Programming Fundamentals Lab

Lab # 12

2D array in C

Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Spring 2023

Department of Computer Science, National University of Computer and Emerging Sciences
FAST Peshawar Campus

Multidimensional Arrays

In the previous lab, you learned about arrays, which is also known as single dimension arrays. These are great, and something you will use a lot while programming in C. However, if you want to store data as a tabular form, like a table with rows and columns, you need to get familiar with multidimensional arrays.

A multidimensional array is basically an array of arrays.

Arrays can have any number of dimensions. In this lab, we will introduce the most common; two-dimensional arrays (2D).

Two-Dimensional Arrays

A 2D array is also known as a matrix (a table of rows and columns). For example,

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

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.

	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]

Example 1 | Matrix Input and Output:

```
#include <stdio.h>
int main()
{
	int rows, cols;
	printf("\nEnter rows: ");
	scanf("%d", &rows);
	printf("\nEnter cols: ");
	scanf("%d", &cols);

	int arr[rows][cols];

	for(int i=0; i<rows; i++)
	{
		for(int j=0; j<cols; j++)
		{
			printf("\nEnter value at index: %d%d  ", i, j);
			scanf("%d", &arr[i][j]);
		}
	}

	printf("\nMatrix: \n");
	for(int i=0; i<rows; i++)
	{
		for(int j=0; j<cols; j++)
		{
			printf("%d ", arr[i][j]);
		}
		printf("\n");
	}
}
```

Output:

```
Enter rows: 2

Enter cols: 3

Enter value at index: 00  1

Enter value at index: 01  4

Enter value at index: 02  2

Enter value at index: 10  6

Enter value at index: 11  7

Enter value at index: 12  4

Matrix:
1 4 2
6 7 4
```

Pass a 2D arrays to a function in C

In C programming, you can pass an entire array to functions.

Example: Sum of two Matrices

```
#include <stdio.h>
void sum(int rows, int cols, int matrix1[rows][cols], int matrix2[rows][cols]) {
	int matrix3[rows][cols];
	for(int i=0; i<rows; i++) {
		for(int j=0; j<cols; j++) {
			matrix3[i][j]=matrix1[i][j]+matrix2[i][j];
		}
	}

	printf("\nSum of two matrices: \n");
	for(int i=0; i<rows; i++) {
		for(int j=0; j<cols; j++) {
			printf("%d ", matrix3[i][j]);
		}
		printf("\n");
	}
}
int main() {

	int rows=2, cols=2;
	int matrix1[rows][cols];
	int matrix2[rows][cols];

	printf("Enter values in matrix 1: \n");
	for(int i=0; i<rows; i++) {
		for(int j=0; j<cols; j++) {
			printf("Enter value at index: %d%d  ", i, j);
			scanf("%d", &matrix1[i][j]);
		}
	}
	printf("\nEnter values in matrix 2: \n");
	for(int i=0; i<rows; i++) {
		for(int j=0; j<cols; j++) {
			printf("\nEnter value at index: %d%d  ", i, j);
			scanf("%d", &matrix2[i][j]);
		}
	}

	sum(rows, cols, matrix1, matrix2);
}
```

Output:

```
Enter values in matrix 1:
Enter value at index: 00  1
Enter value at index: 01  2
Enter value at index: 10  3
Enter value at index: 11  4

Enter values in matrix 2:

Enter value at index: 00  5

Enter value at index: 01  6

Enter value at index: 10  7

Enter value at index: 11  8

Sum of two matrices:
6 8
10 12
```