Lab 04

Topic

Sorting

Objective

To implement and analyze Selection Sort and Bubble Sort algorithms by counting comparisons and swaps, comparing their behavior on different input cases, and finally designing a hybrid sorting algorithm.

Task (1):

Extend your program for Selection Sort to **count and display**:

- Total number of **comparisons** made.
- Total number of **swaps** performed.

Task (2):

Run Selection Sort and Bubble Sort on:

1. A **sorted array**
2. A **reverse-sorted array**
3. A **random array**

Compare and display the **number of comparisons and swaps** for each case.

Task (3):

Modify **Selection Sort so that**

- After each **iteration**, place the **minimum element** at the beginning (like Selection Sort).
- At the same time, place the **maximum element** at the end (like normally Bubble Sort does).

Continue until the array is sorted.

Display the sorted array along with the number of iterations, comparisons, and swaps.