>>> Assignment #3 for Simulation (CAP 4800) <<<
Due on 06/06/13 in class

This assignment covers material from the third week of class lecture.

Problem #1 (25 points)

A typical GET request to a web server requests an HTML page. This HTML page may have embedded images. Each embedded image results in another GET to the server. So, for example, if a given page has two embedded images then there will be two additional GET requests to the server for the images. Suppose that 937 web pages (HTML pages) were characterized for number of embedded images. The below table shows what was observed (that is, 129 pages have 0 embedded images, 247 pages have 1 embedded image, and so on).

Number of embedded images	Number of pages observed
0	129
1	247
2	112
3	332
4	117

Your task is to create a program that generates a simulated workload corresponding to the statistics in the above table. The workload is a series of integer values corresponding to the number of embedded images in a given request. Using your program, generate 1 million simulated requests and determine the mean. Compare the mean of the simulated workload to the expected theoretical mean (which you should calculate in your program) based on the data in the above table. Your program should output the pmf, CDF, and theoretical mean of your empirical distribution, and also the mean of generated requests. Submit your source code and a screen shot of the execution. **Hint:** You may take the printer workload program we discussed in class and modify it for this problem.

Problem #2 (25 points)

Here is a C function for a random number generator. Describe how to evaluate an RNG for "goodness". Evaluate the below RNG.

```
//===========================================================================
//= Superduper RNG for Simulation class (summer 2011)                       =
//===========================================================================
double rand_superduper(void)
{
  const long  a =      33312;
  const long  m = 2147483647;
  const long  q =     250001;
  const long  r =       1111;
  static long x = 1;
  long        x_new, x_div_q, x_mod_q;

  x_mod_q = x % q;
  x_div_q = x / q;
  x_new = a * x_mod_q;
  x_new = x_new - (r * x_div_q);
  if (x_new > 0) x = x_new;
  else x = ((-1) * x_new);

  return((double) x / m);
}
```

Problem #3 (25 points)

Consider a disk drive that transfers data in micro-blocks of 5 bytes. There is a probability of bit error, *p*, for each block transferred (that is, each bit has a probability of being in error of *p*). Bit errors are independent (that is, there is no correlation between bit errors). What is the probability of a block having 0, 1, 2, or 3 bit errors? The Monte Carlo simulation `blockError.c` (which can be found on the class source code page) models this system. Run the simulation for $p = 10^{-2}$. Submit a screenshot of the results. Also, analytically model this system and compare your analytical and simulation results.

Problem #4 (25 points)

Consider the same system of Problem #3. Modify `blockError.c` to model correlated bit errors as follows. If a given bit is not in error, then the probability that the next bit is in error is *p*. If a given bit is in error, then the probability that the next bit is in error is 0.5. Just the same as in Problem #3, for $p = 10^{-2}$ determine the probability of a block having 0, 1, 2, or 3 bit errors. This problem cannot (at least not easily) be modeled analytically. This is an example of a problem well suited for a Monte Carlo simulation model. Submit your source code and a screenshot of the results. Discuss your results – compare them to the results of Problem #3 and speculate on why the difference.