DAA 2

Mayank Rohilla

September 2023

1 Matrix Multiplication

Matrix multiplication is a fundamental operation in linear algebra and mathematics. It involves multiplying two matrices together to produce a new matrix. This operation plays a crucial role in various fields, including physics, computer graphics, engineering, and data analysis. Matrix multiplication allows for the transformation and manipulation of data and is an essential concept in understanding complex systems and solving systems of linear equations. It forms the basis for many algorithms and computations in diverse scientific and technological applications.

1.1 Algorithm

Matrix multiplication, also known as matrix product, is performed by following these steps:

- Matrix Dimensions: Ensure that the matrices you want to multiply are compatible, meaning the number of columns in the first matrix must be equal to the number of rows in the second matrix

- Element-wise Multiplication: Multiply each element of the row of the first matrix by the corresponding element in the column of the second matrix

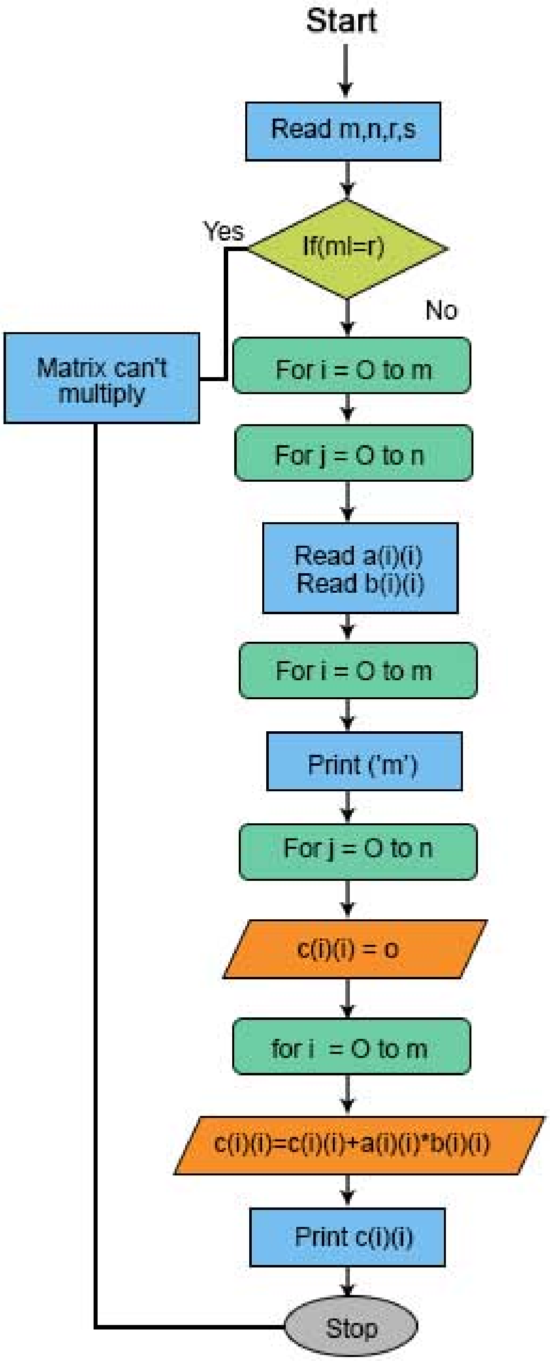
- Summation: Sum up all the products obtained in the previous step for each element of the resulting matrix.

- Resulting Matrix: Repeat steps 2 and 3 for all elements of the resulting matrix. Each element in the resulting matrix is calculated using a row from the first matrix and a column from the second matrix

