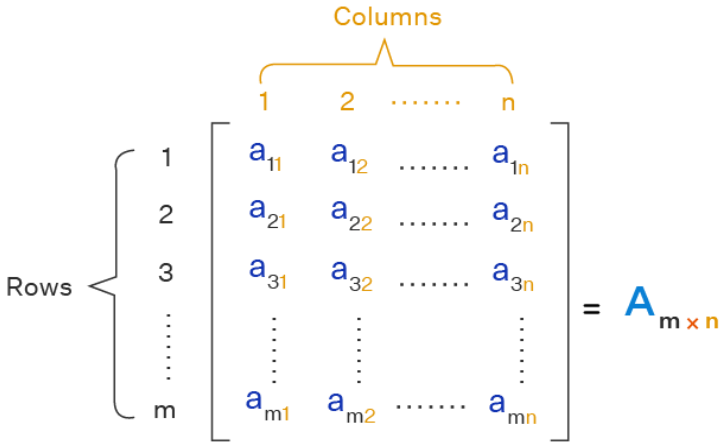
Introduction:



A matrix represents a collection of numbers arranged in an order of rows and columns. It is necessary to enclose the elements of a matrix in parentheses or brackets. For example: A matrix with 9 elements is shown below.

Matrix calculator calculates the resultant matrix when certain arithmetic operations are applied to the two given matrices. In mathematics, Matrix is a grid function or a rectangular array in which the numbers are arranged in ordered rows and columns.

They are extensively used in machine learning models like linear regression, logistic regression, etc.

In real life, Matrix Mathematics is used in optical science to account for refraction and reflection. Matrix is also useful in electrical circuits and quantum physics. Moreover, matrices are used to solve AC network equations in electrical circuits.

Program Input:

This program calculates the addition, subtraction and multiplication of matrices as well as scalar multiplication.

The details of input of the program are given below:

After showing greetings, this program showing a menu that includes several options asks to give input for the specified option.

The menu is given-

Operation Menu

1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices

If we select option 1 or 2
Then the program asks to give the rows and columns of the 1st matrix. Then when we enter the rows and columns, the program asks us to give the second matrix's rows and column numbers. After entering the second matrix row and column number it calculates the resultant matrix and shows the result.

If we select option 3
The program will first ask for the scalar value. After that it will ask for the Matrix dimensions. And after that the program will ask for the matrix elements. After Entering the Matrix elements the program will display the result of scalar multiplication.

If we select option 4,
Then the program will ask for the number of rows and columns in each matrix. If the 1st matrix column number is equal to the 2nd matrix row number, then the program will continue. Then the program will ask us for the elements of the 1st matrix and after that elements of the second matrix.

If the 1st Matrix Number of Column doesn't match with the 2nd Matrix Number of Rows then it will throw an error, because the operation is not mathematically defined.

Process:

Step 1: Start the Program.

Step 2: Enter the row and column of the first (A) matrix.

Step 3: Enter the row and column of the second (B) matrix.

Step 4: Enter the elements of the first (A) matrix.

Step 5: Enter the elements of the second (B) matrix.

Step 6: Print the elements of the first (A) matrix in matrix form.

Step 7: Print the elements of the second (B) matrix in matrix form.

Step 8: Set a loop up to row.

Step 9: Set an inner loop up to the column.

Step 10: Set another inner loop up to the column.

Step 12: Print the final matrix.

Step 13: Stop the Program.

Output:

This program calculates the addition, subtraction and multiplication of matrices as well as scalar multiplication.

Here,the menu is given–

Operation Menu

1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices

If the given input is 1-after show some process we shows the matrices addition.Similarly,when we choice option 2-we shows the matrices subtraction,option 3-we know the scalar multiplication.Again when we select option-4 we find to multiply two matrices but remember that if the 1st Matrix Number of Column doesn't match with the 2nd Matrix Number of Rows then it will throw an error, because the operation is not mathematically defined.When the output is find the program shows"Do you want to calculate again?"if we type y then the program shows(process step-2)or we type N the output shows Goodbye!

Source code:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <ctype.h>

//User Defined Function Declaration
void readMatrix(int array[10][10], int rows, int colums);
void printMatrix(int array[10][10], int rows, int colums);
void matrixAddSub(int arrayone[10][10], int arraytwo[10][10], int rows, int colums, int mul);
void matrixScalarMultiply(int array[10][10], int scalar, int rows, int colums);
void matrixMultiply(int arrayone[10][10], int arraytwo[10][10], int rowsA, int columsA, int
columsB);

int main(void){

    int i, j, k; //used in for loops

    int matrixA[10][10]; // initialized at 10 just to have it initialized
    int matrixB[10][10];
    int rowA, colA;
    int rowB, colB;
    int operation;//used in swtich statements
    char again = 'Y';
    int scalar = 0;
    int add = 1;
    int sub = -1;

    while (again == 'Y'){

        //this is the operation menu just type A, B, C or D to calculate
        printf("\nOperation Menu\n");
        printf("\t1. to Add\n");
        printf("\t2. to Subtract\n");
        printf("\t3. to Scalar Multiply\n");
        printf("\t4. to Multiply two matrices\n");
        printf("Enter yout choice: ");
        scanf(" %d", &operation);

        switch (operation){

        case 1:
```

```
        printf("\nEnter the #rows and #cols for matrix A: ");
        scanf("%d%d", &rowA, &colA);

        printf("Enter the #rows and #cols for matrix B: ");
        scanf("%d%d", &rowB, &colB);

        while ((rowA != rowB) && (colA != colB)){
            printf("\nMatrices must be the same size\n");
            printf("\nEnter the #rows and #cols for matrix A: ");
            scanf("%d%d", &rowA, &colA);

            printf("Enter the #rows and #cols for matrix B: ");
            scanf("%d%d", &rowB, &colB);

