<u>SOLUTIONS FOR EXAM #1 FOR INTRODUCTION TO SIMULATION</u> ***Spring 1998***

Welcome to exam #1 in Introduction to Simulation. Read each problem carefully. There are eight required problems (each worth 12.5 points). Please write your answers on separate sheets of paper. Please put your name on all sheets of paper. You may have one 8.5 x 11 inch sheet of paper with you. On this sheet you may have anything you want (definitions, formulas, etc.) in handwriting or as photocopied text. **Good luck!!!**

<u>Problem #1</u>

a) What is a model, what is the goal in building a model, and why do we build models.

```
A model is a representation (physical, logical, or functional) that mimics
another object under study.  The goal in building a model is to be able to
predict the behavior of the original object.  Building and studying models
is usually cheaper and safer than studying the real object.  Sometimes it
is not possible to study the actual object and a model is thus required.
```

b) Why is probability theory of interest to the modeling of information systems (that is, why do we study probability?)?

```
Most events in information systems are characterized by randomness.
Probability theory is the "tool" needed to study systems characterized by
randomness.
```

c) Describe the three phases of a simulation study and their relationship with each other (a diagram may be a useful presentation tool).

```
  Design    <--------------+
    A                      |
    |                      |
    V                      V
Execution <--------> Output Analysis
```

<u>Problem #2</u>

You have a choice of buying System A or System B. System A consists of 12 components (what the components are does not really matter at this point) where each component has a 99% probability of being "up" on a given day. System B consists of only 4 components where each component has a 98% probability of being "up" on a given day. The state of any component ("up" or "down") is independent of the state of other components and of the state of any given component on any previous day. For the entire system to be "up", <u>all</u> components must be "up". Answer the following:

a) Which system, A or B, is the most reliable (i.e., has the lowest probability of being "down" on a given day)?

```
Pr[A = up on a given day] = (0.99)^12 = 0.8864
Pr[B = up on a given day] = (0.98)^4  = 0.9224   B is the most reliable
```

b) What is the probability of System A being "up" from Monday through Friday given that it was "down" on Saturday and Sunday?

```
Pr[A = up for 5 days in a row] = ((0.99)^12)^5) = 0.5472
```

c) What is the probability of System A being "up" from Monday through Friday given that it was "up" on Saturday and Sunday?

Since all probabilities are independent, the state of the machine on the weekend has no affect on the weekday state. Same answer as (b) (i.e., Pr[A = up for 5 days in a row] = 0.5472)

Problem #3

You have collected data on the number of hours per day employees in your corporation "surf the web". You have 200 employees. The data shows that during the work day 40 employees do not surf at all, 100 of the employees surf for 1 hour per day, 40 employees surf for 2 hours, and 20 employees surf for 3 hours. Let the random variable *X* equal the number of hours of surfing for an employee (i.e., *x* can take on the values 0, 1, 2, and 3 with non-zero probability) and let Pr[*X* = *x*] = the ratio of employees that surf for *x* hours.

a) What is the mean of *X*?

E[X] = (40/200)*0 + (100/200)*1 + (40/200)*2 + (20/200)*3 = 1.2 hours

b) What is the standard deviation of *X*?

E[X^2] = (40/200)*0^2 + (100/200)*1^2 + (40/200)*2^2 + (20/200)*3^2 + =

2.2 hours ^ 2

Var[X] = E[X^2] - E[X]^2 = 0.76 hours ^ 2

Std Dev[X] = sqrt(Var[X]) = 0.8718 hours

c) Plot the pmf (density or, more precisely, mass) function and the PDF (distribution) function for *X*.

```
f(x) |
     |
  0.5|       |
     |       |
     |       |
  0.2|   |   |   |
  0.1|   |   |   |   |
     +------------------------
         0   1   2   3   x= hrs
```

```
F(x)|
 1.0|              +------
    |           +--+
    |           |
    |        +--+
    |        |
    |        |
    |        |
    |    +---+
    |    |
    +---------------------------
         0   1   2   3   x = hrs
```

Problem #4

Answer the following questions:

a) What is the relationship between the Poisson distribution and the exponential distribution?

