# Lecture 13&14

## Multidimensional Array

# Intro Problem

```
Problem Name:"Tracking Monthly Sales"
Problem Statement:"Awesh works at a retail company and is responsible for tracking monthly sales for
different products. He needs an efficient way to store and access this data."
Objective: "Awesh wants to store the sales data for 3 products over 4 months."
Products: "Product A, Product B, and Product C."
Months: "January, February, March, and April."
```

# Naive approach

Declare variable for each product each month.For example ProdAJan, ProdAFeb, ProdBJan ….
In this approach we need to declare 3*4 = 12 variables

# Problem with this approach?

1. **Complexity and Clutter:**
Managing and initialising 12 variables is cumbersome and error-prone.
The code becomes cluttered and harder to read.
2. **Scalability Issues:**
Adding more products or months requires adding many more variables.
This approach doesn’t scale well with increased data.
3. **Inefficient Data Access:**
Accessing and manipulating the data requires multiple lines of code.
Looping through the data for operations (e.g., calculating totals) is impractical.
4. **Reduced Maintainability:**
Updating or modifying the data structure becomes tedious.
High likelihood of making mistakes while handling many variables.
5. **Lack of Structure:**
No clear organisation of data, making it difficult to manage relationships between products and months.
Harder to debug and track issues.

# Better Approach: We need a table

| | January | February | March | April |
|---|---|---|---|---|
| **Product A** | 100 | 150 | 130 | 120 |
| **Product B** | 200 | 180 | 210 | 220 |
| **Product C** | 90 | 110 | 85 | 95 |

# Let's make the table simpler

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | (0,0) | (0,1) | (0,2) | (0,3) |
| 1 | (1,0) | (1,1) | (1,2) | (1,3) |
| 2 | (2,0) | (2,1) | (2,2) | (2,3) |

This is a 2D array !!!

# What is a Multidimensional Array?

**Definition**: An array of arrays, allowing multiple dimensions.

**Types**:
1) 2D Arrays: Arrays of arrays (like a matrix).
2) 3D Arrays: Arrays of 2D arrays.
3) Higher Dimensions: Arrays of arrays of arrays, and so on.

| | 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]` |

# Applications

Mathematical computations:
Matrices, vectors.

Graphics and Images:
Pixels in an image.

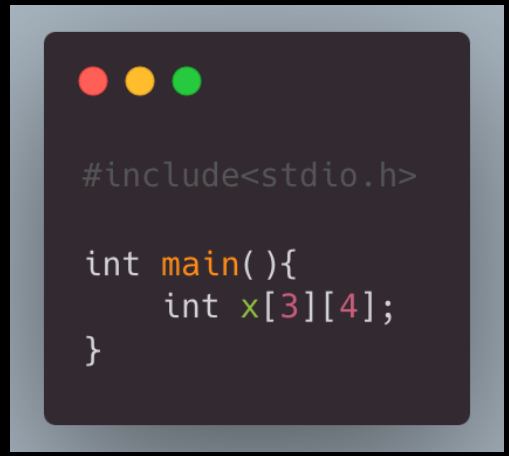
Data Science:
Data tables, spreadsheets.

Game Development:
Grids, maps.