        }

        printf("\n\tEnter elements of Matrix A a %d x %d matrix.\n", rowA, colA); // with the
%d we remember the user the dimentions of the array
        readMatrix(matrixA, rowA, colA);
        printf("\n\t\tMatrix A\n\n");
        printMatrix(matrixA, rowA, colA);


        printf("\n\tEnter elements of Matrix B a %d x %d matrix.\n", rowB, colB); // with the
%d we remember the user the dimentions of the array
        readMatrix(matrixB, rowB, colB);
        printf("\n\t\tMatrix B\n\n");
        printMatrix(matrixB, rowB, colB);


        printf("\nThe Sum of matrixA + matrixB is : \n");
        matrixAddSub(matrixA, matrixB, rowA, colA, add);

        break;

    case 2:

        printf("\nEnter the #rows and #cols for matrix A: ");
        scanf("%d%d", &rowA, &colA);

        printf("Enter the #rows and #cols for matrix B: ");
        scanf("%d%d", &rowB, &colB);
```

```
        while ((rowA != rowB) && (colA != colB)){
            printf("\nMatrices must be the same size\n");
            printf("\nEnter the #rows and #cols for matrix A: ");
            scanf("%d%d", &rowA, &colA);

            printf("Enter the #rows and #cols for matrix B: ");
            scanf("%d%d", &rowB, &colB);
        }

        printf("\n\tEnter elements of Matrix A a %d x %d matrix.\n", rowA, colA); // with the
%d we remember the user the dimentions of the array
        readMatrix(matrixA, rowA, colA);
        printf("\n\t\tMatrix A\n\n");
        printMatrix(matrixA, rowA, colA);

        printf("\n\tEnter elements of Matrix B a %d x %d matrix.\n", rowB, colB); // with the
%d we remember the user the dimentions of the array
        readMatrix(matrixB, rowB, colB);
        printf("\n\t\tMatrix B\n\n");
        printMatrix(matrixB, rowB, colB);

        printf("\nThe difference between matrixA - matrixB is : \n");
        matrixAddSub(matrixA, matrixB, rowA, colA, sub);
        break;

    case 3:

        printf("\nEnter the scalar: ");
        scanf("%d", &scalar);
        printf("\nThe scalar is: %d ", scalar);


        printf("\nEnter the #rows and #cols for matrix A: ");
        scanf("%d%d", &rowA, &colA);

        printf("\n\tEnter elements of Matrix A a %d x %d matrix.\n", rowA, colA); // with the
%d we remember the user the dimentions of the array
        readMatrix(matrixA, rowA, colA);
        printf("\n\t\tMatrix A\n\n");
        printMatrix(matrixA, rowA, colA);
```

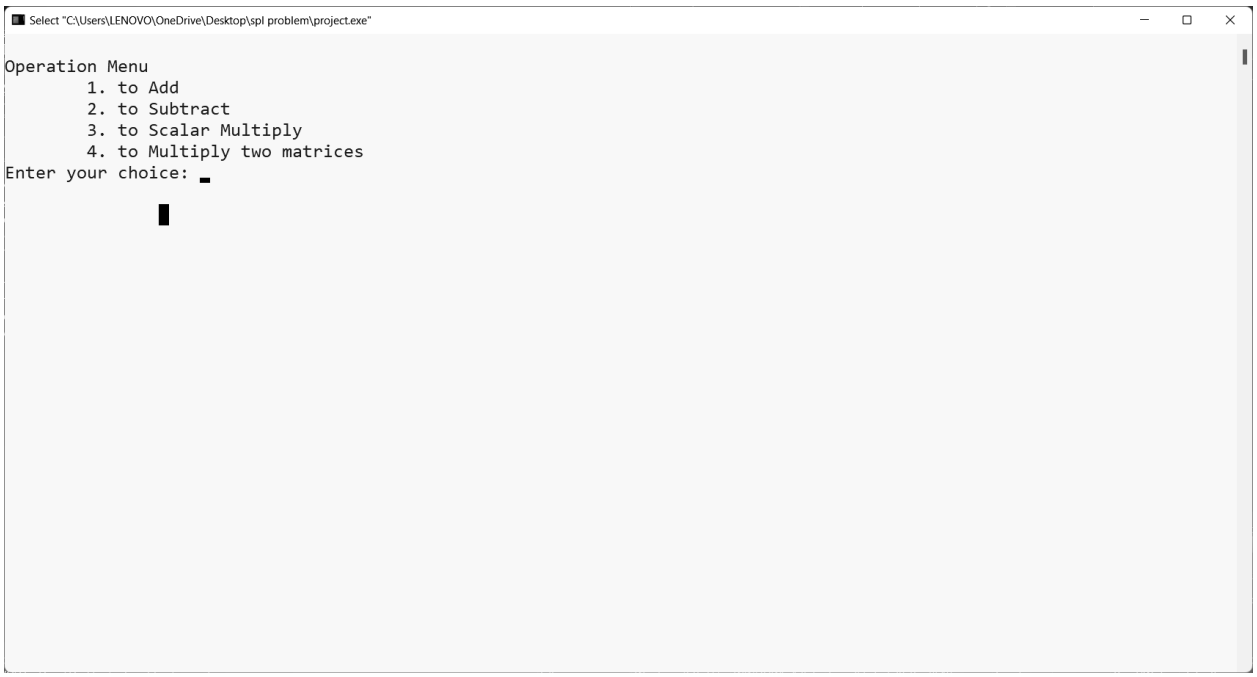
```
            printf("\nThe scalar multiplication between matrixA * %d is: \n", scalar);
            matrixScalarMultiply(matrixA, scalar, rowA, colA);

