SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be **University** *u/s 3 of UGC Act, 1956)*

Under the Guidance of

SUPERVISOR NAME
MS .GEETHA G (ASSISTANT PROFFESSOR(OG),DEPARTMENT OF NETWORKING AND COMMUNICATIONS)

(SUPERVISOR DESIGNATION, DEPARTMENT)

in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in Information Technology

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF

SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

Computer Science and Engineering(SC)-(K1 Section)

18CSS101J- MINI PROJECT

MATRIX CALCULATOR

RA2111028010006

KAMISETTY SATYA CHETAN

OBJECTIVE: The main objective of this matrix calculator is to calculate any matrix to addition or subtraction or scalar multiplication or multiplication by the coding process

SOURCE CODE:

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.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");
```

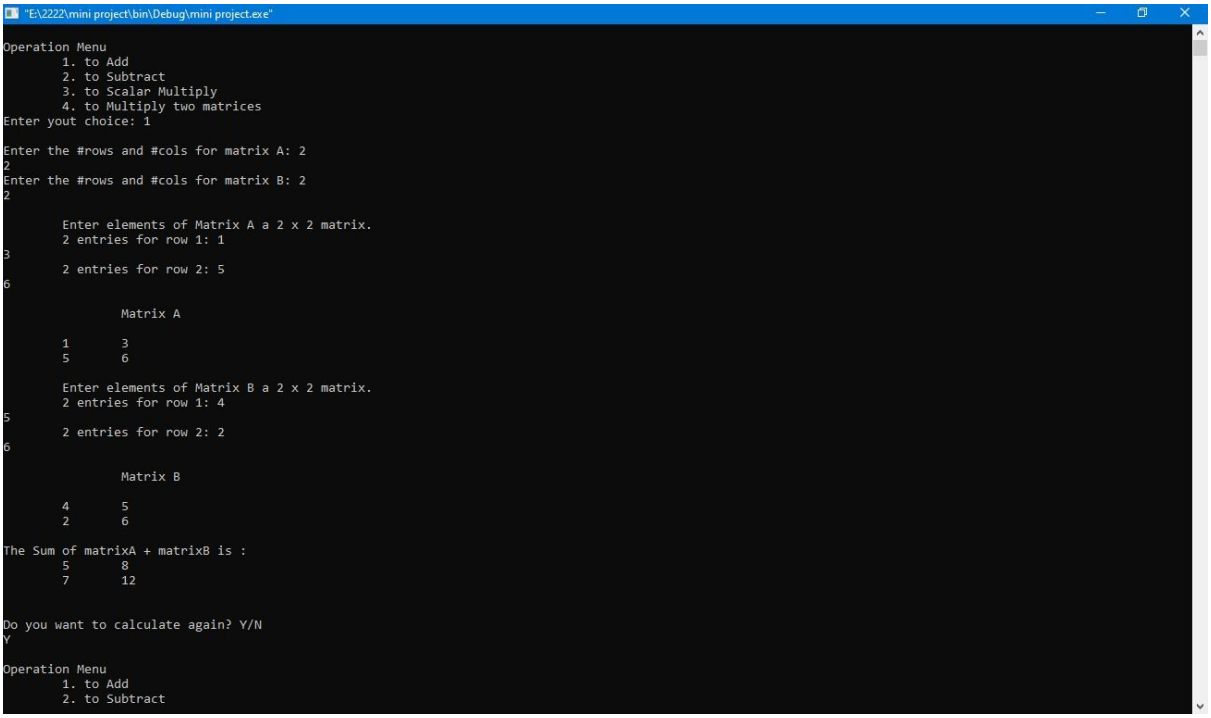
```
   return 0;
}



