CS1160
Lab 7: 2D Arrays

I. Two-Dimensional Arrays

As you now know, an array is a variable that can store multiple values of any type, therefore; an array could also store a variable of type array.

A Two-Dimensional (2D) array is an array of arrays.

- 2D Array declaration:

```
Data_type array_name [array_size] [array_size];
```

Example:

```
int Matrix [3][4];
```

- Initializing 2D Arrays: initialize a 2D array either element by element or using a single statement as follows :

```
Matrix [0][0]=4;
OR

Matrix [3][4]= { {0,1,2,3}, {8, 5, 3, 0}, {-1, 7, 5, 9} };
OR
Matrix [3][4]= {
                 {0,1,2,3},
                 {8, 5, 3, 0},
                 {-1, 7, 5, 9}
                };
```

Look at the array indices order:

	Column 1	Column 2	Column 3	Column 4
Row 1	Matrix [0][0]	Matrix [0][1]	Matrix [0][2]	Matrix [0][3]
Row 2	Matrix [1][0]	Matrix [1][1]	Matrix [1][2]	Matrix [1][3]
Row 3	Matrix [2][0]	Matrix [2][1]	Matrix [2][2]	Matrix [2][3]

Now look at the elements order in the array:

	Column 1	Column 2	Column 3	Column 4
Row 1	0	1	2	3
Row 2	8	5	3	0
Row 3	-1	7	5	9

The total number of elements that can be stored in a 2D array can be calculated by multiplying the number of rows and columns.

2D array Size = number of Rows * number of Columns

So the array Matrix can store a total 3*4 = 12 elements

- Accessing **Elements of 2D Arrays**

Elements in Two-Dimensional arrays are accessed using the row indexes and column indexes.

Example represents the element present in third row and fourth column.

```
int p=  Matrix[2][3];
```

II. Example

This programme will fill a matrix of size 4x3, then it will print the array.

```
1. #include <stdio.h>
2.
3. int main() {
4.
5. int random [4][3];
6.
7. // reading the values and storing them in the array
8. for(int i = 0; i < 4; i++) {
9.    for(int j = 0; j < 3; j++) {
10.
11.      printf("Enter element [%d][%d]: ", i,j);
12.      scanf("%d", &random[i][j]);
13.      }
14. }
15.
16.      printf("Displaying the Matrix: ");
17.      // printing elements of an array
18. for(i = 0; i < 5; ++i) {
19.    for(i = 0; i < 5; ++i) {
20.
21.      printf("%d\t", random[i][j]);
22.     }
23.      printf("\n");
24.
25.   }
26.
27.   return 0;
28. }
```

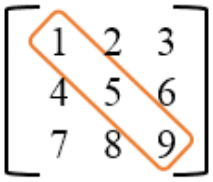
III. Examples

1. Write a C program that **reads** a **float 2D array of size S x G and they are entered by the user**.
   S: Number of students
   G: Number of grades for each student
   The programme will then **find & print** the maximum grade for each students (max number in each row) in the array.

```c
    int S, G;

    // Read number of students and number of grades
    printf("Enter number of students (S): ");
    scanf("%d", &S);

    printf("Enter number of grades per student (G): ");
    scanf("%d", &G);

    float grades[S][G];

    // Read grades from the user
    printf("\nEnter the grades:\n");
    for(int i = 0; i < S; i++) {
        printf("Student %d grades:\n", i + 1);
        for(int j = 0; j < G; j++) {
            printf("Grade %d: ", j + 1);
            scanf("%f", &grades[i][j]);
        }
    }

    // Find and print the maximum grade for each student
    printf("\nMaximum grade for each student:\n");
    for(int i = 0; i < S; i++) {
        float max = grades[i][0];
        for(int j = 1; j < G; j++) {
            if(grades[i][j] > max) {
                max = grades[i][j];
            }
        }
        printf("Student %d: %.2f\n", i + 1, max);
    }

    return 0;
}
```

2. Write a C program that **reads** an **integer 2D array of size n x m. n & m are entered by the user**. It will then **find & print** the main diagonal elements in the array.

Example:



The diagonal of the previous matrix is: **1 5 9**

```c
int main() {
    int n, m;

    // Read dimensions
    printf("Enter number of rows (n): ");
    scanf("%d", &n);

    printf("Enter number of columns (m): ");
    scanf("%d", &m);

    int arr[n][m];

    // Read array elements
    printf("\nEnter the elements of the array:\n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Print the main diagonal
    printf("\nMain diagonal elements:\n");
    // The diagonal runs while i < n and i < m
    int limit = (n < m) ? n : m;
    for(int i = 0; i < limit; i++) {
        printf("%d\t", arr[i][i]);
    }

    return 0;
}
```

3. Write a C program that does the following:
   - Declare integer matrix of size **r*c.**
   - Read the matrix elements from the keyboard.
   - Display the transpose of the matrix

If the user input the following matrix:

$$\begin{bmatrix} 2 & 13 \\ -9 & 11 \\ 3 & 17 \end{bmatrix}$$ The transpose will be as following: $$\begin{bmatrix} 2 & -9 & 3 \\ 13 & 11 & 17 \end{bmatrix}$$

```c
int main() {
    int r, c;

    // Read number of rows and columns
    printf("Enter number of rows (r): ");
    scanf("%d", &r);

    printf("Enter number of columns (c): ");
    scanf("%d", &c);

    int matrix[r][c];

    // Input matrix elements
    printf("\nEnter the elements of the matrix:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &matrix[i][j]);
        }
    }

    // Display transpose
    printf("\nThe transpose of the matrix is:\n");
    for(int i = 0; i < c; i++) {
        for(int j = 0; j < r; j++) {
            printf("%d\t", matrix[j][i]);
        }
        printf("\n");
    }

    return 0;
}
```

III. Tasks

1. **Write a C program that does the following:**

   • Reads an integer 2D array of size **S × G** where **S** is the number of students and **G** is the number of grades for each student (both entered by the user).

   • Calculate and print the **average grade for each student**.

Sample Output:

Student 1 grades: 90 88 75

Student 2 grades: 80 95 60

Average of student 1 = 84.33

Average of student 2 = 78.33

2. Write a C program that does the following:

• Read an integer 2D array of size **n × m** (n and m entered by the user).

• Find and print **the maximum value** in the entire array.

• Also print the **position (row and column index)** of this maximum value.

3 7 2

10 5 16

4 8 9

Sample Output:

Maximum value = 16

Located at row 2 and column 3

3.Write a C program that does the following:

1. Declare a 3×3 integer matrix.
2. Read the matrix elements from the user.
3. Calculate and display the **sum of each row** and **sum of each column**.
4. Display the **overall sum** of all elements in the matrix.

Example: 1 2 3
4 5 6
7 8 9

Sample Output:

Sum of row 1: 6
Sum of row 2: 15
Sum of row 3: 24

Sum of column 1: 12
Sum of column 2: 15
Sum of column 3: 18

Total sum of all elements:
45