**ECE 1111: Engineering Computation I**

**Homework No. 8: Loops and Conditionals**

**Goal:** Looping is a major part of any programming language. In this homework assignment, you will learn how to control program flow using for and while loops. Note that you have already done this quite a few times in previous assignments.

**Description:** There are three tasks in this homework assignment:

1. **For Loop:** Create an array of floats in memory that is 1M elements long. Initialize it with randomly generated values using the drand48() random number generator. Start by doing a simple loop from *[0, N-1]* where *N = 1M*, and use an increment of *1*. Next, change the single loop to a double loop where you increment the outer loop by *M* and the inner loop by *1*. *N* and *M* should be set from the command line using argv[1] and arg[2] respectively. Your loop should work for any combination of the integers *N* and *M*, even if *N* is not evenly divisible by *M*.
2. **While Loop:** Using the same float array, increment from the last sample to the first sample using a while loop. For each iteration of the loop, generate a random integer (you can use drand48() and round it to an integer to make this easy), initialize that number of elements, and decrement your counter by that number of elements. Pseudo-code for this would look something like this:

**i = N;**

**while (i > 0) {**

* **Generate a random value of M**
* **Initialize array values from [i,i-M+1]**
* **Decrement your counter by M**

**}**

Make sure you initialize all *N* elements and do not initialize each element more than once. *N* should be set from the command line using argv[1].

1. **Character Pointers:** Initialize a character string of *N* elements to ASCII values. Copy this string to another string *M* elements at a time. But this time only use character pointers – do not use integers. You can use the integer *M*, but cannot use the integer *N* in your loops. Do the copies manually with a loop. Do not use functions like memcpy or strcpy. *N* and *M* should be set from the command line using argv[1] and arg[2] respectively.

For no. 3, you will be graded on the quality of your code. The more extensive use of pointers you make, the better.

Submit your well-commented and well-documented code as a pdf file in which you follow the homework template and explain how your code works. You will be graded on the thoroughness of your explanations and the detailed way in which you test your code.