Arrays and Strings in C Language

Module 04 - Addition and Multiplication of Two Matrices in C Language

Introduction

In C programming, matrices are represented as two-dimensional arrays. To perform operations like **addition** and **multiplication** on matrices, certain rules and conditions must be met:

Matrix Addition:

Matrix addition is a simple mathematical operation where two matrices of the same dimensions are added together element by element. In C, you can represent a matrix as a 2D array. The sum of two matrices results in a new matrix where each element is the sum of the corresponding elements from the two original matrices.

Key Points:

1. **Matrices must have the same dimensions**: Both matrices should have the same number of rows and columns for addition to be possible.

2. **Element-wise addition**: The elements of two matrices are added together at corresponding positions.

Matrix Addition Process

Suppose you have two matrices, A and B, each with dimensions m x n (m rows and n columns). The sum matrix C will also have the same dimensions, i.e., m x n.

Introduction to 'C' Language

- Matrix **A** and Matrix **B** are given as:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

- The result of adding matrices `A` and `B` to get matrix `C` will be:

$$C = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \dots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \dots & a_{2n}+b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \dots & a_{mn}+b_{mn} \end{bmatrix}$$

How It Works in C:

1. **Input the matrices:** You need to first input the elements of both matrices.
2. **Add corresponding elements:** For each position (i, j), the element of matrix C[i][j] is the sum of the corresponding elements of matrix A[i][j] and matrix B[i][j].
3. **Display the result:** Finally, print the resultant matrix C.

Solved Example: Addition of Two Matrices

```c
#include <stdio.h>
#include <conio.h>  // Include conio.h to use getch()

void main() {
    float a[2][2], b[2][2], c[2][2];
    int i, j;

    printf("Enter the elements of the 1st matrix:\n");
    // Reading the elements of the first matrix
    for(i = 0; i < 2; i++) {
        for(j = 0; j < 2; j++) {
            scanf("%f", &a[i][j]);
        }
    }

    printf("Enter the elements of the 2nd matrix:\n");
    // Reading the elements of the second matrix
    for(i = 0; i < 2; i++) {
        for(j = 0; j < 2; j++) {
            scanf("%f", &b[i][j]);
        }
    }

    // Calculating the sum of the two matrices
    for(i = 0; i < 2; i++) {
        for(j = 0; j < 2; j++) {
            c[i][j] = a[i][j] + b[i][j];  // Add corresponding elements
        }
    }

    // Displaying the result
    printf("Sum of the two matrices:\n");
    for(i = 0; i < 2; i++) {
        for(j = 0; j < 2; j++) {
            printf("%.2f ", c[i][j]);
        }
        printf("\n");
    }

    // Wait for a key press before exiting (using getch())
    getch();  // This will wait for the user to press a key
}
```

Scalar Multiplication of a 2D Array in C

Scalar multiplication in the context of a 2D array involves multiplying each element of a matrix (2D array) by a constant (scalar). The result is a new matrix where each element is the product of the original matrix element and the scalar.

Steps for Scalar Multiplication:

- Input the Matrix: First, you input the matrix (2D array) elements.
- Multiply by Scalar: Multiply each element of the matrix by the scalar value.
- Display the Result: Print the resulting matrix after the scalar multiplication.

Example of Scalar Multiplication:

Let's say we have a 2D matrix `A`:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

If the scalar is 2, then after scalar multiplication, the result will be:

$$B = \begin{bmatrix} 1 \times 2 & 2 \times 2 \\ 3 \times 2 & 4 \times 2 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

Solved Example of Scalar Multiplication

c Copy code

```c
#include <stdio.h>

int main() {
    int m, n, i, j, scalar;

    // Input the number of rows and columns
    printf("Enter the number of rows and columns: ");
    scanf("%d %d", &m, &n);

    // Declare a 2D array for the matrix
    int matrix[m][n];

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

    // Input the scalar value
    printf("Enter the scalar value: ");
    scanf("%d", &scalar);

    // Perform scalar multiplication
    printf("Matrix after scalar multiplication:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%d ", matrix[i][j] * scalar);  // Multiply each element by the scalar
        }
        printf("\n");
    }

    return 0;
}
```

Explanation of the Program:

1. **Input Dimensions**: The program first asks for the dimensions (rows and columns) of the matrix.
2. **Matrix Input**: The elements of the matrix are then entered using a loop.
3. **Scalar Input**: A scalar value is input, which will be used to multiply each element of the matrix.
4. **Scalar Multiplication**: The matrix is displayed after multiplying each element by the scalar.
5. **Output**: The new matrix, with each element multiplied by the scalar, is displayed.

Sample Output:

For example, if you enter the following:

Matrix `A`:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Scalar: `2`

The output will be:

$$B = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$