Data Structures and Algorithms

Lab 07 – Simple Sorting

Exercises/Tasks:

1. Modify the bubble, selection, and insertion sort algorithms and count (and print) the **number of comparisons** and **number of swaps** in each sorting algorithm to see which one has fewer comparisons and swaps.
   Also, try to sort another array with initial values
   a. in descending order and
   b. almost sorted (in ascending order)
   Then, run the above three sorting algorithms and again see the number of comparisons and swaps.
2. Given two arrays with random values, merge them while also sorting them. The result should be stored in the third array. Print the initial two arrays and the third (merged and sorted) array at the end.
3. Given an array of **Student** objects (each containing name, CMS-ID, and GPA), sort the array according to GPA in descending order. Print details (name, CMS-ID, and GPA) before and after sorting in the main method.
4. Sort a linked list using the insertion sort algorithm. Print the list before and after sorting in the main method.
5. A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

   A sentence can be shuffled by appending the **1-indexed word position** to each word and then rearranging the words in the sentence.

   For example, the sentence "**This is a sentence**" can be shuffled as "**sentence4 a3 is2 This1**" or "**is2 sentence4 This1 a3**".

   Given a **shuffled sentence** "**s"** containing no more than 9 words, reconstruct and return *the* ***original sentence***.

   Hint: Sort the words in "**s**" to their original positions "This1 is2 a3 sentence4", then remove the numbers.

   Example 1:
   Input s = "is2 sentence4 This1 a3"

Output: "This is a sentence"

Example 2:
Input s = "Myself2 Me1 I4 and3"
Output: "Me Myself and I"

6. Given an array **nums** with "**n**" objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

   We will use the integers 0, 1, and 2 to represent the colors red, white, and blue, respectively.

   You must solve this problem without using the library's sort function.
7. You are given an array of numbers where half of the numbers are odd and half are even. Your goal is to rearrange the array so that:
   - Every odd number is placed at an odd index (1st, 3rd, 5th, etc.).
   - Every even number is placed at an even index (0th, 2nd, 4th, etc.).

   The output should be any arrangement that satisfies this rule.

   Example 1:
   Input: [1, 2, 3, 4]
   Output: [2, 1, 4, 3]

   Example 2:
   Input: [4, 2, 5, 7]
   Output: [4, 5, 2, 7]

8. Given a string "**s**", sort it in decreasing order based on the frequency of the characters. The frequency of a character is the number of times it appears in the string.

   Return the sorted string. If there are multiple answers, return any of them.

   Example 1:
   Input: s = "tree"
   Output: "eert"
   Explanation: 'e' appears twice while 'r' and 't' both appear once.

So 'e' must appear before both 'r' and 't'. Therefore "eetr" is also a valid answer.

Example 2:
Input: s = "cccaaa"
Output: "aaaccc"
Explanation: Both 'c' and 'a' appear three times, so both "cccaaa" and "aaaccc" are valid answers.
Note that "cacaca" is incorrect, as the same characters must be together.

Example 3:
Input: s = "Aabb"
Output: "bbAa"
Explanation: "bbaA" is also a valid answer, but "Aabb" is incorrect.
Note that 'A' and 'a' are treated as two different characters.

9. You are given the heads of two sorted linked lists **list1** and **list2**.

   Merge the two lists into one sorted list. The list should be made by joining together the nodes of the first two lists.

   Return the head of the merged linked list.

   Example