            break;

        case 4:
            //when mulotiplying arrays matrixA colum # has to equal matrixB row #
            printf("\nEnter the #rows and #cols for matrix A: ");
            scanf("%d%d", &rowA, &colA);

            printf("Enter the #rows and #cols for matrix B: ");
            scanf("%d%d", &rowB, &colB);

            // Column of first matrix should be equal to column of second matrix and
            while (colA != rowB)
            {
                printf("\n\nError! column of first matrix not equal to row of second.\n\n");
                printf("\nEnter the #rows and #cols for matrix A: ");
                scanf("%d%d", &rowA, &colA);

                printf("Enter the #rows and #cols for matrix B: ");
                scanf("%d%d", &rowB, &colB);
            }

            // Storing elements of first matrix.
            printf("\n\tEnter elements of Matrix A a %d x %d matrix.\n", rowA, colA); // with the
%d we remember the user the dimentions of the array
            readMatrix(matrixA, rowA, colA);
            printf("\n\t\tMatrix A\n\n");
            printMatrix(matrixA, rowA, colA);

            // Storing elements of second matrix.
            printf("\n\tEnter elements of Matrix B a %d x %d matrix.\n", rowB, colB); // with the
%d we remember the user the dimentions of the array
            readMatrix(matrixB, rowB, colB);
            printf("\n\t\tMatrix A\n\n");
            printMatrix(matrixB, rowB, colB);

            //multiplyng arrays
            matrixMultiply(matrixA, matrixB, rowA, colA, colB);

            break;
```

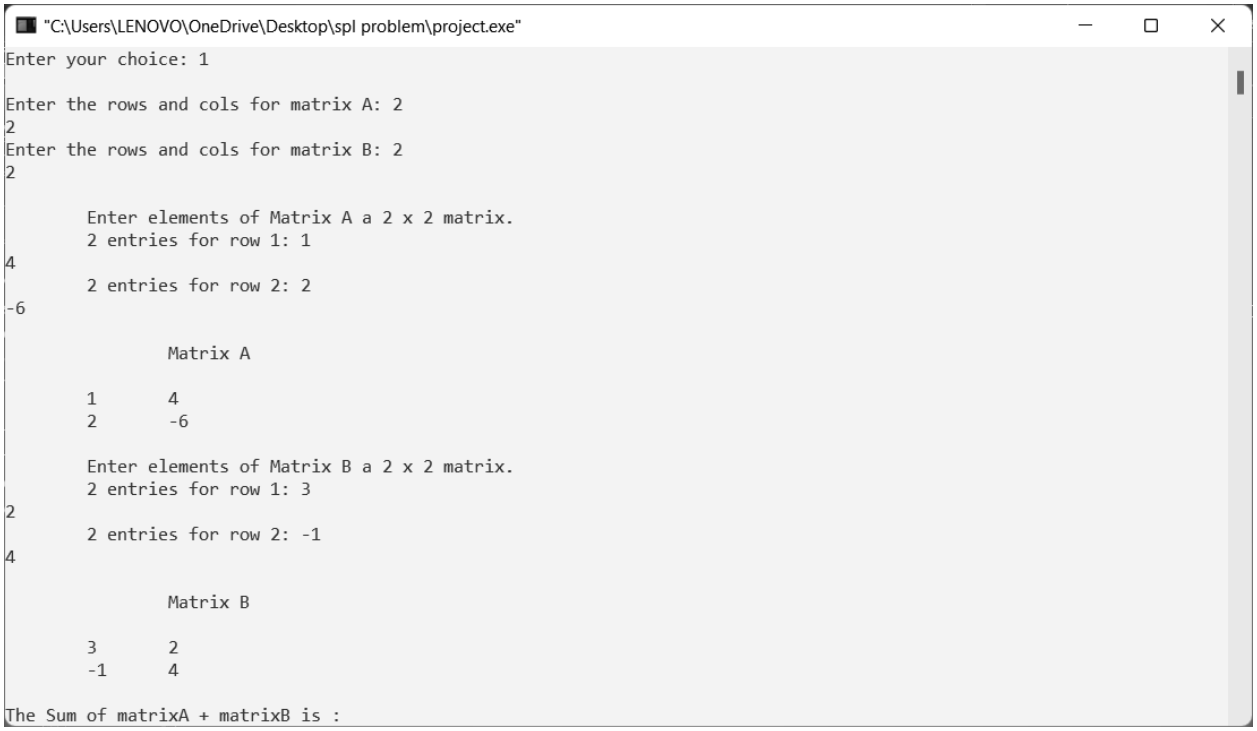
```
        default:
            printf("\nIncorrect option! Please choose a number 1-4.");
            break;
        }

        printf("\n\nDo you want to calculate again? Y/N\n");
        scanf(" %c", &again);
        again = toupper(again);
    }
    printf("\n\nGoodbye!\n\n");

