

Matrix multiplication in C

Overview

Matrix refers to a 2D array of numbers represented in rows and columns. It supports different operations like addition, subtraction, and multiplication. Let's dive deep into the details of matrix multiplication in C.

Introduction to Matrix Multiplication in C

Matrix is an ordered rectangular array of functions or numbers. A matrix with n vertical lines, known as columns, and m horizontal lines, known as rows, is a matrix of order m by n .

Multiplication of two matrices is possible only when the number of columns in the first matrix equals the number of rows in the second matrix. Hence, the order of the product of the two matrices will be equivalent to the number of rows in the first matrix and the number of columns in the second matrix.

The dot product of the first row of the first matrix and the first column of the second matrix leads to the first element of the product matrix. You can perform **matrix multiplication in C using function** or without it.

Algorithm of C Programming Matrix Multiplication

Step-1: Enter the value of m and n , i.e., the order of the first matrix.

Step-2: Enter the value of p and q , i.e., the order of the second matrix.

Step-3: Define a matrix of size $a[m][n]$ and $b[p][q]$.

Step-4: Enter the element of each matrix (row-wise) to calculate **matrix multiplication in C using for loop**.

Step-5: If the number of columns of the first matrix doesn't equal the number of rows of the second matrix, then print the below message "matrix multiplication is incompatible" and exit. If not, move to the next step.

Step-6: Define a third matrix c (result matrix) of size $m \times q$ to save the multiplication result.

Step-7: Create a for loop from $i=0$ to $i=m$.

Step-8: Create an inner loop in the above loop from $j=0$ to $j=q$.

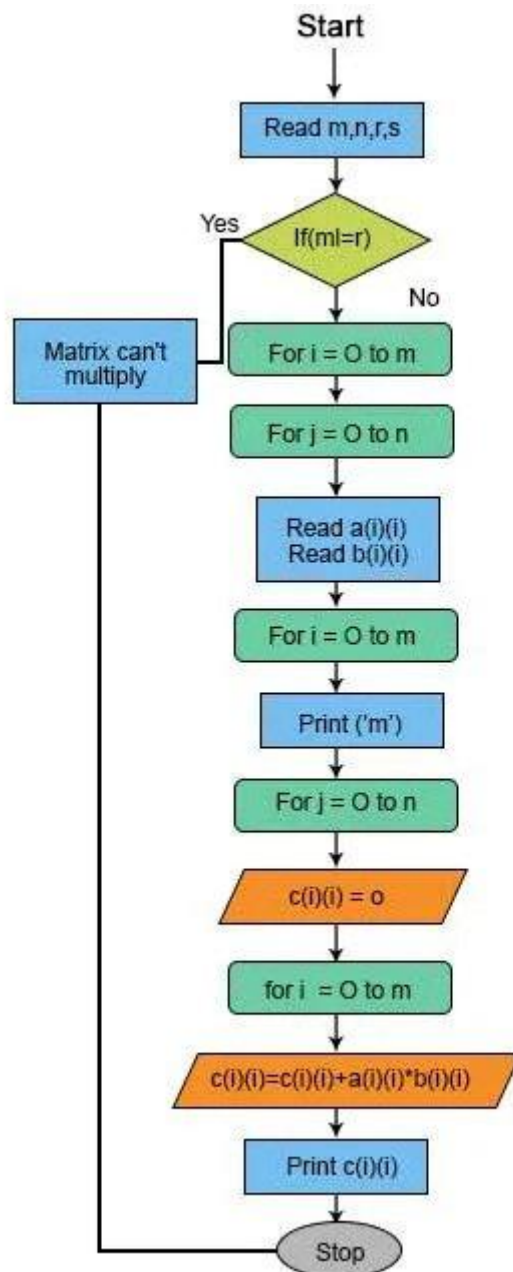
Step-9: Initialise the value of the element i & j of the new matrix to 0.

Step-10: Create an inner loop from $k=0$ to $k=p$ within the above loop.

Step-11: Use add (+) and assign (=) operator to save the value of $a[i][k] * b[k][j]$ in the result matrix $c[i][j]$.

Step-12: Print the output of the result matrix.

Flow Chart of Matrix Multiplication



[Flow Chart of Matrix Multiplication](#)

Multiplication of Two Matrices in C

In C, there are two types of matrices: square matrices and rectangular matrices. Let's begin by exploring how to multiply two square matrices in C.

Multiplication of Square Matrices

A square matrix is one in which the number of rows and columns are the same. You can perform **matrix multiplication in C** of two square matrices only if both the matrices have the same order. The corresponding elements are multiplied and added to the products together.

An easy way to understand this is to develop the code for **2x2 matrix multiplication in C** and **3x3 matrix multiplication in C**.

Here's the code for the multiplication of two square matrices:

```
#include<stdio.h>
int main() {
    int a[5][5], b[5][5], c[5][5], n, x, y, z;

    printf("Enter the value of N (N <= 5): ");
    scanf("%d", & n);
    printf("Enter the elements of Matrix-1: \n");

    for (x = 0; x < n; x++) {
        for (y = 0; y < n; y++) {
            scanf("%d", & a[x][y]);
        }
    }

    printf("Enter the elements of Matrix-2: \n");

    for (x = 0; x < n; x++) {
        for (y = 0; y < n; y++) {
            scanf("%d", & b[x][y]);
        }
    }

    for (x = 0; x < n; x++) {
        for (y = 0; y < n; y++) {
            c[x][y] = 0;
            for (z = 0; z < n; z++) {
                c[x][y] += a[x][z] * b[z][y];
            }
        }
    }
}
```

```

    }

    printf("The product of the two matrices is: \n");
    for (x = 0; x < n; x++) {
        for (y = 0; y < n; y++) {
            printf("%d\t", c[x][y]);
        }
        printf("\n");
    }
    return 0;
}

```

Output:

```

Enter the value of N (N <= 5): 2
Enter the elements of Matrix-1:
1 3
3 4
Enter the elements of Matrix-2:
4 6
1 2
The product of the two matrices is:
7    12
16   26

```

In the above code, the first step defines the input, output, and resultant matrix sizes. The user is prompted to enter the elements of matrix-1 and matrix-2. Subsequently, the program runs the for loop to multiply and add the elements of the matrices. The loop runs until 'n' (the size of a square matrix).

