MIT-WPU

MIT-WORLD PEACE UNIVERSITY

F. Y. B. Tech

Trimester: I/II/III Subject: Programming and Problem Solving

Name: Krishnaraj Thadesar **Division:** 9

Roll No.: 109054 **Batch:** I3

Experiment No.: 4

Name of the Experiment: Menu Driven Matrix Calculator using Switch Case.

Performed on: 13th January 2022

Submitted on: 21st January 2022

AIM: Write an algorithm and draw a flowchart to Make a menu driven program in C to perform addition and Subtraction of Matrices using Switch Case.

OBJECTIVE:

1. To learn and understand arrays in C.
2. To learn and understand two dimensional arrays and operation on it.

THEORY:

What is an Array?

An Array is a collection of similar type of data items stored at contiguous memory locations. It is a variable that can store multiple values.

Types of Arrays:

1. *One Dimensional Array:*

   In C programming language, one dimensional arrays are used tos teore a list of values of the same data type.

   ```
   Eg. Int marks[2] = {99, 98};
   ```

2. *Two Dimensional Array:* An array having more than one dimensions is known as a multi-dimensional array. If the array has 2 dimensions, it is known as a two dimensional array.

```
Eg. int matrix[2][2] = {
                {1, 2}
                {2. 3}
                }
```

3. *Multi-dimensional Arrays*: An array having more than one dimensions is known as a multi-dimensional array

Declaration of a 1D Array

1. The declaration must have a data type (int, char, float, double, etc), variable name, and square brackets with the number of elements of the array inside it.
2. Subscript or the square brackets represents the size of the array.
3. Array index always starts from 0
4. Each element in stored in a separate memory location.

Declaration of a 2D Array

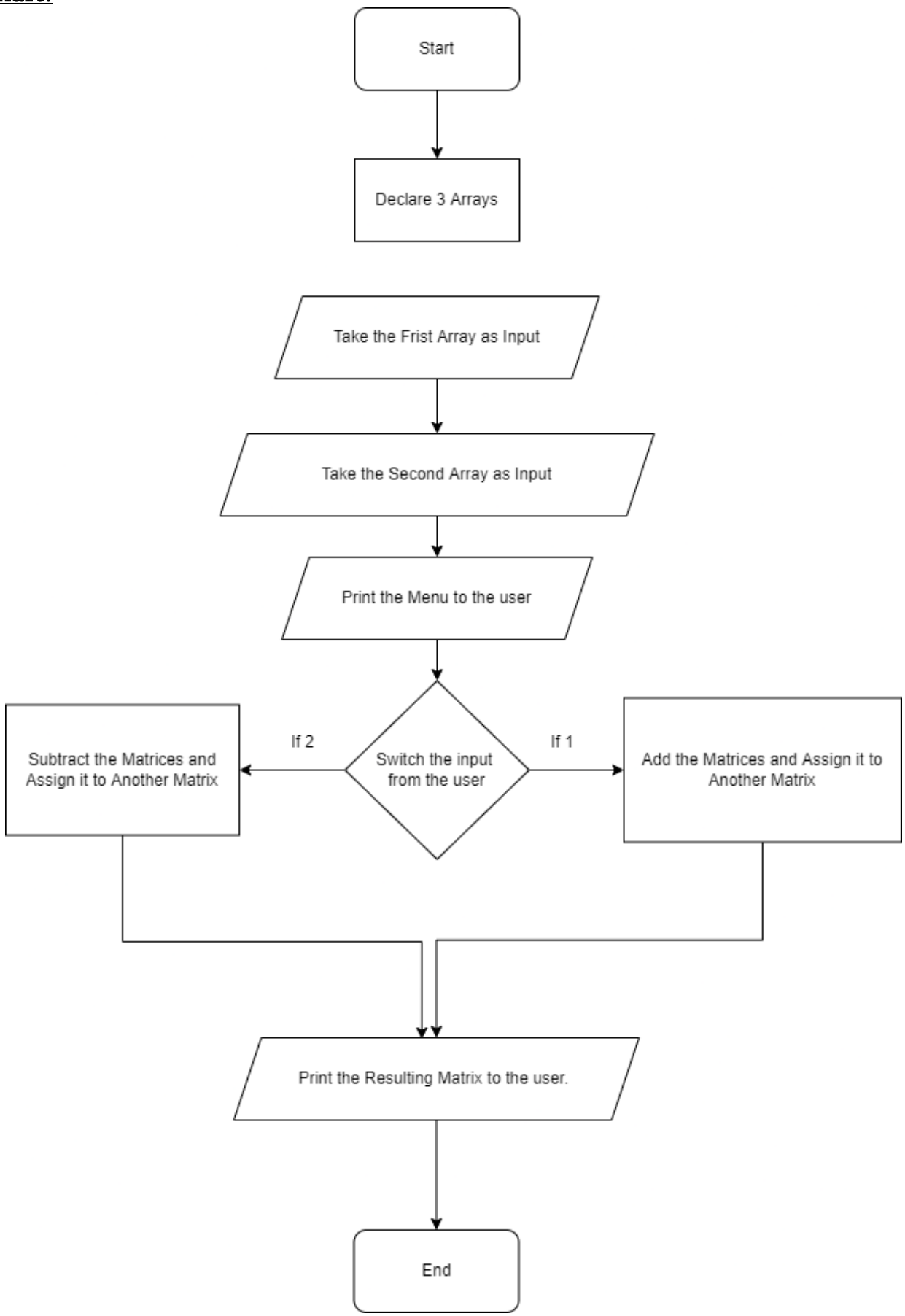
```
Syntax: data_type array-name[row-size][col-size]
Initialization: int matrix[2][2] = {
                {1, 2}
                {2. 3}
                }
```