# Declaring and Initialising 2D Array

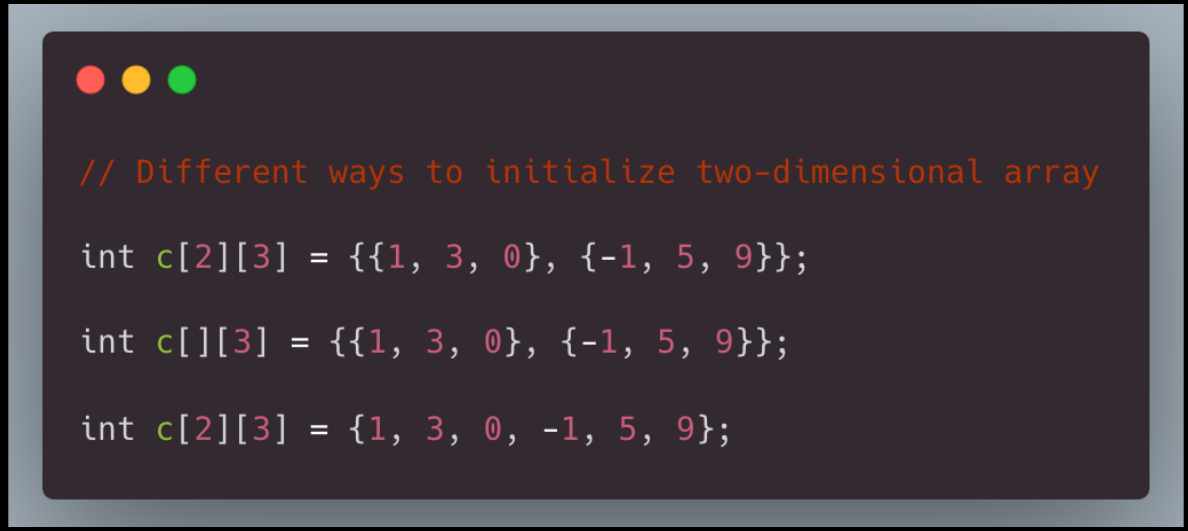
Declaration:

*dataType name[row][column];*



```c
#include<stdio.h>

int main(){
    int x[3][4];
}
```

Initialisation:



```c
// 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};
```

# Access Array Elements

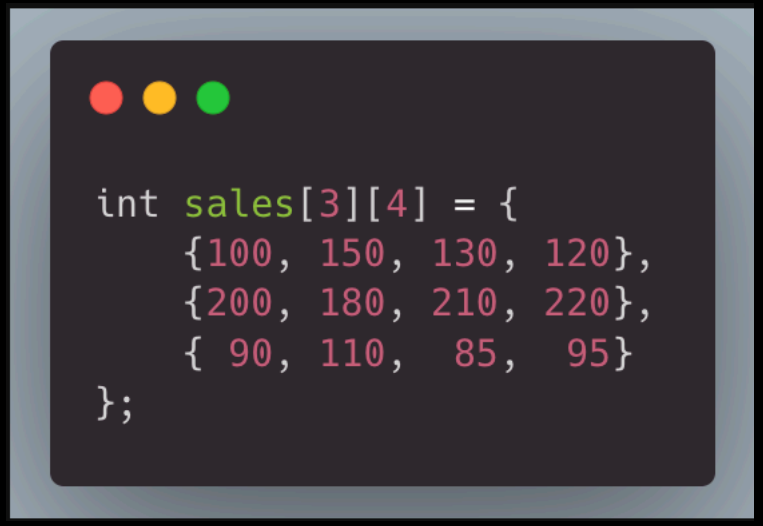
```
arr[1][0]   =   5;
```

Row number
(starts from 0)

Column number
(starts from 0)

| | 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] |

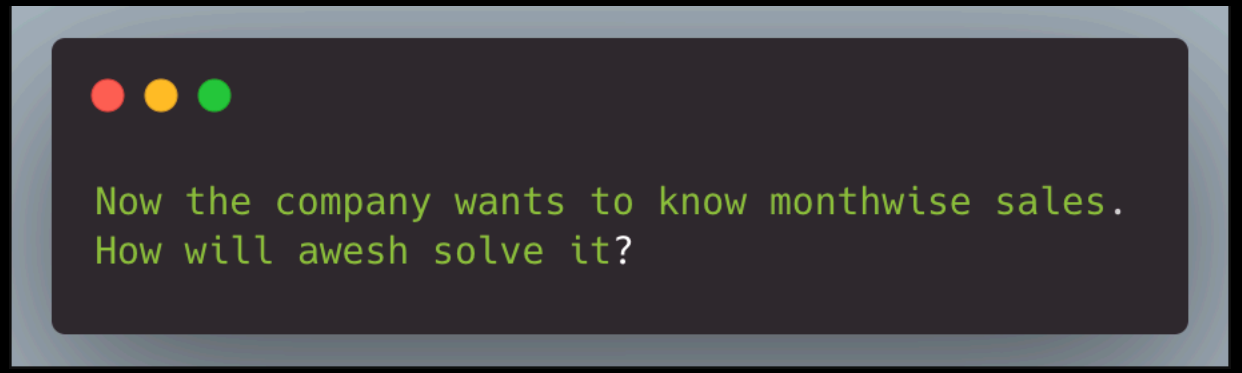
# Lets solve the intro problem



```
int sales[3][4] = {
    {100, 150, 130, 120},
    {200, 180, 210, 220},
    { 90, 110,  85,  95}
};
```

