SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

COURSE NAME : 23ITT101- PROBLEM SOLVING & C PROGRAMMING

I YEAR /I SEMESTER

Unit II – ARRAYS AND STRINGS

Topic : Array

INTRODUCTION

Arrays

- Two-dimensional arrays
- Initializing of two dimensional arrays
- Multi –Dimensional Arrays

Arrays

- An array in C is a collection of elements of the same data type stored in contiguous memory locations.
- It is a fundamental data structure that helps to store and manage multiple values efficiently.
- Arrays are the **derived data type in C**

General Form:

```
datatype arrayName [size ];
```

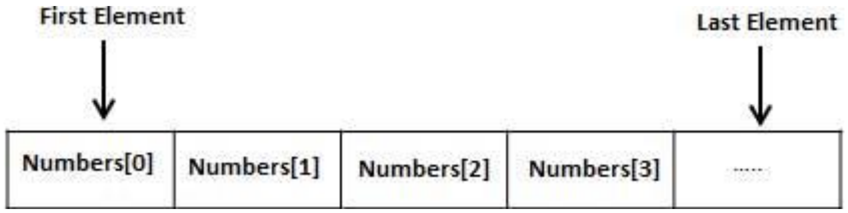
data_type: Type of data to be stored in the array. Here data_type is valid C data type

arrayName: Name of the array

size: Size of the array dimensions

- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

int Numbers[10];



Initialization :

At the time of declaration:

```
int arr[5] = {1, 2, 3, 4, 5};
```

Default initialization (all elements set to 0 for global/static arrays):

```
int arr[5] = {0};
```

Partially initialized (remaining elements set to 0):

```
int arr[5] = {1, 2};
```

Accessing Elements:

Array elements are accessed using an index (starting from 0).

```
arr[0] = 10; // Sets the first element

int x = arr[1]; // Retrieves the second element
```

Types of Arrays:

One-Dimensional Array - A single row of elements.

```
#include <stdio.h>
int main() {
    int arr[5] = {10, 20, 30, 40, 50};
   printf("Elements of the array:\n");
    for (int i = 0; i < 5; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Two-Dimensional Array

- An array of arrays, often visualized as a matrix
- The basic form of declaring a two-dimensional array of size x, y:

Syntax:

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

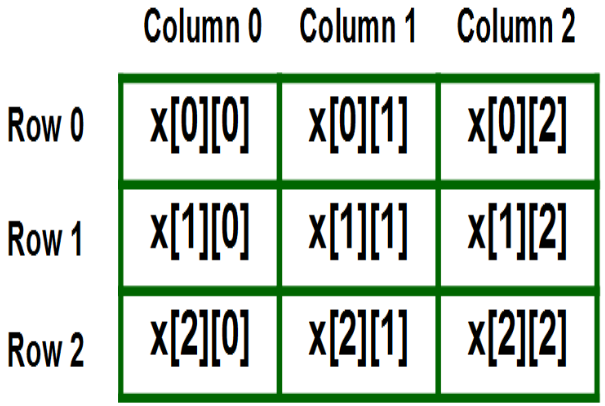
data_type: Type of data to be stored. Valid C/C++ data type.

can declare a two dimensional integer array say ‘x’ of size 10,20 as:

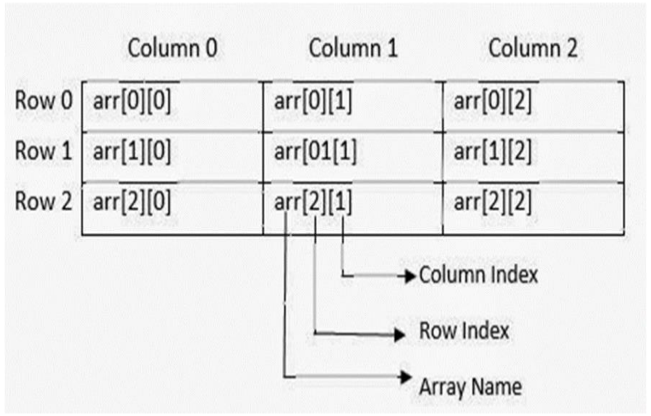
```
int x[10][20];
```

Elements in two-dimensional arrays are commonly referred by x[i][j] where i is the row number and ‘j’ is the column number.

- A two – dimensional array can be seen as a table with ‘x’ rows and ‘y’ columns where the row number ranges from 0 to (x-1) and column number ranges from 0 to (y-1).
- A two – dimensional array ‘x’ with 3 rows and 3 columns is shown below:



	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]



	Column 0	Column 1	Column 2
Row 0	arr[0][0]	arr[0][1]	arr[0][2]
Row 1	arr[1][0]	arr[01[1]	arr[1][2]
Row 2	arr[2][0]	arr[2][1]	arr[2][2]

Declaration and Initialization:

```
int arr[3][3] = {

    {1, 2, 3},

    {4, 5, 6},

    {7, 8, 9}

};
```

Accessing Elements:

Access element in the 2nd row, 3rd column

```
int x = arr[1][2];
```

Two-dimensional Array – Compile Time Initialization Example

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

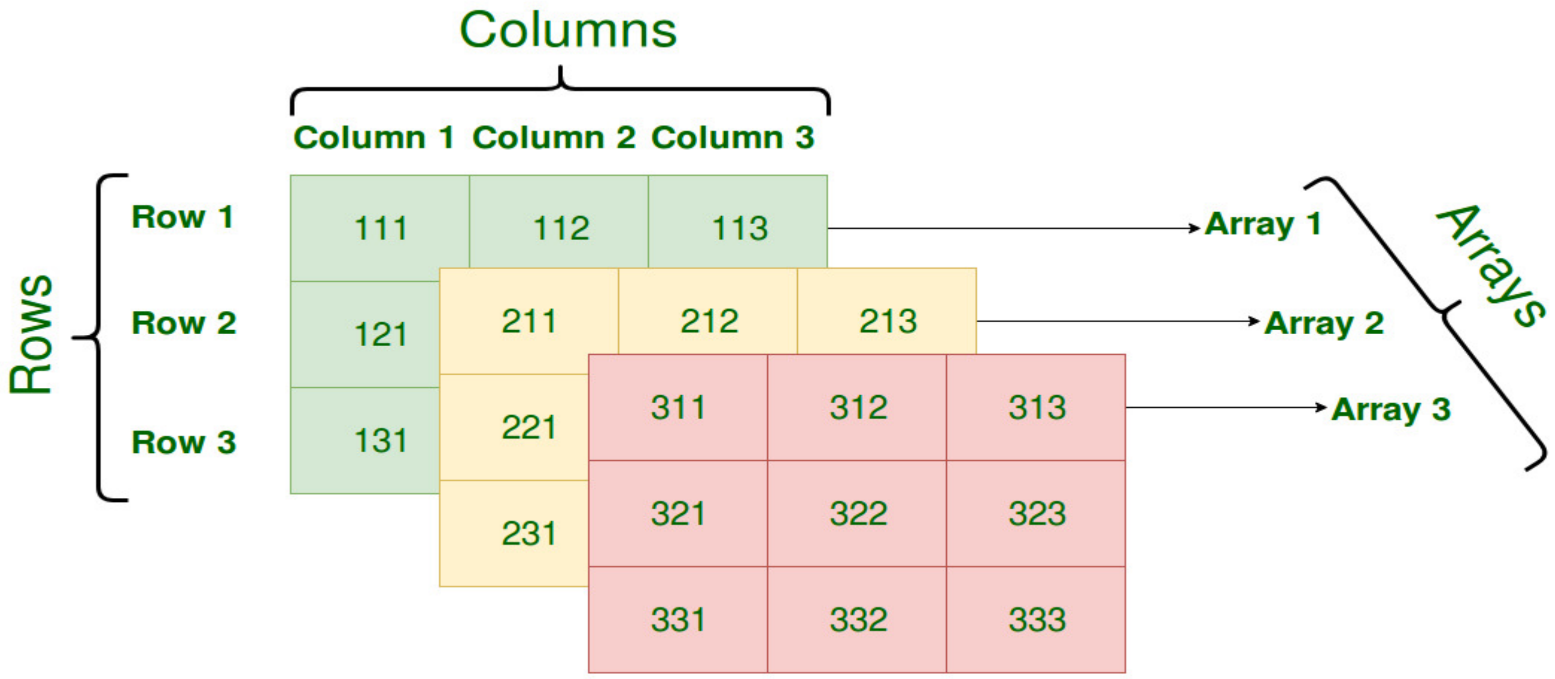
	column[0]	column [1]	column [2]
row[0]	1	2	3
row[1]	4	5	6
row[2]	7	8	9

EXAMPLE

```
#include <stdio.h>
int main()
{
    int matrix[2][2] = {
        {1, 2},
        {3, 4}
    };

    printf("Elements of the 2x2 matrix:\n");
    for (int i = 0; i < 2; i++)
   {
        for (int j = 0; j < 2; j++)
        {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Multidimensional Arrays in C



Initializing Three-Dimensional Array

- Initialization in Three-Dimensional array is same as that of Two-dimensional arrays.
- The difference is as the number of dimension increases so the number of nested braces will also increase.

Method 1:

```
int x[2][3][4] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19,
 20, 21, 22, 23};
```

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.

  Two – dimensional array is the simplest form of a multidimensional array.

Key Points

- Fixed Size:

  The size of an array must be specified at the time of declaration (for static arrays).

- Contiguous Memory:

  Elements are stored in contiguous memory locations.

- Indexing:

  Array indices start from 0 and go up to size - 1.

- Boundary:

  Accessing out-of-bounds indices leads to undefined behavior.

Summary

- An array is **a variable that can store multiple values**. A **one dimensional array** is a set of single dimensional **arrays** of same size and type allocated in adjacent memory allocations.
- A **two dimensional array** is also defined as a table of data of same type arranged in rows and columns
- Total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

Thank you