET KIND = S.PHYS.QUEUE(KIND)
            JUMP BACK
OTHERWISE
  FILE PATIENT IN PHYS.QUEUE
RETURN END
```

```
ENCOUNTER
17.96 23.83 13.55 4.34
VOLUME
156
OPENING.TIME
0800
CLOSING.TIME
2400
PERCENTAGE
.55
REFERRALS
.31
ITERATIONS
200
PRELOADING
7.0 4.0
AMOSIST.SCH
5 0800. 5. 1100. 3. 1300. 5. 1530. 7. 1630. 2.
PHYSICIANS
5 0800. 2. 1100. 1. 1200. 2. 1530. 3. 1630. 1.
ARRIVAL.FCN
32 810. 22.5 820. 23.5 840. 24.5 850. 23.5 900. 22.5 910.
21.5 920. 20.5 930. 19.6 945. 18.6 1000. 17.6 1015. 16.6
1030. 14.7 1045. 13.35 1100. 12.05 1115. 10.95 1130. 9.9
1145. 8.9 1200. 7.9 1215. 7.0 1245. 6.2 1300. 7.2 1330 13.6
1400. 12.5 1430. 11.1 1500. 9.85 1530. 8.8 1630. 7.55 1730.
6.3 1930. 5.8 2130. 4.4 2330. 2.8 2400. .01
DELAY.VALUE
15.0
END.OF.DATA
```

SAMPLE INPUT

BIBLIOGRAPHY

1. Butterworth, Richard W., "Determining Staffing Levels for Outpatient Walk-In Clinics," Proceedings of the Forum on Ambulatory Care Systems, National Cooperative Services Center for Hospital Management Engineering, San Francisco, California, June 1975.

2. Emshoff, James R. and Sisson, Roger C., Design and Use of Computer Simulation Models, Macmillan Publishing Co., Inc., New York, 1970.

3. Hogg, Robert V. and Craig, Allen T., Introduction to Mathematical Statistics, 3rd ed., Macmillan Publishing Co., Inc., New York, 1970.

4. Kiviat, P. J., Villanueva, R., and Markowitz, H. M., Simscript II.5 Programming Language, Consolidated Analysis Centers, Inc., Los Angeles, 1968.

5. Naval Postgraduate School Report NPS55Bd 75061, On Congestion in Outpatient Walk-In Clinics, by Richard W. Butterworth, June 1975.

6. Van Asdlen, David L., and Wahlig, Leonard O., A Numerical Solution for Time-Dependent, Multi-Channel Queues and An Application to the Acute Minor Illness Clinic, Silas B. Hayes Hospital, Fort Ord, California. M. S. Thesis, Naval Postgraduate School, Monterey, California, September 1974.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	2
4.	Professor R. Butterworth Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	10
5.	Associate Professor J. Brown Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	Project Officer Family Practice Models Health Care Studies Division Army Academy of Health Sciences Fort Sam Houston, Texas 78234	1
7.	Dr. J. D. Bloom, Capt, MC, USN Naval Regional Medical Center San Diego, California	1
8.	Dr. J. F. Powers, Col, MC, USA Chief, Ambulatory Care Division Headquarters, United States Army Health Services Command Fort Sam Houston, Texas 78234	1
9.	Dr. W. G. Peard, Col, MC, USA Chief, Health Care Studies Division Academy of Health Sciences, U. S. Army Fort Sam Houston, Texas 78234	1

10. Dr. L. J. Legters, Col, MC, USA 1
Box 115, Student Detachment
U. S. Army War College
Carlisle Barracks, Pennsylvania 17013

11. Major L. R. Woods, MSC, USA 1
Chief, Management Division
Silas B. Hayes Hospital
Fort Ord, California 93941

12. Dr. G. A. Gorczyca, Col, MC, USA 1
Chief, Ambulatory Services
Silas B. Hayes Hospital
Fort Ord, California 93941

13. LT E. S. Matheson, USN 1
OASD (HDee) Health Personnel Task Force
Room 707, Lynn Building
101 19th Street
Rosslyn, Virginia 22209

14. LT B. B. Culmer, USN 1
Fleet Coordinating Group Two
Little Creek
Norfolk, Virginia 27722

15. Dr. Osvaldo Bustos, LTCOL, MC, USA 1
Chief, Acute Minor Illness Clinic
Silas B. Hayes Hospital
Fort Ord, California 93941

16. LT Dave Van Asdlen 1
13404 Portofino Drive
Del Mar, California 92014

17. LT L. O. Wahlig 1
Bureau of Naval Personnel
(Pers 21141)
Navy Department
Washington, D. C. 20370

162579
Thesis
C92565
c.1
Culmer
A simulation model for multi-channel, time-dependent queueing systems and an application to test and evaluate analytical model of the U. S. Army Acute Minor Illness Clinics.