| | January (Col 0) | February (Col 1) | March (Col 2) | April (Col 3) |
|---|---|---|---|---|
| **Product A (Row 1)** | 100 | 150 | 130 | 120 |
| **Product B (Row 2)** | 200 | 180 | 210 | 220 |
| **Product C (Row 3)** | 90 | 110 | 85 | 95 |

# Intro problem update-1



# Accessing column elements

```c
for (int i = 0; i < 3; i++) {
    printf("Product %c, February: %d\n", 'A' + i, sales[i][1]);
}
```

0 for January
2 for march
3 for April

# Intro problem update-1 Solve

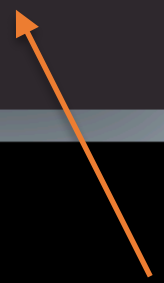
```c
#include<stdio.h>
#include<string.h>
int main(){
  int sales[3][4] = {
        {100, 150, 130, 120},
        {200, 180, 210, 220},
        { 90, 110,  85,  95}
  };
  char month_name[10];

  //Outer loop to decide month
  for(int j = 0;j < 4;j++){

    //Inner loop to decide which product
    for (int i = 0; i < 3; i++) {

        //Decide month name
        if(i == 0) strcpy(month_name,"January");
        else if(i == 1) strcpy(month_name,"February");
        else if(i == 2) strcpy(month_name,"March");
        else strcpy(month_name,"April")

        printf("Product %c, %s: %d\n", 'A' + i,month_name, sales[i]
[j])}
  }
}
```

# Intro problem update-2

Now company wants productwise sales which means how much of product A is sold in total 4 months.
How will awesh solve it?

# Accessing row elements

```c
for (int j = 0; j < 4; j++) {
    printf("Product C, Month %d: %d\n", j+1, sales[2][j]);
}
```



0 for Product A
1 for Product B

# Intro problem update-2 Solve

Solve it Yourself

# Problem-2 2D array I/O

Awesh works in a weather forecast company. He needs to make an app, which takes tempratures as inpute and shows them in your terminal.

# Solve

```c
// 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;
}
```

# Problem Sum of matrices

Add two 3x3 matrices and store the result in a third matrix.

```c
#include <stdio.h>

// Function to add two 3x3 matrices
void sumMatrices(int matrix1[3][3], int matrix2[3][3], int result[3][3])
{   for (int i = 0; i < 3; i++) { // Loop through rows
        for (int j = 0; j < 3; j++) { // Loop through columns
            // Sum corresponding elements
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}
```

# Problem Sum of matrices

```c
int main() {
    // Define two 3x3 matrices
    int matrix1[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int matrix2[3][3] = {
        {9, 8, 7},
        {6, 5, 4},
        {3, 2, 1}
    };

    // Array to store the result
    int result[3][3];

    // Call the function to sum the matrices
    sumMatrices(matrix1, matrix2, result);

    // Print the result
    printf("Sum of the two matrices:\n");
    for (int i = 0; i < 3; i++) { // Loop through rows
        for (int j = 0; j < 3; j++) { // Loop through columns
            printf("%d ", result[i][j]); // Print each
element }
        printf("\n"); // Newline for next row
    }

    return 0;
}
```

# Problem Matrix Transpose

```c
#include <stdio.h>

// Function to transpose a 3x3 matrix
void transposeMatrix(int matrix[3][3], int transposed[3][3])
{    for (int i = 0; i < 3; i++) { // Loop through rows
        for (int j = 0; j < 3; j++) { // Loop through columns
            // Assign the transposed element
            transposed[j][i] = matrix[i][j];
        }
    }
}
```