    return 0;
}



//User Defined Function Definition
void readMatrix(int array[10][10], int rows, int colums){
    int i, j;
    for (i = 0; i < rows; i++){
        printf("\t%d entries for row %d: ", colums, i + 1);
        for (j = 0; j < colums; j++){
            scanf("%d", &array[i][j]);
        }
    }

    return;
}

void printMatrix(int array[10][10], int rows, int colums){
    int i, j;

    for (i = 0; i < rows; i++) {
        for (j = 0; j < colums; j++){
            printf("\t%d", array[i][j]);
        }
        printf("\n");
    }
}
```

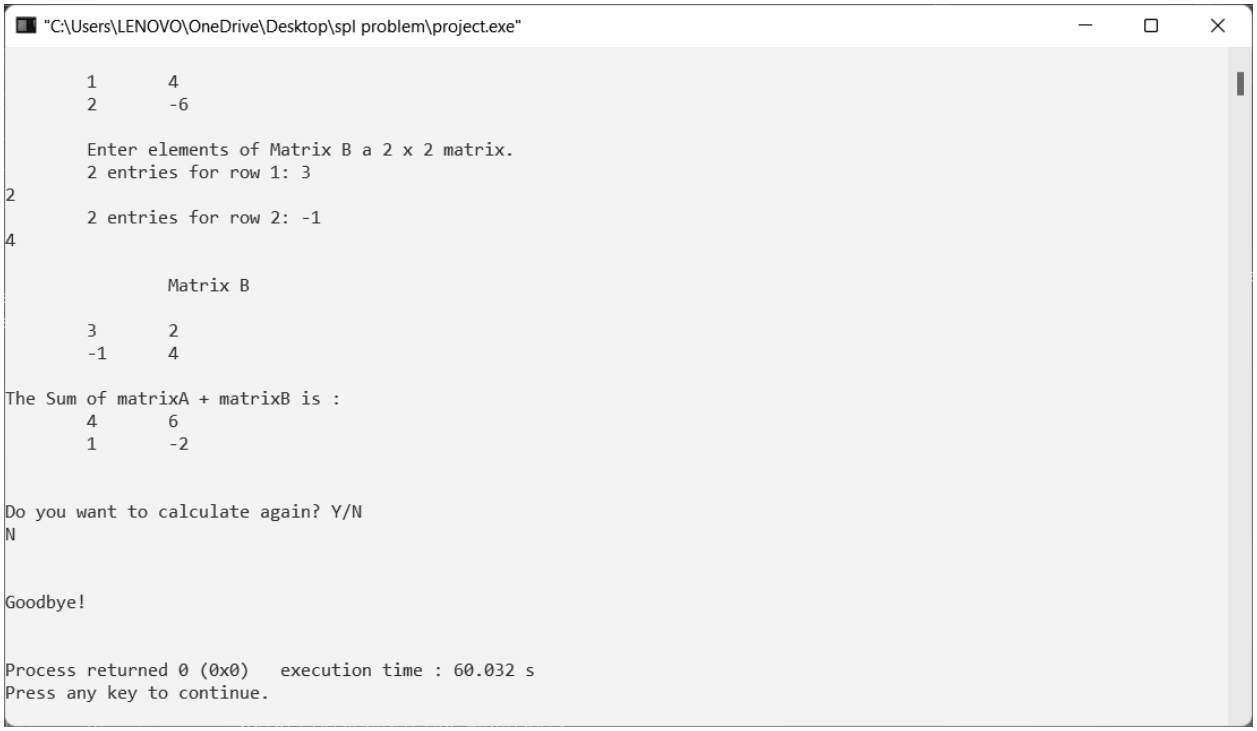
```
void matrixAddSub(int arrayone[10][10], int arraytwo[10][10], int rows, int colums, int mul){
    int i, j;
    int sumM[10][10];
    int scaM[10][10];
    for (i = 0; i < rows; i++){
        for (j = 0; j < colums; j++){
            scaM[i][j] = mul * arraytwo[i][j];
            }
        }


    for (i = 0; i < rows; i++){
        for (j = 0; j < colums; j++){
            sumM[i][j] = arrayone[i][j] + scaM[i][j];
            printf("\t%d", sumM[i][j]);
        }
        printf("\n");
    }
}


void matrixScalarMultiply(int array[10][10], int scalar, int rows, int colums){
    int i, j;
    int scaM[10][10];
    for (i = 0; i < rows; i++){
        for (j = 0; j < colums; j++){
            scaM[i][j] = scalar * array[i][j];
            printf("%d\t", scaM[i][j]);
        }
        printf("\n");
    }
}

void matrixMultiply(int arrayone[10][10], int arraytwo[10][10], int rowsA, int columsA,int
columsB){
    int i, j, k;
    int mulM[10][10];

