One Dimensional Arrays

An array which has only one subscript is known as one dimensional array.

Declaration of Array

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

data_type array_name [array_size];

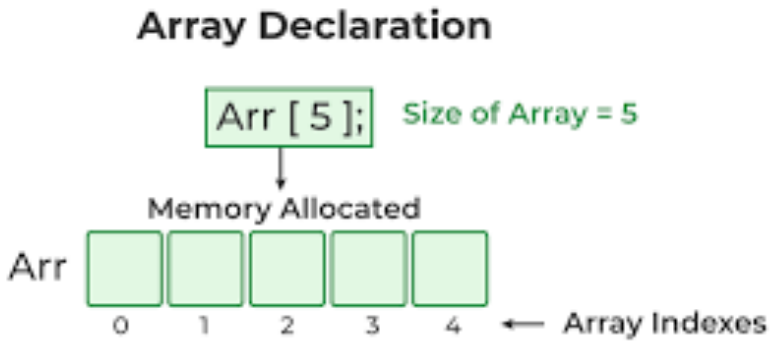
Where

- **data-type** - all the array elements should have the same data-type. Any valid data-type of C can be taken.
- **array-name** - is the name of the array.
- **array-size** - it specifies the size of the array. It tells how many elements are present in the array.

Example

int arr[5];

Here, int is the data_type, Arr is the array_name, 5 is the array_size.



Array Initialization

1. Array Initialization with Declaration

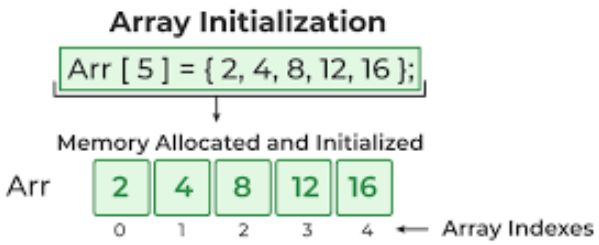
In this method, we initialize the array along with its declaration. We use an initializer list to initialize multiple elements of the array. An initializer list is the list of values enclosed within braces **{ }** separated b a comma.

Syntax:-

data_type array_name [*size*] = {value1, value2, ... valueN};

Example:-

arr[5]={2,4,8,12,16};



2. Array Initialization with Declaration without Size

If we initialize an array using an initializer list, we can skip declaring the size of the array as the compiler can automatically deduce the size of the array in these cases. The size of the array in these cases is equal to the number of elements present in the initializer list as the compiler can automatically deduce the size of the array.

Example:-

```
data_type array_name[] = {1,2,3,4,5};
```

3. Array Initialization after Declaration (Using Loops)

An array can also be initialized using a loop. The loop iterates from 0 to (size - 1) for accessing
all indices of the array starting from 0. The following syntax uses a “for loop” to initialize the array elements. This is the most common way to initialize an array in C.

```
// declare an array.
int my_array[5];
for (int i = 0; i < N; i++)
{
    array_name[i] = value;
}
```

Access Array Elements

We can access any element of an array in C using the array subscript operator **[ ]** and the index value ***i*** of the element.

array_name [*index*];

Example:-

```
#include<stdio.h>
main()
{
    int arr[5],n;
    printf("Enter the size of array :");
    scanf("%d",&n);
    printf("Enter the elements of array:");
    for (int i = 0;i < 5; i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Elements of Array: ");
    for (int i = 0; i < 5; i++)
    {
        printf("%d ",arr[i]);
    }
}
```

Output:-

```
Enter the size of array : 5
```

Enter the elements of array: 10 20 30 40 55
Elements of Array: 10 20 30 40 55

Find the min and max of a 1-D integer array.

```
#include <stdio.h>
int main()
{
	int a[1000],i,n,min,max;
	printf("Enter size of the array : ");
	scanf("%d",&n);
	printf("Enter elements in array : ");
	for(i=0; i<n; i++)
	{
		scanf("%d",&a[i]);
	}
	min=max=a[0];
	for(i=1; i<n; i++)
	{
		if(min>a[i])
		min=a[i];
		if(max<a[i])
		max=a[i];
	}
	printf("minimum element of array is : %d",min);
	printf("\n maximum element of array is : %d",max);
	return 0;
}
```

Output:-

Enter size of the array : 5
Enter elements in array : 23
21
56
43
8
minimum element of array is : 8
maximum element of array is : 56