# Problem multiplication of matrix

Multiply two 2x2 matrix

```c
#include <stdio.h>

// Function to multiply two 2x2 matrices
void multiplyMatrices(int matrix1[2][2], int matrix2[2][2], int result[2][2]) {
    for (int i = 0; i < 2; i++) { // Loop through rows of the first matrix
        for (int j = 0; j < 2; j++) { // Loop through columns of the second
matrix
            result[i][j] = 0; // Initialize the result matrix element
            for (int k = 0; k < 2; k++) { // Loop to calculate dot product
                result[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
}
```

# Problem multiplication of matrix

Multiply two 3x3 matrix

Solve it Yourself

# Array of strings

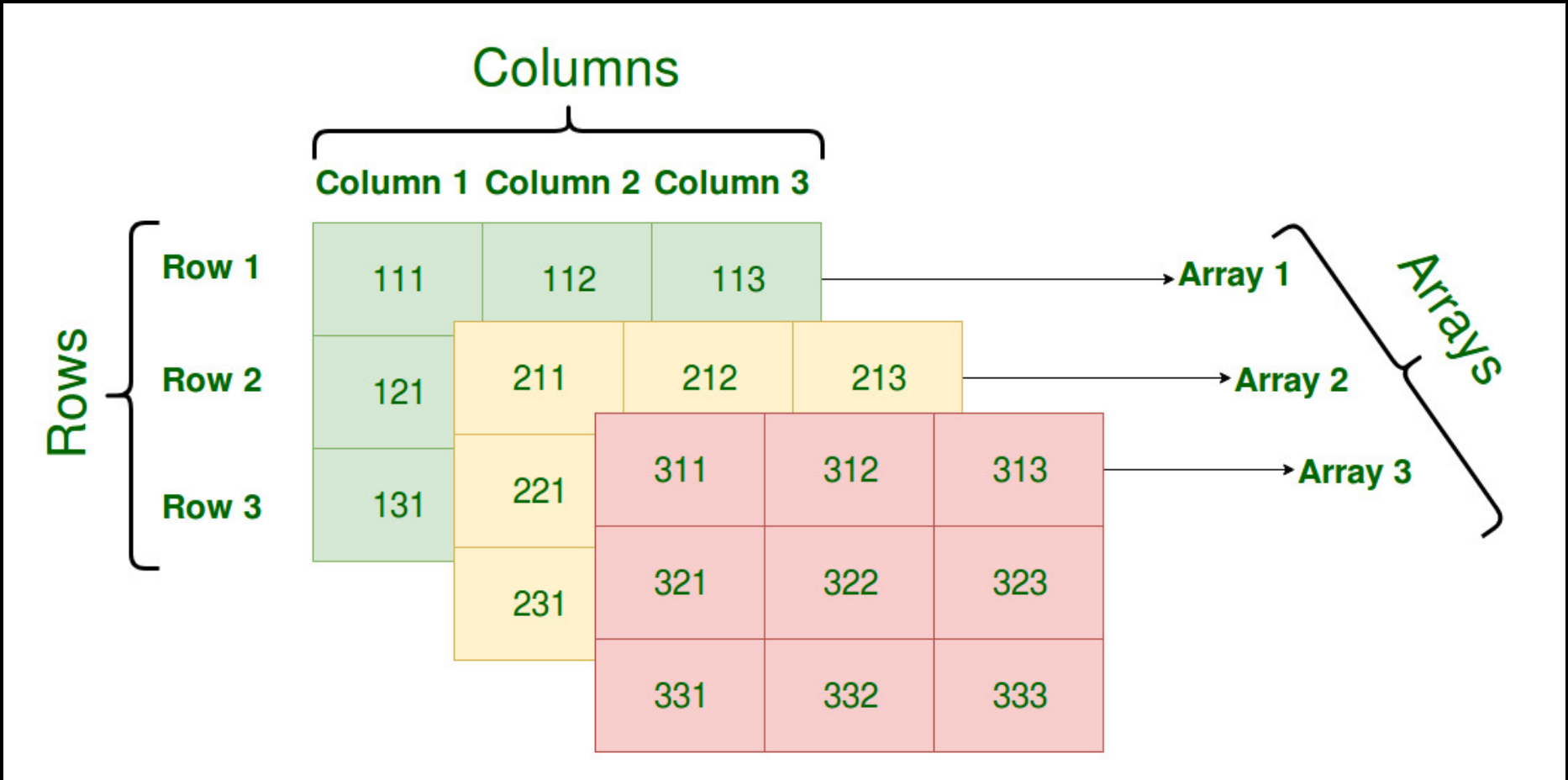
Problem statement: "Awesh wants to store the names of all the countries in the South Asian Association for Regional Cooperation (SAARC). He needs a way to efficiently store and access these country names in his C program."
Objective: "Awesh needs to store the names of the 8 SAARC countries."
Countries: "Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, Sri Lanka."