    // Initializing all elements of result matrix to 0
    for (i = 0; i<rowsA; ++i)
        for (j = 0; j<columsB; ++j)
        {
```

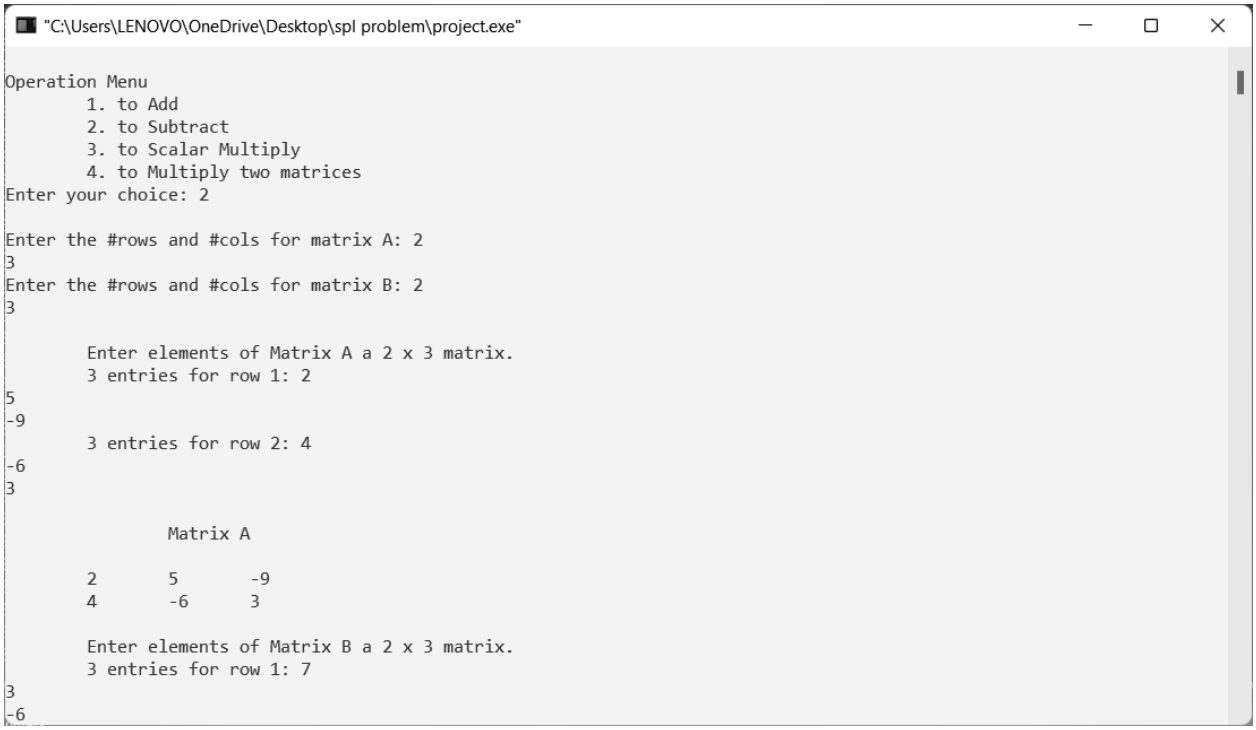
```
            mulM[i][j] = 0;
        }