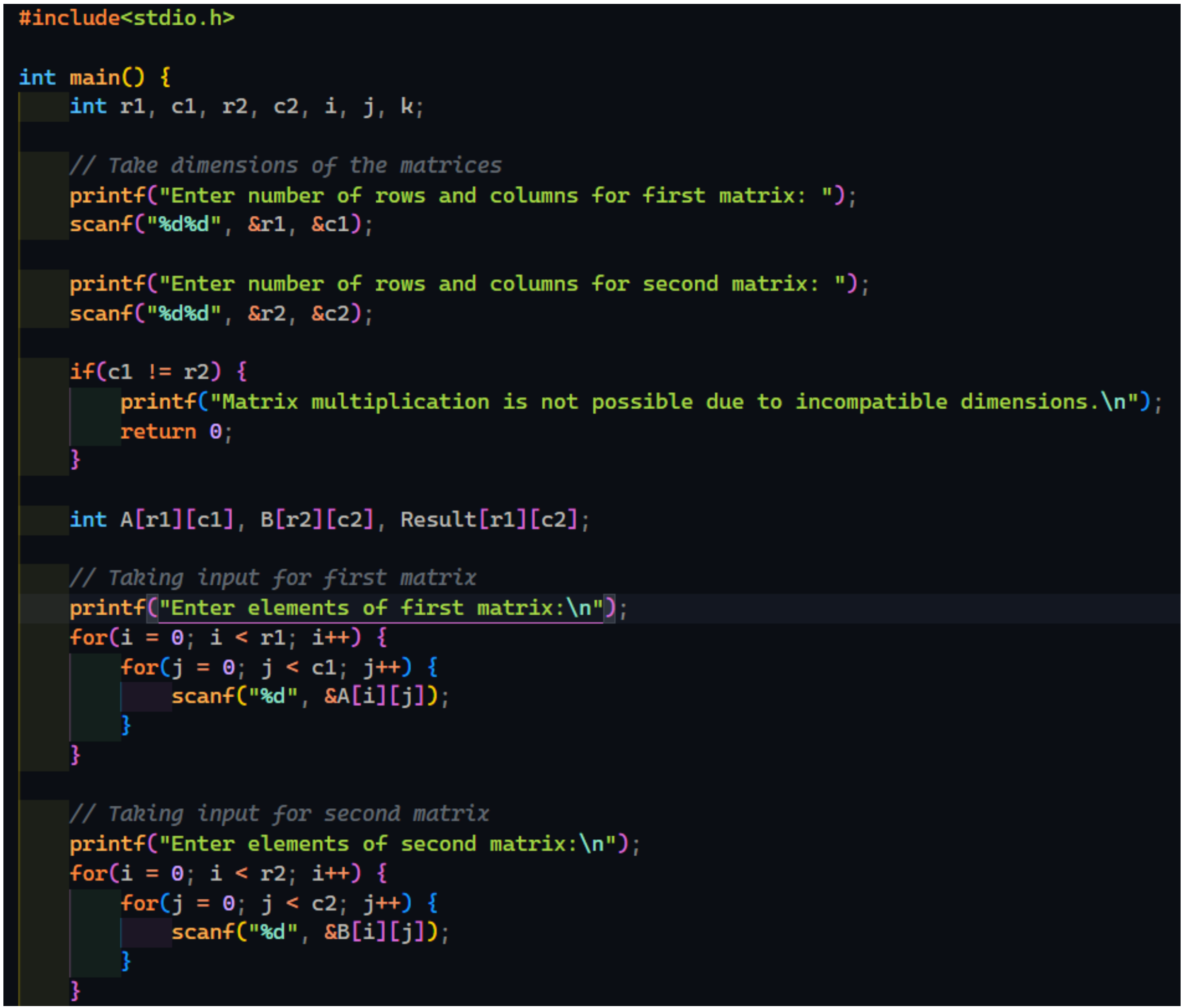
- Result: The resulting matrix will have dimensions determined by the number of rows from the first matrix and the number of columns from the second matrix

- Matrix multiplication is not commutative, meaning the order of multiplication matters, and the result can be different if the order of the matrices is reversed.

