PROBLEMS

1.1. Describe what you think would be the most effective way to study each of the following systems, in terms of the possibilities in Fig. 1.1, and discuss why.
(*a*) A small section of an existing factory
(*b*) A freeway interchange that has experienced severe congestion
(*c*) An emergency room in an existing hospital
(*d*) A pizza-delivery operation
(*e*) The shuttle-bus operation for a rental-car agency at an airport
(*f*) A battlefield communications network

1.2. For each of the systems in Prob. 1.1, suppose that it has been decided to make a study via a simulation model. Discuss whether the simulation should be static or dynamic, deterministic or stochastic, and continuous or discrete.

1.3. For the single-server queueing system in Sec. 1.4, define $L(t)$ to be the *total* number of customers in the system at time t (including the queue and the customer in service at time t, if any).
(*a*) Is it true that $L(t) = Q(t) + 1$? Why or why not?
(*b*) For the same realization considered for the hand simulation in Sec. 1.4.2, make a plot of $L(t)$ vs. t (similar to Figs. 1.5 and 1.6) between times 0 and $T(6)$.
(*c*) From your plot in part (*b*), compute $\hat{\ell}(6)$ = the time-average number of customers in the system during the time interval $[0, T(6)]$. What is $\hat{\ell}(6)$ estimating?
(*d*) Augment Fig. 1.7 to indicate how $\hat{\ell}(6)$ is computed during the course of the simulation.

1.4. For the single-server queue of Sec. 1.4, suppose that we did not want to estimate the expected average delay in queue; the model's structure and parameters remain the same. Does this change the state variables? If so, how?

1.5. For the single-server queue of Sec. 1.4, let W_i = the *total* time in the system of the ith customer to finish service, which includes the time in queue plus the time in service of this customer. For the same realization considered for the hand simulation in Sec. 1.4.2, compute $\hat{w}(m)$ = the average time in system of the first m customers to exit the system, for $m = 5$; do this by augmenting Fig. 1.7 appropriately. How does this change the state variables, if at all?

1.6. From Fig. 1.5, it is clear that the maximum length of the queue was 3. Write a general expression for this quantity (for the n-delay stopping rule), and augment Fig. 1.7 so that it can be computed systematically during the simulation.

1.7. Modify the code for the single-server queue in Sec. 1.4.4 to compute and write in addition the following measures of performance:
(*a*) The time-average number in the system (see Prob. 1.3)
(*b*) The average total time in the system (see Prob. 1.5)
(*c*) The maximum queue length (see Prob. 1.6)
(*d*) The maximum delay in queue
(*e*) The maximum time in the system
(*f*) The proportion of customers having a delay in queue in excess of 1 minute
Run this program, using the random-number generator given in App. 7A.

1.8. The algorithm in Sec. 1.4.3 for generating an exponential random variate with mean β was to return $-\beta \ln U$, where U is a $U(0, 1)$ random variate. This algorithm could validly be changed to return $-\beta \ln(1 - U)$. Why?

1.9. Run the single-server queueing simulation of Sec. 1.4.4 ten times by placing a loop around most of the main program, beginning just before the initialization and ending just after invoking the report generator. Discuss the results. (This is called *replicating* the simulation 10 times independently.)

1.10. For the single-server queueing simulation of Sec. 1.4, suppose that the facility opens its doors at 9 A.M. (call this time 0) and closes its doors at 5 P.M., but operates until all customers present (in service or in queue) at 5 P.M. have been served. Change the code to reflect this stopping rule, and estimate the same performance measures as before.

1.11. For the single-server queueing system of Sec. 1.4, suppose that there is room in the queue for only two customers, and that a customer arriving to find that the queue is full just goes away (this is called *balking*). Simulate this system for a stopping rule of exactly 480 minutes, and estimate the same quantities as in Sec. 1.4, as well as the expected number of customers who balk.

1.12. Consider the inventory simulation of Sec. 1.5.
(*a*) For this model with these parameters, there can never be more than one order outstanding (i.e., previously ordered but not yet delivered) at a time. Why?
(*b*) Describe specifically what changes would have to be made if the delivery lag were uniformly distributed between 0.5 and 6.0 months (rather than between 0.5 and 1.0 month); no other changes to the model are being considered. Should ordering decisions be based only on the inventory level $I(t)$?

