



Daffodil
International
University

ASSIGNMENT

Course Code: SE215

Course Title: Algorithm Analysis and Design Lab

Topic Name: Linear Search & Binary Search

Submitted To:

Name: Ishrat Sultana

Designation: Lecturer

**Department: Department of Software
Engineering**

Daffodil International University

Submitted By:

Name: Monira Islam

ID: 0242320005341017

Section: 41 - A1

Semester: Spring 2025

**Department: Department of Software
Engineering**

Daffodil International University

Submission Date: 2025-01-29

1) Implement of "Linear Search" Algorithm - Create a function named -LinearSearch()



```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int LinearSearch(const vector<int> &arr, int target)
5  {
6      for (int i = 0; i < arr.size(); i++)
7      {
8          if (arr[i] == target)
9          {
10             return i;
11          }
12      }
13      return -1;
14  }
15
16  int main()
17  {
18      vector<int> v;
19      int sz, element, target;
20
21      cout << "enter the number of elements: ";
22      cin >> sz;
23
24      cout << "enter the elements: ";
25      for (int i = 0; i < sz; i++)
26      {
27          cin >> element;
28          v.push_back(element);
29      }
30
31      cout << "enter target for Linear Search: ";
32      cin >> target;
33
34      int linearResult = LinearSearch(v, target);
35
36      if (linearResult != -1)
37      {
38          cout << "Element found at index " << linearResult << endl;
39      }
40      else
41      {
42          cout << "element not found" << endl;
43      }
44
45      return 0;
46  }
```

Output:

```
Idba@DESKTOP-EQEQ37R MINGW64 /d/Algorithmn Lab/class_1/output (main)
$ ./"linearSearch.exe"
● enter the number of elements: 7
  enter the elements: 11 88 96 2 77 67 57
  enter target for Linear Search: 2
  Element found at index 3

Idba@DESKTOP-EQEQ37R MINGW64 /d/Algorithmn Lab/class_1/output (main)
○ $ 
```

2) Apply the recursive method of "Binary Search" Algorithm

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int BinarySearchRecursive(const vector<int> &arr, int left, int right, int target)
5  {
6      if (left > right)
7      {
8          return -1;
9      }
10
11     int mid = left + (right - left) / 2;
12
13     if (arr[mid] == target)
14     {
15         return mid;
16     }
17
18     if (arr[mid] > target)
19     {
20         return BinarySearchRecursive(arr, left, mid - 1, target);
21     }
22
23     return BinarySearchRecursive(arr, mid + 1, right, target);
24 }
25
26 int main()
27 {
28     vector<int> v;
29     int sz, element, target;
30
31     cout << "enter arr size: ";
32     cin >> sz;
33
34     cout << "enter elements -> ";
35     for (int i = 0; i < sz; i++)
36     {
37         cin >> element;
38         v.push_back(element);
39     }
40
41     sort(v.begin(), v.end());
42     cout << "your sorted array -> ";
43     for (int i = 0; i < sz; i++)
44     {
45         cout << v[i] << " ";
46     }
47     cout << endl;
48
49     cout << "enter target for binary search: ";
50     cin >> target;
51
52     int binaryResult = BinarySearchRecursive(v, 0, v.size() - 1, target);
53
54     if (binaryResult != -1)
55     {
56         cout << "element found at index " << binaryResult << endl;
57     }
58     else
59     {
60         cout << "element not found" << endl;
61     }
62
63     return 0;
64 }
```

Output:

```
Idba@DESKTOP-EQEQ37R MINGW64 /d/Algorithmn Lab/class_1/output (main)
$ ./"binarySearch.exe"
enter arr size: 8
enter elements -> 2 3 5 23 33 66 77 90
your sorted array -> 2 3 5 23 33 66 77 90
● enter target for binary search: 90
  element found at index 7

Idba@DESKTOP-EQEQ37R MINGW64 /d/Algorithmn Lab/class_1/output (main)
○ $ █
```