

UNIT III

ARRAYS AND FUNCTIONS

ARRAYS (Lecture 3)

Recapitulations

- Compile time Array initialization
- Runtime Array initialization
- Advantages of an Array in C
- Disadvantages of an Array in C
- Access elements out of its bound

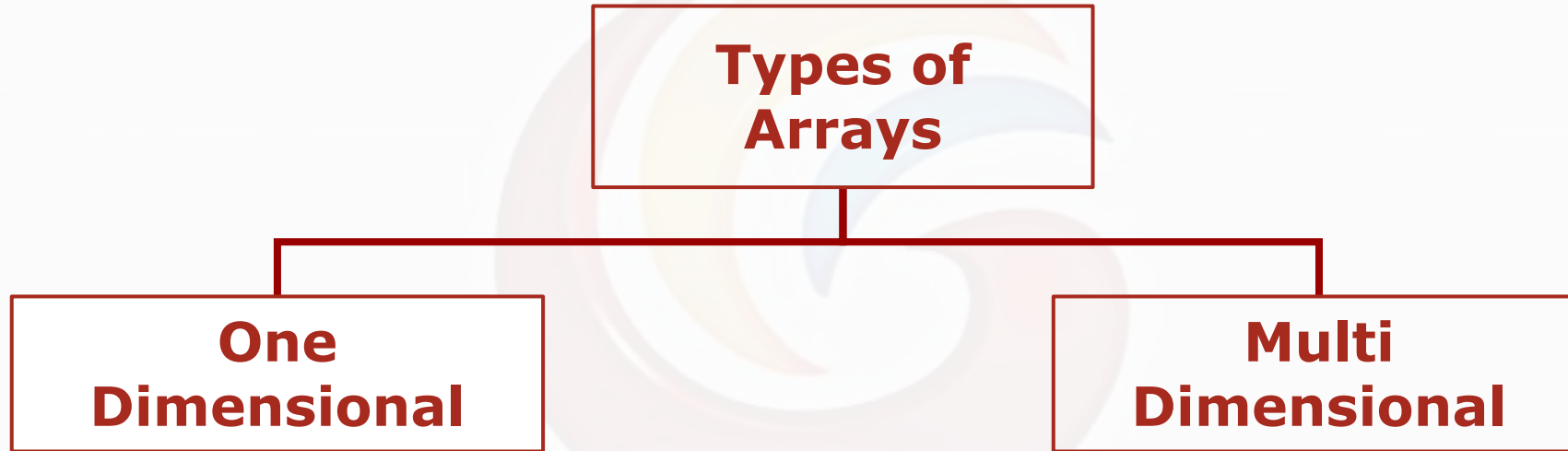
GALGOTIAS
UNIVERSITY

Objective

- Types of Arrays
- One- dimensional Array
- Multi- dimensional Array
- Examples of multidimensional Arrays

