Write a C program for the following
Pass matrices as parameters

1. Matrix addition and subtraction
2. Multiplication
3. Sum of Rows and Columns
4. Sum of principle diagonal and non principle diagonal
5. Print transpose of given matrix
6. Check if matrix is symmetric or not

Code:

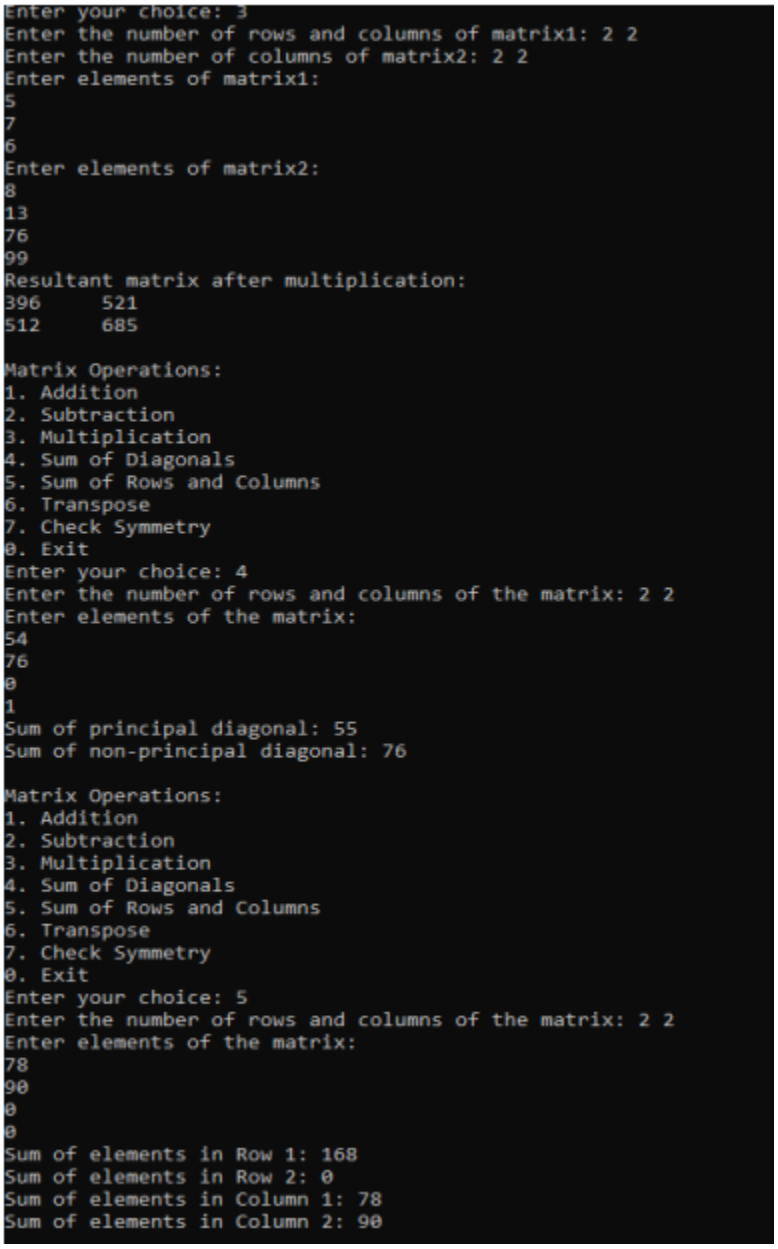
```
#include<stdio.h>
#include<conio.h>
void add(int matrix1[3][3], int matrix2[3][3])
{
   int sum[3][3];
   for(int i=0;i<3;i++)
      for(int j=0;j<3;j++)
         sum[i][j]=matrix1[i][j]+matrix2[i][j];
   for(int i=0;i<3;i++){
      printf("\n");
      for(int j=0;j<3;j++){
         printf("\t%d",sum[i][j])
         }
}
}

void subtract(int matrix1[3][3], int matrix2[3][3])
{
   int diff[3][3];
   for(int i=0;i<3;i++)
      for(int j=0;j<3;j++)
         diff[i][j]=matrix1[i][j]-matrix2[i][j];
   for(int i=0;i<3;i++){
      printf("\n");
      for(int j=0;j<3;j++){
         printf("\t%d",diff[i][j]);
         }
}
}

void multiply(int matrix1[3][3], int matrix2[3][3])
{
   int product[3][3];
```

```
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            product[i][j]=0;
            {
                for(int k=0;k<3;k++){
                    product[i][j]+=matrix1[i][k]*matrix2[k][j];
                }
            }
        }
    }
     for(int i=0;i<3;i++){
        printf("\n");
        for(int j=0;j<3;j++){
            printf("\t%d",product[i][j]);
            }
}
}

void sumOfRowsColumns(int matrix[3][3])
{
    int row_sum[3][4],column_sum[3][4],rowsum,columnsum;
    for(int i=0;i<3;i++){
        rowsum=0,columnsum=0;
        for(int j=0;j<3;j++){
            rowsum+=matrix[i][j];
            columnsum+=matrix[j][i];
        }
        row_sum[i][4]=rowsum;
        column_sum[i][4]=columnsum;

    }
    for(int i=0;i<3;i++){
        printf("\n");
        for(int j=0;j<4;j++){
            printf("\t%d",row_sum[i][j]);
            }
}
    for(int i=0;i<3;i++){
        printf("\n");
        for(int j=0;j<4;j++){
            printf("\t%d",column_sum[i][j]);
            }
}
```

```
}

void transpose(int matrix[3][3])
{
   int transpose[3][3];
   for(int i=0;i<3;i++){
      for(int j=0;j<3;j++){
         transpose[i][j]=matrix[j][i];
      }
   }
   for(int i=0;i<3;i++){
      printf("\n");
      for(int j=0;j<3;j++){
         printf("\t%d",transpose[i][j]);
         }
}
}

void checkSymmetric(int matrix[3][3])
{
    for(int i=0;i<3;i++){
      for(int j=0;j<3;j++){
         if(matrix[i][j]!=matrix[j][i]){
            printf("\nAsymmetric matrix");
            return;
         }
      }
   }
   printf("\nMatrix is symmetric");
}

void sumOfDiagonals(int matrix[3][3])
{
   int sum=0,a=0;
    for (int i=0;i<3;++i) {

         sum = sum + matrix[i][i];
         a = a + matrix[i][3 - i - 1];
      }


      printf("\nMain diagonal elements sum is = %d\n", sum);
      printf("Off-diagonal elements sum is = %d\n", a);
   }
```