//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");
   }
}
```

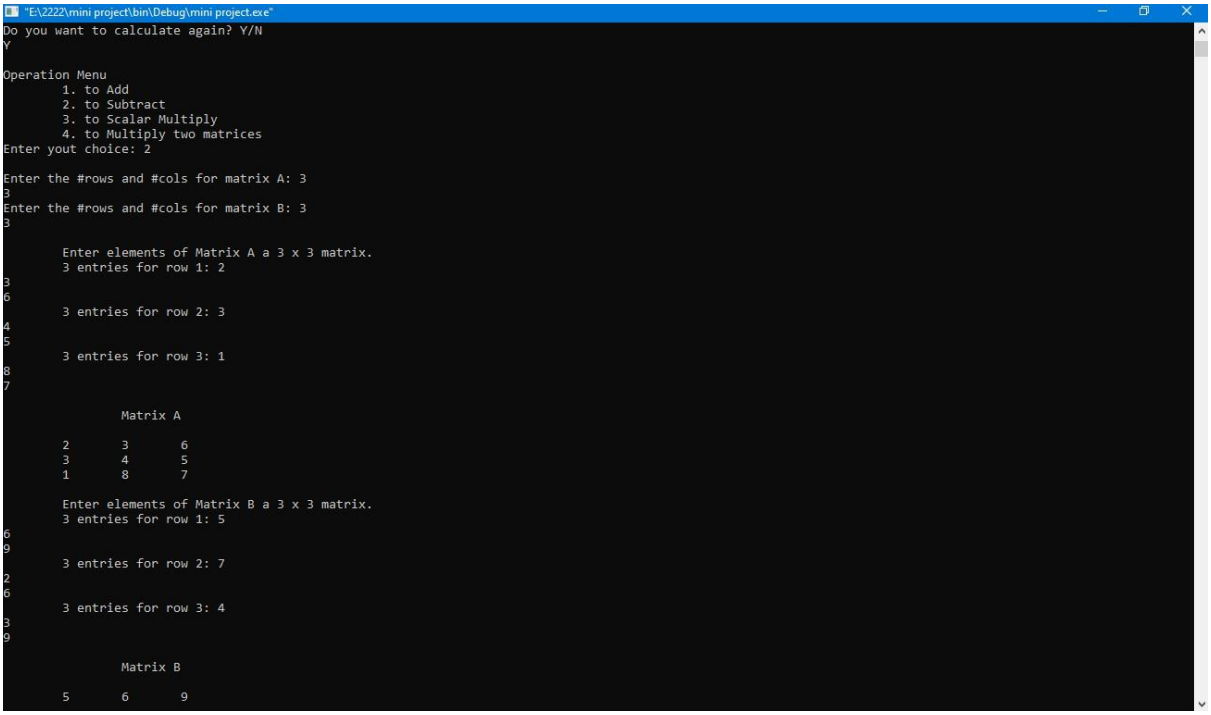
```
void matrixAddSub(int arrayone[10][10], int arraytwo[10][10], int rows, int colums, int
mul){
   int i, j;
   int sumM[10][10];
int scaM[10][10];
   for (i = 0; i < rows; i++){       for (j =
0; j < colums; j++){          scaM[i][j] =
mul * arraytwo[i][j];
         }
      }


