Bass the matrices as the parameter in all progre b) 17117 Sub 1 1 310 France 1 plin Money d) Sum of diagonal I non diagonal c) Sum of some & columns e) check if the given matrices is symmetric #include < 8 td.o. w # define MAX_ 8 I 26 100" 1 3 18 XAM void infort matrix (unt matrix [MAX SIZE] [MAX - SIZE] unt sous unt colo) & for (i = 0 : i < round: it t) & for (: 0 : i < rows: i+) { for (i :0: i cols; j++) f. De Buss Scanf (" 1.d", & matrix (;][;]): 3 Void fried Matrix (int matrix [MAX_SIZE] [MAX_SIZE] int bows sintercols) of the land and all the land frint (matrice In): 1 alos the I sevore for (i=0; i < 9,0008; i++) } for (j=0: j < cole; j++) for on = | 2 | 2) friends (" of d", matrix cidly) frintf (" Im"):
LE XAMI LOSEE XAMI Share Noid add Matrices (int matrix 1 LMAX SIZE] [MAX SIZE], but matrix & [MAX_SIZE] [MAX_SIZE], int rows, introles

```
int result [MAX SIZE] [MAX SIZE]
reportor (1=0) 12 Roles; 111) i
      for (j. 0; j = cole; j++) $
        result [i][j] = matrix ( [i][j] + matrix ? [i][j]
                emol I non diagonal
   Brinds ("Addition of matrices; (n);
    for int matrix ( nexult from cols)
Void Subtract Motorices Coint matrix 1 [MAX-SIZE] [MAX.SIZ
                                                    No d
 int matrix EMAX-SIZE] [MAX-SIZE], int source
     int result [ MAX. SIZE] [MAX_ SIZE]
 int co (8, 2) of
 · for (i=0; i < 8000 : 1++) }
       for (j=0; j&coto salj++)to saturd ) le
          result [i] [j] motrix [i] [j] - motrix e [i] [j]
        Scanf (" of d" , I matris (: Ili I):
     forint ("Subtraction of matrices; \n");
     for I matrix ( sesult, 2000, cols):
  " d fort motors ( and motors [ max size] [ max size] &
      multiply matrices ( int matrix) [MAX SIZE] [MAX SIZE
      rows 1, int cols 1, int matrix & [ mAX - 8 I Z & ] [ max ]
      1 rows 2, int cols 2) f. sound > 1 = 0 -1) -07
      Cools! = nows &) {++1 + 3/00 > 1 +0-10 +03
     forint ( Green: matrices cannot be multiplied !
      selver !
         result [MAX-8] ZEJ [MAX-8] ZEJ;
for (i=0; ; < Rows 1; i++) &
       for (j=0; j=color j+t) f
                 LOSIZ KAM ] LOSTS THAT SXINTON
```

```
result listif = 0;
       for (K=0; K<cols1; K++) $
         result [i][j] + = matrix | [i][k] * matrix e[k][j] =
  frint ( Mulliplication of matrices: In ):
frint Matrin ( assult, nower, cols a);
 Void Sum Digonal Non Diagonal Cut matrix [MAX_SIZE][MAX_SIZE]
 just nows, int cols- chou choice) of
   ent sun : 0;
   if Cchoice == '0' 11 choice == 'd')f
     for (: 0 ; i < 90ws; i++) f
         sun += matrix (i)[i];
      Board f (" Sum of diagonal elements: "Id In sun).
    gelee if Cchoice == 'N' Il choice == 'n' I'
     for ( = 0: 1 + nows; it ) &
         for (j :0; j = cols; j++) {
          if (: ! = j) }
               Sum 1 = matrix Cilly 7:
  Chipme for a property of the of
   frinds (" Sum of non-diagonal elements: " ! d \n', sum)
  g else &
     for at (" Invalid choice. Please enter Dor N. In
Void Sum Rows Columns Cint matrix [ max SIZE] [MAX
ant rows, int cols ) &
    unt now Sun [MAX_SIZE] = {03 ;
```

```
int colsum [MAX_SIZE] = 809;
     for Ci=0 1 1 2 hours 5 i+1) {
        for (jos je cols ; j++) {
          row Sum Lidt: matrix [idlid]
         colsum bj ] + = matrix [:][j];
  frints ( Sum of non-diagonal elevents - 1-dlm sum)
    for If ( Invalid Choice Please enter Dov N. In)
   frintf (" Sun of rows: In"):
    for ( := 0 , ; < hows ; ; ++ ) {
      for Cj=0; j zcols; j++) pr. of ( Row 1.d: 1/d)
        now Sum [i] + = matrix Lifty J; &
     Brint ("Sum of colemns; \m");
for (j=0; j < cols; j++)s
                      1/d: 1/d \m", j+1, colsum[j]);
Void transpose motor ( int motor x [MAX-8IZE][MAXIN
    sous . int cols ) {
   int bourfosed EMAX-SIZE] [MAX-SIZE];
    for ( := 0; i x now ; i+ + ) }
       for G: 0; j < cols; j++) }
transposed GjJ [i] = matrix [i][j]:
  for tf ( Transposed matrix: In'):
```

```
forint matrix (transposed, cols, rows);
int is Symmetric Matrix Circle matrix (MAX. SIZE] [MAY. SIZE]
 int rows, int cols ) +
  if ( some ! = cols) }
                  m/ - Lanton - ) Itag
     return o
  for ( i=0; i < nows; i+1) f
     for (j. o; j < cols; j+1) f
      if (matrix [i][j]! = matrix [j][i]) }
          return 0:
  net un 1?
     int choice:
   Brindf (" Matrix Operations: \n");
     fority ("1. Addition \n");
     Britof (" 2. Subtraction la ):
     Brists (" 3. Multiplication In"):
   for: alf ("4. Sun of diagonal or non diagonal elevals
     for the (-5. Sem of room and columns in"):
      for tf ("6. Transfose of matrix (");
     Bristf (" 7. Check if motion is symmetric In");
    for If ( Enter your choice: "):
      Scoul ( 1. d " , & choice):
   Scanf (" toler the no of your in motrices;");
     int gows, cols:
```

