**Scoring Rubric for Project 2 : Mergesort**

*Due 9/19/2019 @ 5 pm*

|  |
| --- |
| Student Name: Lauren Childers |

|  |  |  |
| --- | --- | --- |
|  | **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 MergeSortedLists (can use assert or printed output) | 2.5 | **5** |
| MergeSortedLists works for both even and odd sized input | 5 | **5** |
| MergeSortedLists works for both left and right lists emptying first | 5 | **5** |
| Implements MergeSort base case correctly (may be implicit) | 5 | **5** |
| MergeSort recursive case follows pseudocode: two recursive calls followed by call to MergeSortedLists (may have indexing bugs) | 5 | **5** |
| MergeSort passes 5 different unit tests (2 points each) | 10 | **10** |
| **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):** | 97.5 | **100** |

Notes:

Your unit test for MergeSortedLists is incorrect. You are storing the result in a vector called “test” but you assert on “v” which is still all 0’s. Also, the purpose of the function is to merge together two lists that are already sorted. So, the left half and right half of “test” should be sorted already, independently of each other.