   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]);
      }
```

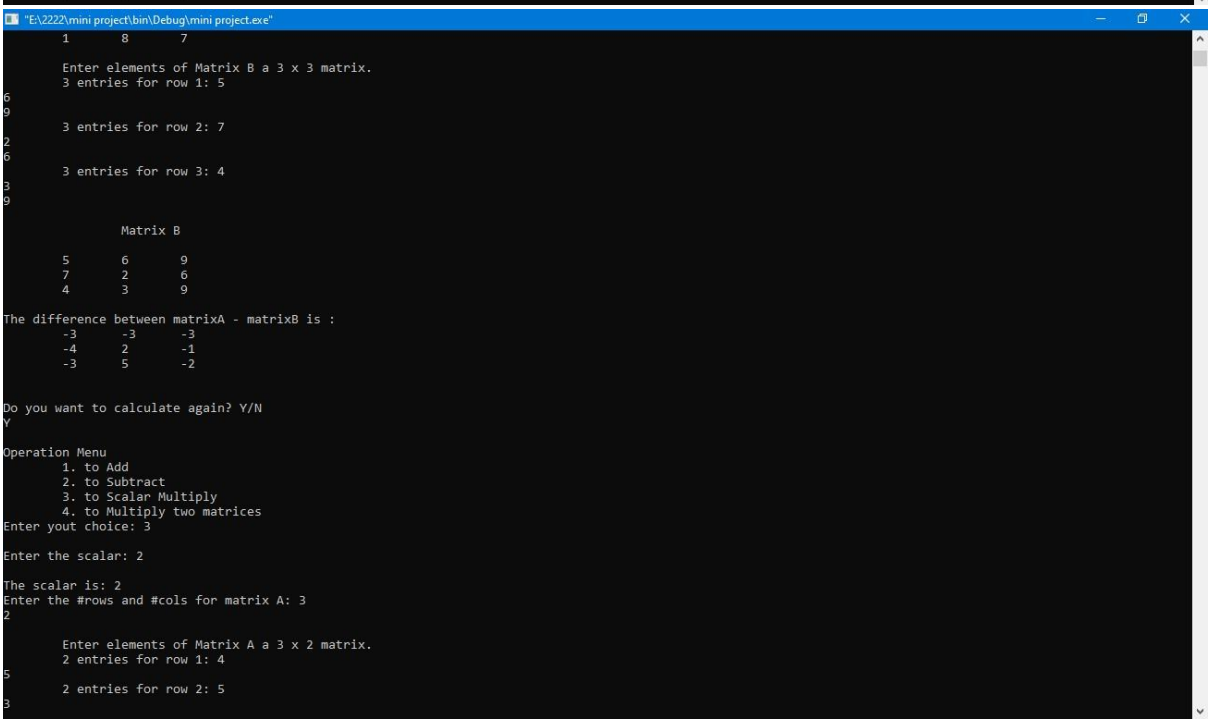
```
      printf("\n");
   }

