SEARCHING

LINEAR SEARCH :

- Also known as SEQUENTIAL SEARCH, is a simple algorithm used to find a target value within a list or array.
- It works by sequentially checking each element in the list until the desired element is found. (or) the end of the list is reached.

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

int linearSearch (vector <int>& arr, int target) {
    for (int i=0; i < arr.size(); i++)
        if (arr[i] == target)
            return i;

    return -1;
}

int main () {
    // using vector for dynamic array
    vector <int> numbers = {10,25,3,40,15,7};
    int searchTarget = 40;

    int result = linearSearch (numbers, searchTarget);

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

    return 0;
}
```

BINARY SEARCH :

- It is an efficient algorithm for finding a target value within a SORTED array or list.
- It works by repeatedly dividing the search interval in half.

```
#include <bits/stdc++.h>
using namespace std;
int binSearchIterative(vector<int>& arr, int target) {
    int left = 0;
    int right = arr.size()-1;
    while (left <= right) {
        int mid = (left + right) / 2;
        if (arr[mid] == target)
            return mid;          // Target found at index mid
        else if (arr[mid] < target)
            left = mid+1;        // Search in Right half
        else
            right = mid-1;       // Search in Left half
    }
    return -1;                   // Target not found
}

int binSearchRecursive(vector<int>& arr, int target,
                       int left, int right) {
    if (left > right)
        return -1;               // Target not found
    int mid = (left + right) / 2;
    if(arr[mid] == target)
        return mid;
    else if (arr[mid] < target)
        return binSearchRecursive(arr, target, mid+1, right);
    else
        return binSearchRecursive(arr, target, left, mid-1);
}
```

```
int main() {
    vector <int> sortedArray = {3,4,6,7,9,12,16,17};
    target = 9;

    // Using iterative approach
    int index = binarySearchIterative (sortedArray, Target);
    if (index != -1)
        cout << " Target found at " << index << endl;
    else
        cout << " Target not found " << endl;

    ---------------------------------------

    // Using recursive approach
    int index = binSearchRecursive (sortedArray, Target, 0,
                                    sortedArray.size() -1);

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

    ---------------------------------------

    // Using STL function
    if (binary_search (sortedArray.begin(), sortedArray.end(),
                       target))
        cout << " Target is Found " << endl;
    else
        cout << " Target not found " << endl;

    return 0;
}
```