Searching Algorithms -

Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored. Based on the type of search operation, these algorithms are generally classified into two categories:

1) Sequential Search
2) Interval Search

Sequential Search -

In this, the list or array is traversed sequentially and every element is checked.
For example: Linear Search.

0	1	2	3	4
55	62	7	11	173

Interval Search -

These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear Search as they repeatedly target the center of the search structure and divide the search space in half.
For Example: Binary Search.

0	1	2	3	4
7	11	55	62	173

Linear Search Algorithm -

```
1. Take input array from user
2. Take element you want to search
   from user
3. Start from 1st element in
   array to last
4. if match found
        then -> print message.
        stop process.
    else
        move to next element in array
5. END
```

0	1	2	3	4
55	62	7	11	173

linearSearch.cpp | linearSearch1.cpp

```cpp
# include <iostream>
using namespace std;

void linearSearch(int a[], int n)
{
    int temp = -1;

    for(int i=0;i<5;i++)
    {
        if(a[i]==n)
        {
            cout<<"Element found at location: "<<i<<endl;
            temp=0;
        }
    }
    if(temp==-1)
    {
        cout<<"No Element Found"<<endl;
    }

}


int main()
{
    int arr[5]={1,22,34,5,7};
    cout<<"Please enter an element to search"<<endl;
    int num;
    cin>>num;

    linearSearch(arr,num);
```

linearSearch.cpp | [*] Untitled1

```cpp
# include <iostream>
using namespace std;

void linearSearch(int a[], int n)
{
    int temp = -1;

    for(int i=0;i<5;i++)
    {
        if(a[i]==n)
        {
            cout<<"Element found at location: "<<i<<endl;
        }
    }

}


int main()
{
    int arr[5]={1,22,34,5,7};
    cout<<"Please enter an element to search"<<endl;
    int num;
    cin>>num;

    linearSearch(arr,num);

    return 0;
}
```