CS – 121

Colby Blair

Due Date: May 2, 2008

Production Template

Lab/Assignment: Assignment 08

I. Design Time: 1 hour

a. Rough pseudo code:

Prompt user for the data file

Do a mock input from the file to count the amount of data elements coming in, calculating the array size.

Reset the input point to the beginning of the data file, clear input errors, and read from the file and copy to all data structures needed until the end of the file.

Insertion sort –

Find the lowest value in the array, copy it to a temp array, and continue through both until all remaining values have been found

If a min exists smaller than something in the new temp array, find where it goes in the temp array, and shift everything left from that point. Then insert the value.

Finally, copy the temp array back into the original.

Selection sort –

Find the min value, and switch it with the remaining unchecked values starting at the front of the array, until all values have been checked.

Mergesort –

Divide the array in half until it is parts of one. Then recombine, comparing and sorting each subpart, until it is put together again.

Quicksort –

Break the array into halfs using pivot points, and switch the values compared if they are out of order.

b. Estimated time of completion: 6 hours

II. Program Source

III. Output

IV. Program Log

a. Design time: 1 hours
b. Implementation time: 6.5 hours
c. Conclusions/Reflections:

Merge sort for some reason was easy to conceptualize, but really hard to think in terms of code. Recursion definitely got the best of me here, and I will most likely go back and remake this code to make it cleaner. Stat taking was also a bit messy, I would like to greatly redesign the program for simplicity. I was happy with the data input functions and the first 2 sorting methods, but the second 2 were not clear to me.