1.13. Modify the inventory simulation of Sec. 1.5 so that it makes five replications of each (s, S) policy; see Prob. 1.9. Discuss the results. Which inventory policy is best? Are you sure?

1.14. A service facility consists of two servers in series (tandem), each with its own FIFO queue (see Fig. 1.52). A customer completing service at server 1 proceeds to server 2, while a customer completing service at server 2 leaves the facility. Assume that the interarrival times of customers to server 1 are IID exponential random variables with mean 1 minute. Service times of customers at server 1 are IID exponential random variables with mean 0.7 minute, and at server 2 are IID exponential random variables with mean 0.9 minute. Run the simulation for exactly 1000 minutes and estimate for each server the expected average delay in queue of a customer, the expected time-average number of customers in queue, and the expected utilization.

1.15. In Prob. 1.14, suppose that there is a travel time from the exit from server 1 to the arrival to queue 2 (or to server 2). Assume that this travel time is distributed uniformly

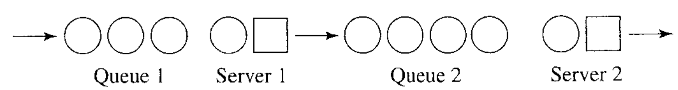

FIGURE 1.52
A tandem queueing system.

between 0 and 2 minutes. Modify the simulation and rerun it under the same conditions to obtain the same performance measures. What is the required dimension (i.e., length) of the event list?

1.16. In Prob. 1.14, suppose that no queueing is allowed for server 2. That is, if a customer completing service at server 1 sees that server 2 is idle, she proceeds directly to server 2, as before. However, a customer completing service at server 1 when server 2 is busy with another customer must stay at server 1 until server 2 gets done; this is called *blocking*. While a customer is blocked from entering server 2, she receives no additional service from server 1, but prevents server 1 from taking the first customer, if any, from queue 1. Furthermore, "fresh" customers continue to arrive to queue 1 during a period of blocking. Compute the same six performance measures as in Prob. 1.14.

1.17. For the inventory system of Sec. 1.5, suppose that if the inventory level $I(t)$ at the beginning of a month is less than zero, the company places an *express order* to its supplier. [If $0 \le I(t) < s$, the company still places a normal order.] An express order for Z items costs the company $48 + 4Z$ dollars, but the delivery lag is now uniformly distributed on [0.25, 0.50] month. Run the simulation for all nine policies and estimate the expected average total cost per month, the expected proportion of time that there is a backlog, that is, $I(t) < 0$, and the expected number of express orders placed. Is express ordering worth it?

1.18. For the inventory simulation of Sec. 1.5, suppose that the inventory is *perishable*, having a shelf life distributed uniformly between 1.5 and 2.5 months. That is, if an item has a shelf life of l months, then l months after it is placed in inventory it spoils and is of no value to the company. (Note that different items in an order from the supplier will have different shelf lives.) The company discovers that an item is spoiled only upon examination before a sale. If an item is determined to be spoiled, it is discarded and the next item in the inventory is examined. Assume that items in the inventory are processed in a FIFO manner. Repeat the nine simulation runs and assume the same costs as before. Also compute the proportion of items taken out of the inventory that are discarded due to being spoiled.

1.19. Consider a service facility with s (where $s \ge 1$) parallel servers. Assume that interarrival times of customers are IID exponential random variables with mean $E(A)$ and that service times of customers (regardless of the server) are IID exponential random variables with mean $E(S)$. If a customer arrives and finds an idle server, the customer begins service immediately, choosing the leftmost (lowest-numbered) idle server if there are several available. Otherwise, the customer joins the tail of a *single* FIFO queue that supplies customers to all the servers. (This is called an $M/M/s$ queue; see App. 1B.) Write a general program to simulate this system that will estimate the expected average delay in queue, the expected time-average number in queue, and the expected utilization of each of the servers, based on a stopping rule of n delays having been completed. The quantities s, $E(A)$, $E(S)$, and n should be input parameters. Run the model for $s = 5$, $E(A) = 1$, $E(S) = 4$, and $n = 1000$.

