matrix

Tuesday, January 14, 2025 9:53 PM

Source Code:

```c
#include <stdio.h>


void getUserInput(int array[2][2], int row, int column, const char* name) {
    int input = 0;
    int result = 0; // Result indicator for input
    while (result != 1) { // Loop until valid input is entered: EXTRA CREDIT
        printf("%s%d%d = ", name, row + 1, column + 1);
        result = scanf("%d", &input);
        if (result == 1) {
            array[row][column] = input;
            break;
        } else {
            printf("ERROR: Invalid input. Please enter numbers only!\n");
            while (getchar() != '\n');
        }
    }
}
int main() {
    // Declaring matrices
    int a[2][2] = {{0,0}, {0,0}};
    int b[2][2] = {{0,0}, {0,0}};
    // Prompting for first matrix
    printf("Enter the first matrix(a) that will be multiplied:\n");
    // Looping through rows and columns of matrix
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            getUserInput(a, i, j,"a");
        }
    }
    printf("\n\n");
    // Repeating process above for Matrix B
    printf("Enter the first matrix(b) that will be multiplied:\n");
    for(int i =0; i < 2; i++) {
        for (int j =0; j< 2; j++) {
            getUserInput(b, i, j, "b");
        }
    }
    int result[2][2] = {{0,0}, {0,0}};
    // Providing Matrix Multiplication Results
    printf("Result: \n");
    result[0][0] = a[0][0] * b[0][0] + a[0][1] * b[1][0];
    result[0][1] = a[0][0] * b[0][1] + a[0][1] * b[1][1];
    result[1][0] = a[1][0] * b[0][0] + a[1][1] * b[1][0];
    result[1][1] = a[1][0] * b[0][1] + a[1][1] * b[1][1];
    printf("%d %d\n", result[0][0], result [0][1]);
    printf("%d %d\n", result[1][0], result [1][1]);
    return 0;
}
```

Comments include a concise, but meaningful representation of the code written. My main focus when designing was to write as little code as possible, while also keeping the solution simple. I did this by creating the getUserInput() function below that takes an array to changed, with row and column inputs, as well as a name character that would be used for the display.

```
void getUserInput(int array[2][2], int row, int column, const char* name) {
    int input = 0;
    int result = 0; // Result indicator for input
    while (result != 1) { // Loop until valid input is entered: EXTRA CREDIT
        printf("%s%d%d = ", name, row + 1, column + 1);
        result = scanf("%d", &input);
        if (result == 1) {
            array[row][column] = input;
            break;
        } else {
            printf("ERROR: Invalid input. Please enter numbers only!\n");
            while (getchar() != '\n');
        }
    }
}
```

I printed the name, row + 1, then column + 1 (since array indexes start at 0 instead of 1) with an equal sign with the format provided, as well as handled incorrect type inputs by using the state of the scanf function that returns either 0 or 1 using

Result = scanf("%d", &input);

After that, I just made sure to iterate through all rows and columns of both 2x2 matrices, then computed the result to

print in the correct format.