1) Write a C or C++ program to do the following pass the matrices as the parameters in all progm.
a) Matrices Addition
b) -11- Sub
c) -11- Multip
d) Sum of diagonal / non diagonal
e) Sum of rows & columns
d) Transpose of matrices
e) check if the given matrices is Symmetric or not

```
#include <stdio.h>
#define MAX_SIZE 100

void inputMatrix(int matrix[MAX_SIZE][MAX_SIZE],
int rows, int cols) {
    printf("Enter the elements of the matrix:\n");
    for(i=0; i<rows; i++){
        for(j=0; j<cols; j++){
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMatrix(int matrix[MAX_SIZE][MAX_SIZE],
int rows, int cols){
    printf("Matrix:\n");
    for(i=0; i<rows; i++){
        for(j=0; j<cols; j++){
            printf("%d", matrix[i][j]);
        }
        printf("\n");
    }
}

void addMatrics(int matrix1[MAX_SIZE][MAX_SIZE],
int matrix2[MAX_SIZE][MAX_SIZE], int rows, int cols){
```

```
    int result[MAX_SIZE][MAX_SIZE];
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++)
            result[i][j] = matrix1[i][j] + matrix2[i][j];
    }
    printf("Addition of matrices:\n");
    printMatrix(result, rows, cols);
}

void subtractMatrices(int matrix1[MAX_SIZE][MAX_SIZE],
int matrix2[MAX_SIZE][MAX_SIZE], int rows,
int cols) {
    int result[MAX_SIZE][MAX_SIZE];
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++)
            result[i][j] = matrix1[i][j] - matrix2[i][j];
    }
    printf("Subtraction of matrices:\n");
    printMatrix(result, rows, cols);
}

void multiplyMatrices(int matrix1[MAX_SIZE][MAX_SIZE],
int rows1, int cols1, int matrix2[MAX_SIZE][MAX_SIZE],
int rows2, int cols2) {
    if (cols1 != rows2) {
        printf("Error: Matrices cannot be multiplied\n");
        return;
    }
    int result[MAX_SIZE][MAX_SIZE];
    for (i=0; i<rows1; i++) {
        for (j=0; j<cols2; j++) {
```

```
            result[i][j] = 0;
            for (k=0; k<cols1; k++){
                result[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
    printf("Multiplication of matrices:\n");
    printMatrix(result, rows1, cols2);

Void SumDigonal NonDiagonal(int matrix[MAX_SIZE][MAX_SIZE]
int rows, int cols, char choice){
    int sum = 0;
    if (choice == 'D' || choice == 'd'){
        for (i=0; i<rows; i++){
            sum += matrix[i][i];
        }
        printf("Sum of diagonal elements: %d\n", sum);
    } else if (choice == 'N' || choice == 'n'){
        for (i=0; i<rows; i++){
            for (j=0; j<cols; j++){
                if (i!=j){
                    sum += matrix[i][j];
                }
            }
        }
        printf("Sum of non-diagonal elements: %d\n", sum);
    } else{
        printf("Invalid choice. Please enter D or N.\n");
    }
}

Void SumRowsColums(int matrix[MAX_SIZE][MAX
int rows, int cols){
    int rowSum[MAX_SIZE] = {0};
```

```
int colSum [MAX_SIZE] = {0};
for (i=0; i<rows; i++) {
    for (j=0; j<cols; j++) {
        rowSum [i] += matrix[i][j];
        colSum [j] += matrix [i][j];
    }
}
printf ("Sum of non-diagonal elements : %d\n", sum);
} else {
    printf ("Invalid Choice. Please enter D or N. \n");
}
}
Void
    printf ("Sum of rows : \n");
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++) printf ("Row %d : %d\n
            i+1, rowSum[i]);
        rowSum [i] += matrix[i][j]; }
        colSum [j] += matrix [i][j];
    printf ("Sum of columns : \n");
    for (j=0; j<cols; j++) {
        printf ("column %d : %d \n", j+1, colSum[j]);
}
}
Void transposeMatrix (int matrix [MAX_SIZE][MAX_SIZE]
int rows, int cols) {
    int transposed [MAX_SIZE] [MAX_SIZE];
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++) {
            transposed [j] [i] = matrix [i] [j];
        }
}
printf ("Transposed matrix : \n");
```

```
    printMatrix(transposed, cols, rows);
}

int isSymmetricMatrix(int matrix[MAX_SIZE][MAX_SIZE],
    int rows, int cols) {
    if (rows != cols) {
        return 0;
    }
    for (i=0; i<rows; i++) {
        for (j=0; j<cols; j++) {
            if (matrix[i][j] != matrix[j][i]) {
                return 0;
            }
        }
    }
    return 1;
}

int main() {
    int choice;
    printf("Matrix Operations:\n");
    printf("1. Addition\n");
    printf("2. Subtraction\n");
    printf("3. Multiplication\n");
    printf("4. Sum of diagonal or non-diagonal elements\n");
    printf("5. Sum of rows and columns\n");
    printf("6. Transpose of matrix\n");
    printf("7. Check if matrix is Symmetric\n");
    printf("Enter your choice:");
    scanf("%d", &choice);

    int rows, cols;
    printf("Enter the no of rows in matrices:");
    scanf("%d", &rows);
```

```
printf("Enter the no of columns in the matrices: ");
scanf("%d", &cols);
int matrix1 [MAX_SIZE][MAX_SIZE];
int matrix2 [MAX_SIZE][MAX_SIZE]
switch(choice){
case 1: printf("Matrix 1: \n");
        inputMatrix(matrix1, rows, cols);
        printf("Matrix 2: \n");
        inputMatrix(matrix2, rows, cols);
        addMatrices(matrix1, matrix2, rows, cols);
        break;
case 2: printf("Matrix 1: \n");
        inputMatrix(matrix1, rows, cols);
        printf("Matrix 2: \n");
        inputMatrix(matrix2, rows, cols);
        SubtroutMatrices(matrix1, matrix2, rows, cols)
        break;
case 3: printf("Matrix 1: \n");
        inputMatrix(matrix1, rows, cols);
        printf("Matrix 2: \n");
        inputMatrix(matrix2, cols, rows);
        multiplyMatrices(matrix1, rows, cols, matrix2
                         cols, rows);
        break;
case 4: printf("Matrix: \n");
        inputMatrix(matrix1, rows, cols);
        printf("Enter 'D' for Diagonal elements
          or 'N' for non-diagonal elements: ");
        char sumChoice;
        scanf("%c", &sumChoice);
        SumDiagonalNonDiagonal(matrix1, rows
                               cols, sumChoice);
```

```
        break;
case 5: printf("Matrix : \n");
        inputMatrix(matrix1, rows, cols);
        transposeMatrix(matrix1, rows, cols);
        break;
case 7: printf("Matrix: \n");
        inputMatrix(matrix1, rows, cols);
        if(isSymmetricMatrix(matrix1, rows, cols)){
          printf("Symmetric \n");
        }else{
          printf("Not Symmetric \n");

        }
        break;
default: printf("Invalid choice \n");
         break;

    }
    return 0;

}
```