GALGOTIAS
UNIVERSITY

Types of Arrays

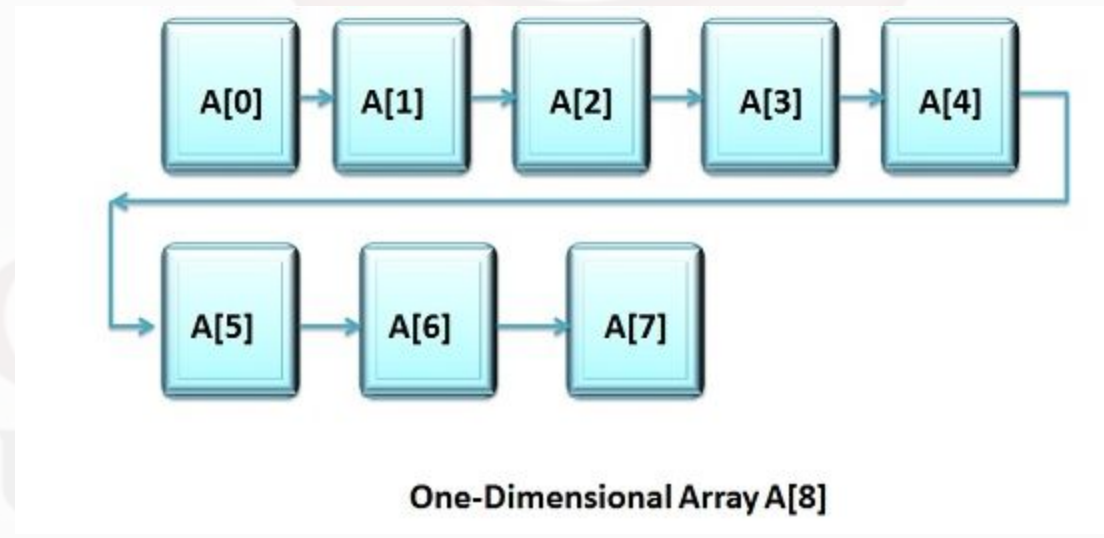


GALGOTIAS
UNIVERSITY

One Dimensional Array (1D)

One-Dimensional or Single-Dimensional array is considered as the “list of variables of similar data types”, and each variable can be distinctly accessed by specifying its index in square brackets preceded by the name of that array.

Conceptually you can think of a one-dimensional array as a row, where elements are stored one after another.

