# CST8130: Data Structures Lab #3- Sorting and Searching

***DUE: demonstrate in lab during week February 5– February 11 (or earlier)***

## Problem Description:

Add to your solution for Lab 1 and Lab 2:

1. add a menu option (#7) to sort the array of numbers. You should implement **Insertion Sort** (per lecture notes).
2. add another menu option (8) to display the count of how many float values in the array are contained in a range of values. So this option should prompt the user to enter two values (say lower value and higher value) – then should efficiently search through the array counting how many values in the array are in this range and display that count. (ie how many numbers are > low limit and also < high limit).

***Lab 3 Test Plan:***

**Menu Testing**

|  |  |  |
| --- | --- | --- |
| Condition | Input | Result |
| Menu input of 7 first | 7 | Array should be sorted (see testing below)– then back to menu |
| Menu input of 4 | 4 | Values should print 0.0 0.0 0.0 |
| Menu input of 8 | 8 | Error Message – “cannot complete this option without sorting array first” |
| Menu input of 2 | 2 (then enter 3 for size of new array) | Input accepted, back to menu |
| Menu input of 3 | 3 (then values -1 -3 -2) | User should be prompted to enter 3 numbers |
| Menu input of 5 | 5 | Average of -2.0 should be displayed |
| Menu input of 7 | 7 | Array should be sorted – then back to menu |
| Menu input of 4 | 4 | Values should print -3.0 -2.0 -1.0 |
| Menu input of 8 | User should be prompted to enter two float values | Count of how many values are in the entered range is displayed (see testing below) |
| Invalid selection | 10, a, -2 | Error message |
| Menu input of 9 | 9 | Program ends |

**Sorting Testing**

|  |  |  |
| --- | --- | --- |
| Condition | Input | Result |
| Values are in order | 1 2 3 | Sorted should be 1.0 2.0 3.0 |
| Values are backwards | 3 2 1 | Sorted should be 1.0 2.0 3.0 |
| Values are mixed up | 3 1 2 | Sorted should be 1.0 2.0 3.0 |
| Some values are same | 1 2 1 | Sorted should be 1.0 1.0 2.0 |
| Positive and negative and zero values | 1 -1 0 | Sorted should be -1.0 0.0 1.0 |

**Search/count Testing**

|  |  |  |
| --- | --- | --- |
| Condition | Input | Result |
| User enters good values | Low: 1.1 high: 2.2 and numbers are 1, 2, 3 | Count displays as 1 |
| User enters good values – includes endpoints | Low: 1.0 high: 3.0 and numbers are 1, 2, 3 | Count displays as 1 \*\*note – testing for equality with floats can be problematic – this option should test > low and < high |
| User enters good numbers – no values in range | Low: -1 high: 0 and numbers are 1, 2, 3 | Count displays as 0 |
| User enters good numbers – no values in range | Low: 4 high: 5 and numbers are 1, 2, 3 | Count displays as 0 |
| Low value is higher than high value | Low: -1 high: -4 | Error message – high must be larger than low…reenter |
| Low or high value is invalid | A for either | Error message |
| Positive and negative and zero values | 1 -1 0 | Sorted should be -1.0 0.0 1.0 |