Nicholas Skinner

CS260

Merge Sort

Design:

Most of the design plans came from the video shown in class, where we take an inital array and divide it up in half over and over. After that, we will take two elements at a time, compare them, and determine what one needs to be in what position, after that it will be on a larger scale, incorperating two different sections so we have 4 total values to sort, and so on to higher up parts of the array.

Testing:

Test plans this time around included changing values that were in the array, number of values in the array, and duplicates. I did not include any inputs from the user, so I do not need to use input validation.

Normal Run:

All good, prints out important parts of the sort.

more values:

Seems to work, but only does 3 tiers of sorting (compare 2, compare 4, compare all)

Duplicates:

correctly sorts into the correct position.

Reflection:

This was a fun program to make, I'm not too familar with sort methods, but it was fun to listen to the methods from the video and play it by ear on what I need to do to ensure that I get the program to where it needs to be. Most of these other programs I've had to look up external assistance, but this one was straightforward enough to implement that I could do it in an afternoon. The only regret I have from my designs is that I didnt make it generic, it depends on the inital array, I feel like I could have done some cheeky tricks to make it dependent on one (slightly longer) for loop, and be agnostic of the array size.