}

Output:


Enter your choice: 3
Enter the number of rows and columns of matrix1: 2 2
Enter the number of columns of matrix2: 2 2
Enter elements of matrix1:
5
7
6
Enter elements of matrix2:
8
13
76
99
Resultant matrix after multiplication:
396 521
512 685
Matrix Operations:
1. Addition
2. Subtraction
3. Multiplication
4. Sum of Diagonals
5. Sum of Rows and Columns
6. Transpose
7. Check Symmetry
0. Exit
Enter your choice: 4
Enter the number of rows and columns of the matrix: 2 2
Enter elements of the matrix:
54
76
0
1
Sum of principal diagonal: 55
Sum of non-principal diagonal: 76
Matrix Operations:
1. Addition
2. Subtraction
3. Multiplication
4. Sum of Diagonals
5. Sum of Rows and Columns
6. Transpose
7. Check Symmetry
0. Exit
Enter your choice: 5
Enter the number of rows and columns of the matrix: 2 2
Enter elements of the matrix:
78
90
0
0
Sum of elements in Row 1: 168
Sum of elements in Row 2: 0
Sum of elements in Column 1: 78
Sum of elements in Column 2: 90

```
Matrix Operations:
1. Addition
2. Subtraction
3. Multiplication
4. Sum of Diagonals
5. Sum of Rows and Columns
6. Transpose
7. Check Symmetry
0. Exit
Enter your choice: 1
Enter the number of rows and columns of the matrices: 2 2
Enter elements of matrix1:
4
7
8
3
Enter elements of matrix2:
12
78
55
23
Resultant matrix after addition:
16      85
63      26

Matrix Operations:
1. Addition
2. Subtraction
3. Multiplication
4. Sum of Diagonals
5. Sum of Rows and Columns
6. Transpose
7. Check Symmetry
0. Exit
Enter your choice: 2
Enter the number of rows and columns of the matrices: 2 2
Enter elements of matrix1:
34
78
99
24
Enter elements of matrix2:
65
55
88
11
Resultant matrix after subtraction:
-31     23
11      13
```

```
Enter your choice: 6
Enter the number of rows and columns of the matrix: 2 2
Enter elements of the matrix:
1
43
56
7
Transpose of the matrix:
1       56
43      7

Matrix Operations:
1. Addition
2. Subtraction
3. Multiplication
4. Sum of Diagonals
5. Sum of Rows and Columns
6. Transpose
7. Check Symmetry
0. Exit
Enter your choice: 7
Enter the number of rows and columns of the matrix: 2 2
Enter elements of the matrix:
15
87
5
3
The matrix is not symmetric.
```

