**ECE 1111: Engineering Computation I**

**Homework No. 10: Command Line Arguments and  
Dynamic Memory Allocation**

**Goal:** The data structure used to store command line arguments, *argv*, is one of the more interesting routine data structures you will be see in C. This assignment teaches you the basics of manipulating structures like this and how important it is to be careful about memory management.

**Description:** This assignment is deceptively simple: write a C program that creates a copy of argv in memory. Your program should:

* Implement a function to print the contents of argv to stdout;
* Create a function that allocates space for a complete copy of argv (not just the pointers, but space for the actual strings stored in argv);
* Copy argv to this structure, which we will call *argv1*;
* Create a function that deallocates space for this same structure;
* Create a second copy of argv, called *argv2*;
* Copy the contents of *argv1* to *argv2*;
* Call your deallocation method to free space for *argv1* (and hopefully not corrupt *argv2* in the process);
* Print the contents of *argv2* using your print function (demonstrating it is not corrupt);
* Deallocate *argv2*;
* Demonstrate that there are no memory leaks in your code.

You must use make files to implement this task. You must demonstrate that you can compile and link your programs using make files.

Submit your well-commented and well-documented code by creating a directory using your last name – all lowercase – here:

/data/courses/ece\_1111/current/homework/hw\_10

Be sure to change the protection on your directory so only you can view it (e.g., “rwx--------”).

**You will be graded on both the functionality of your code and the degree to which it is documented.**