PLATFORM: *Windows 11 64 Bit*

ALGORITHM:

Step 1: Start
Step 2: Declare 3 Matrices of Size [3][3] each and assign them to zero.
Step 3: Input the First Matrix
Step 4: Input the Second Matrix
Step 5: Write another nested For loop
Step 6: Switch the Choice variable inside the for loop
Step 7: If choice is 1, add the values, if choice is 2, subtract the values and assign them to the third matrix C.
Step 6: Output the Third Matrix
Step 7: End

Flowchart:


Start
Declare 3 Arrays
Take the Frist Array as Input
Take the Second Array as Input
Print the Menu to the user
Switch the input from the user
If 2
If 1
Subtract the Matrices and Assign it to Another Matrix
Add the Matrices and Assign it to Another Matrix
Print the Resulting Matrix to the user.
End

CODE:

```c
#include <stdio.h>

int main()
{
    int a[3][3], b[3][3], c[3][3];
    int choice = 0;
    printf("Enter the First Matrix: \n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("\nEnter the Second Matrix: \n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            scanf("%d", &b[i][j]);
        }
    }
    printf("Select what Operation you want to do [1, 2] :\n\
            1. Addition of Matrices\n\
            2. Subtraction of Matrices\n");
    scanf("%d", &choice);

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            switch (choice)
            {
            case 1:
                c[i][j] = a[i][j] + b[i][j];
                break;
            case 2:
                c[i][j] = a[i][j] - b[i][j];
                break;
```

```
            default:
                printf("Invalid Input");
                return 0;
            }
        }
    }
    printf("\nThe Resulting Matrix is: \n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%d ", c[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT

Addition

```
Enter the First Matrix:
1 2 3
2 3 4
5 5 8

Enter the Second Matrix:
1 5 4
7 5 9
5 7 2
Select what Operation you want to do [1, 2]:
            1. Addition of Matrices
            2. Subtraction of Matrices
1

The Resulting Matrix is:
2 7 7
9 8 13
10 12 10
```

Subtraction

```
Enter the First Matrix:
1 2 3
2 3 4
5 5 8

Enter the Second Matrix:
1 5 4
7 5 9
5 7 2
Select what Operation you want to do [1, 2] :
            1. Addition of Matrices
            2. Subtraction of Matrices
2

The Resulting Matrix is:
0 -3 -1
-5 -2 -5
0 -2 6
```

CONCLUSION:

The working, concept and implementation of single and multi-dimensional arrays was understood in detail and implemented using switch case in a menu driven program.

FAQs:

Q1. What are the different types of arrays and how do we define them?

Ans. There are 2 Types of arrays:

1. *One dimensional arrays*
2. *Two dimensional Arrays*

One Dimensional Arrays:

They are arrays that have a single subscript.

```
Syntax: data_type array-name[size]
```

Multi-dimensional Arrays:

An array having more than one dimensions is known as a multi-dimensional array.

```
Syntax: data_type array-name[row-size][col-size]
```

Q2. How are arrays initialized and processed?

Ans. Initialization (static):

```
int matrix[2][2] = {
                {1, 2}
                {2. 3}
                }
```

Initialization (dynamic):

```
int matrix[2];
matrix[0] = 1;
matrix[1] = 2;
```

Q3. How are elements accessed in a 2D array?

Elements in a 2-dimensional array are accessed using row and column indices.

For eg.

```
M = {{1, 2}
     {2, 3}}
M[0][1] = 2;
```

This shows that the element in the 1nd row and 2nd column is 2.