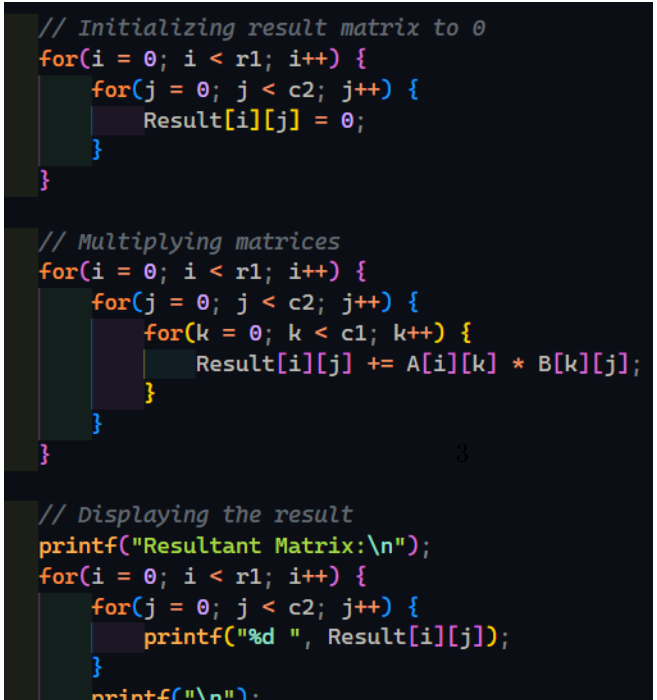
1.2 Flow Chart



1.3 Code

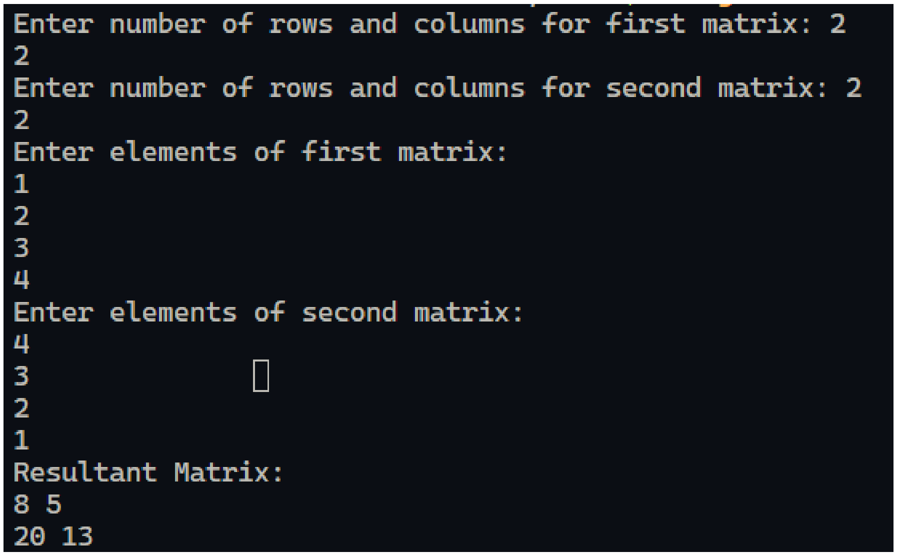


```c
#include<stdio.h>

int main() {
    int r1, c1, r2, c2, i, j, k;

    // Take dimensions of the matrices
    printf("Enter number of rows and columns for first matrix: ");
    scanf("%d%d", &r1, &c1);

    printf("Enter number of rows and columns for second matrix: ");
    scanf("%d%d", &r2, &c2);

    if(c1 != r2) {
        printf("Matrix multiplication is not possible due to incompatible dimensions.\n");
        return 0;
    }

    int A[r1][c1], B[r2][c2], Result[r1][c2];

    // Taking input for first matrix
    printf("Enter elements of first matrix:\n");
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c1; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    // Taking input for second matrix
    printf("Enter elements of second matrix:\n");
    for(i = 0; i < r2; i++) {
        for(j = 0; j < c2; j++) {
            scanf("%d", &B[i][j]);
        }
    }
```



```c
    // Initializing result matrix to 0
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            Result[i][j] = 0;
        }
    }

    // Multiplying matrices
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            for(k = 0; k < c1; k++) {
                Result[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    // Displaying the result
    printf("Resultant Matrix:\n");
    for(i = 0; i < r1; i++) {
        for(j = 0; j < c2; j++) {
            printf("%d ", Result[i][j]);
        }
        printf("\n");
```

1.4 Output



```
Enter number of rows and columns for first matrix: 2
2
Enter number of rows and columns for second matrix: 2
2
Enter elements of first matrix:
1
2
3
4
Enter elements of second matrix:
4
3
2
1
Resultant Matrix:
8 5
20 13
```