EE 2372 Software Design I
Assignment #6: Functions Programs
Due Date: Monday December 4 at 11:59 PM

Objectives:

1) Learn to read and use command line arguments
2) Learn to dynamically allocate memory using malloc
3) Learn to read binary data from a file

Description

In this programming assignment you will write a program that reads binary data from a file. The first value in the file is an integer that indicates the number of floating point values remaining in the file. The remaining data is the actual floating point values. Your program must do the following:

1. The name of the input file is to be passed into the program as a command line argument.
2. The program will open this binary file, and read the first integer in the file. It will then dynamically create an array of floats of this size using the malloc function.
3. The program will then read the floating point values and store them into this newly created array.
4. Finally, the program will print out the floating point values in this array in reverse order, one value per line.

Files data1 and data2 are now on the course web site. The file data1 contains the following:
4, 1.1, 2.2, 3.3, 4.4

The output from your program ($./a.out data1 or $binread data1) should give

```
4.400000
3.300000
2.200000
1.100000
```

Deliverables:

1) The output produced by file data2 **as an attachment**. Name the file data2out.txt
2) The C source code **as an attachment**. Name the file binread.c
3) Mail the requested files to our TA Pavithra Pochamreddy at ppochamredd@miners.utep.edu.
4) The subject line must be “EE 2372 Assignment 6” (don't include the quotes).

Scoring:

Your grade for this assignment will be determined by three criteria. The first criterion determines if your program compiles and runs producing the correct result. The correct result must adhere to what is specified in the **Tasks** section. The second criterion is whether the program follows the interface specification outlined in the **Tasks** section. The third criterion determines if your source code is well documented. Your source code must include (at the top) your name, class section, due date, assigned date, and a small description of your program. For this assignment, each line of code should have a descriptive comment.

Operation/Successful Demonstration	**60%**
Was a C source file submitted? 10%	
Does the program compile on Linux? 15%	
Does the program run correctly on Linux? 35%	
Adherence to Interface Specification	**30%**
Does the program adhere to the interface specification in the tasks section? 30%	
Comments	**10%**
Is the source code well-documented? 10%	
Lateness	**10%** per day (including weekends and holidays)