

Solution Review: Multiply Two Matrices

Let's go over the solution review of the challenge given in the previous lesson.

We'll cover the following

- Solution
- Explanation
 - multiplication function

Solution

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

// multiplication function
void multiplication(int arr1[][2], int row1, int col1, int arr2[][2], int row2, int col2, int result[][2]) {
    // Check if col of first array equal to row of second array
    if (col1 == row2) {
        // Traverse first array row
        for (int x = 0; x < row1; x++) {
            // Traverse second array columns
            for (int y = 0; y < col2; y++) {
                // Traverse first array columns and second array rows
                for (int z = 0; z < col1; z++) {
                    // Multiplication
                    result[x][y] = result[x][y] + arr1[x][z] * arr2[z][y];
                }
            }
        }
    }
    else{
        // Traverse first array row
        for (int x = 0; x < row1; x++) {
            // Traverse second array columns
            for (int y = 0; y < col2; y++) {
                // Fill the elements of array by -1
                result[x][y] = -1;
            }
        }
    }
}

// print_array function
```

```

void print_array(int arr[3][2], int row, int column) {
    // Outer loop
    for (int i = 0; i < row; i++) {

        // Inner loop
        for (int j = 0; j < column; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}

// main function
int main() {
    // Initialize arr1
    int arr1[3][2] = {{1,2},{3,4},{5,6}};
    // Initialize arr2
    int arr2[2][2] = {{10,20},{30,40}};
    // Initialize result
    int result[3][2] = {{0,0},{0,0},{0,0}};
    // Call function multiplication
    multiplication(arr1,3,2,arr2,2,2,result);
    // Call function print_array
    print_array(result,3,2);
    return 0;
}

```



Explanation

multiplication function

The `multiplication` function takes three 2D arrays `arr[][]` of type `int` and its `row` and `column` of type `int` in its input parameters, and, it returns nothing in the output.

Matrix multiplication is only possible if the number of columns of the 1st matrix is equal to the number of rows of the 2nd matrix. We first check for this condition. If `col1` is not equal to `row2`, we fill each element of the matrix with `-1`. If matrices can be multiplied, traverse through the 2D array (matrix) using two nested loops. In matrix multiplication, each value at a certain `[row][column]` is computed by multiplying each element of the row index of the first matrix with its corresponding column index elements of the 2nd matrix. For this, we use a 3rd nested loop to iterate over each row element of the first array (equal to the number of columns in the first matrix). We initialize `z` from `0` to `col1` and multiply the elements in each row of the first matrix to the element in each column of the second matrix and add their products.

Let's wrap up this chapter by completing a quiz in the upcoming lesson.