Multiplication of Rectangular Matrices

In a rectangular matrix, the number of rows and the number of columns are unequal. Unlike the square matrix, the rectangular matrix allows **matrix multiplication in C with different dimensions**. The number of columns in the first matrix must equal the number of rows in the second matrix. Otherwise, you will get an error

Here's the code for the multiplication of two rectangular matrices:

```

#include <stdio.h>
void multiplyMatrices(int mat1[][3], int row1, int col1, int mat2[][2], int row2, int col2, int result[][2]) {
    int x, y, z;
    // Performs matrix multiplication

```

```

    for (x= 0; x < row1; x++) {
    for (y = 0; y < col2; y++) {
    result[x][y] = 0;
    for (z = 0; z < col1; z++) {
    result[x][y] += mat1[x][z] * mat2[z][y];
    }
    }
    }
}

void showMatrix(int mat[][2], int row, int col) {
    int x, y;

    printf("Resultant Matrix:\n");
    for (x = 0; x < row; x++) {
    for (y = 0; y < col; y++) {
    printf("%d ", mat[x][y]);
    }
    printf("\n");
    }
}

int main() {
    int row1, col1, row2, col2;
    printf("Enter the number of rows and columns for Matrix 1: ");
    scanf("%d %d", &row1, &col1);
    printf("Enter the number of rows and columns for Matrix 2: ");
    scanf("%d %d", &row2, &col2);

    if (col1 != row2) {
    printf("Matrix multiplication is not possible!\n");
    return 0;
    }

    int mat1[row1][col1];
    int mat2[row2][col2];
    int result[row1][col2];

    int x, y;

    printf("Enter the elements of Matrix 1:\n");
    for (x = 0; x < row1; x++) {
    for (y = 0; y < col1; y++) {
    scanf("%d", &mat1[x][y]);

```

```

    }
}

printf("Enter the elements of Matrix 2:\n");
for (x = 0; x < row2; x++) {
    for (y = 0; y < col2; y++) {
        scanf("%d", &mat2[x][y]);
    }
}

multiplyMatrices(mat1, row1, col1, mat2, row2, col2, result);
showMatrix(result, row1, col2);

return 0;
}

```

In the above code of rectangular **matrix multiplication in C**, the first step initialises the rows and columns of the input matrices and the result matrix. The user then inputs the elements of each matrix. The program runs the for loop to multiply and add the elements of each of the input matrices. If the number of columns in the first matrix is not equal to the number of rows in the second matrix, it displays a message that matrix multiplication is not possible.

Output

```

Enter the number of rows and columns for Matrix 1: 2 3
Enter the number of rows and columns for Matrix 2: 3 4
Enter the elements of Matrix 1:
1 2 3
2 3 4
Enter the elements of Matrix 2:
1 2 3 4
2 3 4 5
1 5 4 2
Resultant Matrix:
13 19 19 28
19 28 28 37

```

Let's try running the above program in the case when the number of columns in the first matrix doesn't equal the number of rows in the second matrix. Here's the output.

Output:

```

Enter the order of first matrix
1 4

```

```
Enter the order of second matrix
```

```
14
```

```
Matrix multiplication is not possible
```

Multiply Matrices by Passing it to a Function

The code for **matrix multiplication in C without function** may be long. You can pass a function to simplify the code for matrix multiplication. The method of passing a function involves allowing the user to input the matrix using the `input()` function. The `display()` function is used to show the matrix's result. You can name a function `multiply()` and use it to calculate the multiplication of two matrices.

Benefits of C Programming Matrix Multiplication

- **Matrix multiplication in C** simplifies the manual task of calculating the multiplication of two matrices.
- It helps you to easily determine the solutions to linear equations.
- It lets you learn the applications of mathematical logic in a coding algorithm.
- The C programming language considers a matrix as a data type and consumes less memory when processing it.
- Storing values in a matrix instead of individual variables makes sure the C program can manipulate data more efficiently.
- It makes it simpler to retrieve information about object rotation.

Conclusion

Using **matrix multiplication in C** ensures the C program can access and accomplish operations on the data more efficiently. Understanding its syntax and practising it helps you to apply mathematical logic to coding algorithms.

Tutorials significantly help you to master core concepts, pursuing courses like upGrad's [Executive Post Graduate Programme in Software Development](#) by IIT-B can help your career skyrocket in the developing field of STEM.

[Enrol now](#) to commence your journey!

FAQs

Q. Is matrix multiplication commutative?

Generally, the matrix multiplication is not commutative. The reason is when we multiply the two matrices, the elements of the first row of matrix-1 are multiplied by the elements of the first column of matrix-2. So, altering the order will alter the corresponding elements of matrix-1 and matrix-2.

Q. When is matrix multiplication in C possible?

The matrix multiplication in C is possible if the number of columns in the first matrix and the number of rows in the second matrix are equal.

Q. Is matrix multiplication the same as cross product or dot product?

Matrix multiplication is not the same as the dot product or cross product. The dot product accepts two vectors as input and outputs a scalar. The cross-product inputs two R3 vectors, and the output is another R3 vector. On the other hand, matrix multiplication accepts a matrix and a vector as input and outputs a vector.