

Week 2: Binary Search and Linear Search

1. **Aim:** Implement Linear Search using array data structure.

Input Format

- First line contains number of total elements **n**
- second line contain **n** array elements
- Last line contains **x** , a value that to be searched from the array

Output Format

- If **x** is present in the array, then its index will be printed.
- If **x** is not present in the array, then this line must be printed "Element **x** is not present in the array."
- Last line of output shows the number of comparisons required to find the element **x**

Program:

```
// ITERATIVE APPROACH

#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    int n;
    cin>>n;
    int arr[n];
    for(int i =0;i<n;i++)
    {
        cin>>arr[i];
    }
}
```

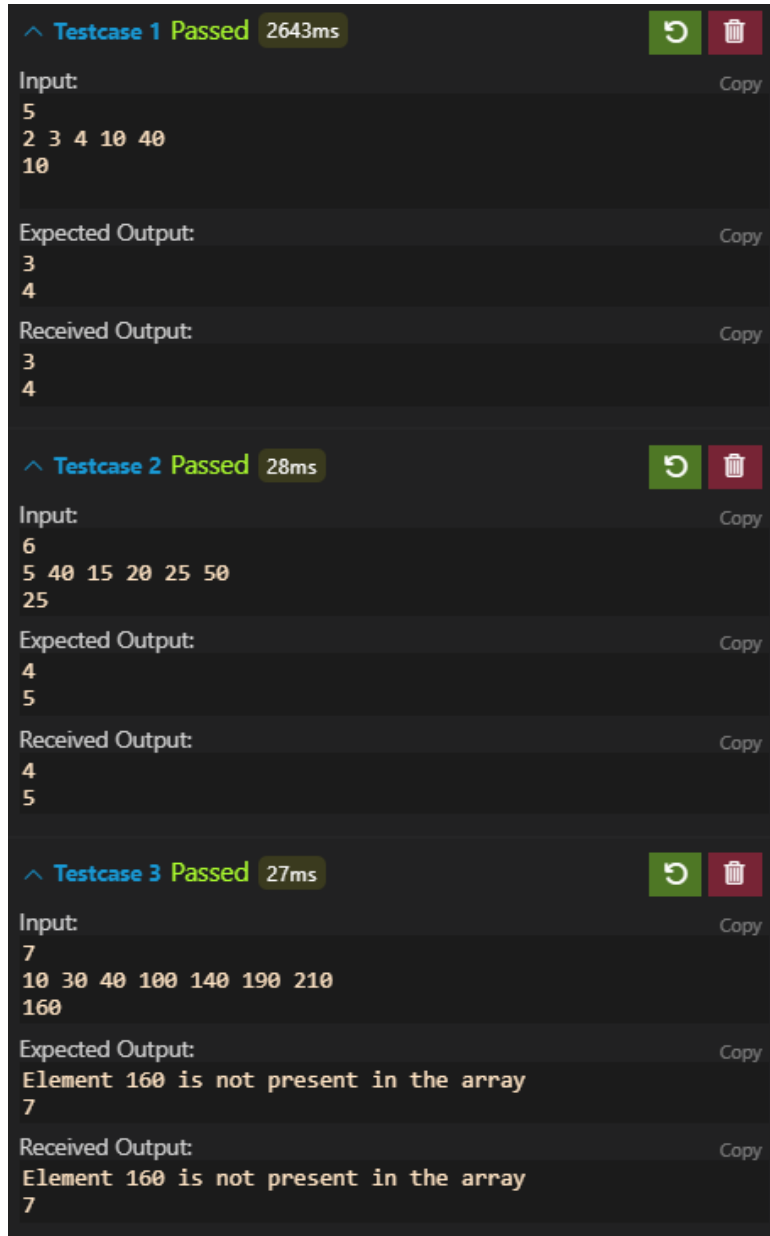
```
int x;
cin>>x;
int cnt =0;
for(int i =0;i<n;i++)
{
    if(arr[i] == x)
    {
        cnt = 1;
        cout<<i<<endl<<i+1;
        break;
    }
}
if(cnt == 0)
{
    cout<<"Element "<<x<< " is not present in the
array"<<endl<<n;
}
return 0;
}
```

```
// RECURSIVE APPROACH
```