    // Multiplying matrices a and b and
    // storing result in result matrix
    for (i = 0; i<rowsA; ++i)
        for (j = 0; j<columsB; ++j)
            for (k = 0; k<columsA; ++k)
            {
                mulM[i][j] += arrayone[i][k] * arraytwo[k][j];
            }
    printf("\nOutput Matrix:\n");
    for (i = 0; i<rowsA; ++i)
        for (j = 0; j<columsB; ++j)
        {
            printf("\t%d ", mulM[i][j]);
            if (j == columsB - 1)
                printf("\n\n");
        }
}
```

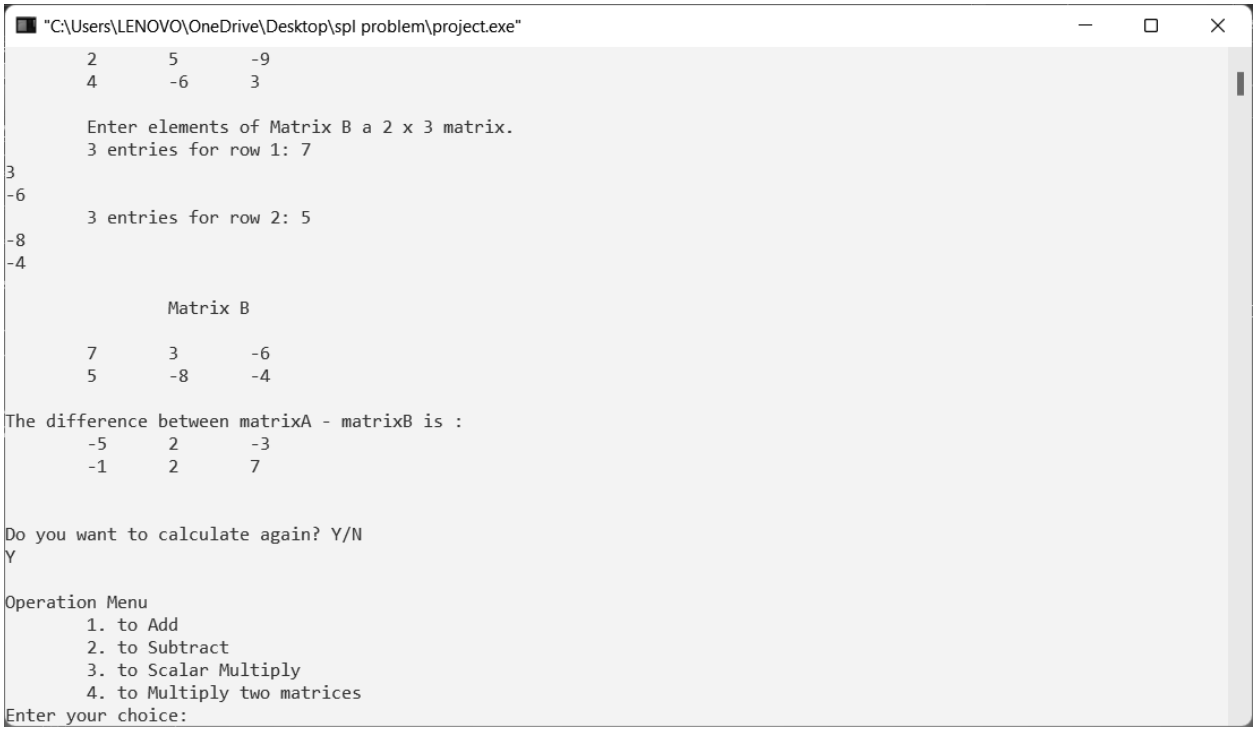
Output:

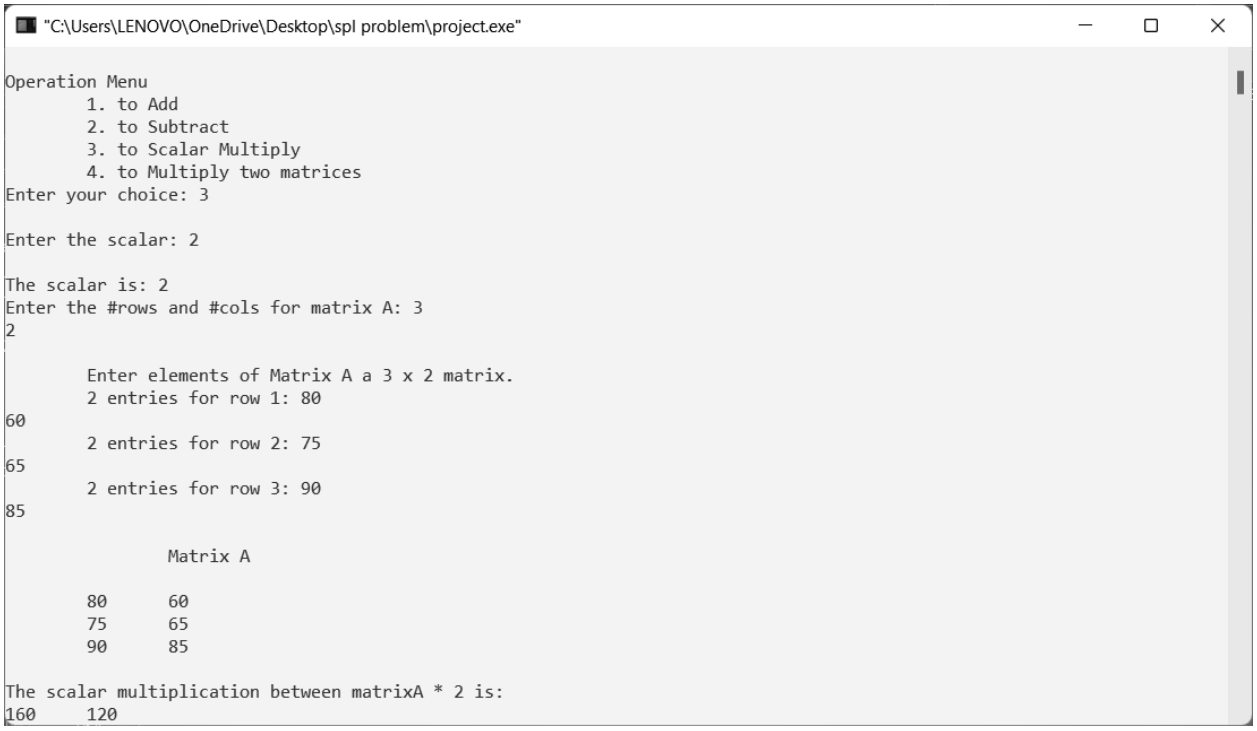


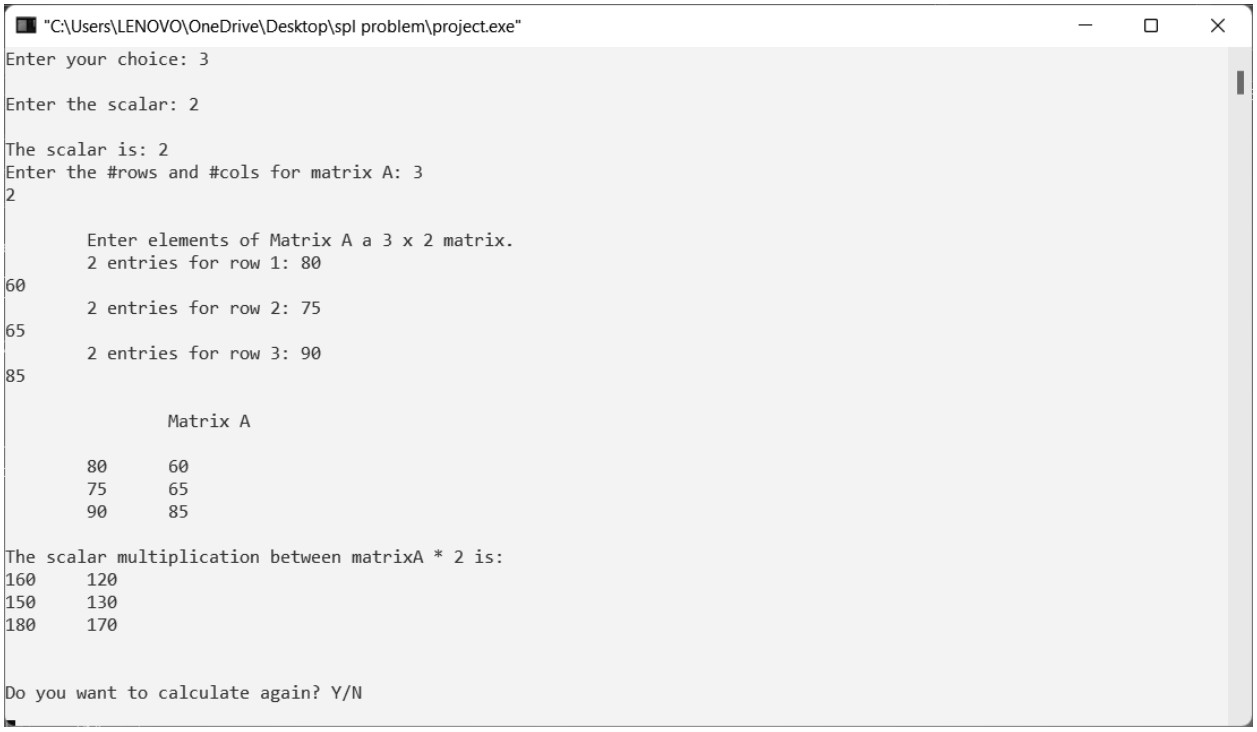

"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
Enter your choice: 1
Enter the rows and cols for matrix A: 2
2
Enter the rows and cols for matrix B: 2
2
Enter elements of Matrix A a 2 x 2 matrix.
2 entries for row 1: 1
4
2 entries for row 2: 2
-6
Matrix A
1 4
2 -6
Enter elements of Matrix B a 2 x 2 matrix.
2 entries for row 1: 3
2
2 entries for row 2: -1
4
Matrix B
3 2
-1 4
The Sum of matrixA + matrixB is :

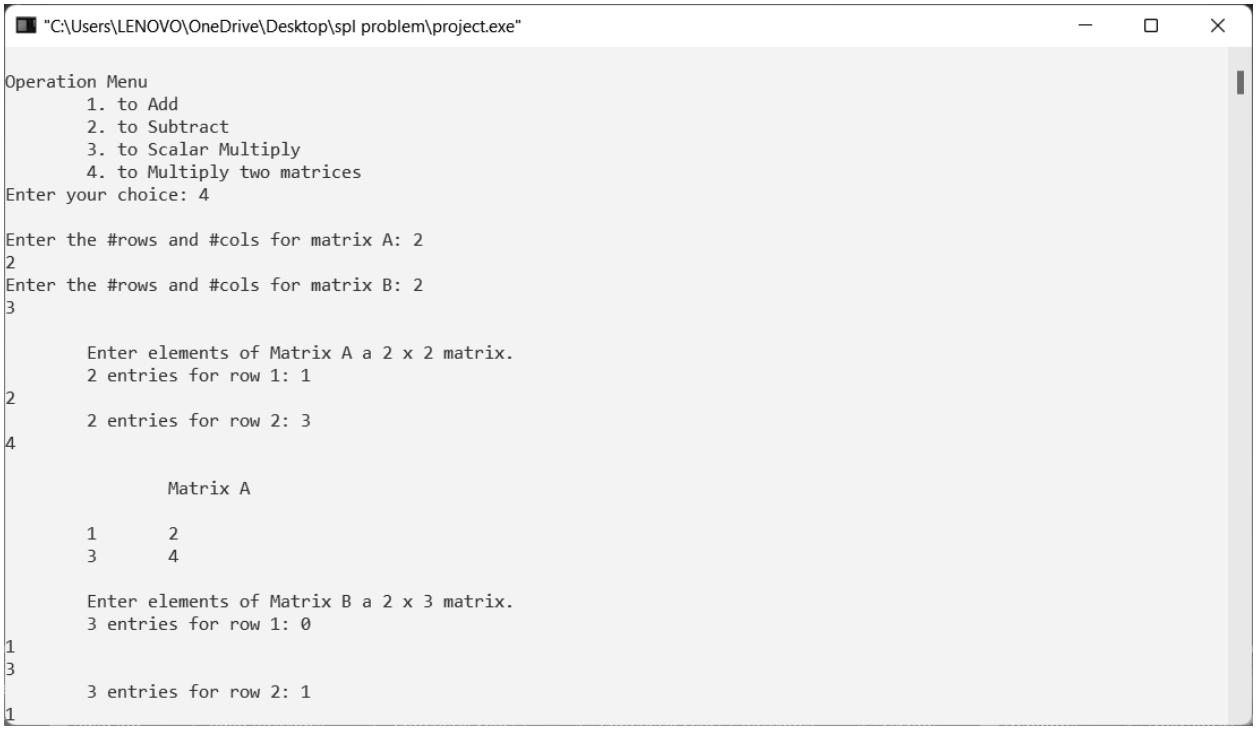

"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
1 4
2 -6
Enter elements of Matrix B a 2 x 2 matrix.
