#17 DAY

C++ COMPLETE BOOTCAMP

INSPIRE CLUB, MANIT BHOPAL

BRINGS

C++ Complete Bootcamp

Learn How To Apply Problem Solving Skills

Hello CPPBudddies

DAY #17

Welcome

To

C++ COMPLETE BOOTCAMP

Your Guide To A Solid Foundataion in C++

Let us begin

LAST CLASS: Pointers & Memory

DISCUSSED: POINTERS & DOUBLE POINTERS

Program #3

Write a program to input **N numbers** and get the **largest** and the **smallest** number.

Program #4

WAP to input a number and print the sum of all positive and negative numbers separately and find the difference in their sums

Program #5

Simple Searching In Array

Given an array of n elements
search for a particular
element k in the array.
Print "YES" or "NO" accordingly.

Advantages of C++ Array

- Code Optimization (less code)
- Random Access
- Easy to traverse data
- Easy to manipulate data
- Easy to sort data etc.

Disadvantages of C++ Array

- Fixed size

INTERVIEW PROBLEM

Write a program to reverse an array

[1, 2, 3, 4, 5] => [5, 4, 3, 2, 1]

[2, 4, 1, 3] => [3, 1, 4, 2]

C++ Array Types

There are 2 types of arrays in C++ programming:

