Group 6 Dean Pektas

**Question 5 Summary**

I began by including some necessary libraries at the top of my project file. I initialized variables that I knew would be very important to have within my sorting, counter and remaining integer variables. Besides the main function, I created some prerequisite functions that would also be important to have within my sorting functions, which were the swap and printArr functions. The swap array is designed to take in the addresses of two integer variables, store the address of the first parameter into a temp variable, set the address of the second variable into the first and then inserting what’s stored in temp back into the second variable. Afterwards, I developed the main function so that I could establish how the instructions would be sent to the sorting functions. I hard coded an array that, in the case of what’s currently saved in the project file, has 14 entries of random numbers. I also initialized a variable n that would take in the size of the array as well. I made it so that every time something important needs to be displayed to the user, I’d print it. That means that I wanted to print the unsorted array first. Then, I created a function called decide that would determine if it should be passed onto sort X or sort Y based on the length of the array. Given that it has more than 10 entries, the decide function will pass in the array and the length n of the array to the sortone function, which is based off of selection sort. I made print statements within this function letting the user know where the decide function has determined to move to. I also set the variable counter (initialized in the very beginning) to n so that as the selection sort finishes comparing a specific pivot number to the rest of the numbers in the array, it will decrease the counter variable by 1. The selection sort keeps track of this with an if statement, so that if it reaches below 10 elements, it now knows that the program must break out of the selection sort and pass the incompletely sorted array to the sorttwo function. Within the beginning of the first sort, the first element of the array is set as the minimum number, and as the nested for loop compares each number to the first, if it finds a smaller number it will set that number as the minimum instead. This will continue until it has reached the end of the array, at which case it will perform the swap between the final set minimum value and the pivoting element. Once the program passes the array to sorttwo, the function will perform a check to see if it has been checked by the first sort function, and if so it will set the remaining variable (also initialized in the very beginning) to the value of counter subtracted from the length of the array. This second sort function is based off of bubble sort and shares some similarities with the first sort function in that it uses nested for loops and also does a comparison within the second for loop as well as a swap based on that condition. Finally, once that is all over the program will return to the main function and print the newly sorted array.