2 entries for row 1: 3
2
2 entries for row 2: -1
4
Matrix B
3 2
-1 4
The Sum of matrixA + matrixB is :
4 6
1 -2
Do you want to calculate again? Y/N
N
Goodbye!
Process returned 0 (0x0) execution time : 60.032 s
Press any key to continue.

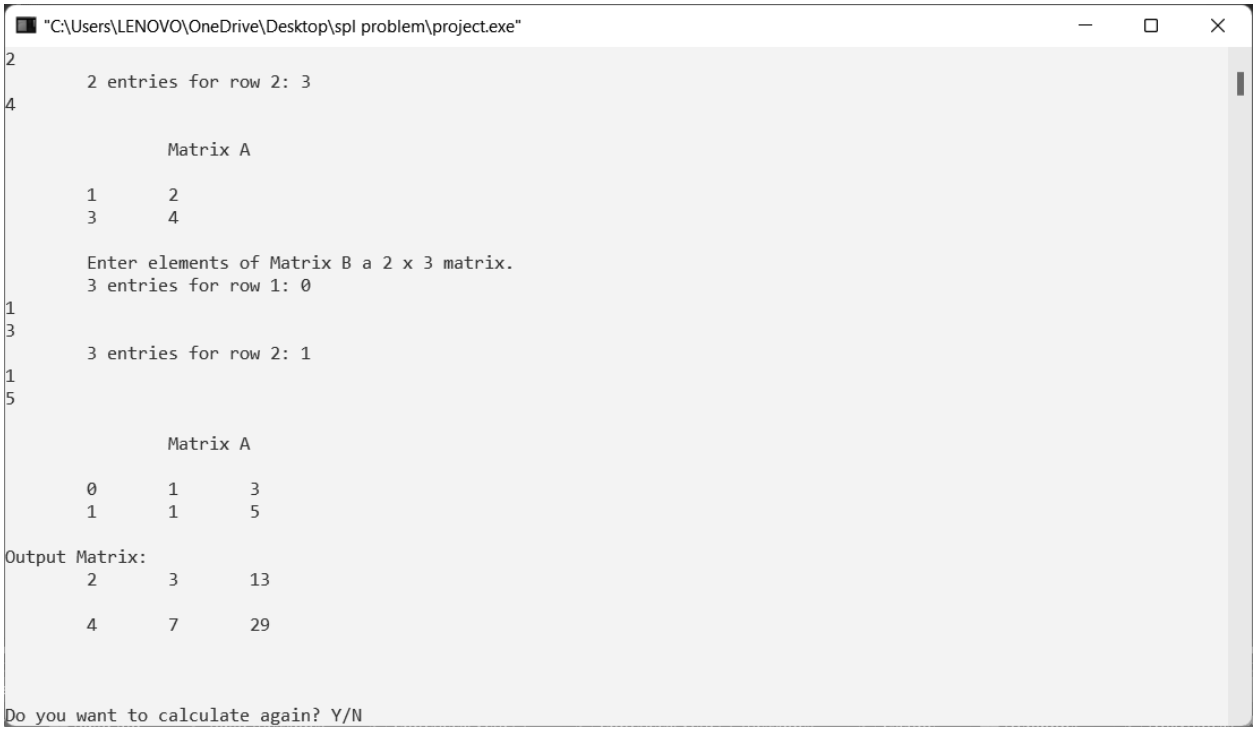

"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter your choice: 2
Enter the #rows and #cols for matrix A: 2
3
Enter the #rows and #cols for matrix B: 2
3
Enter elements of Matrix A a 2 x 3 matrix.
3 entries for row 1: 2
5
-9
3 entries for row 2: 4
-6
3
Matrix A
2 5 -9
4 -6 3
Enter elements of Matrix B a 2 x 3 matrix.
3 entries for row 1: 7
3
-6


"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
2 5 -9
4 -6 3
Enter elements of Matrix B a 2 x 3 matrix.
3 entries for row 1: 7
3
-6
3 entries for row 2: 5
-8
-4
Matrix B
7 3 -6
5 -8 -4
The difference between matrixA - matrixB is :
-5 2 -3
-1 2 7
Do you want to calculate again? Y/N
Y
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter your choice:


"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter your choice: 3
Enter the scalar: 2
The scalar is: 2
Enter the #rows and #cols for matrix A: 3
2
Enter elements of Matrix A a 3 x 2 matrix.
2 entries for row 1: 80
