**Bubble Sort**

**Aim :**

To write a C++ program that accepts n elements from the user, stores them in an array, and sorts the array in ascending order using the **Bubble Sort algorithm**.

**Program:**

#include <iostream>

using namespace std;

void bubbleSort(int a[], int n)

{

for (int i = 0; i < n - 1; i++)

for (int j = 0; j < n - i - 1; j++)

if (a[j] > a[j + 1])

{

int temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

int main()

{

int a[20], n;

cout << "Enter the value of n: ";

cin >> n;

cout << "Enter " << n << " values one by one: ";

for (int i = 0; i < n; i++)

cin >> a[i];

cout << "Original array: ";

for (int i = 0; i < n; i++)

cout << a[i] << " ";

cout << endl;

bubbleSort(a, n);

cout << "Sorted array: ";

for (int i = 0; i < n; i++)

cout << a[i] << " ";

cout << endl;

return 0;

}

Result:

The program was successfully executed.

**Linear Search**

**Aim :**

To write a C++ program that accepts n elements from the user, stores them in an array, and searches for a given element using the **Linear Search technique**.

**Program:**

#include <iostream>

using namespace std;

int linearSearch(int a[], int n, int key)

{

for (int i = 1; i < =n; i++)

{

if (a[i] == key)

return i;

}

return -1;

}

int main()

{

int a[20], n, key;

cout << "Enter the number of elements: ";

cin >> n;

cout << "Enter " << n << " elements: ";

for (int i = 0; i < n; i++)

cin >> a[i];

cout << "Enter the element to search: ";

cin >> key;

int result = linearSearch(a, n, key);

if (result == -1)

cout << "Element not found in the array." << endl;

else

cout << "Element found at position " << result + 1 << " (index " << result << ")." << endl;

return 0;

}

Result:

The program was successfully executed.

**Binary Search**

Aim :

To write a C++ program that accepts n sorted elements from the user, stores them in an array, and searches for a given element using the **Binary Search technique**.

Program:

#include <iostream>

using namespace std;

int binarySearch(int a[], int n, int key)

{

int low = 0, high = n - 1;

while (low <= high)

{

int mid = (low + high) / 2;

if (a[mid] == key)

return mid;

else if (a[mid] < key)

low = mid + 1;

else

high = mid - 1; }

return -1; // not found

}

int main()

{

int a[20], n, key;

cout << "Enter the number of elements: ";

cin >> n;

cout << "Enter " << n << " elements in sorted order: ";

for (int i = 0; i < n; i++)

cin >> a[i];

cout << "Enter the element to search: ";

cin >> key;

int result = binarySearch(a, n, key);

if (result == -1)

cout << "Element not found in the array." << endl;

else

cout << "Element found at position " << result + 1 << " (index " << result << ")." << endl;

return 0;

}

Result:

The program was successfully executed.