

UNIT–III : Arrays & Strings

Arrays

Definition: An array is a collection of elements of same type that share a common name.

The elements of an array are stored in contiguous memory locations. In 'C' language, arrays are classified into two types. They are

1. One Dimensional Arrays
2. Multi Dimensional Arrays

One dimensional array

When an array is declared with only one dimension (subscript) then it is called "Onedimensional array" or "single dimensional array".

Declaration of one dimensional Array:-

The general format of declaring a one dimensional array is as follows:

Syntax:-

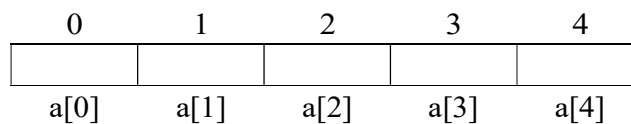
datatype arrayname[size] ;

In the above syntax,

- The datatype is any valid data type of 'C' language. An array can hold all the values based on the specified data type.
- The 'arrayname' is an identifier that specifies name of the array variable. All the elements will share this variable name.
- The 'size' indicates maximum number of elements that an array can hold. It must be a positive integer constant.

Example-1: int a[5];

The above declaration reserves 5 contiguous memory locations of integer data type for the array 'a'. The conceptual view for the above declaration is as shown below:



Storing values into the Array

To store the array elements, we follow the following two methods. They are:

- Initialization of the array
- Reading the array elements

Initialization of the Array:-

While declaring an array, we can initialize it with some values. The general format to initialize one dimensional array is as follows:

Syntax:-

datatype arrayname[size] = {V₀, V₁, ..., V_{size-1}} ;

Here, **V₀, V₁, ..., V_{size-1}** are the initial values of the specified array. If we omit the values then an array contains garbage values. The number of values must be less than or equal to the size of the array. When there are few values then the remaining elements are assigned with zeros.

Ex:

int a[5] = {78, 34, 98, 90, 124}; □

a[0]	a[1]	a[2]	a[3]	a[4]
78	34	98	90	124

Reading the arrays elements: To read the array elements thru keyboard, we write and execute the following example

```
for(i=0;i<n;i++)
scanf("%d",
&num[i]);
```

Accessing elements of the Array :-

Once we store the elements of an array, we may access the elements of the array. To access the array elements, we use the index numbers of the array. The index numbers are used to identify a particular location of the array. The following example is used to accessing the elements of an array and prints them on the screen.

```
main()
{
    int a[]={ 5, 6, 9, 7, 3};
    i
    n
    t
    i
    ;
    c
    l
    r
    s
    c
    r
```

```

(
)
;

printf("The
Elements are \n");
for(i=0 ; i<5; i++)

    printf("
%d \n", a[i]);
    getch();

}

```

Two-Dimensional Arrays

When an array uses only two subscripts then it is called “two-dimensional array”. A two-dimensional array is useful for matrix operations.

Declaration of Two-Dimensional Array:-

The general format of declaring a two dimensional array is as follows:

Syntax:-

datatype arrayname[rowsize][columnsize] ;

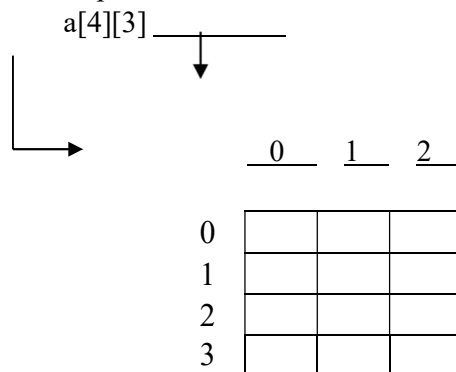
In the above syntax,

- The datatype is any valid data type of ‘C’ language. An array can hold all the values based on the specified data type.
- The ‘arrayname’ is an identifier that specifies name of the array variable. All the elements will share this variable name.
- The ‘rowsize’ indicates maximum number of rows and ‘columnsize’ indicates number of columns in a row.

Example: int a[4][3];

Here, it reserves 12 (4 rows x 3 columns) integer type memory locations for the array ‘a’.

The conceptual view for the above declaration is as shown below:



Initialization of Two Dimensional Array:-

While declaring an array, we can initialize it with some values. The general format to initialize two dimensional array is as follows:

Syntax:-

datatype arrayname[rowsize][columnsize] = {{row1 values}, {row2 values}, ..., } ;

Here, If we omit the values then an array contains garbage values. The number of values must be less than or equal to the size of the array. When there are few values then the remaining elements are assigned with zeros.

Example-1:

```
int a[2][3] = { {4, 6, 8}, {1, 3, 5}};
```

This example initializes the array 'a' with the given values as shown below:

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

Example-1:

```
int a[2][3] = { {4, 8}, {5}};
```

This example initialises the array 'a' with the given values as shown below:

a	0	1	2
0	4	8	0
1	5	0	0

Accessing of two dimensional arrays:-

Each two dimensional array element is referred or accessed by specifying the array name followed by two subscripts. The subscripts must be enclosed within square brackets.

For example,

a[0][0] refers to the first element in the two dimensional array. a[rowsize-1][colsize-1] refers to the last element of the array.

The remaining elements can be accessed as a[0][1], a[2][3], a[1][2] and so on.

The two dimensional array that contains 'm' rows and 'n' columns can be read using loops as follows:

```
for(i=0;i<m;i++)
for(j=0;j<n;j++)
)
scanf("%d",&a[i][j]);
```