Insertion Sort – Practice Set

Basic Questions

1. Dry run Insertion Sort on the array **[7, 4, 5, 2]** (Ascending order).
2. Show how Insertion Sort sorts the array **[10, 30, 20, 40, 50]** step by step.
3. Apply Insertion Sort on **[9, 8, 7, 6, 5]** and show the array after each pass.
4. Sort the array **[3, 1, 4, 1, 5]** using Insertion Sort.
5. After applying Insertion Sort on **[2, 2, 1, 3]**, what will the final sorted array look like?

Medium Level Questions

6. Implement Insertion Sort in C++ to sort an array of **10 integers** entered by the user.
7. Modify Insertion Sort to sort an array in **descending order**. Dry run it on **[15, 12, 18, 6, 3]**.
8. Count the total number of **shifts** made by Insertion Sort on the array **[12, 11, 13, 5, 6]**.
9. Apply Insertion Sort on **[20, 35, 15, 40, 50, 10]** and show the array after each pass.
10. For the array **[5, 1, 2, 3, 4]**, how many comparisons and shifts will Insertion Sort perform?

Hard/Conceptual Questions

11. Prove that Insertion Sort runs in **O(n)** time when the array is already sorted.
12. Why is Insertion Sort considered a **stable sorting algorithm**? Give an example with duplicate elements.

13. If Insertion Sort is applied on a **linked list** instead of an array, how would the shifting operation be different?

14. Compare Insertion Sort with Selection Sort:

- Which one performs fewer shifts?
- Which one is better for nearly sorted arrays?

15. Apply Insertion Sort on **[64, 25, 25, 12, 22, 11, 11, 90]** and check whether duplicate elements preserve their relative order.