EECE/CS 115 Programming Assignment 4 (15 points)

Goals

1. Learn how to read from a file.
2. Learn how to use arrays.
3. Learn how to implement Bubble Sort.
4. More practice with functions, loops, if statements, printing to screen.

Assignment

In this assignment, you will be implementing bubble sort. You will read 10 integers from a file, display those 10 integers, unsorted, then sort ascending (low to high) and display that array and finally you will sort the array descending (high to low) and display that array.

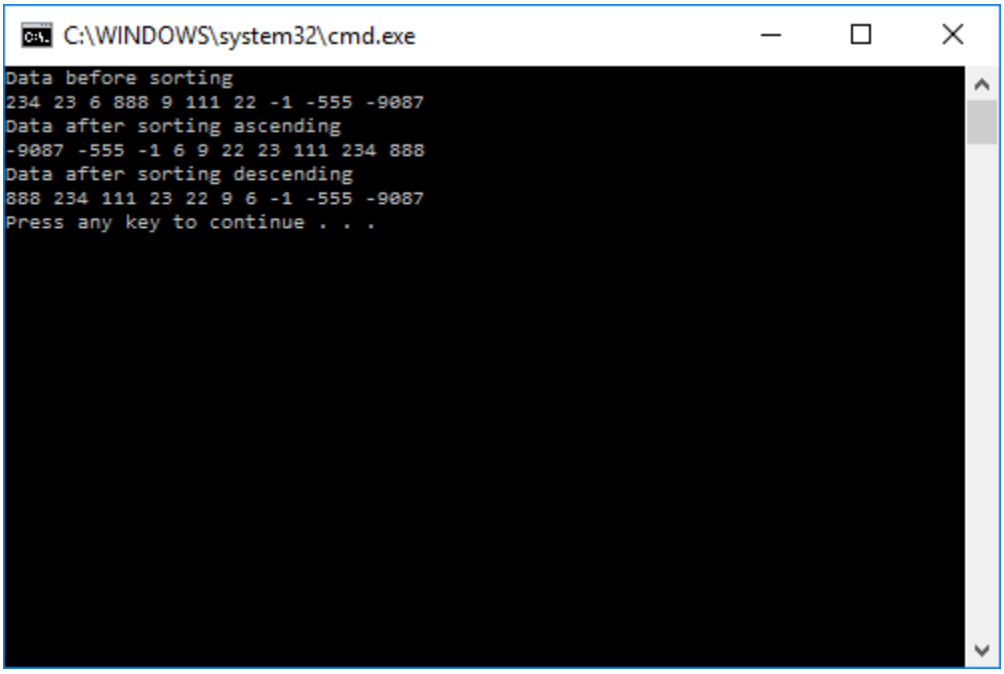
Requirements

1. Create cpp file bubbleSort.cpp.
2. Create main function.
3. Read "data.txt" file – included in zip file.
4. You can assume there are 10 integers in the file (1 in each line).
5. Read all 10 integers from file into array of size 10 (1 integer in each index of array).
6. Print array to screen (see PrintArray function).
7. Sort array ascending (low to high).
8. Print sorted array (PrintArray).
9. Sort array descending (high to low).
10. Print sorted array (PrintArray).
11. **Functions to create**
    a. *void SortAscending(int data[], int count);*
        i. data[] is the array of integers
        ii. count is the size of the array (10 in this case).
        iii. Implement bubble sort to sort low to high.
        iv. You can do this with a double for loop
            1. Outer loop goes until count
            2. Inner loop goes until count-1
        v. Create a temporary variable (int tmp).
        vi. In the interior loop, check if current index of array is greater than next index of array
            1. Set temp variable equal to current index of array.
            2. Set current index of array equal to next index.
            3. Set next index of array to temporary variable.
    b. *void PrintArray(int data[], int count);*
        i. Display all 10 integers in data. This is the only function that should print anything.

ii. Do not print anything in any of the other functions.
iii. This function should only be called from main.

c. *void SortDescending(int data[], int count);*
   i. This function is identical to SortAscending, except for step *f*. Instead of checking to see if current index of array is greater than next, check to see if it is less, then implement the next step.

12. **SEE IMAGE BELOW FOR WHAT OUTPUT SHOULD LOOK LIKE**
13. **Submit the CPP file ONLY (named bubbleSort.cpp) to blackboard. DO NOT ZIP IT.**



Grading (16 points total for +1 extra credit):

- 1 point: Correct submission format
- 4 points: Read from file correctly
- 4 points: Correct SortAscending
- 4 points: Correct SortDescending
- 2 points: Correct PrintArray
- 1 point: Correctly calling functions in main only