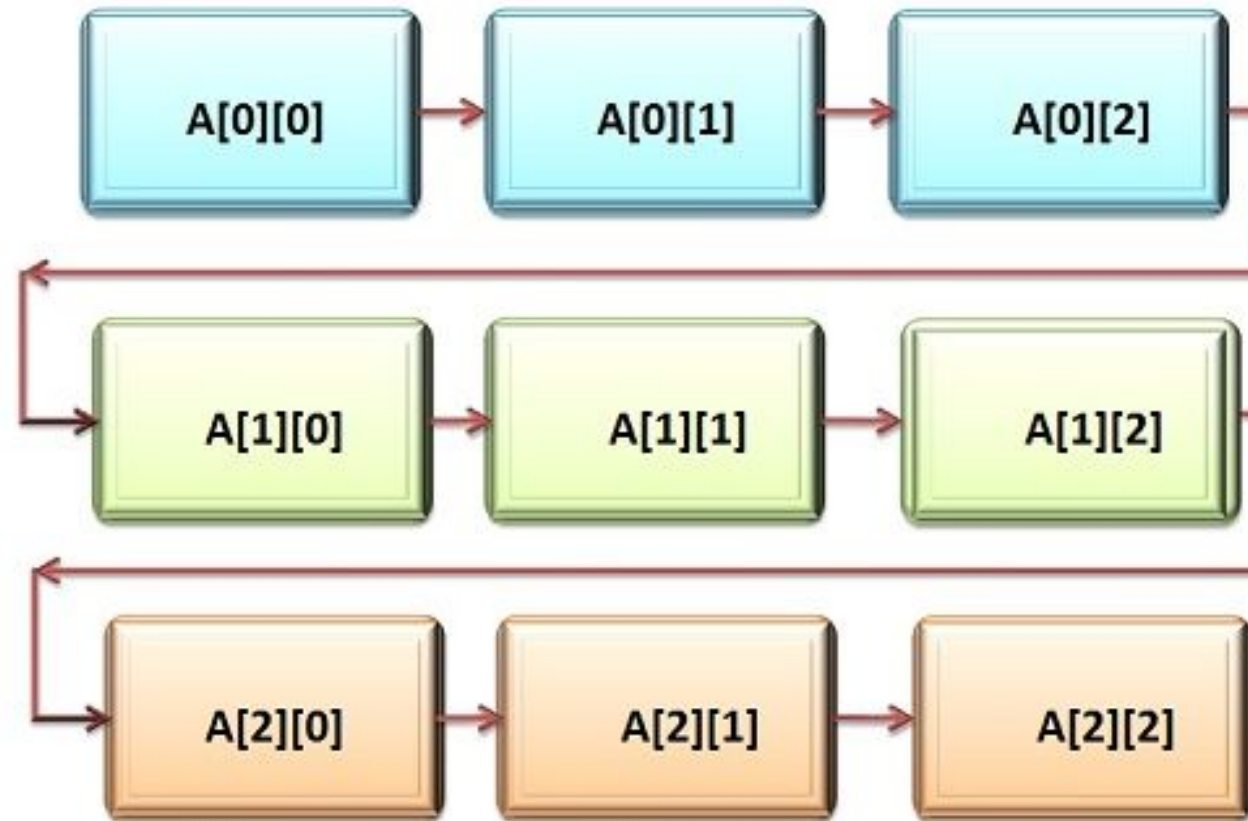


Multidimensional Arrays

- In C programming, you can create an array of arrays. These arrays are known as multidimensional arrays
- One of the simplest forms of a multidimensional array is a two-dimensional array or 2-D array.
- A two-dimensional array is stored in the form of the row-column matrix, where the first index designates the row and second index shows the column.
- You can also declare 3- D Arrays

GALGOTIAS
UNIVERSITY

Two Dimensional Array



Two-Dimensional Array A[3][3]

Declaring Two Dimensional Array

Example:

```
float x[3][4];
```

| | Column 1 | Column 2 | Column 3 | Column 4 |
|-------|-----------|-----------|-----------|-----------|
| Row 1 | x [0] [0] | x [0] [1] | x [0] [2] | x [0] [3] |
| Row 2 | x [1] [0] | x [1] [1] | x [1] [2] | x [1] [3] |
| Row 3 | x [2] [0] | x [2] [1] | x [2] [2] | x [2] [3] |

Here, x is a two-dimensional (**2d**) array. The array can hold **12** elements. You can think the array as a table with 3 rows and each row has 4 columns.

Similarly, you can declare a three-dimensional (**3d**) array.

Example:

Here, the array y can hold **24** elements.

```
float y[2][4][3];
```


Initializing a multidimensional array

Different ways to initialize two-dimensional array

```
int c[2][3] = {{1, 3, 0}, {-1, 5, 9}};
```

```
int c[][3] = {{1, 3, 0}, {-1, 5, 9}};
```

```
int c[2][3] = {1, 3, 0, -1, 5, 9};
```

Initializing a multidimensional array

You can initialize a **three-dimensional** array in a similar way like a two-dimensional array. Here's an example,

```
int test[2][3][4] = {  
    {{3, 4, 2, 3}, {0, -3, 9, 11}, {23, 12, 23, 2}},  
    {{13, 4, 56, 3}, {5, 9, 3, 5}, {3, 1, 4, 9}}};
```

GALGOTIAS
UNIVERSITY

Example: Two-dimensional array to store and print values

```
// C program to store temperature of two cities of a week and display it.
#include <stdio.h>
const int CITY = 2;
const int WEEK = 7;
int main()
{
    int temperature[CITY][WEEK];

    // Using nested loop to store values in a 2d array
    for (int i = 0; i < CITY; ++i)
    {
        for (int j = 0; j < WEEK; ++j)
        {
            printf("City %d, Day %d: ", i + 1, j + 1);
            scanf("%d", &temperature[i][j]);
        }
    }
    printf("\nDisplaying values: \n\n");

    // Using nested loop to display vlues of a 2d array
    for (int i = 0; i < CITY; ++i)
    {
        for (int j = 0; j < WEEK; ++j)
        {
            printf("City %d, Day %d = %d\n", i + 1, j + 1, temperature[i][j]);
        }
    }
    return 0;
}
```

Program Name: B.Tech (CSE)

Summary

- ❖ Types of Arrays - One dimensional & Multidimensional
- ❖ A two-dimensional array is stored in the form of the row-column matrix
- ❖ To access the entire 2D array we have to do looping against the rows and columns

GALGOTIAS
UNIVERSITY

References

- E. Balaguruswamy 7th Edition, Programming ANSI C, McGraw-Hill
- Brian W. Kernighan and Dennis M. Ritchie, The C programming Language, Prentice-Hall in 1988
- Byron Gottfried, Programming with C, Schaum's Outline

GALGOTIAS
UNIVERSITY

Dr. Saumya Chaturvedi

saumya.chaturvedi@galgotiasuniversity.edu.in

GALGOTIAS
UNIVERSITY



Thank You