DSA LAB 12:

Name: Saif Majid Khan

SAP-ID: 57114

11/11/2024

GitHub:

https://github.com/saif01234567/Lab-Tasks-DS

TASK#1:

```
#include<iostream>
using namespace std;

int main() {
    int n = 5;
    int a[n];

    // Take input for the array
    cout << "Enter 5 elements of the array: " << endl;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
```

```
cout << "Unsorted Array: " << endl;
    for (int k = 0; k < n; k++) {
        cout << a[k] << "\t";
    }
    cout << endl;

    cout << "Insertion Sort in Descending Order" << endl;

    // Insertion Sort in descending order
    for (int i = 1; i < n; i++) {
        int temp = a[i];
        int j = i - 1;
        while (j >= 0 && a[j] < temp) {  // Change condition for descending order
            a[j + 1] = a[j];
            j--;
        }
        a[j + 1] = temp;
```

```
// Display array after each iteration
        cout << "Array after iteration " << i << ": ";
        for (int k = 0; k < n; k++) {
            cout << a[k] << "\t";
        }
        cout << endl;
    }

    cout << "Sorted Array after Insertion Sort in Descending Order: " << endl;
    for (int k = 0; k < n; k++) {
        cout << a[k] << "\t";
    }
    cout << endl;

    return 0;
}
```

Output

Clear

```
Enter 5 elements of the array:
1
2
6
3
9
Unsorted Array:
1   2   6   3   9
Insertion Sort in Descending Order
Array after iteration 1: 2  1   6   3   9
Array after iteration 2: 6  2   1   3   9
Array after iteration 3: 6  3   2   1   9
Array after iteration 4: 9  6   3   2   1
Sorted Array after Insertion Sort in Descending Order:
9   6   3   2   1


=== Code Execution Successful ===
```

TASK#2:

```
#include<iostream>
using namespace std;

int main() {
    int n = 9;
    int a[n];

    // Take input for the array
    cout << "Enter 9 elements of the array: " << endl;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
```

```
cout << "Unsorted Array:" << endl;
    for (int k = 0; k < n; k++) {
        cout << a[k] << " ";
    }
    cout << endl;

    cout << "Bubble Sort" << endl;
    int temp;
    bool swapped;
```

```
// Bubble Sort with early exit if the array is sorted
    for (int i = 0; i < n - 1; i++) {
        swapped = false;  // Reset the flag for each pass
        for (int j = 0; j < n - i - 1; j++) {
            if (a[j] > a[j + 1]) {
                // Swap elements if they are in the wrong order
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
                swapped = true;  // Set the flag to true if a swap occurred
            }
        }
 // If no elements were swapped, the array is already sorted
        if (!swapped) {
            break;
        }
```

```
        // Display the array after each pass
        cout << "Array after pass " << i + 1 << ": ";
        for (int k = 0; k < n; k++) {
            cout << a[k] << " ";
        }
        cout << endl;
    }

    cout << "Sorted Array after Bubble Sort is: " << endl;
    for (int k = 0; k < n; k++) {
        cout << a[k] << " ";
    }
    cout << endl;

    return 0;
}
```

Output

Clear

```
Enter 9 elements of the array:
2
1
6
8
3
4
5
9
7
Unsorted Array:
2 1 6 8 3 4 5 9 7
Bubble Sort
Array after pass 1: 1 2 6 3 4 5 8 7 9
Array after pass 2: 1 2 3 4 5 6 7 8 9
Sorted Array after Bubble Sort is:
1 2 3 4 5 6 7 8 9


=== Code Execution Successful ===
```