>>> Assignment #2 for Simulation (CIS 4930) <<<
Due on 05/30/13 in class

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

Problem #1

Answer the following probability theory questions:

a) Roll a pair of dies (dice). What is the probability that the sum of the dots is 6? What is the probability of getting a 1-1, 2-2, or 3-3?

b) A lottery is based on 50 balls each with a number on it (1, 2, …, 50) placed in a bucket and thoroughly mixed-up. Six balls are pulled from a bucket without replacement. What is the probability of getting 1-2-3-4-5-6? What is the probability of getting 47-23-7-13-2-36? Explain why the probabilities are the same or different (whichever the case may be). Note that in a lottery it does not matter what order the numbers appear in, just that the set of selected numbered are all "pulled".

c) Assume that the probability of a PC being "dead on arrival" (DOA) due to an independent manufacturing defect is 1 in 100. If you order 300 PCs for (say) your business, what is the probability that the first PC you unpack and install is DOA?

Problem #2

Design of reliable data transfer methods is a challenging problem. The simplest reliable data transfer method is to transfer all data blocks N times ($N > 1$) each with N "large enough" to insure successful data delivery with an acceptably high probability. Assume that the probability of a data block being corrupted is p, errors occur independently, and corrupted data blocks can always be detected at the receiver. Define a success rate, S, to be the probability that a given data transfer is successful (that is, that at least one of the N blocks transferred is not corrupt, or bad, at the receiver).

a) If p is 0.05, what is the minimum N for S to be no less than 99.99%, 99.999%, and 99.9999%? Note that N must be an integer! **Hint:** What is the probability that all N blocks transferred are bad? Knowing this you can easily find the probability of one or more blocks transferred being good (not bad).

b) For a given S and p, solve for N. That is, find a general expression for N in terms of S and p.

c) A Monte Carlo simulation of this problem is `as3_2_1.c` (available on the class source code page for download here: `http://www.csee.usf.edu/~christen/class3/source3.html`). Run this simulation (you will need to change the parameter values in the program – you cannot just compile and run it) and see if the answers you get match with what you computed for (a). Show your run time results. How do the results from this simulation vary as a function of NUM_ITER. Why?

Problem #3

A Monte Carlo simulation of the dice roll problem we discussed in class (that is, the probability of rolling a pair) is `diceSim2.c` (available here: `http://www.csee.usf.edu/~christen/class3/source3.html`). Run it can convince yourself that the probability of rolling a pair is 1/6. Modify this program to model three die (say, red, blue, and green) and the event of rolling a triple (so, 1-1-1, 2-2-2, …, 6-6-6) and determine the probability of this event occurring. You are now writing (albeit simple) simulations!

Problem #4

Assume you flip a fair coin 4 times – call this a trial. You can have 0, 1, 2, 3, or 4 heads appear in each trial. What is the expected (mean) number of heads that will appear given an infinite number of trials? What is the variance of the number of heads that will appear? Plot the probability mass function for the number of heads that appear.

Problem #5

Derive the expression for variance of the continuous uniform probability distribution (we derived the expression for mean in class). Show your work. Using `genunifc.c` from the Christensen tools page generate 1 million samples with $a = 1.0$ and $b = 10.0$. Find the mean and variance of the samples (that is, the experimental mean and variance). Compute the relative error of the experimental mean and variance compared to the theory mean and variance (that is, as calculated from the derived expressions)? **Hint:** the program `summary1.c` from the Christensen tools page can be used to compute summary statistics for a series of values (be sure to read the header in the program to learn how to use it!).