Scarf (" fater the no of columns [max-size] [max-size] [max-size]; [MAX_SIZE] [MAX_SIZE Switch Coho ce) of case 1: Paint (- Matrix 1: \n'): input Matrix (mostrix 1, rows, cols); frit (Matrix 2: (m)); input Matrix (matrix &, rows, cols); addmatrices (matrix), matrix &, rows, colo) break ; case &: frint (Matrix 1: \n'); input matrix (matrix), rows, cols); forint (matrix 9; 1 m); input matrix (matrix R, vows, cols); Subtrout matrices (matrix 1, matrix 2, rows, cols) break ; l'a/ avoitorego cases: frint ("matri Adistri) infaut Mate: x (mate & to vo, we co le) for the (Matrix &:) "]; infort motor (materix 2 cols, rows); multiply Motrices (matrix 1, rows, cols, met break; case 4: for not (Matin : 12"); input Madrix (matrix 1, rows, cots); frent (Enter D' for D'agonal element ov N' for non-diagonal elements! char sum Choice sconfl 1.0, 2 sumchoice); Sum Diagonal Non Diagonal (matrix 1, row cols, sum Choice)

```
case 5: printf ("Matrix: In");
            infort Mat x (mat x 1, rows. cols );
            transpose Matrix (matrix), rows, cols);
            break ;
  case 7: printf ("matrix: (m);
             input Matrix (matrix 1, rows, cols);
            of (is Symnetic Matrix (modrix 1, rows, cols)) f
            fornt f ( Synnetric 1 m);
              Sprintf (" Not Symmetric \");
         default: Brint (Invalid choice (m');
      netwn 0;
 4
Output: Enter & matrix
    A = \begin{bmatrix} 1 & R & 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 & R & 3 \\ 4 & 6 & 6 \\ 7 & 8 & 10 \end{bmatrix}
A + B : \begin{bmatrix} 1 & R & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}
    A-B: [0 & 3]
4 4 6
7 8 8
     Transpose of B is [100]
                 B is Symmetric
```

Sum of diagonal elements of A = 15
Sum of non diagonal elements of A = 15
Sum of now of A = 1 & 3 6
4 5 6 15 Sum of column of A

OUTPUT:

```
Matrix Operations:
1. Addition
Subtraction
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:
987
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:
123
4 6 8
790
Matrix 2:
Enter the elements of the matrix:
1 3 6
679
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
678
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.
```