

Arrays

An array is a collection of elements of the same type that are referenced by a common name.

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

Array Length = 9

First Index = 0

Last Index = 8

Basically, there are two types of arrays:

One dimensional array

Multidimensional array

One Dimensional Array:

Declaration:

```
dataType arrayname[arraySize];
```

```
int mark[5];
```

Access Array Elements

You can access elements of an array by indices.

Suppose you declared an array `mark` .

The first element is `mark[0]`

second element is `mark[1]` and so on.

`mark[0]` `mark[1]` `mark[2]` `mark[3]` `mark[4]`

--	--	--	--	--

Notes:

Arrays have 0 as the first index, not 1. In this example, `mark[0]` is the first element.

If the size of an array is n , to access the last element, the $n-1$ index is used. In this example, `mark[4]`

Suppose the starting address of `mark[0]` is 2120.
Then, the address of the `mark[1]` will be 2122.
the address of `mark[2]` will be 2124 and so on.

This is because the size of a int is 2 bytes.

initialize an array

Gf:

Datatype array-name[size]={value1,value2,.....,value size-1}

Example 1:

```
int mark[5] = {19, 10, 8, 17, 9};
```

Example 2:

```
int mark[] = {19, 10, 8, 17, 9};
```

Here, we haven't specified the size. However, the compiler knows its size is 5 as we are initializing it with 5 elements.

```
int x[5]={1,2,3,4,5,6,7,8,9,10};
```

Error: Too many initializers

```
int x[5]={1,2,3,4,5};  
clrscr();  
for(i=0;i<10;i++)  
printf("%d",x[i]);
```

Output:

123450344000


```
int x[10]={1,2,3,4,5};  
clrscr();  
for(i=0;i<10;i++)  
printf("%d",x[i]);
```

Output:

1234500000

Change Value of Array elements

```
int mark[5] = {19, 10, 8, 17, 9}
```

```
// make the value of the third element to -1  
mark[2] = -1;
```

```
// make the value of the fifth element to 0  
mark[4] = 0;
```

1. Write a program to read n elements and then find the sum of array elements.
2. Write a program to read n elements and then find the average of array elements.
3. Write a program in C to copy the elements of one array into another array.
4. Program to print an array in reverse order
5. C Program to Put Even & Odd Elements of an Array in 2 Separate Arrays
6. C Program to Sort the Array in an Ascending Order
7. Write a program in C to find the maximum and minimum element in an array

Two Dimensional Array

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns.

Size of multidimensional arrays

Total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

For example:

The array **int x[10][20]** can store total $(10 * 20) = 200$ elements.

Similarly array **int x[5][10][20]** can store total $(5 * 10 * 20) = 1000$ elements.

declaring a two-dimensional array of size x, y:

Syntax:

```
data_type array_name[x][y];
```

Example

```
int x[10][20];
```

Column 0 Column 1 Column 2

Row 0

x[0][0]

x[0][1]

x[0][2]

Row 1

x[1][0]

x[1][1]

x[1][2]

Row 2

x[2][0]

x[2][1]

x[2][2]

Initializing Two – Dimensional Arrays:

There are two ways in which a Two-Dimensional array can be initialized.

First Method:

```
int x[3][4] = {0, 1 ,2 ,3 ,4 , 5 , 6 , 7 , 8 , 9 , 10 , 11}
```

The above array have 3 rows and 4 columns.

The elements will be filled in the array in the order, first 4 elements from the left in first row, next 4 elements in second row and so on

Better Method:

```
int x[3][4] = {{0,1,2,3}, {4,5,6,7}, {8,9,10,11}};
```

This type of initialization make use of nested braces.

Each set of inner braces represents one row. In the above example there are total three rows so there are three sets of inner braces.

```
int Employees[ ][ ] = { {10, 20, 30}, {15, 25, 35},  
{22, 44, 66}, {33, 55, 77} };
```

Here, We haven't mentioned the row size and column size.

The compiler calculate the size by checking the number of elements inside the row and column.

We can also write this two dimensional array as:

```
int Employees[ ][3] = { {10, 20, 30}, {15, 25, 35},  
{22, 44, 66}, {33, 55, 77} };
```


