Searching and Sorting Project

Name: _____

Period: _____PC #: _____

1. ____ / 10

Pull up the youtube video "XoaX Search". Watch and take notes. Use the *pseudocode* in the video to finish the programs `LinearSearch.java` and `BinarySearch.java`.

2. ____ / 5

Pull up the youtube video "XoaX Selection Sort". Watch and take notes. Use the pseudocode from the teacher to finish the program `SelectionSort.java`.

3. ____ / 5

Pull up the youtube video "XoaX Insertion Sort". Watch and take notes. Use the pseudocode in the video to finish the programs `InsertionSort.java`.

4. ____ / 5

Write a program `DogNames.java` that reads in a list of dog names and outputs them sorted by alpha. You can assume that the max number of names you are going to read in is 200 and the last line in the input is [END]. To avoid a `java.lang.NullPointerException` set all of the values in the array to an empty string [""] before reading in any input. Use the file *dog_names.txt* for input.

5. ____ / 5

Write a program `Merging.java` that reads in two separate arrays of sorted positive integers and adds them to a new array so that it is also sorted. This program will read in how many integers are in the first array and then read in its values. This process will be repeated for the second array. Here is an example

```
How many integers for the first array: 5
Enter the first Array: 3 5 8 11 20

How many integers for the second array: 7
Enter the second Array: 1 4 7 10 12 18 27

The new sorted array:
1 3 4 5 7 8 10 11 12 18 20 27
```

To get full credit, you must sort the numbers as you enter them into the new array. You cannot use a known sorting algorithm.

4. ____ / 5

Below is two listings of the `Selection Sort` method. These methods are currently the same, and they sort the array from least to greatest. Modify each method two different ways so that they sort the integers from greatest to least.

```
public static void selectionSort(int[] elements)
{
   for(int j=0; j<elements.length-1; j++)
   {
      int minIndex=j;

      for(int k=j+1; k<elements.length; k++)
      {
         if(elements[k]<elements[minIndex])
         {
            minIndex=k;
         }
      }

      int temp = elements[j];
      elements[j] = elements[minIndex];
      elements[minIndex] = temp;
   }
}
```

```
public static void selectionSort(int[] elements)
{
   for(int j=0; j<elements.length-1; j++)
   {
      int minIndex=j;

      for(int k=j+1; k<elements.length; k++)
      {
         if(elements[k]<elements[minIndex])
         {
            minIndex=k;
         }
      }

      int temp = elements[j];
      elements[j] = elements[minIndex];
      elements[minIndex] = temp;
   }
}
```