CS 135
Exercise #11
Point value: 40
Date due: email your source file (.cpp file) to your lab instructor by 11:59pm Sunday, April 23

New Skills Practiced (Learning Goals)

- Problem solving and debugging.
- User-defined functions.
- C++ strings.
- Arrays.
- Sorting.
- C++ filestreams.

Design a C++ program using functions (as described below) that will

- interactively prompt the user for and read the name of an input file that contains a maximum of 25 floating point numbers.
- open the input file
- read the numbers and store them in an array, counting as they are read (see get_data function in lecture notes)
- sort them into ascending order
- interactively prompt the user for the name of an output file and read it
- open the output file
- write the following to the output file
    - your name, lecture and lab section #s and exercise #
    - a list of the numbers (with 1 digit to the right of the decimal) in ascending order (1 per line) - label the list
    - the average of the numbers (with label and 3 digits to the right of the decimal)
    - the median of the numbers (with label and 3 digits to the right of the decimal)
- close all files

To receive full credit for this exercise,

- the program must make use of functions and pass parameters (**no global variables. no goto statements**)
- the input file can only be opened and read **ONE** time
- there must be a function to sort the numbers (bubblesort recommended)
- there must be a function to compute the average (void or value-returning)
- there must be a function to compute the median (void or value-returning)
- any additional functions are optional
- an array **MUST** be used to store the numbers from the file

NOTES:

- The median is the "middle" value in a set of sorted values. If there are an even number of values in the set, the median is the average of the 2 "middle" values.
- Assumptions about input: (you do not have to test for these conditions)
    - data file will exist, it will not be empty
    - maximum number of values in the file will be 25
    - numbers will be separated by blanks and/or linefeeds
    - the last line in the data file will be terminated with a linefeed ('\n')
- Program must be designed to read and write to filestream variables.
- Include all header files for library functions used by the program.

When the program compiles and runs correctly, use the mail utility to email a copy of the program file to your lab instructor. Make sure the subject line of your email includes your name, lecture and lab section #s, and the exercise # if you wish to receive full credit.

NOTES:

- Make sure you choose enough test data to ensure that your program meets all the requirements.
- It is a good idea to send a carbon copy to yourself (-c option) of all emails sent to your lab or course instructor when using the mail utility.
- A comment with your name, lecture section#, lab section#, and exercise# should be at the start of your program file.

Sample terminal session:

```
[lee@bobby keys]$ more data4eleven
3.0 7.0 11.5  16.2
15.3 22.8
[lee@bobby keys]$ g++ exercise11.cpp
[lee@bobby keys]$ ./a.out
Enter name of input file
data4eleven
Enter name of output file
output4eleven
[lee@bobby keys]$ more output4eleven
Lee Misch LecSec#10__ LabSec#10__ Exercise #11

List of Numbers
3.0
7.0
11.5
15.3
16.0
22.8

Average = 12.633
Median = 13.400
```

Return to exercises list