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
```

```
int linear_search(int *arr,int k,int i,int n)
{
    if(i == n)
        return -1;
    if(arr[i] == k)
        return i;
    return linear_search(arr,k,i+1,n);
}

int main() {
    int n;
    cin>>n;
    int arr[n];
    for(int i =0;i<n;i++)
    {
        cin>>arr[i];
    }
    int key;
    cin>>key;
    int k  = linear_search(arr,key,0,n);
    if(k!=-1)
    {
        cout<<k<<endl<<k+1;
    }
    else
    {
        cout<<"Element "<<key<<" is not present in the
array"<<endl<<n;
    }
    return 0;
}
```

Input & Output:

```
^ Testcase 1 Passed 2643ms
Input:
5
2 3 4 10 40
10
Expected Output:
3
4
Received Output:
3
4

^ Testcase 2 Passed 28ms
Input:
6
5 40 15 20 25 50
25
Expected Output:
4
5
Received Output:
4
5

^ Testcase 3 Passed 27ms
Input:
7
10 30 40 100 140 190 210
160
Expected Output:
Element 160 is not present in the array
7
Received Output:
Element 160 is not present in the array
7
```

Conclusion:

In this Program I have learned the Algorithm called linear search which is used to search the element in unsorted array

Time complexity:

For Iterative : $O(N)$

For Recursive: $O(N)$

Space Complexity

For Iterative : $O(1)$

For Recursive: $O(N)$

2. **Aim:** Instructor has the sorted list (according to roll numbers in ascending order) in which roll numbers and submission time are stored of those students who have submitted the program after its deadline.

Help Instructor to check whether particular student has submitted the program or not. If yes, then submission time should be displayed. If not, then “Not submitted” message should be printed.

Explanation :

Sample Input:

n=5

roll numbers : 67,89,95,96,99

submission times: 65,69,62,64,61

roll_no_tosearch : 99

Sample output:

99 has submitted program in **61** minutes.

Sample Input:

n=5

roll numbers : 67,89,95,96,99

submission times: 65,69,62,64,61

roll_no_tosearch : 100

Sample output:

" Not submitted".

Input Format

- First line contains the integer, the number of students n
- next lines have two integers , *roll_no* and *submission time* for n students.
- Third Line is interger, *roll_no_tosearch*

Constraints

- $1 < n < 50$
- $1 < \text{roll_no} < 190$
- $60 < \text{submission time} < 300$
- $1 < \text{roll_no_tosearch} < 190$

Output Format

- First line contains the *submissiontime* of the *roll_no_tosearch*.
- Second line contains number of comparisons required in Binary search.
- If the *roll_no_tosearch* is not in the list, then "Not Submitted" message is the output.

Program:

```
// ITERATIVE APPROACH

#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int n;
    cin >> n;
    int a[n];
    int b[n];
    for(int i = 0; i < n; i++)
    {
        cin >> a[i];
        cin >> b[i];
    }
    int k;
    cin >> k;
    int s = 0;
    int e = n-1;
    int mid = s + (e-s)/2;
    int cnt = 0;
```

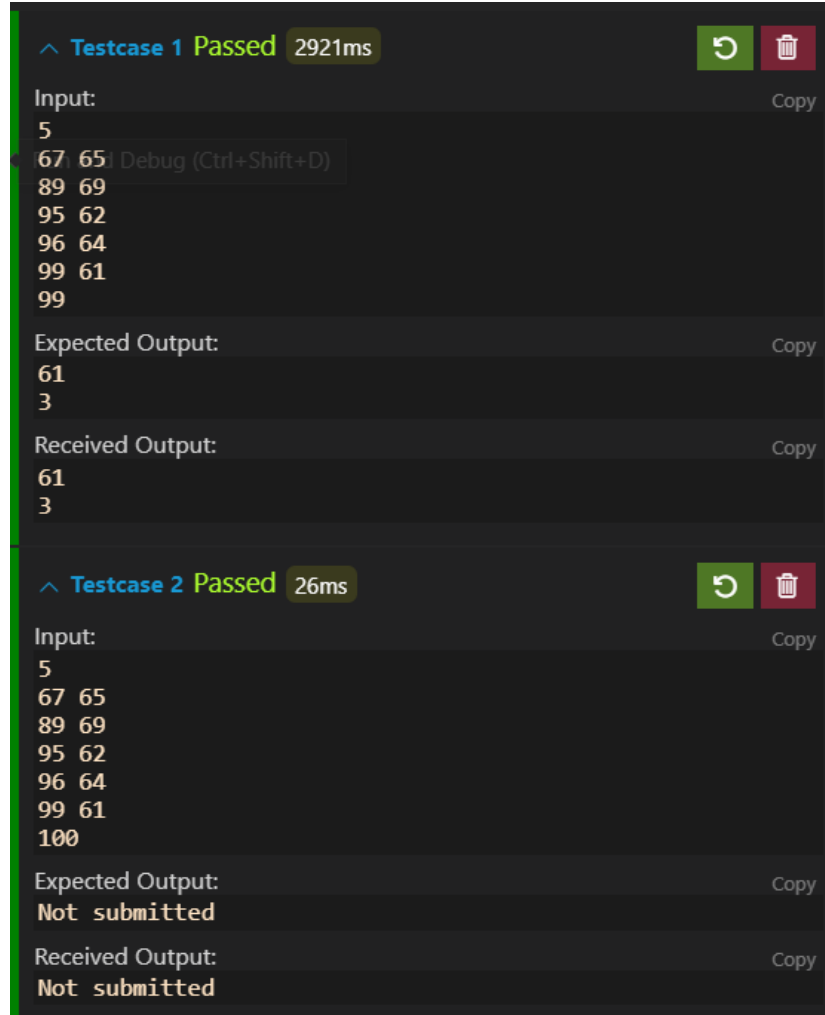
```
int flag = 0;
while(s<=e){
    cnt++;
    if(a[mid] == k)
    {
        cout<<b[mid]<<endl<<cnt;
        flag = 1;
        break;
    }
    else if(a[mid]>k)
        e =mid-1;
    else
        s = mid +1;
    mid = s + (e-s)/2;
}
if(flag == 0)
    cout<<"Not submitted"<<endl;
return 0;
}
```

```
//RECURSIVE APPROACH

#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

int binary_search(int *arr,int key,int s,int e,int &i)
{
    i++;
    if(s>e)
        return -1;
    int mid = s+ (e-s)/2;
    if(arr[mid] == key)
        return mid;
    else if(arr[mid]>key)
        return binary_search(arr,key,s,mid-1,i);
    else
        return binary_search(arr,key,mid+1,e,i);
}
```

```
int main() {  
    /* Enter your code here. Read input from STDIN. Print output to STDOUT  
*/  
    int n;  
    cin>>n;  
    int arr[n];  
    int b[n];  
    for(int i =0;i<n;i++)  
    {  
        cin>>arr[i];  
        cin>>b[i];  
    }  
    int key;  
    cin>>key;  
    int i =0;  
    int ind =binary_search(arr,key,0,n-1,i);  
    if(ind == -1)  
        cout<<"Not submitted";  
    else  
    {  
        cout<<b[ind]<<endl<<i;  
    }  
    return 0;  
}
```


Input & Output:

```
^ Testcase 1 Passed 2921ms
Input:
5
67 65
89 69
95 62
96 64
99 61
99
Expected Output:
61
3
Received Output:
61
3

^ Testcase 2 Passed 26ms
Input:
5
67 65
89 69
95 62
96 64
99 61
100
Expected Output:
Not submitted
Received Output:
Not submitted
```

Conclusion:

In this program I have learned the algorithm called Binary search which is used to search the element in sorted array.

Time complexity:

For Iterative : $O(\log(N))$

For Recursive: $O(\log(N))$

Space Complexity

For Iterative : $O(1)$

For Recursive: $O(\log(N))$