}

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)
      {
```

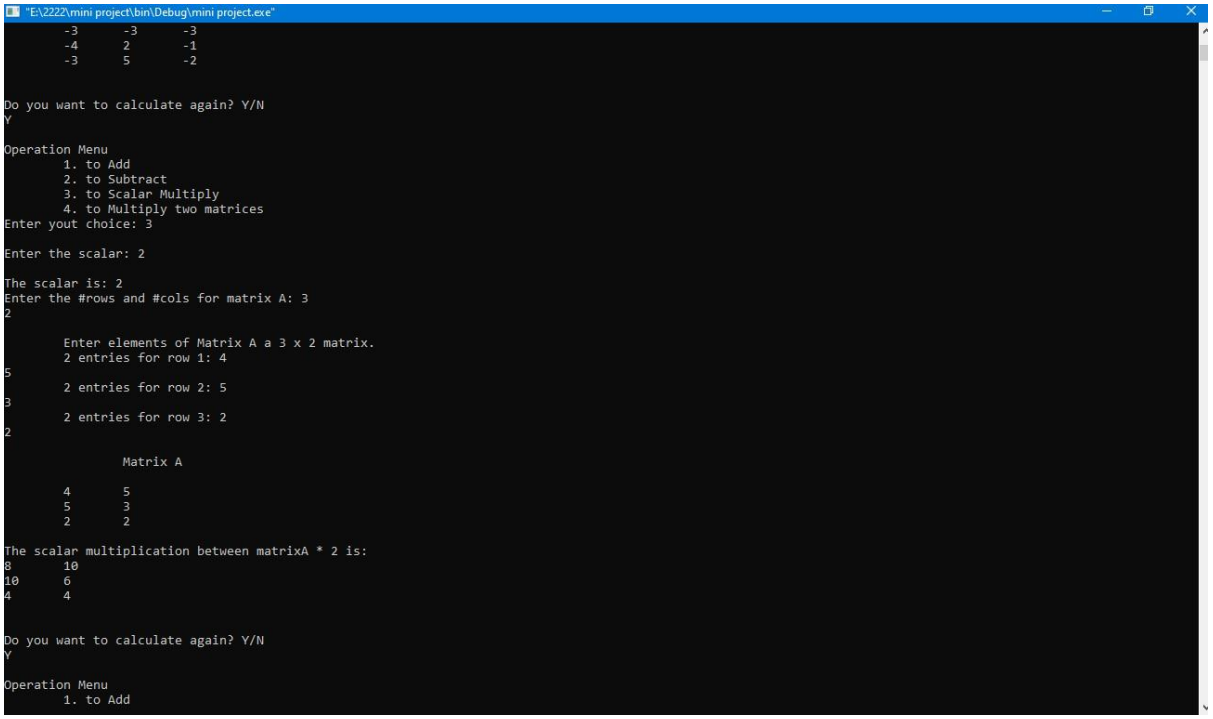
```
            printf("\t%d ", mulM[i][j]);
if (j == columsB - 1)
printf("\n\n");
        }
}
```

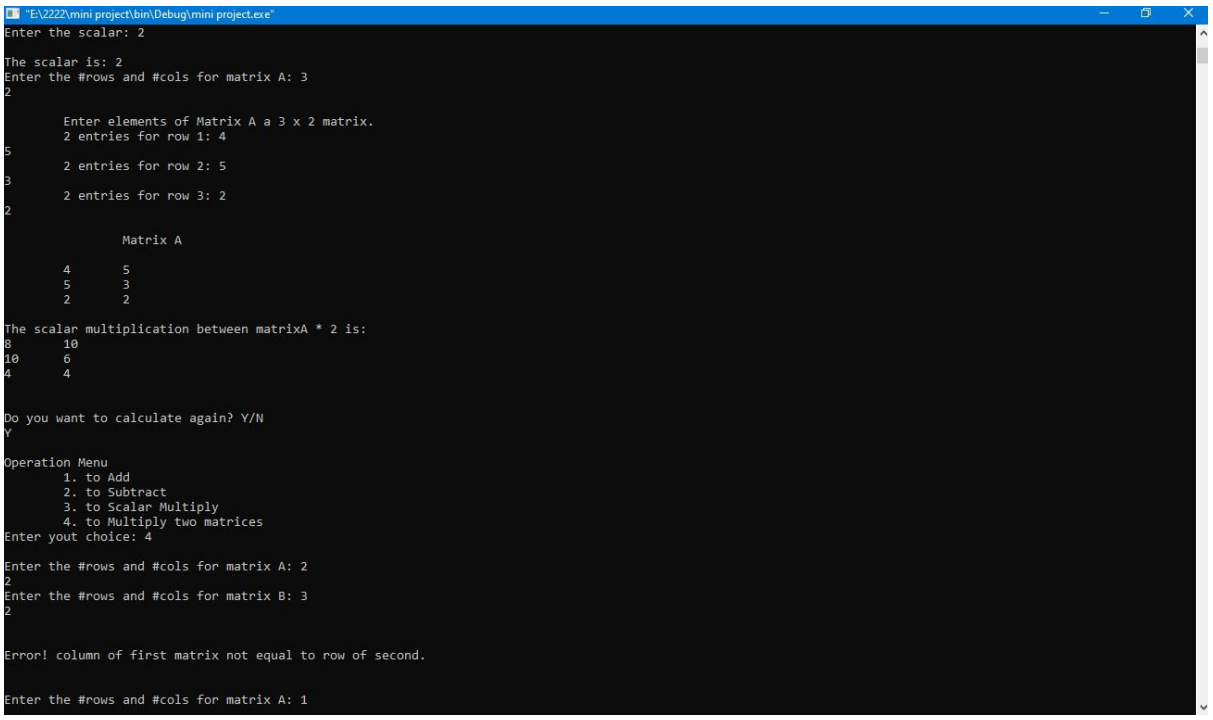
OUTPUT:

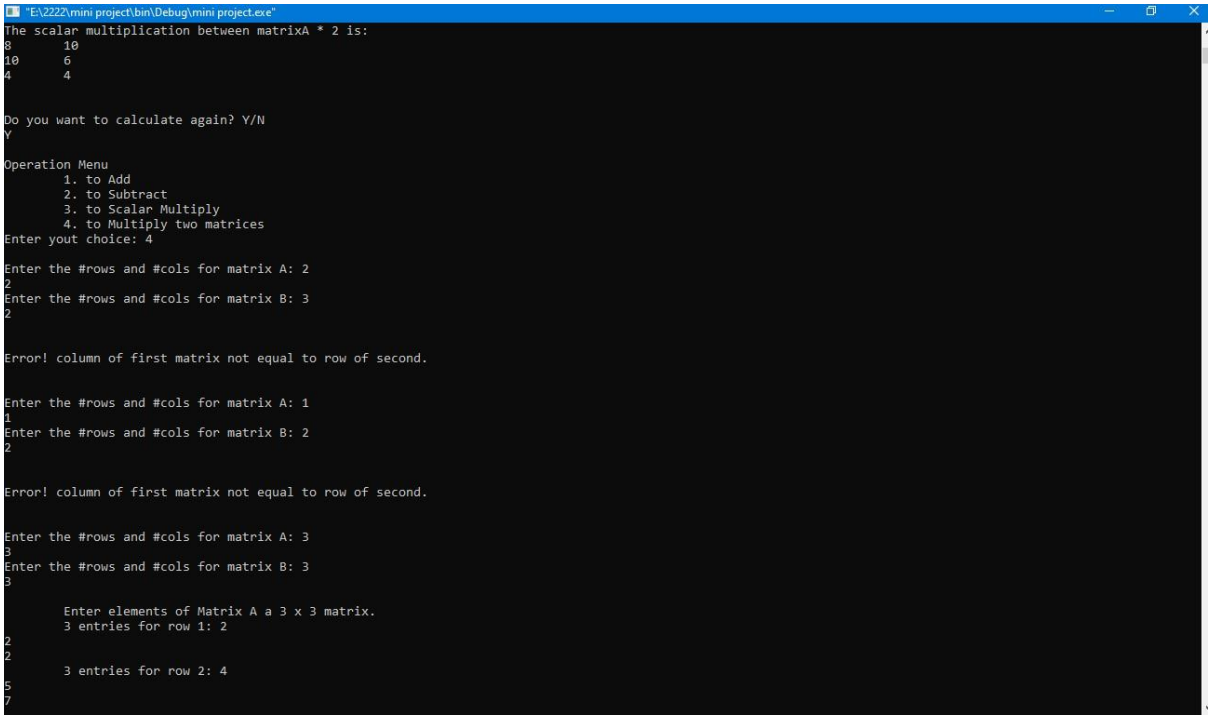


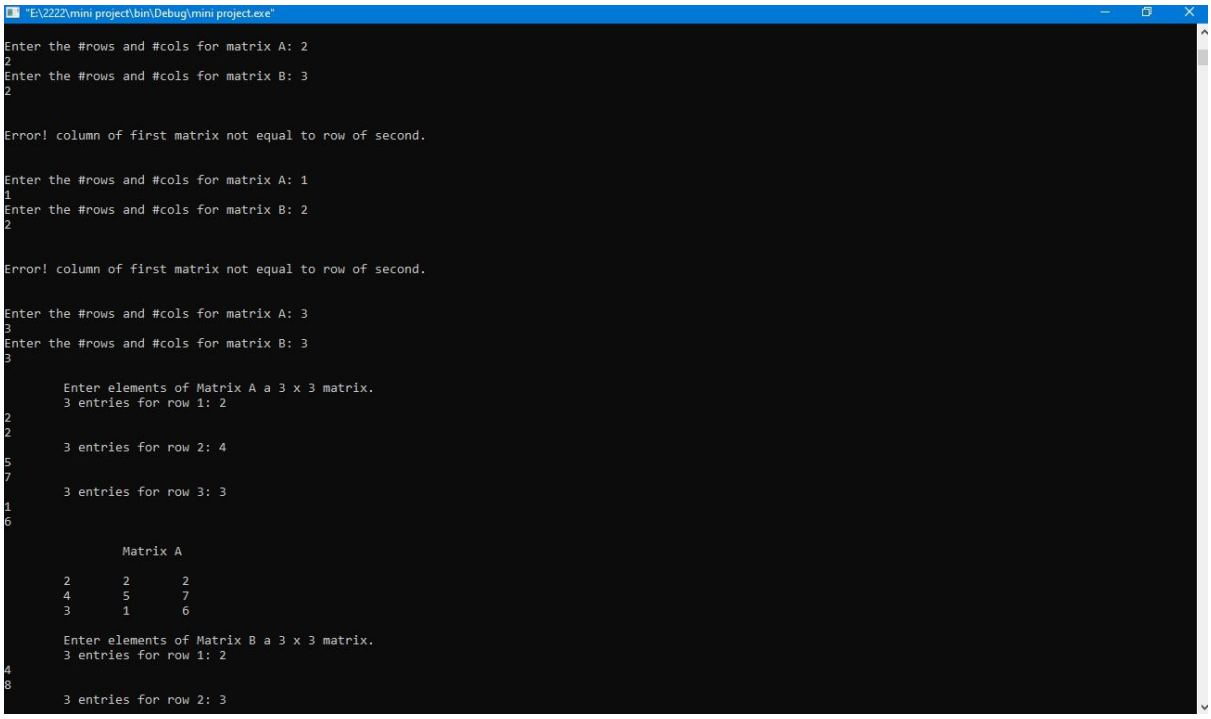

"E:\2222\mini project\bin\Debug\mini project.exe"
Do you want to calculate again? Y/N
Y
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter yout choice: 2
Enter the #rows and #cols for matrix A: 3
3
Enter the #rows and #cols for matrix B: 3
3
Enter elements of Matrix A a 3 x 3 matrix.
3 entries for row 1: 2
3
6
3 entries for row 2: 3
4
5
3 entries for row 3: 1
8
7
Matrix A
2 3 6
3 4 5
1 8 7
Enter elements of Matrix B a 3 x 3 matrix.
3 entries for row 1: 5
6
9
3 entries for row 2: 7
2
6
3 entries for row 3: 4
3
9
Matrix B
5 6 9

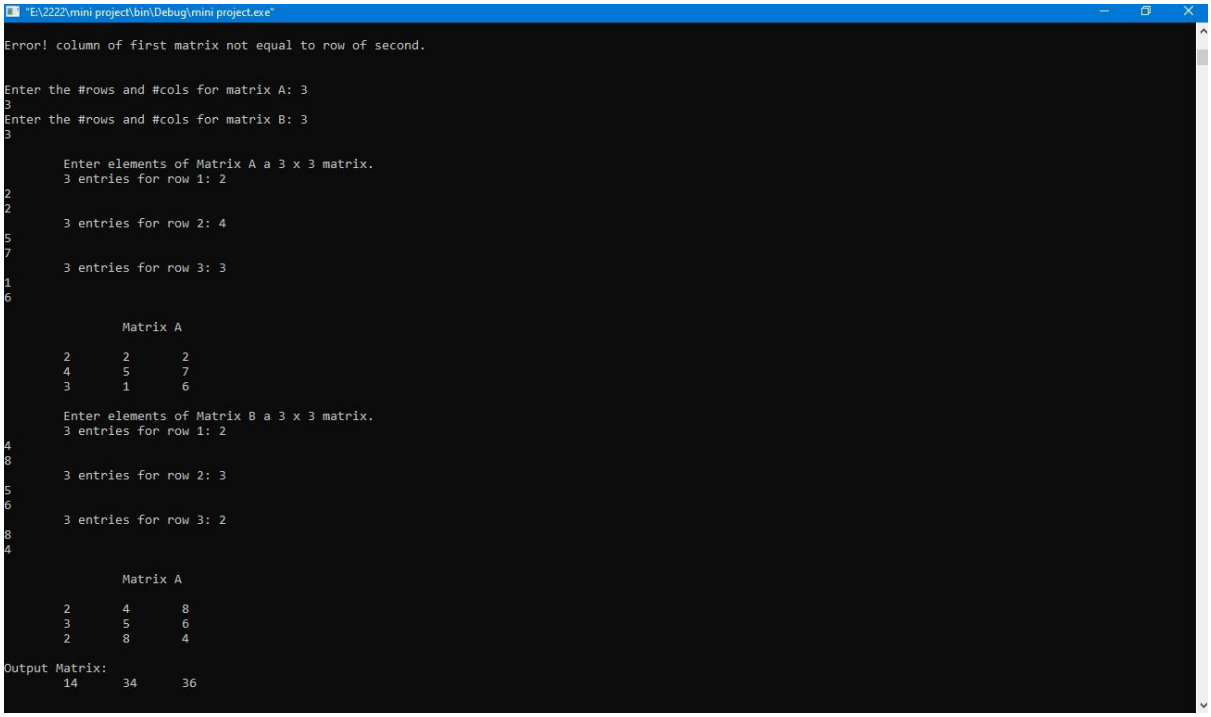

