CS3350 Assignment 2 Due: Sunday, Feb 19 11:59 p.m.

You are given a Password.txt, which lists the most used 10k passwords in the world. Your goal is to sort the passwords in the file in an ascending order (characters are compared by the ASCII codes) by using various sorting algorithms and make a comparison of the algorithms.

**Your Tasks:**

1. Write a single assignment2\_yourLastname. cpp file that includes **A**. a main function to call the sorting algorithms to sort the Password.txt, **B**. the implementation of all the following sorting algorithms (some of them are already in Zybook and you can re-use the codes).
2. Bubble sort
3. Selection sort (Select the smallest in each iteration)
4. Insertion sort
5. Merge sort
6. Quick Sort

For quick sort, you must use the **median-of-three pivot selection** introduced in class: you need to sort the first, middle and last elements and then use the middle element as the pivot.

1. Set two counters for each sorting algorithm.
2. Counter 1 is used to record the **number of comparisons**. Whenever you have a comparison of two passwords during the sorting process, you need to increase one for this counter. Print the counter value after the sorting is done.
3. Counter 2 is used to record the **number of movements.** Whenever you move one password to a different location (a swap typically costs 3 movements), you need to increase one for this counter. Print the counter value after the sorting is done.

The output sample is as follows. Your program will be outputting the number of comparisons and movements that replace “XXX” in this sample.

Text

Description automatically generated

**Zip & Submit**

**Submit the Assignment2\_** **yourLastname.cpp** to the Assignment2 link on Canvas.

**Grading Rubric**

\_\_\_\_\_\_\_ (3) proper documentation/comments where appropriate

\_\_\_\_\_\_\_ (3) implementation of Bubble sort

\_\_\_\_\_\_\_ (3) implementation of Selection sort

\_\_\_\_\_\_\_ (3) implementation of Insertion sort

\_\_\_\_\_\_ (4) implementation of Merge sort

\_\_\_\_\_\_\_ (4) implementation of Quicksort

**The number of comparisons and number of movements must be the correct values in order to get full marks for each sorting algorithm. Each value is 1 pt.**