Practical-1

AIM: Installation of VS Code. Implement Linear Search and Binary Search using array data structure.

- Program

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

//using iterative function
int linearSearch(int Arr[], int n, int key){
    for(int i=0; i<n; i++){
        if(Arr[i] == key){
            return i;
        }
    }
    return -1;
}

//Using recursive function
int linearSearch(int Arr[], int n, int key, int i){
    if(i>n){
        return -1;
    }
    else if(Arr[i] == key){
        return i;
    }
    else if(Arr[i] != key){
        return linearSearch(Arr, n, key, ++i);
    }
}

void linearSearchSelected(){
    int n;
    int key;
    cout << "Enter number of elements : ";
```

```
        cin >> n;
        int Arr[n];
        for(int i=0; i < n; i++){
            cout << "Enter " << i+1 << " element : ";
            cin >> Arr[i];
        }
        cout << "Enter element you want to find : ";
        cin >> key;

        int result = linearSearch(Arr, n, key);
        if(result == -1){
            cout << "Element does not found in array." << endl;
        }
        else{
            cout << "Element founded on index number " << result
<< endl;
        }
    }

    //Using iterative function
    int binarySearch(int Arr[], int n, int key){
        int l = 0;
        int h = n-1;
        int mid = 0;
        while(l<=h){
            mid = (l+h)/2;
            if(Arr[mid] == key){
                return mid;
            }
            else if(Arr[mid] < key){
                l = mid+1;
            }
            else if(Arr[mid] > key){
```

```
            h = mid-1;
        }
    }
    return -1;
}

//Using recursive function
int binarySearch(int Arr[], int n, int key, int l, int h){
    int mid = (l+h)/2;
    if(l>h){
        return -1;
    }
    else if(Arr[mid] == key){
        return mid;
    }
    else if(Arr[mid] < key){
        return binarySearch(Arr, n, key, mid+1, h);
    }
    else if(Arr[mid] > key){
        return binarySearch(Arr, n, key, l, mid-1);
    }
}

void binarySearchSelected(){
    int n;
    int key;
    cout << "Enter number of elements : ";
    cin >> n;
    int Arr[n];
    cout << "Enter array in sorted form " << endl << endl;
    for(int i=0; i < n; i++){
        cout << "Enter element A[" << i << "] : " ;
```

```
            cin >> Arr[i];
            cout << endl;
        }
        cout << "Enter element you want to find : ";
        cin >> key;

        int result = binarySearch(Arr, n, key, 0, n-1); //Using
iterative function or recursive function
        if(result == -1){
            cout << "Element does not found in array." << endl;
        }
        else{
            cout << "Element founded on index number " << result
<< endl;
        }
    }

    int main(){
        int choice = 0;
        cout << "This program is developed by
22CE097_ShivangPatel" << endl << endl;
        cout << "1. Linear Search " << endl;
        cout << "2. Binary Search " << endl;
        cout << "Select searching method : ";
        cin >> choice;
        switch (choice)
        {
            case 1:
                linearSearchSelected();
                break;
            case 2:
                binarySearchSelected();
                break;
```

```
            default:
                cout << "Select either 1 or 2";
                main();
                break;
        }
    }
```

Output

```
This program is developed by 22CE097_ShivangPatel

1. Linear Search
2. Binary Search
Select searching method : 1
Enter number of elements : 5
Enter 1 element : 24
Enter 2 element : 46
Enter 3 element : 23
Enter 4 element : 65
Enter 5 element : 76
Enter element you want to find : 76
Element founded on index number 4
```

```
This program is developed by 22CE097_ShivangPatel

1. Linear Search
2. Binary Search
Select searching method : 2
Enter number of elements : 5
Enter array in sorted form

Enter element A[0] : 23

Enter element A[1] : 45

Enter element A[2] : 76

Enter element A[3] : 87

Enter element A[4] : 89

Enter element you want to find : 87
Element founded on index number 3
```

- Conclusion