- **Single Dimensional Array**
- **Multidimensional Array**

MULTI-DIMENSIONAL ARRAYS

2-D Arrays

It is an array of 1D-arrays

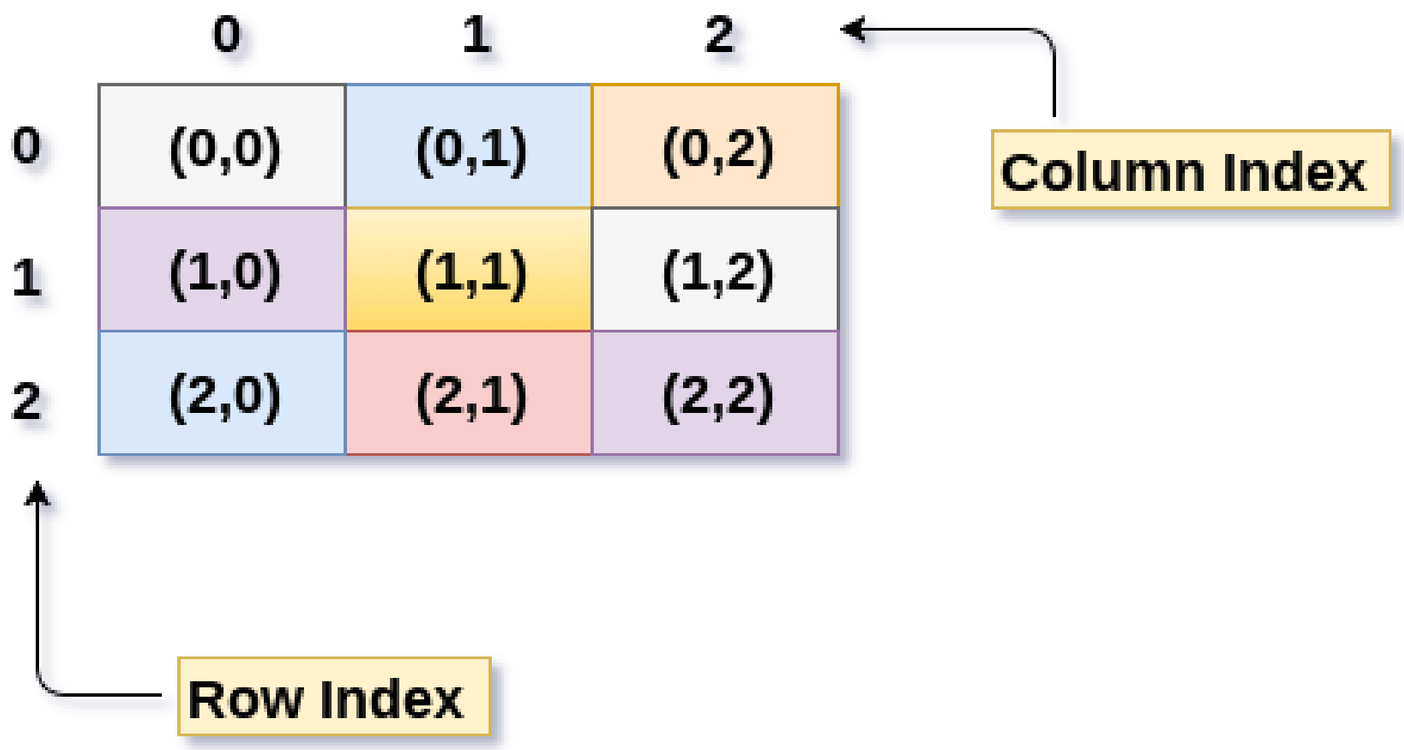
It is known as a matrix

It has M * N elements

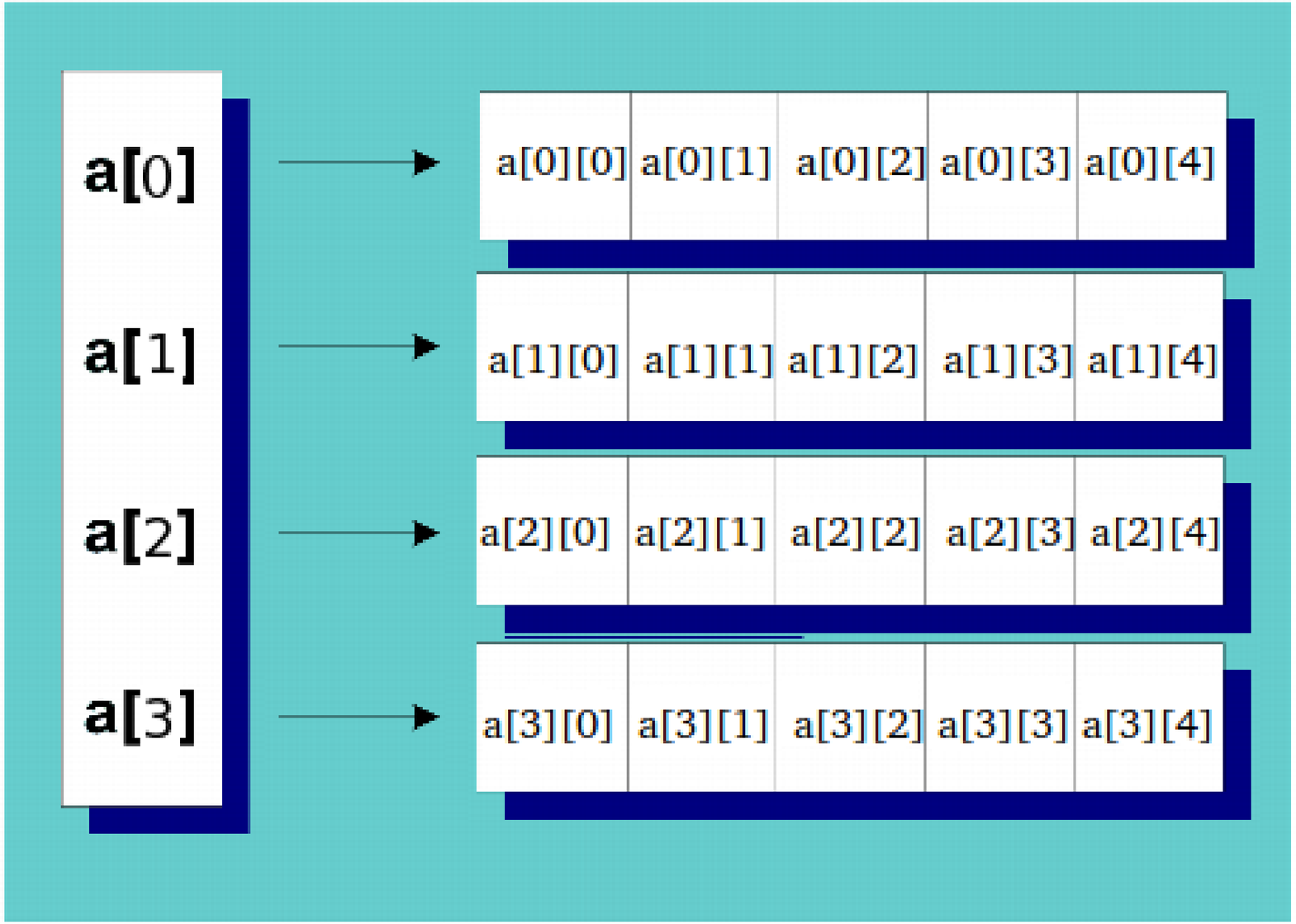
M = no. of rows

N = no. of colums

int arr[3][3];



2D Array: It is an array of 1D-arrays



How is a 2D array stored in the memory

1 D ARRAY:

C	O	D	I	N	G	E	E	K
0	1	2	3	4	5	6	7	8

single row of elements

2 D ARRAY:

i \ j	col 0	col 1	col 2
row 0	A	A	A
row 1	B	B	B
row 2	C	C	C

column

array elements

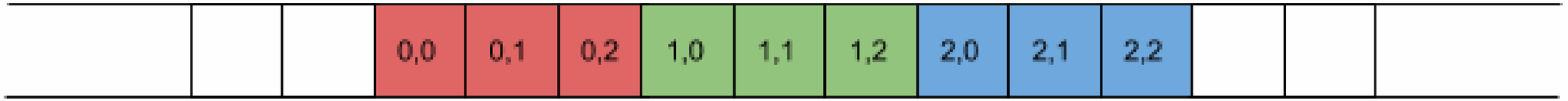
rows

In memory, a 2d array is also stored in a **contiguous** fashion

Row - Major Order

row,col

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2



Declaration

SYNTAX: type name[size_row][size_col];

int arr[2][3];

string values[3][5];

double cows[5][10];

