

Project Three

Due: Sunday 10/05 by 11:59pm.

Objective:

1. Become familiar with the use of arrays in C++.
2. Develop familiarity of using arrays as part of function arguments.
3. Gain familiarity of using command line arguments.

Description:

Write a program that uses Bubble Sort to sort integers in an array in ascending order.

Background:

The basic idea is that larger numbers "sink" to the bottom (end) of the array while smaller numbers "float" towards the top (beginning) of the array gradually. At each step two neighboring array elements are compared and, if required, swapped to be arranged in the correct order. This step of comparison and swapping continues while any pair of elements can be swapped. When no more swaps can be performed, the array contains its elements in sorted order from smallest to largest.

Input:

Since the concept of file I/O has not been introduced yet, all data can be obtained by the way of the stdin prompt in an interactive fashion. You can then feed the data to your program from a file by using file redirection. The input file will contain the matrix of numbers.

Output:

The final result should be the sorted array.

Specific Requirements:

1. Your program should contain at least one array.
2. Your program should contain a main.cpp as the main entry to the program execution.
3. Your program should have a library file called myArray (myArray.h and myArray.cpp)
4. Your your library (myArray) should contain at least the following three functions:
 1. ReadData(int [], int size): should read in the values from the user and store them in a 2 dimensional array.
 2. BubbleSort(int [], int size): should receive an unsorted 2 dimensional array of integers and sort them via Bubble Sort.
 3. PrintData(int [], int size): should receive the sorted array and print it to the console in the format shown in the samples.

4. ReverseData(int [], int size): should reverse the order of the array elements.
5. Your program must be prepared to work for any array size. The size of the array needs to be passed to the program via the command line argument. The example below is for an array of size 5 and the command line argument would consist of `"/a.out 5"`
6. You can use the sample input file to test.

1.

Sample:

Input:

```
21 22 23 24 25
11 12 13 14 15
06 07 08 09 10
01 02 03 04 05
16 17 18 19 20
```

Output:

```
01 02 03 04 05
06 07 08 09 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```