"E:\2222\mini project\bin\Debug\mini project.exe"
1 8 7
Enter elements of Matrix B a 3 x 3 matrix.
3 entries for row 1: 5
6
9
3 entries for row 2: 7
2
6
3 entries for row 3: 4
3
9
Matrix B
5 6 9
7 2 6
4 3 9
The difference between matrixA - matrixB is :
-3 -3 -3
-4 2 -1
-3 5 -2
Do you want to calculate again? Y/N
Y
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter yout 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: 4
5
2 entries for row 2: 5
3

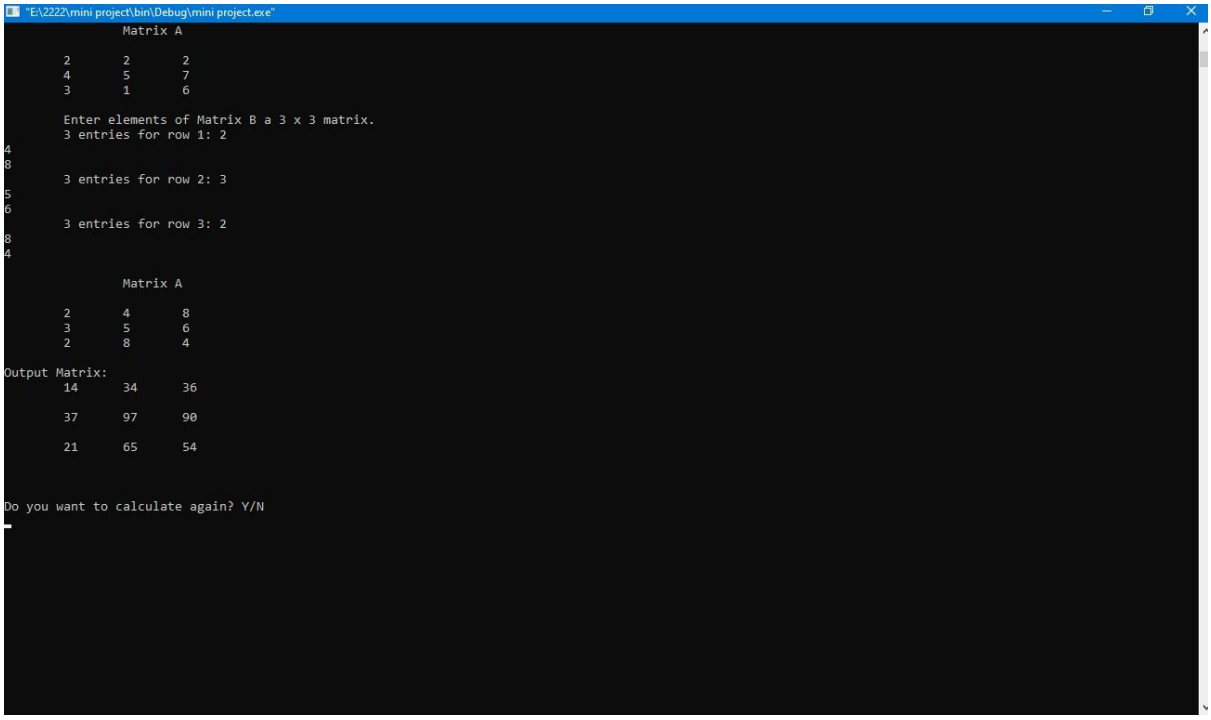

"E:\2222\mini project\bin\Debug\mini project.exe"
-3 -3 -3
-4 2 -1
-3 5 -2
Do you want to calculate again? Y/N
Y
Operation Menu
1. to Add
2. to Subtract
3. to Scalar Multiply
4. to Multiply two matrices
Enter yout choice: 3
Enter the scalar: 2
The scalar is: 2
Enter the #rows and #cols for matrix A: 3
2
Enter elements of Matrix A a 3 x 2 matrix.
2 entries for row 1: 4
5
2 entries for row 2: 5
3
2 entries for row 3: 2
2
Matrix A
4 5
