**CS2750 System programming and Tools Fall 2019**

**Project #5 [50 points]**

---------------------------------------------------------------------------------------------------------

Due date is 11/19/2019.

**Task:**

In this project you are being asked to write a program that will read in a set of numbers, format it in a particular way, and then output the result.

The command line arguments that your program should take are -h and -n. -h should output a help message indicating exactly what the program is expecting. The next argument, -n, will specify the minimum number that your script file can take in, to be explained later. Note that these arguments (-h and -n X) should be handled using getopt (example of the use of getopt is on delmar at /home/hauschildm/scripts/C/stringsio/opt.c )

An example of running this project would be, if your executable was called numpalin,

./numpalin -n 5 57 17 8 3 57 19 57

This would indicate that you were running your program with an argument of -n 5, and arguments of 57, 17, 8, 3, 19 and 57.

If the -h option is given, your program should just terminate with the help message indicating how it should run.

Now your program should then read in all of those numbers into an array. However, when doing so you should ensure that you ignore all numbers that are less than the argument to -n. In addition, you should ignore duplicates after the first (that is, only allow one duplicate, so if a list could have two 5’s in it but not 3 5’s). So given the above example, the numbers you would read in would be 5 57 17 8 57 19, as we would ignore anything below 5 and would ignore the third duplicate.

After storing these in an array, you must then pass this array to a function, which should dynamically allocate a new array inside and copy the old array to it, then return that array out through the return statement. So if we passed the array of [5 57 17 8 19] to it, then the function would dynamically allocate an array of this size and return a pointer to main with this array storing [19 8 17 57 5].

In your main body you should then compare the new array to your old array and report if it is the same backwards and forwards. Output both arrays element by element and then notify the user if it is the same in reverse.

**Details:**

You must implement this project using at least 3 source files, numpalin.c, functions.h, functions.c, You also must have a makefile that will compile the project for me. If you do not include a makefile your project will not be run. Submit the 3 source files and your makefile.