1.20. Repeat Prob. 1.19, but now assume that an arriving customer finding more than one idle server chooses among them with equal probability. For example, if $s = 5$ and a customer arrives to find servers 1, 3, 4, and 5 idle, he chooses each of these servers with probability 0.25.

1.21. Customers arrive to a bank consisting of three tellers in parallel.

(*a*) If there is a single FIFO queue feeding all tellers, what is the required dimension (i.e., length) of the event list for a simulation model of this system?

(*b*) If each teller has his own FIFO queue and if a customer can *jockey* (i.e., jump) from one queue to another (see Sec. 2.6 for the jockeying rules), what is the required dimension of the event list? Assume that jockeying takes no time.

(*c*) Repeat part (*b*) if jockeying takes 3 seconds.

Assume in all three parts that no events are required to terminate the simulation.

1.22. A manufacturing system contains m machines, each subject to randomly occurring breakdowns. A machine runs for an amount of time that is an exponential random variable with mean 8 hours before breaking down. There are s (where s is a fixed, positive integer) repairmen to fix broken machines, and it takes one repairman an exponential amount of time with mean 2 hours to complete the repair of one machine; no more than one repairman can be assigned to work on a broken machine even if there are other idle repairmen. If more than s machines are broken down at a given time, they form a FIFO "repair" queue and wait for the first available repairman. Further, a repairman works on a broken machine until it is fixed, regardless of what else is happening in the system. Assume that it costs the system \$50 for each hour that each machine is broken down and \$10 an hour to employ each repairman. (The repairmen are paid an hourly wage regardless of whether they are actually working.) Assume that $m = 5$, but write general code to accommodate a value of m as high as 20 by changing an input parameter. Simulate the system for exactly 800 hours for each of the employment policies $s = 1, 2, \ldots, 5$ to determine which policy results in the smallest expected average cost per hour. Assume that at time 0 all machines have just been "freshly" repaired.

1.23. For the facility of Prob. 1.10, suppose that the server normally takes a 30-minute lunch break at the first time after 12 noon that the facility is empty. If, however, the server has not gone to lunch by 1 P.M., the server will go after completing the customer in service at 1 P.M. (Assume in this case that all customers in the queue at 1 P.M. will wait until the server returns.) If a customer arrives while the server is at lunch, the customer *may* leave immediately without being served; this is called *balking*. Assume that whether such a customer balks depends on the amount of time remaining before the server's return. (The server posts his time of return from lunch.) In particular, a customer who arrives during lunch will balk with the following probabilities:

Time remaining before server's return (minutes)	Probability of a customer's balking
[20, 30)	0.75
[10, 20)	0.50
[0, 10)	0.25

(The random-integer-generation method discussed in Sec. 1.5.2 can be used to determine whether a customer balks. For a simpler approach, see Sec. 8.4.1.) Run the simulation and estimate the same measures of performance as before. (Note that the server is not busy when at lunch and that the time-average number in queue is computed including data from the lunch break.) In addition, estimate the expected number of customers who balk.

1.24. For the single-server queueing facility of Sec. 1.4, suppose that a customer's service time is known at the instant of arrival. Upon completing service for a customer, the server chooses from the queue (if any) the customer with the smallest service time. Run the simulation until 1000 customers have completed their delays and estimate the expected average delay in queue, the expected time-average number in queue, and the expected proportion of customers whose delay in queue is greater than 1 minute. (This priority queue discipline is called *shortest job first.*)

1.25. For the tandem queue of Prob. 1.14, suppose that with probability 0.2, a customer completing service at server 2 is *dissatisfied* with her overall service and must be completely served over again (at least once) by both servers. Define the delay in queue of a customer (in a particular queue) to be the total delay in that queue for all of that customer's passes through the facility. Simulate the facility for each of the following cases (estimate the same measures as before):
(*a*) Dissatisfied customers join the tail of queue 1.
(*b*) Dissatisfied customers join the head of queue 1.

