

# One dimensional Array

In the simplest terms, a one-dimensional array in C is a list. Each element of the list contains only one value, whether that value be an int, char, or float. To **declare** an array, three values are typically needed: the data type, name of the array, and the number of elements to be contained in the array.

## Declaration of one dimensional Array

**We can declare an array in the c language in the following way.**

```
data_type array_name[array_size];
```

Now, let us see the example to declare the array.

```
int marks[5];
```

Here, int is the *data\_type*, marks are the *array\_name*, and 5 is the *array\_size*.

## Initialization of one dimensional Array

**The simplest way to initialize an array is by using the index of each element. We can initialize each element of the array by using the index. Consider the following example.**

1. marks[0]=80;//initialization of array
2. marks[1]=60;
3. marks[2]=70;
4. marks[3]=85;
5. marks[4]=75;

80	60	70	85	75
----	----	----	----	----

marks[0] marks[1] marks[2] marks[3] marks[4]

## **Initialization of Array**

## C array example

1. `#include<stdio.h>`
2. `int main(){`
3. `int i=0;`
4. `int marks[5];//declaration of array`
5. `marks[0]=80;//initialization of array`
6. `marks[1]=60;`
7. `marks[2]=70;`
8. `marks[3]=85;`
9. `marks[4]=75;`
10. `//traversal of array`
11. `for(i=0;i<5;i++){`
12. `printf("%d \n",marks[i]);`
13. `}//end of for loop`
14. `return 0;`
15. `}`

## Output

```
80
60
70
85
75
```

## one dimensional Array: Declaration with Initialization

We can initialize the c array at the time of declaration. Let's see the code.

```
int marks[5]={20,30,40,50,60};
```

In such case, there is **no requirement to define the size**. So it may also be written as the following code.

```
int marks[]={20,30,40,50,60};
```

Let's see the C program to declare and initialize the array in C.

1. `#include<stdio.h>`
2. `int main(){`
3. `int i=0;`
4. `int marks[5]={20,30,40,50,60};`//declaration and initialization of array
5. `//traversal of array`
6. `for(i=0;i<5;i++){`
7. `printf("%d \n",marks[i]);`
8. `}`
9. `return 0;`
10. `}`

### Output

```
20
30
40
50
60
```

### SEARCHING ELEMENT IN ONE DIMENSIONAL ARRAY

```
#include<stdio.h>
int main()
{
    int a[20],i,x,n;
    printf("how many elements?");
    scanf("%d",&n);
    printf("enter array elements:n");
    for(i=0;i<n;++i)
        scanf("%d",&a[i]);
    printf("nenter element to search:");
    scanf("%d",&x);
    for(i=0;i<n;++i)
        if(a[i]==x)
            break;
    if(i<n)
        printf("element found at index %d",i);
    else
        printf("element not found");
```

```
    return 0;  
}
```

**OUTPUT:**

How many elements ? 10

Enter array elements :

21

22

34

12

66

77

87

99

09

39

Enter element to search : 87

Element found at index 6