**Scoring Rubric for Project 3 : BubbleSort**

*Due 10/03/2019 @ 3:30 pm*

|  |
| --- |
| Student Name: Nathan Maynard |

|  |  |  |
| --- | --- | --- |
|  | **Score** | **Maximum** |
| **Execution (50 pts):** | | |
| Program compiles without errors (warnings are okay) | 50 | **50** |
| **Implementation (40 pts):** | | |
| Uses function declarations as provided | 5 | **5** |
| Main function includes at least one unit test for Swap (can use assert or printed output) | 5 | **5** |
| BubbleSort works for input size of 42 and 47 (all or nothing) | 5 | **5** |
| Use a dynamically allocated array for BubbleSort | 5 | **5** |
| Free the allocated array at the end of Main function | 2.5 | **5** |
| Complete the BubbleSort unit test | 0 | **5** |
| Use command line arguments to read the array size and the seed | 5 | **5** |
| Measure the execution times of MergeSort and BubbleSort and plot them on a graph | 5 | **5** |
| **Style (10 pts):** | | |
| The driver and functions are easy to follow based on the use of comments | 6 | **6** |
| Easily identifiable variable names | 4 | **4** |
| **Total (100 pts):** | 92.5 | **100** |

Notes:

Would be nice to have both the times for merge sort and bubble sort plotted on the same graph so it’s easy to compare how they perform.

You need to add array = nullptr; to finish deallocating the array. This will ensure that array will not be left as a dangling pointer.

You need to check that the output of bubblesort is correct similar to the way it is checked with mergesort.

Technically, the second command line argument should be the length of the array, not how many times to iterate. I didn’t take any points off since it’s clear you did that so you can easily compute the timings.