60
2 entries for row 2: 75
65
2 entries for row 3: 90
85
Matrix A
80 60
75 65
90 85
The scalar multiplication between matrixA * 2 is:
160 120


"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
Enter your choice: 3
Enter the scalar: 2
The scalar is: 2
Enter the #rows and #cols for matrix A: 3
2
Enter elements of Matrix A a 3 x 2 matrix.
2 entries for row 1: 80
60
2 entries for row 2: 75
65
2 entries for row 3: 90
85
Matrix A
80 60
75 65
90 85
The scalar multiplication between matrixA * 2 is:
160 120
150 130
180 170
Do you want to calculate again? Y/N


"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter your choice: 4
Enter the #rows and #cols for matrix A: 2
2
Enter the #rows and #cols for matrix B: 2
3
Enter elements of Matrix A a 2 x 2 matrix.
2 entries for row 1: 1
2
2 entries for row 2: 3
4
Matrix A
1 2
3 4
Enter elements of Matrix B a 2 x 3 matrix.
3 entries for row 1: 0
1
3
3 entries for row 2: 1
1


"C:\Users\LENOVO\OneDrive\Desktop\spl problem\project.exe"
2
2 entries for row 2: 3
4
Matrix A
1 2
3 4
Enter elements of Matrix B a 2 x 3 matrix.
3 entries for row 1: 0
1
3
3 entries for row 2: 1
1
5
Matrix A
0 1 3
1 1 5
Output Matrix:
2 3 13
4 7 29
Do you want to calculate again? Y/N