14/6/23 Exp-1.

Q1.] Write C or C++ program to do the following

i] Pass the matrices as parameters

ii] Addition / substraction

iii] Sum of row & columns

iv] Multiplication

v] Sum of principle / non-principle diagonal elements

vi] Print the transpose of a given matrix

vii] Symmetric or not

Soln:-

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

```
printf("\n");
  }
}
void addMatrices(int matrix1[MAX-SIZE][MAX-SIZE], int
                 matrix2[MAX-SIZE][MAX-SIZE], int rows, int cols){

  int result [MAX-SIZE][MAX-SIZE];
  for(int i =0; i<rows; i++){
   for(int j=0; j<cols; j++){
    result[i][j] = matrix1[i][j] +matrix2[i][j];
   }
  }
  printf("\n");
 }
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 (int i=0; i<rows; i++) {
  for(int j=0; j<cols; j++){
   result [i][j] = matrix1[i][j] - matrix2[i][j];
  }
 }
 printf("\n");
}
void multiplyMatrices(int matrix1[MAX-SIZE][MAX-SIZE],
  int rows1, int cols1, matrix2 [MAX-SIZE][MAX-SIZE],
  int rows2, int cols2) {
  if (cols1 != rows2) {
   printf("Error");
   return;
  }
 int result [MAX-SIZE][MAX-SIZE];
 for (int i=0; i<rows1; i++){
  for (int j=0; j<cols2; j++) {
```

```
result [i][j] = 0;
  for (int  k = 0; k < cols1 ; k++) {
    result [i][j] + = matrix1[i][k] * matrix2[k][j];
  }
 }
}
printf ("\n");
}
void sumDiagonalNonDiagonal (int matrix[MAX-SIZE][MAX-SIZE]
                int rows, int choice) {
  int sum = 0;
 if (choice == 'D' || choice = = 'd') {
   for (int i = 0; i < rows; i++) {
     sum += matrix [i][i];
   }
  printf("sum of diagonal elements: %d\n", sum);
 }
 else if (choice == 'N' || choice = = 'n') {
    for (int i = 0; i < rows; i++) {
      sum += matrix [i][j];
    }
  printf ("sum of diagonal elements: %d\n", sum);
 }
 else if (choice == 'N' || choice = = 'n') {
   for(int i = 0; i < rows; i ++) {
     ~~for (int j = 0; j < cols; j++) {~~
       if (i != j);
       sum += matrix [i][j];
     }
   }
  }
 printf ("non-diagonal elements : %d\n", sum);
```

```
}else{
  printf("Invalid choice, please enter D or N\n");
 }
}
void transposeMatrix(int matrix[MAX-SIZE][MAX-SIZE],
  int rows, int cols){
  int transposed[MAX-SIZE][MAX-SIZE];
  for(int i=0; i<rows; i++){
   for(int j=0; j<cols; j++){
     transposed[j][i] = matrix[i][j];
   }
  }
  printf("\n");
 }
int SymmetrixMatrix(int matrix[MAX-SIZE][MAX-SIZE],
                    int rows, int cols){
   if(rows != cols){
     return 0;
   }
  for (int i=0; i<rows; i++){
     for(int j=0; j <cols; j++){
     if(matrix[i][j] != matrix[j][i]){
       return 0;
     }
    }
  }
 }
return 1;
}
```

```
int main(){
  int choice;
 printf("1.Addition  2.Substraction  3.Multiplaction
       4. Sum of diagonal elements  ~~8. Sum of rows &~~
       5. Transpose matrix  6. IS Symmetrix");

 printf("Enter your choice");
 scanf("%d", &choice);
 int rows, cols;
 printf("enter the number of rows");
 scanf("%d", & rows);
 printf("enter the number of columns");
 scanf("%d", &cols);
 int matrix1[MAX-SIZE][MAX-SIZE];
 int matrix2[MAX-SIZE][MAX-SIZE];
switch(choice){
  case 1: inputMatrix(matrix1, rows, cols);
          inputMatrix(matrix2, rows, cols);
          addMatrices(matrix1, matrix2, rows, c
          break;
  case 2: inputMatrix(matrix1, rows, cols);
          inputMatrix(matrix2, rows, cols);
          substractMatrices (matrix1, matrix2, rows
          break;
 case 3: inputMatrix(matrix1, rows, cols);
          inputmatrix (matrix2, rows, cols);
          multiplyMatrices (matrix1, rows, cols,
             matrix2, cols, rows);
        break;
case 4: inputMatrix(matrix1, rows, cols);
      printf("Enter 'D' for diagonal elements or 'N'
      for non-diagonals elements: ");
      char sumChoice;
      scanf(" %c ", &sumchoice);
      sumDiagonalNonDiagonal(matrix1, rows, cols,
            sum choice);
    break;
se 5: inputMatrix(matrix1, rows, cols);
      sumRowsColumns(matrix1, rows, cols);
      break;
se 6: inputMatrix(matrix1, cols, rows);
      transposeMatrix(matrix1, rows, cols);
      break;
e 7: inputMatrix(matrix1, rows, cols);
     if (isSymmetrixMatrix(matrix1, rows, cols)){
      printf("symmetrix");
    }else{
      printf("non-symmetrix");
    }
    break;
efault : printf("Invalid choice");
         break;
  }
return 0;
```

✓ [illegible] 17-6-23