

LAB TASK : 12

Name: Abdul Moiz

SAP I'd: 54482

Section: 3rd

LAB: DSA

TASK 1:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int arr[5];
```

```
    // Taking input from the user
```

```
    cout << "Enter 5 elements in the array: ";
```

```
    for (int i = 0; i < 5; i++) {
```

```
        cin >> arr[i];
```

```
    }
```

```
    // Insertion Sort in Descending Order
```

```

for (int i = 1; i < 5; i++) {
    int key = arr[i];
    int j = i - 1;

    // Move elements of arr[0..i-1] that are smaller than key
    // to one position ahead of their current position
    while (j >= 0 && arr[j] < key) {
        arr[j + 1] = arr[j];
        j--;
    }
    arr[j + 1] = key;

    // Displaying array after each iteration
    cout << "Iteration " << i << ": ";
    for (int k = 0; k < 5; k++) {
        cout << arr[k] << " ";
    }
    cout << endl;
}

// Displaying the sorted array
cout << "Sorted array in descending order: ";

```

```

for (int i = 0; i < 5; i++) {
    cout << arr[i] << " ";
}

cout << endl;

return 0;
}

```

Dry Run

Let's assume the user inputs the array as: 30, 10, 50, 20, 40

1. **Initial Array:** 30, 10, 50, 20, 40

Iteration 1 (i = 1):

- **Key:** 10
- arr[1] is 10. Since $30 > 10$, no shifting is required.
- **Array after iteration 1:** 30, 10, 50, 20, 40

Iteration 2 (i = 2):

- **Key:** 50
- arr[2] is 50. Since $50 > 30$, shift 30 to the right.
- Move 10 as well since $50 > 10$.
- **Array after iteration 2:** 50, 30, 10, 20, 40

Iteration 3 (i = 3):

- **Key:** 20
- arr[3] is 20. Since $20 < 30$ but > 10 , move only 10 to the right.
- **Array after iteration 3:** 50, 30, 20, 10, 40

Iteration 4 (i = 4):

- **Key:** 40
- arr[4] is 40. Since $40 < 50$ but > 30 , shift 30 and 20 to the right.
- **Array after iteration 4:** 50, 40, 30, 20, 10

Final Output

After all iterations, the sorted array in descending order is:

50, 40, 30, 20, 10

TASK 2:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int arr[9];
```

```
    // Taking input from the user
```

```
    cout << "Enter 9 elements in the array: ";
```

```
    for (int i = 0; i < 9; i++) {
```

```
        cin >> arr[i];
```

```
    }
```

```
    // Bubble Sort with early termination if already sorted
```

```
    bool isSorted;
```

```
    for (int i = 0; i < 9 - 1; i++) {
```

```
isSorted = true; // Assume the array is sorted at the start of each
pass
```

```
// Perform a single pass of Bubble Sort
```

```
for (int j = 0; j < 9 - i - 1; j++) {
```

```
    if (arr[j] > arr[j + 1]) {
```

```
        // Swap arr[j] and arr[j+1] if they are in the wrong order
```

```
        int temp = arr[j];
```

```
        arr[j] = arr[j + 1];
```

```
        arr[j + 1] = temp;
```

```
        isSorted = false; // If a swap is made, array is not sorted
```

```
    }
```

```
}
```

```
// Displaying array after each pass
```

```
cout << "After pass " << i + 1 << ": ";
```

```
for (int k = 0; k < 9; k++) {
```

```
    cout << arr[k] << " ";
```

```
}
```

```
cout << endl;
```

```

// If no swaps were made in the pass, the array is already sorted
if (isSorted) {
    cout << "Array is already sorted. Stopping early." << endl;
    break;
}
}

// Displaying the sorted array
cout << "Sorted array in ascending order: ";
for (int i = 0; i < 9; i++) {
    cout << arr[i] << " ";
}
cout << endl;

return 0;
}

```

Dry Run

Let's assume the user enters the array: 5, 2, 9, 1, 5, 6, 3, 7, 4

Pass 1:

- Array after pass 1: 2, 5, 1, 5, 6, 3, 7, 4, 9
(Swaps were made, so `isSorted = false`)

Pass 2:

- Array after pass 2: 2, 1, 5, 5, 3, 6, 4, 7, 9
(Swaps were made, so `isSorted = false`)

Subsequent Passes:

The passes continue until the array is sorted in ascending order. If a pass completes without swaps, the program stops early, indicating the array is already sorted.

Final Output

After sorting, the array might look like:

1, 2, 3, 4, 5, 5, 6, 7, 9