Accessing & Assigning & Changing values

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

// Assigning Values
x[0][0] = 10;
x[1][0] = 20;

// Accessing Values
cout<< x[0][0] << " " << cout << x[1][0] << endl;

// Changing Values
x[0][0] = 30;

// Accessing Values
cout<< x[0][0] << " " << cout << x[1][0] << endl;
```

Pictorial Representation

	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]

Declaration + Initialization

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

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

Two dimensional array:

int two_d[10][20];

Three dimensional array:

int three_d[10][20][30];

Size of multidimensional arrays

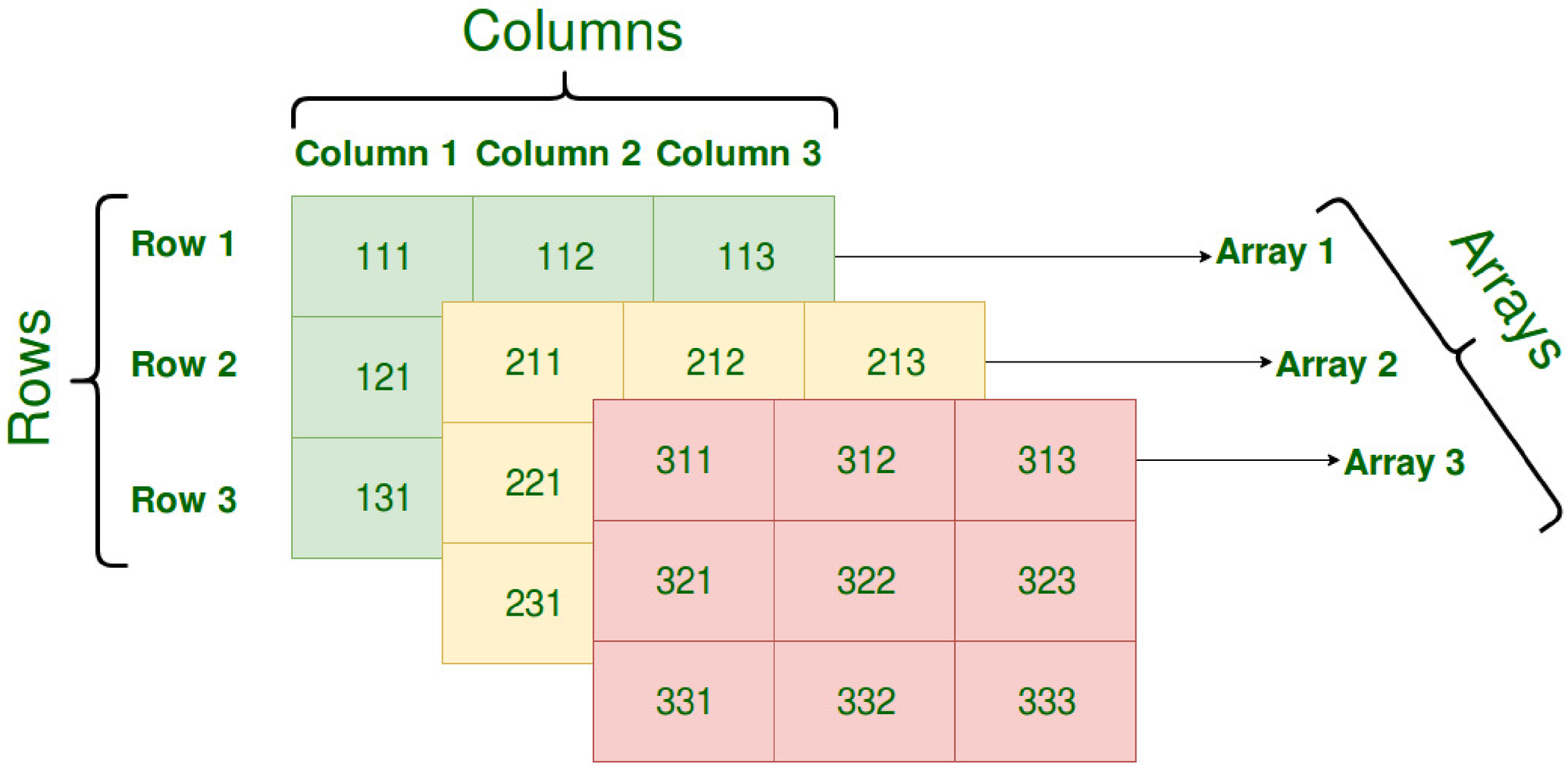
Total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

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.

3D Array: It is an array of 2D arrays



We will focus primarily on
1D Array (Array)
2D Array (Matrix)

We will not work with 3D arrays

THANK YOU

keep calm,
wear mask,
and
study hard

ISSUED IN PUBLIC INTEREST

whoami

AKASH MAJI
[TCS DIGITAL]
Your Mentor