The time between arrivals in a Poisson process is exponentially distributed (i.e., interarrivals times are drawn from an exponential distribution).

b) Why is the Poisson distribution an "important distribution" (that is, why do we care about the Poisson distribution other than to get the right answer on the exam)?

```
The Poisson process occurs frequently in real life as the result of
multiple random events or streams coming together.  For example, time
between user input to a computer system is often exponentially
distributed and hence the input stream (of user requests) is then a
Poisson stream.  The Poisson process also has several "nice" properties
(e.g., combination of Poisson streams is Poisson, splitting of Poisson
streams is Poisson) that no other distribution has.  Finally, the
memoryless property of the exponential distribution enables relatively
simple analytical models to be built.
```

c) Assume that random variable X has a Poisson distribution with rate $\lambda = 3$. Solve for the $Pr[X > 2]$.

```
Pr[X > 2] = 1 - Pr[X = 0] - Pr[X = 1] - Pr[X = 2] =

  1 - ((3^0)/0!)*e^-3 - ((3^1)/1!)*e^-3 - ((3^2)/2!)*e^-3 = 0.5768
```

Problem #5

You have made measurements (shown below) on CPU time required for compute jobs arriving into a compute server. Write a C function (prototyped as `double cpu_time(void);`) that returns a simulated CPU time from an empirical distribution based on your measurements. Assume that the CPU time requirements for arriving jobs are independent (i.e., that a previous job's CPU time has no effect on the probability of the next jobs CPU time requirements). You may omit the header and other documentation in your C function, but any included documentation may help your grade if your code is wrong. Hint: Less than 10 lines of code ought to do it.

Required CPU Time	Number of jobs	Ratio
1 sec	800	0.5000
2	300	0.1875
3	200	0.1250
4	200	0.1250
5	100	0.0625

```c
double cpu_time(void)
{
  double z;
  z = rand() / RAND_MAX;
  if (z < 0.5000) return(1.0);
  if (z < 0.6875) return(2.0);
  if (z < 0.8125) return(3.0);
  if (z < 0.9375) return(4.0);
  return(5.0);
}
```

Problem #6

Assume that you can somehow watch file requests arriving into a file server and also observe the requested files being transmitted out on a network. You have a means of associating each file request with its resulting file "response". Assume that the rate of file requests is 100 request per second and that the mean time between a given request arrival and its associated file transmission is 0.5 seconds. What is the average number of queued-up file requests in the server?

```
Length = arrival rate * wait, so 100*0.5 = 50 queued-up requests (average)
```

Problem #7

Consider the following single-server queueing system from time $t = 0$ to $t = 10$ sec. Arrivals and service times are:

- Customer #1 arrives at $t = 1$ and requires 3 seconds of service time
- Customer #2 arrives at $t = 3$ and requires 4 seconds of service time
- Customer #3 arrives at $t = 4$ and requires 1 second of service time
- Customer #4 arrives at $t = 5$ and requests 3 seconds of service time

Solve for system throughput (X), total busy time (B), mean service time (T_s), utilization, (U), mean system time (W), and mean number in the system (L). You must show your work to receive credit!

```
Number in System |
                 |
               3 |                   +-----------+
                 |                   |           |
               2 |           +-------+           +---+
                 |           |                       |
               1 |   +-------+                       +--------
                 |   |
                 +----------------------------------------------
                     1   2   3   4   5   6   7   8   9   10  time

                   X  = C/T = 3/10 = 0.3333 cust/sec
                   B  = 9 seconds
                   Ts = B/C = 9/3 = 3 sec/cust
                   U  = B/T = 9/10 = 0.90 or 90%
                   W = Sum of w / C = 18 / 3 = 6 sec
                   L = Sum of w / T = 18 / 10 = 1.8 customers
```

Problem #8

Write a Monte-Carlo simulation to find an approximate value of π. Some hints… The area of a circle is πr^2 where r is the radius and the formula for a circle is $r^2 = x^2 + y^2$. It might be easiest to just consider part of a circle (e.g., the part for postive x and y) and then adjust the final result appropriately. Note that if $r = 1$, the area of the circle is simply π. You may omit the header and other documentation in your C function, but any included documentation may help your grade if your code is wrong. Less than 20 lines of code ought to do it.

```
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
  double   r1, r2;
  long int accept = 0;
  long int reject = 0;
  long int i;

  for (i=0; i<1000000; i++)
  {
    r1 = (double) rand() / RAND_MAX;
    r2 = (double) rand() / RAND_MAX;
    if ((r1*r1 + r2*r2) <= 1.0)
      accept++;
    else
      reject++;
  }

  printf("The value of pi is approximately %f \n",
    4*((double) accept / (accept + reject)));
}
```