

[Home](#) » [C programming](#) » [C programs](#) » Linear search in C

## Linear search in C

### Linear search

Array

6	3	0	5	1	2	8	-1	4
---	---	---	---	---	---	---	----	---

Element to search: 8

Linear search in C to find whether a number is present in an array. If it's present, then at what location it occurs. It is also known as a sequential search. It is straightforward and works as follows: we compare each element with the element to search until we find it or the list ends. Linear search for [multiple occurrences](#) and using a [function](#).



### Linear search program in C

```
#include <stdio.h>

int main()
{
    int array[100], search, c, n;

    printf("Enter number of elements in array\n");
    scanf("%d", &n);

    printf("Enter %d integer(s)\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    printf("Enter a number to search\n");
    scanf("%d", &search);

    for (c = 0; c < n; c++)
    {
        if (array[c] == search)    /* If required element is found */
        {
            printf("%d is present at location %d.\n", search, c+1);
        }
    }
}
```

```
        break;
    }
}
if (c == n)
    printf("%d isn't present in the array.\n", search);

return 0;
}
```



Output of program:

Download [Linear search](#) program.

[C program for binary search](#)

## Linear search C program for multiple occurrences

In the code below we will print all locations at which required element is found and also the number of times it occurs in the list.

```
#include <stdio.h>

int main()
{
    int array[100], search, c, n, count = 0;

    printf("Enter number of elements in array\n");
    scanf("%d", &n);

    printf("Enter %d numbers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    printf("Enter a number to search\n");
    scanf("%d", &search);

    for (c = 0; c < n; c++) {
        if (array[c] == search) {
            printf("%d is present at location %d.\n", search, c+1);
            count++;
        }
    }
    if (count == 0)
        printf("%d isn't present in the array.\n", search);
    else
```

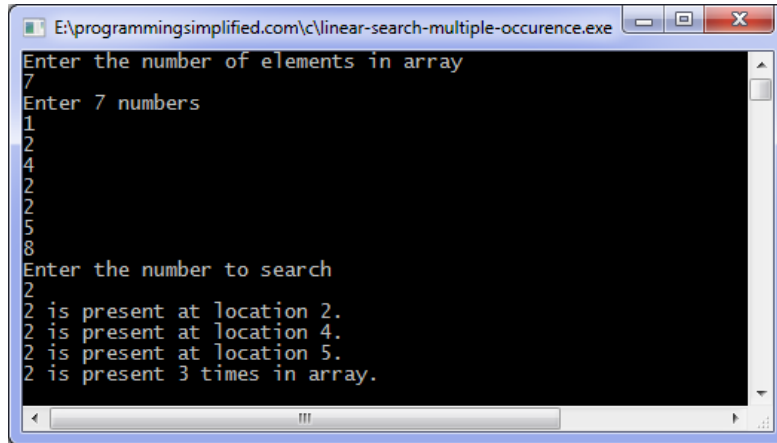
```

        printf("%d is present %d times in the array.\n", search, count);
    }
    return 0;
}

```

Download [Linear search](#) multiple occurrence program.

Output of code:



```

E:\programmingsimplified.com\c\linear-search-multiple-occurrence.exe
Enter the number of elements in array
7
Enter 7 numbers
1
2
4
2
2
5
8
Enter the number to search
2
2 is present at location 2.
2 is present at location 4.
2 is present at location 5.
2 is present 3 times in array.

```

## C program for linear search using a function

```

#include <stdio.h>

long linear_search(long [], long, long);

int main()
{
    long array[100], search, c, n, position;

    printf("Input number of elements in array\n");
    scanf("%ld", &n);

    printf("Input %d numbers\n", n);

    for (c = 0; c < n; c++)
        scanf("%ld", &array[c]);

    printf("Input a number to search\n");
    scanf("%ld", &search);

    position = linear_search(array, n, search);

    if (position == -1)
        printf("%d isn't present in the array.\n", search);
    else
        printf("%d is present at location %d.\n", search, position+1);

    return 0;
}

long linear_search(long a[], long n, long find) {
    long c;

    for (c = 0 ; c < n ; c++ ) {
        if (a[c] == find)
            return c;
    }

    return -1;
}

```

## Linear search function using pointers

```

long linear_search(long *p, long n, long find) {
    long c;

    for (c = 0; c < n; c++) {
        if (*(p+c) == find)
            return c;
    }

    return -1;
}

```

The time required to search an element using the algorithm depends on the size of the list. In the best case, it's present at the beginning of the list, in the worst-case, element is present at the end. Its time complexity is  $O(n)$ .

## Add new comment

Your name

Subject

Comment \*

[More information about text formats](#)

Plain text

- No HTML tags allowed.
- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

Save

Preview

- [C programs](#)
- [C programming](#)
- [C graphics](#)
- [C graphics programs](#)
- [C source codes](#)
- [Tutorials](#)
- [Data structures](#)
- [C++ programs](#)
- [Java programs](#)



## C programs

C Hello world  
Print Integer  
Addition of two numbers  
Even odd  
Add, subtract, multiply and divide  
Check vowel  
Roots of quadratic equation  
Leap year program in C  
Sum of digits  
Factorial program in C  
HCF and LCM  
Decimal to binary in C  
nCr and nPr  
Add n numbers  
Swapping of two numbers  
Reverse a number  
Palindrome number  
Print Pattern  
Diamond  
Prime numbers  
Armstrong number  
Armstrong numbers  
Fibonacci series in C  
Floyd's triangle in C  
Pascal triangle in C  
Addition using pointers  
Maximum element in array  
Minimum element in array  
Linear search in C  
Binary search in C  
Reverse array  
Insert element in array  
Delete element from array  
Merge arrays  
Bubble sort in C  
Insertion sort in C  
Selection sort in C  
Add matrices

- [Subtract matrices](#)
- [Transpose matrix](#)
- [Matrix multiplication in C](#)
- [Print string](#)
- [String length](#)
- [Compare strings](#)
- [Copy string](#)
- [Concatenate strings](#)
- [Reverse string](#)
- [Palindrome in C](#)
- [Delete vowels](#)
- [C substring](#)
- [Subsequence](#)
- [Sort a string](#)
- [Remove spaces](#)
- [Change case](#)
- [Swap strings](#)
- [Character's frequency](#)
- [Anagrams](#)
- [C read file](#)
- [Copy files](#)
- [Merge two files](#)
- [List files in a directory](#)
- [Delete file](#)
- [Random numbers](#)
- [Add complex numbers](#)
- [Print date](#)
- [Get IP address](#)
- [Shutdown computer](#)

Programming Simplified is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](#).

[Home](#) | [About](#) | [Contact](#) | [Programmer Resources](#) | [Sitemap](#) | [Privacy](#) | [Facebook](#)