5 3
2 2
The scalar multiplication between matrixA * 2 is:
8 10
10 6
4 4
Do you want to calculate again? Y/N
Y
Operation Menu
1. to Add


"E:\2222\mini project\bin\Debug\mini project.exe"
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: 4
5
2 entries for row 2: 5
3
2 entries for row 3: 2
2
Matrix A
4 5
5 3
2 2
The scalar multiplication between matrixA * 2 is:
8 10
10 6
4 4
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 yout choice: 4
Enter the #rows and #cols for matrix A: 2
2
Enter the #rows and #cols for matrix B: 3
2
Error! column of first matrix not equal to row of second.
Enter the #rows and #cols for matrix A: 1


"E:\2222\mini project\bin\Debug\mini project.exe"
The scalar multiplication between matrixA * 2 is:
8 10
10 6
4 4
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 yout choice: 4
Enter the #rows and #cols for matrix A: 2
2
Enter the #rows and #cols for matrix B: 3
2
Error! column of first matrix not equal to row of second.
Enter the #rows and #cols for matrix A: 1
1
Enter the #rows and #cols for matrix B: 2
2
Error! column of first matrix not equal to row of second.
Enter the #rows and #cols for matrix A: 3
3
Enter the #rows and #cols for matrix B: 3
3
Enter elements of Matrix A a 3 x 3 matrix.
3 entries for row 1: 2
2
2
3 entries for row 2: 4
5
7


"E:\2222\mini project\bin\Debug\mini project.exe"
Enter the #rows and #cols for matrix A: 2
2
Enter the #rows and #cols for matrix B: 3
2
Error! column of first matrix not equal to row of second.
Enter the #rows and #cols for matrix A: 1
1
Enter the #rows and #cols for matrix B: 2
2
Error! column of first matrix not equal to row of second.
Enter the #rows and #cols for matrix A: 3
3
Enter the #rows and #cols for matrix B: 3
3
Enter elements of Matrix A a 3 x 3 matrix.
3 entries for row 1: 2
2
2
3 entries for row 2: 4
5
7
3 entries for row 3: 3
1
6
Matrix A
2 2 2