Output: Enter 2 matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 2 & 2 & 3 \\ 4 & 6 & 6 \\ 7 & 8 & 10 \end{bmatrix} \qquad A - B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

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

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 colum of A:

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

N
21/6/23

OUTPUT:

```
Matrix Operations:
1. Addition
2. Subtraction
3. Multiplication
4. Sum of diagonal or non-diagonal elements
5. Sum of rows and columns
6. Transpose of matrix
7. Check if matrix is symmetric
Enter your choice: 1
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix 1:
Enter the elements of the matrix:
 1 3 4
 2 6 8
 5 7 9
Matrix 2:
Enter the elements of the matrix:
9 8 7
6 9 6
3 8 2
Addition of matrices:
Matrix:
10 11 11
8 15 14
8 15 11
```

```
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix 1:
Enter the elements of the matrix:
1 2 3
4 6 8
7 9 0
Matrix 2:
Enter the elements of the matrix:
1 3 6
6 7 9
2 4 7
Subtraction of matrices:
Matrix:
0 -1 -3
-2 -1 -1
5 5 -7
```

```
Enter your choice: 3
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix 1:
Enter the elements of the matrix:
1 2 3
2 4 5
6 7 8
Matrix 2:
Enter the elements of the matrix:
2 4 7
3 7 8
1 2 6
Multiplication of matrices:
Matrix:
11 24 41
21 46 76
41 89 146
```

```
Enter your choice: 4
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix:
Enter the elements of the matrix:
1 2 3
3 4 6
2 3 6
Enter 'D' for diagonal elements or 'N' for non-diagonal elements: D
Sum of diagonal elements: 11
```

```
Enter your choice: 5
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix:
Enter the elements of the matrix:
1 2 3
4 6 7
3 7 2
Sum of rows:
Row 1: 6
Row 2: 17
Row 3: 12
Sum of columns:
Column 1: 8
Column 2: 15
Column 3: 12
```

```
Enter your choice: 6
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix:
Enter the elements of the matrix:
1 4 5
2 3 5
6 5 3
Transposed matrix:
Matrix:
1 2 6
4 3 5
5 5 3
```

```
Enter your choice: 7
Enter the number of rows in the matrices: 3
Enter the number of columns in the matrices: 3
Matrix:
Enter the elements of the matrix:
1 4 5
2 3 6
6 7 8
The matrix is not symmetric.
```