1.26. A service facility consists of two type A servers and one type B server (not necessarily in the psychological sense). Assume that customers arrive at the facility with interarrival times that are IID exponential random variables with a mean of 1 minute. Upon arrival, a customer is determined to be either a type 1 customer or a type 2 customer, with respective probabilities of 0.75 and 0.25. A type 1 customer can be served by any server but will choose a type A server if one is available. Service times for type 1 customers are IID exponential random variables with a mean of 0.8 minute, regardless of the type of server. Type 1 customers who find all servers busy join a single FIFO queue *for type 1 customers*. A type 2 customer requires service from *both* a type A server *and* the type B server *simultaneously*. Service times for type 2 customers are uniformly distributed between 0.5 and 0.7 minute. Type 2 customers who arrive to find both type A servers busy *or* the type B server busy join a single FIFO queue *for type 2 customers*. Upon completion of service of *any* customer, preference is given to a type 2 customer if one is present and if both a type A and the type B server are then idle. Otherwise, preference is given to a type 1 customer. Simulate the facility for exactly 1000 minutes and estimate the expected average delay in queue and the expected time-average number in queue for each type of customer. Also estimate the expected proportion of time that each server spends on each type of customer.

1.27. A supermarket has two checkout stations, regular and express, with a single checker per station; see Fig. 1.53. Regular customers have exponential interarrival times with mean 2.1 minutes and have exponential service times with mean 2.0 minutes. Express customers have exponential interarrival times with mean 1.1 minutes and exponential service times with mean 0.9 minute. The arrival processes of the two types of customers are independent of each other. A regular customer arriving to find at least one checker idle begins service immediately, choosing the regular checker if both are idle; regular customers arriving to find both checkers busy join the end of the regular queue. Similarly, an express customer arriving to find an idle checker goes right into service, choosing the express checker if both are idle; express customers arriving to find both checkers busy join the end of the express queue, even if it is longer than the regular queue. When either checker finishes serving a customer, he takes the next customer from his queue, if any, and if his queue is empty but the other one is not, he takes the first customer from the other queue. If both queues are empty, the checker becomes idle. Note that the mean service time of a customer is determined by the customer type, and not by whether the checker is the regular or express one. Initially, the system is empty and idle, and the simulation is to run for exactly 8 hours. Compute the average delay in each queue, the time-average number in each queue, and the utilization of each checker. What recommendations would you have for further study or improvement of this system? (On June 21, 1983, the Cleveland *Plain Dealer*, in a story entitled "Fast Checkout Wins over Low Food Prices," reported that "Supermarket shoppers think fast checkout counters are more important than attractive prices, according to a survey [by] the Food Marketing Institute. . . . The biggest group of shoppers, 39 percent, replied 'fast checkouts,' . . . and 28 percent said good or low prices . . . [reflecting] growing irritation at having to stand in line to pay the cashier.")

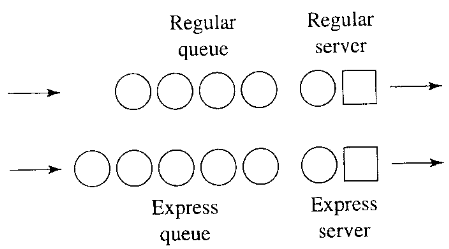

FIGURE 1.53
A supermarket checkout operation.

1.28. A one-pump gas station is always open and has two types of customers. A police car arrives every 30 minutes (exactly), with the first police car arriving at time 15 minutes. Regular (nonpolice) cars have exponential interarrival times with mean 5.6 minutes, with the first regular car arriving at time 0. Service times at the pump for all cars are exponential with mean 4.8 minutes. A car arriving to find the pump idle goes right into service, and regular cars arriving to find the pump busy join the end of a single queue. A police car arriving to find the pump busy, however, goes to the front of the line, ahead of any regular cars in line. [If there are already other police cars at the front of the line, assume that an arriving police car gets in line ahead of them as well. (How could this happen?)] Initially the system is empty and idle, and the simulation is to run until exactly 500 cars (of any type) have completed their delays in queue. Estimate the expected average delay in queue for each type of car separately, the expected time-average number of cars (of either type) in queue, and the expected utilization of the pump.