4 5 7
3 1 6
Enter elements of Matrix B a 3 x 3 matrix.
3 entries for row 1: 2
4
8
3 entries for row 2: 3


"E:\2222\mini project\bin\Debug\mini project.exe"
Error! column of first matrix not equal to row of second.
Enter the #rows and #cols for matrix A: 3
3
Enter the #rows and #cols for matrix B: 3
3
Enter elements of Matrix A a 3 x 3 matrix.
3 entries for row 1: 2
2
2
3 entries for row 2: 4
5
7
3 entries for row 3: 3
1
6
Matrix A
2 2 2
4 5 7
3 1 6
Enter elements of Matrix B a 3 x 3 matrix.
3 entries for row 1: 2
4
8
3 entries for row 2: 3
5
6
3 entries for row 3: 2
8
4
Matrix A
2 4 8
3 5 6
2 8 4
Output Matrix:
14 34 36


"E:\2222\mini project\bin\Debug\mini project.exe"
Matrix A
2 2 2
4 5 7
3 1 6
Enter elements of Matrix B a 3 x 3 matrix.
3 entries for row 1: 2
4
8
3 entries for row 2: 3
5
6
3 entries for row 3: 2
8
4
Matrix A
2 4 8
3 5 6
2 8 4
Output Matrix:
14 34 36
37 97 90
21 65 54
Do you want to calculate again? Y/N