Programming Fundamentals Lab

Lab # 13

2D array in C, passing array to a function

Instructor: Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2022

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 | Sum of two matrices

```c
// C program to find the sum of two matrices of order 2*2
#include <stdio.h>
int main()
{
 int a[2][2], b[2][2], result[2][2],i,j;
 // Taking input using nested for loop
 printf("Enter elements of 1st matrix\n");
 for (i = 0; i < 2; ++i)
   for (j = 0; j < 2; ++j)
   {
     printf("Enter a[%d%d] : ", i, j);
     scanf("%d", &a[i][j]);
   }
 // Taking input using nested for loop
 printf("Enter elements of 2nd matrix\n");
 for (i = 0; i < 2; ++i)
   for (j = 0; j < 2; ++j)
   {
     printf("Enter b[%d%d] : ", i, j);
     scanf("%d", &b[i][j]);
   }
 // adding corresponding elements of two arrays
 for (i = 0; i < 2; ++i)
   for (j = 0; j < 2; ++j)
   {
     result[i][j] = a[i][j] + b[i][j];
   }
 // Displaying the sum
 printf("\nSum Of Matrix:\n");
```

```
 for (i = 0; i < 2; ++i){
   for (j = 0; j < 2; ++j)
   {
     printf("%d\t", result[i][j]);
   }
       printf("\n");
 }
 return 0;
}
```

Output

```
Enter elements of 1st matrix
Enter a[00] : 2
Enter a[01] : 3
Enter a[10] : 4
Enter a[11] : 5
Enter elements of 2nd matrix
Enter b[00] : 1
Enter b[01] : 7
Enter b[10] : 8
Enter b[11] : 9
Sum Of Matrix:
3       10
12      14
```

Pass arrays to a function in C

In C programming, you can pass an entire array to functions. Before we learn that, let's see how you can pass individual elements of an array to functions.

Example 2 | Pass Individual Array Elements

```c
#include <stdio.h>
void display(int age1, int age2) {
 printf("%d\n", age1);
 printf("%d\n", age2);
}

int main() {
 int ageArray[] = {2, 8, 4, 12};

 // pass second and third elements to display()
 display(ageArray[1], ageArray[2]);
 return 0;
}
```

Output

```
8
4
```

Here, we have passed array parameters to the display() function in the same way we pass variables to a function.

Example 3 | Pass Arrays to Functions

```c
//Program to calculate the sum of array elements by passing to a function
#include <stdio.h>
float calculateSum(float num[]);

int main() {
 float result, num[] = {23.4, 55, 22.6, 3, 40.5, 18};
 // num array is passed to calculateSum()
 result = calculateSum(num);
 printf("Result = %.2f", result);
 return 0;
}

float calculateSum(float num[]) {
 float sum = 0.0;
 for (int i = 0; i < 6; ++i) {
   sum += num[i];
 }
 return sum;
}
```

Output

```
Result = 162.50
```

To pass an entire array to a function, only the name of the array is passed as an argument.

```c
result = calculateSum(num);
```

However, notice the use of [] in the function definition.

```c
float calculateSum(float num[]) {
... ..
}
```

Example 4 | Pass two-dimensional arrays

```c
#include <stdio.h>
void displayNumbers(int num[2][2]);

int main() {
 int num[2][2];
 printf("Enter 4 numbers:\n");
 for (int i = 0; i < 2; ++i) {
   for (int j = 0; j < 2; ++j) {
     scanf("%d", &num[i][j]);
   }
 }
 // pass multi-dimensional array to a function
 displayNumbers(num);
 return 0;
}
void displayNumbers(int num[2][2]) {
 printf("Displaying:\n");
 for (int i = 0; i < 2; ++i) {
   for (int j = 0; j < 2; ++j) {
     printf("%d\n", num[i][j]);
   }
 }
}
```

Output

```
Enter 4 numbers:
4
5
6
7
```

```
Displaying:
4
5
6
7
```

References:

https://www.programiz.com/c-programming/c-multi-dimensional-arrays