DAY 29/180

Q1- Use Insertion Sort Algorithm to sort the array of integers in decreasing order.

```cpp
#include <bits/stdc++.h>  // Include necessary libraries
using namespace std;

void insertionsort(vector<int>& arr, int n) {
    for (int i = 1; i < n; i++) {
        // Start from the second element and work towards the end of the array
        for (int j = i; j > 0; j--) {
            if (arr[j] > arr[j - 1]) {
                // If the current element is greater than the previous element, swap them
                swap(arr[j], arr[j - 1]);
            } else {
                // If the current element is smaller or equal to the previous element, stop the inner loop
                break;
            }
        }
    }
}

int main() {
    int n;
    cin >> n;  // Input the number of elements in the array
    vector<int> arr(n);  // Create a vector to store the elements

    for (int i = 0; i < n; i++) {
        cin >> arr[i];  // Input the elements of the array
    }

    insertionsort(arr, n);  // Call the insertion sort function to sort the array

    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";  // Output the sorted elements
    }
}
```

Q2-Insertion Sort Algorithm to sort the array of integers in increasing order if we start from the last element of the array. Question was explained in the class.

```cpp
#include <bits/stdc++.h>
using namespace std;

void insertionsort(vector<int>& arr, int n) {
    for (int i = n - 1; i >= 1; i--) {
        // Start from the last element and work towards the first element
        for (int j = i; j < n; j++) {
            if (arr[j] < arr[j - 1]) {
                // If the current element is smaller than the previous element, swap them
                swap(arr[j], arr[j - 1]);
            } else {
                // If the current element is greater or equal to the previous element, stop the inner loop
                break;
            }
        }
    }
}

int main() {
    int n;
    cin >> n;  // Input the number of elements in the array
    vector<int> arr(n);  // Create a vector to store the elements

    for (int i = 0; i < n; i++) {
        cin >> arr[i];  // Input the elements of the array
    }

    insertionsort(arr, n);  // Call the insertion sort function to sort the array

    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";  // Output the sorted elements
    }
}
```