Observation:

1 Write c / c++ program for the following
Pass the matrices as parameters.

(i) Matrix Addition & Subtraction
(ii) Multiplication
(iii) Sum of Row & column.
(iv) Sum of principle diagonal & Non-principle diagonal.
(v) Print transpose of give matrix.
(vi) Check if matrix is Symmetric or not.

```
Void Add ( int A[3][3], int B[3],[3], m) {
    int i=0, j=0;
    int C[m][m];
    for (i=0; i<m; i++) {
        for (j=0; j<m; j++) {
            C[i][j]= A[i][j] + B[i][j];
    }  }
    display (c, m);
}

Void Sub( int A[3][3], int B[3][3], m) {
    int i=0;
    int j=0;
    int C[m][m];
    for (i=0 ; i<m; i++) {
        for (j=0; j<m; j++) {
            C[i][j]= A[i][j] - B[i][j];
    }  }
    display (c, m);
}

Void multiply (int A[3][3], int B[3][3], int m) {
    int i=0, j=0, k=0;
    int C[m][m];
    for (i=0; i<m; i++) {
        for (j=0; j<m; j++) {
            C[i][j] = 0;
    }  }
```

```
for (i=0; i<m; i++){
    for (j=0; j<m; j++){
        for (k=0; k<m; k++){
            C[i][j] += A[i][k] * B[k][j];
        }
    }
}
display (C, m);

void Dsum (int A[m][m], int B[m][m], int m){
    int i=0, j=0;
    int DS=0, NS=0;
    for (i=0; i<m; i++){
        DS = A[i][i] + B[j][j];
    }
    Display (DS);
    for (i=0; i<m; i++){
        NS = A[i][m-1-i] + B[i][m-1-i];
    }
    display (NS);

void transpose (int A[3][3], int m){
    int i, j;
    for (i=0; i<m; i++){
        for (j=0; j<m; j++){
            Printf ("%d", A[j][i]);
        }
    }
}
```

Output

Enter 2 matrix.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A+B = \begin{bmatrix} 2 & 2 & 3 \\ 4 & 6 & 6 \\ 7 & 8 & 10 \end{bmatrix}$$

$$A - B = \begin{bmatrix} 0 & 2 & 3 \\ 4 & 4 & 6 \\ 7 & 8 & 8 \end{bmatrix}$$

$$A = B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Transpose of B is $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

B is Symmetric.

Sum of diagonal Elements of A = 15

Sum of non-diagonal Elements of A = 15

Sum of row of $A = \begin{bmatrix} 1 & 2 & 3 & 6 \\ 4 & 5 & 6 & 15 \\ 7 & 8 & 9 & 24 \end{bmatrix}$

Sum of column of $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 12 & 15 & 18 \end{bmatrix}$

21/6/23