

1. Write a C++ programs to implement recursive and non-recursive

i) Linear search

ii) Binary search

1) Linear Search

Non-Recursive (Iterative) Linear Search:

```
#include <iostream>
```

```
using namespace std;
```

```
int linearSearchIterative(int arr[], int n, int key) {
```

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

```
        if (arr[i] == key) {
```

```
            return i;
```

```
        }
```

```
    }
```

```
    return -1;
```

```
}
```

```
int main() {
```

```
    int arr[] = {2, 4, 6, 8, 10};
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    int key = 8;
```

```
    int result = linearSearchIterative(arr, n, key);
```

```
    if (result != -1)
```

```
        cout << "Element found at index " << result << endl;
```

```
    else
```

```
        cout << "Element not found." << endl;
```

```
    return 0;
```

```
}
```

Recursive Linear Search:

```
#include <iostream>

using namespace std;

int linearSearchRecursive(int arr[], int n, int key) {
    if (n == 0)
        return -1;
    if (arr[n - 1] == key)
        return n - 1;
    return linearSearchRecursive(arr, n - 1, key);
}

int main() {
    int arr[] = {2, 4, 6, 8, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 8;

    int result = linearSearchRecursive(arr, n, key);

    if (result != -1)
        cout << "Element found at index " << result << endl;
    else
        cout << "Element not found." << endl;

    return 0;
}
```

ii) Binary Search

Non-Recursive (Iterative) Binary Search:

```
#include <iostream>
```

```
using namespace std;
```

```
int binarySearchIterative(int arr[], int n, int key) {
```

```
    int low = 0, high = n - 1;
```

```
    while (low <= high) {
```

```
        int mid = low + (high - low) / 2;
```

```
        if (arr[mid] == key)
```

```
            return mid;
```

```
        else if (arr[mid] < key)
```

```
            low = mid + 1;
```

```
        else
```

```
            high = mid - 1;
```

```
    }
```

```
    return -1;
```

```
}
```

```
int main() {
```

```
    int arr[] = {2, 4, 6, 8, 10};
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    int key = 8;
```

```
    int result = binarySearchIterative(arr, n, key);
```

```
    if (result != -1)
```

```

        cout << "Element found at index " << result << endl;
    else
        cout << "Element not found." << endl;

    return 0;
}

```

Recursive Binary Search:

```

#include <iostream>
using namespace std;

int binarySearchRecursive(int arr[], int low, int high, int key) {
    if (low > high)
        return -1;

    int mid = low + (high - low) / 2;

    if (arr[mid] == key)
        return mid;
    else if (arr[mid] < key)
        return binarySearchRecursive(arr, mid + 1, high, key);
    else
        return binarySearchRecursive(arr, low, mid - 1, key);
}

int main() {
    int arr[] = {2, 4, 6, 8, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 8;

    int result = binarySearchRecursive(arr, 0, n - 1, key);
}

```

```
if (result != -1)
    cout << "Element found at index " << result << endl;
else
    cout << "Element not found." << endl;

return 0;
}
```