



Tutorial Link <https://codequotient.com/tutorials/Searching an Element - Linear Search/5a12edf146765b2b63e3476b>

## TUTORIAL

# Searching an Element - Linear Search

## Chapter

### 1. Searching an Element - Linear Search

#### Topics

#### 1.2 Linear Search

#### 1.5 Recursive implementation of linear

The process of identifying or finding a particular record is called Searching. You often spend time in searching for any desired item. If the data is kept properly in sorted order, then searching becomes very easy and efficient. Any search is said to be successful or unsuccessful depending upon whether the element that is being searched is found or not. In this article you will get to know the basic concepts of searching in sorted and unsorted arrays that is used in data structures. Search can be done popularly in two ways: -

## Linear Search

If we start from the first element of list, and compare each element with the element we are searching, it is called linear search. This method can be performed on a sorted or an unsorted list (usually arrays). In both cases, the search will start from array index 0 and each time it match the searched element and the element at current index. If they match it will return the index otherwise, it will move to next index. If the whole array is passed to a function and the searched element does not match then it will return a negative index. The time

complexity of Linear search is  $O(n)$  as it may have to search for all the elements in worst case. In case of an array the general algorithm for linear search is as follows: -

```
X = searched_Element
index = 0
While(index < length_of_array)
    If: X == array[index] then RETURN index
    Else: index = index + 1
End
End
RETURN -1
```

Following is the iterative implementation of linear search: -

```
1  #include<stdio.h>
2
3  int linear_search(int arr[], int n, int x)
4  {
5      int i;
6      for (i=0; i<n; i++)
7          if (arr[i] == x)    // Check each element of the
8              return i;      // array.
9      return -1;             // if found return the position
10 }                           // otherwise return -1
11
12 int main()
13 {
14     int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
15
16     x=25;    // Searched Element.
17
18     loc=linear_search(array, 10, x);    // Call the search
19                                         function
20     if(loc != -1)
        printf("Element found at location : %d",loc);
```

```
21
22     else
23         printf("Element not present in the array.");
24     return 0;
25 }
26
```

```
1  import java.util.Scanner;
2  // Other imports go here
3  // Do NOT change the class name
4  class Main{
5      static int linear_search(int arr[], int n, int x)
6      {
7          int i;
8          for (i=0; i<n; i++)
9              if (arr[i] == x)    // Check each element of the
array.
10                 return i;      // if found return the position
11                 return -1;     // otherwise return -1
12     }
13     public static void main(String[] args)
14     {
15         int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
16
17         x=25;    // Searched Element.
18
19         loc=linear_search(array, 10, x);    // Call the
search function
20
21         if(loc != -1)
22             System.out.print("Element found at location : " +
loc);
23         else
24             System.out.print("Element not present in the
array.");
25     }
26 }
```

Java

The output of above program is as below for different runs: -

Element found at location :5

## Recursive implementation of linear

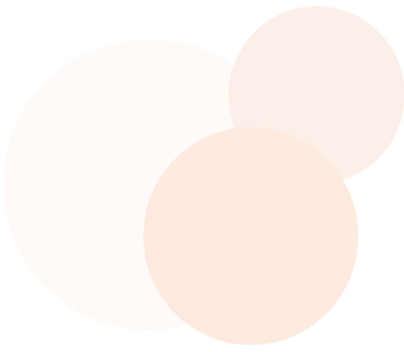
```
1  #include<stdio.h>
2
3  int rec_linear_search(int arr[], int left, int right, int
4  x)
5  {
6      int result;
7      if (right < left)          // The array is exhausted so
8      return -1
9      return -1;
10     if (arr[left] == x)        // If element found return
11     position
12     return left;
13     // Call the function again with new subarray from next
14     element.
15     result = rec_linear_search(arr, left+1, right, x);
16     return result;    // return the result to the calling
17     function.
18 }
19
20 int main()
21 {
22     int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
23     x=13;                // element to be searched in the array
24     loc=rec_linear_search(array,0,10,x);
25     if(loc != -1)
26         printf("Element found at location : %d",loc);
27     else
28         printf("Element not present in the array.");
29     return 0;
30 }
```

```
1  import java.util.Scanner;
2  // Other imports go here
3  // Do NOT change the class name
```

```
4 class Main{
5 static int rec_linear_search(int arr[], int left, int
right, int x)
6 {
7     int result;
8     if (right < left)          // The array is exhausted so
return -1
9     return -1;
10    if (arr[left] == x)        // If element found return
position
11    return left;
12    // Call the function again with new subarray from next
element.
13    result = rec_linear_search(arr, left+1, right, x);
14    return result;    // return the result to the calling
function.
15 }
16 public static void main(String[] args)
17 {
18     int loc,x,array[]={10,11,12,13,14,25,26,37,48,29};
19     x=13;    // Searched Element.
20     loc = rec_linear_search(array, 0, 10, x);    // Call
the search function
21     if(loc != -1)
22         System.out.print("Element found at location : " +
loc);
23     else
24         System.out.print("Element not present in the
array.");
25 }
26 }
```

The output of above program is as below for different runs: -

```
Element found at location :3
```



Tutorial by codequotient.com | All rights reserved, CodeQuotient 2020