# Array of strings

```c
#include <stdio.h>
 int main()
 {
     char saarc[8][100] = {"Bangladesh", "India", "Pakistan", "Sri Lanka", "Nepal", "Bhutan", "Maldives",
                           "Afganistan"};
     int row;
     for (row = 0; row < 8; row++) {
         printf("%s\n", saarc[row]);
     }
     return 0;
 }
```

# 3D Array

**Definition**: An array of 2D arrays.



# Declaration & Initialisation

```
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} }
 };
```

# Visualisation

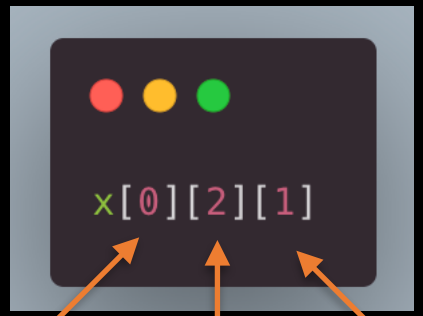
| | Column-0 | Column-1 | Column-2 | Column-3 |
|---|---|---|---|---|
| Row-0 | 0 | 1 | 2 | 3 |
| Row-1 | 4 | 5 | 6 | 7 |
| Row-2 | 8 | 9 | 10 | 11 |

Page-0

| | Column-0 | Column-1 | Column-2 | Column-3 |
|---|---|---|---|---|
| Row-0 | 12 | 13 | 14 | 15 |
| Row-1 | 16 | 17 | 18 | 19 |
| Row-2 | 20 | 21 | 22 | 23 |

Page-1

# Accessing 3D array elements




Page number
(starts from 0)

Row number
(starts from 0)

Column number
(starts from 0)

| | Column-0 | Column-1 | Column-2 | Column-3 |
|---|---|---|---|---|
| Row-0 | 0 | 1 | 2 | 3 |
| Row-1 | 4 | 5 | 6 | 7 |
| Row-2 | 8 | 9 | 10 | 11 |

Page-0

| | Column-0 | Column-1 | Column-2 | Column-3 |
|---|---|---|---|---|
| Row-0 | 12 | 13 | 14 | 15 |
| Row-1 | 16 | 17 | 18 | 19 |
| Row-2 | 20 | 21 | 22 | 23 |

Page-1

# 3D array input output

```
Problem Statement:
"Awesh runs a small hotel and needs a way to manage room bookings across different floors and days. He wants to keep track of whether each room is booked or available. Awesh decides to use a 3D array in C to handle this task.""

The Scenario:
Objective: "Awesh wants to create a booking system for his hotel.""
Hotel Structure: "The hotel has 3 floors, each with 4 rooms. He wants to track bookings for 7 days."
Requirement: "Awesh needs a program to input booking data and display the booking status for each room on each floor for the week."
```

# Bonus Problem

User will input a day (Friday,Saturday…)
Find a room for him. Provide floor no and room number.
Current state of hotel:

```
int hotel[3][4][7] = {
        {{1, 0, 1, 0, 0, 1, 1},
         {0, 1, 0, 0, 1, 1, 0},
         {1, 1, 0, 0, 0, 1, 0},
         {1, 0, 1, 0, 0, 1, 1}},
        {{1, 1, 1, 0, 1, 1, 1},
         {1, 1, 1, 0, 1, 1, 1},
         {0, 0, 0, 0, 0, 0, 0},
         {1, 0, 0, 0, 0, 0, 0}},
        {{1, 1, 1, 0, 1, 0, 0},
         {1, 0, 1, 0, 1, 0, 1},
         {1, 0, 1, 0, 0, 1, 1},
         {1, 0, 1, 0, 0, 1, 1}}
    };
```

# Thank You!