1.29. Of interest in telephony are models of the following type. Between two large cities, A and B, are a fixed number, n, of long-distance lines or circuits. Each line can operate in either direction (i.e., can carry calls originating in A or B) but can carry only one call at a time; see Fig. 1.54. If a person in A or B wants to place a call to the other city and

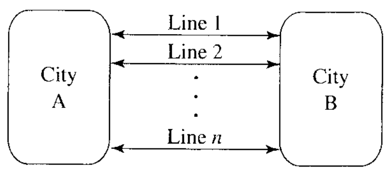

FIGURE 1.54
A long-distance telephone system.

a line is open (i.e., idle), the call goes through immediately on one of the open lines. If all n lines are busy, the person gets a recording saying that she must hang up and try later; there are no facilities for queueing for the next open line, so these *blocked* callers just go away. The times between attempted calls from A to B are exponential with mean 10 seconds, and the times between attempted calls from B to A are exponential with mean 12 seconds. The length of a conversation is exponential with mean 4 minutes, regardless of the city of origin. Initially all lines are open, and the simulation is to run for 12 hours; compute the time-average number of lines that are busy, the time-average proportion of lines that are busy, the total number of attempted calls (from either city), the number of calls that are blocked, and the proportion of calls that are blocked. Determine approximately how many lines would be needed so that no more than 5 percent of the attempted calls will be blocked.

1.30. City busses arrive to the maintenance facility with exponential interarrival times with mean 2 hours. The facility consists of a single inspection station and two identical repair stations; see Fig. 1.55. Every bus is inspected, and inspection times are distributed uniformly between 15 minutes and 1.05 hours; the inspection station is fed by a single FIFO queue. Historically, 30 percent of the busses have been found during inspection to need some repair. The two parallel repair stations are fed by a single FIFO queue, and repairs are distributed uniformly between 2.1 hours and 4.5 hours. Run the simulation for 160 hours and compute the average delay in each queue, the average length of each queue, the utilization of the inspection station, and the utilization of the repair station (defined to be half of the time-average number of busy repair stations, since there are two stations). Replicate the simulation 5 times. Suppose that the arrival rate of busses quadrupled, i.e., the mean interarrival time decreased to 30 minutes. Would the facility be able to handle it? Can you answer this question without simulation?

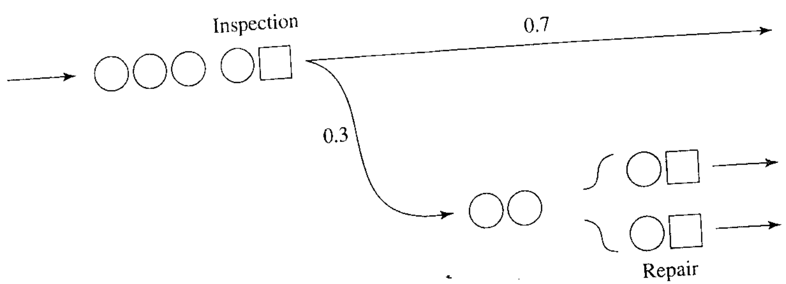

FIGURE 1.55
A bus maintenance depot.

CHAPTER 2

Modeling Complex Systems

Recommended sections for a first reading: 2.1 through 2.5

2.1 INTRODUCTION

In Chap. 1 we looked at simulation modeling in general, and then modeled and coded two specific systems. Those systems were very simple, and it was possible to program them directly in a general-purpose language, without using any special simulation software or support programs (other than a random-number generator). Most real-world systems, however, are quite complex, and coding them without supporting software can be a difficult and time-consuming task.

In this chapter we first discuss an activity that takes place in most simulations: list processing. A group of ANSI-standard C support functions, simlib, is then introduced, which takes care of some standard list-processing tasks as well as several other common simulation activities, such as processing the event list, accumulating statistics, generating random numbers and observations from a few distributions, as well as calculating and writing out results. We then use simlib in four example simulations, the first of which is just the single-server queueing system from Sec. 1.4 (included to illustrate the use of simlib on a familiar model); the last three examples are somewhat more complex.

Our purpose in this chapter is to illustrate how more complex systems can be modeled, and to show how list processing and the simlib utility functions can aid in their programming. Our intention in using a package such as simlib is purely pedagogical; it allows the reader to move quickly into modeling more complex systems and to appreciate how real simulation-software packages handle lists and other data. We do not mean to imply that simlib is as comprehensive or efficient as, or in any other way comparable to, the modern commercial simulation software discussed in