classmate
Date 14th Jun.
Page

Write a c or c++ program to do the following pass the matrices as parameters as all this programs.

(1) Matrices addition or substraction and multiplication.
(2) Sum principle diagonal & non-principle diagonal.
(3) Sum of rows & columns.
(4) Print the transpose of matrices.
(5) Check if the given matrix is symmetric or not. (sq. matrix)

→
```
#include <stdio.h>
#define max_size 15
void printMatrix (int Matrix [][Max_size], int size)
{
for (i=0; i< size; i++)
 {
for (j=0; j< size ; j++)
  {
  Printf ("%d", matrix[i][j]);
  }
 }
}
void add (int matrixA [][max_size] , int matrixB [][max_size]
, int result [][max_size], int size)
{
for (i=0; i< size; i++)
 {
for (j=0; j<size ; j++) {
  result[i][j] = matrix A[i][j] + matrixB[i][j];
  }
 }
}
void sub (int matrix A [][max_size] , int matrixB [] [max_size]
, int result [ ] [max_size ], int size)
{
 for (i=0; i< size; i++) {
  for (j=0; j< size; j++) {
```

```
    result[i][j] = matrix A[i][j] - matrix B[i][j];
}
}
}
void multiply(int matrix A[][max_size], int matrix B[]
[max_size], int result[][max_size], int size){
  for(int i=0; i<size; i++)
{
for(int j=0; j<size; j++)
{
  result[i][j]=0;
for(int k=0; k<size; k++)
{
result[i][j] += matrix A[i][k] * matrix B[k][j];
}
}
}
}
int sumprincipledigonal(int matrix[][max_size], int size)
{
int sum=0;
for(int i=0; i<size; i++)
{
sum += matrix[i][i];
}
return sum;
}
int sumnonprincipledigonal(int matrix[][max_size], int size)
{
int sum=0;
for(int i=0; i<size; i++)
{
sum += matrix[i][size-i-1];
}
return sum;
}
```

```
void sumRows (int matrix [] [MAX_SIZE], int size) {
for (int i=0; i<size; i++)
 int sum=0;
for (int j=0; j<size; j++)
 {
 sum+ = matrix [i][j];
 }
Printf ("sum of elements of row %d: %d \n", i+1, sum);
}
}
void sumcolumns (int matrix [] [MAX_SIZE], int size) {
for (int i=0; i<size; i++)
{
 int sum=0;
for (int j=0; j<size; j++)
{
sum += matrix [j][i];
}
Printf ("sum of elements in column %d: %d \n", i+1, sum);
}
}
void transposematrix (int matrix [] [MAX_SIZE], int size)
{
int temp;
for (int i=0; i<size; i++)
{
int sum=0;  for (j=i+1; j<size; j++) {
for (int j=0; j<size; j++) temp= matrix [i] [j];
{
sum+ = matrix [j] [i];  matrix [i] [j] = matrix [j] [i];
}  matrix [j] [i] = temp;
}
print
}
}
```

```
int isSymmetric(int matrix[][MAX_SIZE], int size)
{
for(int i=0; i<size; i++){
for(int j=i+1; j<size; j++){
if(matrix[i][j] != matrix[j][i]){
return 0;
}
}
}
return 1;
}
int main()
{
int matrixA[MAX_SIZE][MAX_SIZE];
int matrixB[MAX_SIZE][MAX_SIZE];
int result[MAX_SIZE][MAX_SIZE];
int size;
Printf("Enter size of matrices");
Scanf("%d", &size);
Printf("Enter elements of matrix A\n");
for(int i=0; i<size; i++)
{
for(int j=0; j<size; j++)
Scanf("%d", &matrixA[i][j]);
}
}
Printf("Enter elements of matrix B\n");
for(int i=0; i<size; i++)
{
for(int j=0; j<size; j++)
Scanf("%d", &matrixB[i][j]);
}
}
```

```
int choice;
int flag=0;
while (!flag)
{
Print ("\n matrix operation\n");
Printf("1. Add");
Printf("2. sub");
   " ("3. Multiply");
   " ("4. sum of diagonal");
   " ("5. Sum of non-diagonal");
   " ("6. Sum of rows");
   " ("7. sum of column");
   " ("8. Transpose\n");
   " ("9. Symmetry\n");
    Printf("0. Exit\n");
Printf("Enter choice");
Scanf("%d", &choice);
Switch (choice){
case 0:
  flag=1;
  break;
case 1:
  Printf("Matrix A:\n");
  Print Matrix (matrix A, size);
  printf("Matrix B:\n");
  Print Matrix (matrix B, size);
  Printf("\n Adding matrices:\n");
  add (matrix A, matrix B, result, size);
  print Matrix (result, size);
  break;
case 2:
  Printf("Matrix A:\n");
  Print Matrix (matrix A, size);
  Printf("Matrix B:\n");
```

```
printMatrix(matrixB, size);
Printf("\n substracting matrices: \n");
Substract(matrixA, matrixB, result, size);
printMatrix(result, size);
break;
case 3:
Printf("Matrix A: \n");
Print Matrix(matrix A, size);
Printf("Matrix B: \n");
Print Matrix(matrix B, size);
Printf("\n Multiplying matrices: \n");
multiply(matrix A, matrix B, result, size);
print Matrix(result, size);
break;
case 4:
Printf("Matrix: \n");
Print Matrix(matrix A, size);
Print f("\n sum of Principle diagonal: %d \n", sum
diagonal(matrix A, size));
break;
case 5:
Printf("Matrix: \n");
PrintMatrix(matrix A, size);
Print f("\n sum of non-principal diagonal: %d \n",
sum non-diagonal(matrix A, size));
break;
case 6:
print f("Matrix: \n");
Print Matrix (matrix A, size);
Print f("\n sum of rows: \n");
Sum Rows(matrix A, size);
break;
case 7:
Printf("Matrix: \n");
```

```
printMatrix(matrixA, size);
Printf("\n sum of columns:\n");
SumColumns(matrix A, size);
break;
case 8:
Printf("Matrix:\n");
PrintMatrix(MatrixA, size);
Printf("\n Transpose of Matrix:\n");
transposeMatrix(matrix A, size);
printMatrix(matrixA, size);
break;
case 9:
Printf("Matrix:\n");
printmatrix(matrixA, size);
if(isSymmetric(matrixA, size)){
Printf("\n The matrix is symmetric\n");
}
else
{
Printf("\n The matrix is not symmetric\n");
}
break;
default;
Printf("Invalid choice:\n");
break;
}
}
return 0;
}.
```

Output.

Matrix operations

1. Add
2. Substract.
3. Multiply
4. Sum of diagonal.
5. Sum of rows & columns
6. Transpose
7. Check symmetry
8. Exit.

Enter your choice : 1.

Enter no. of rows & columns 2 2

Enter elements of matrix 1

4 7 8 3

Enter elements of matrix 2

12 98 55 23

Resultant matrix

16 85

63 26

Menu.

Enter choice: 2

Enter elements of matrix 1

34 78 99 29

Enter elements of matrix 2

65 55 88 11

Resultant matrix

-31 23

11 13

Menue

Enter your choice: 3

Enter elements of matrix 1

5 7 6 9

Enter element of matrix 2

8 13 76 99

Resultant matrix

396 521

512 685

Enter choice 4

Enter elements of matrix 1

59 76 0 1

sum of principle diagonal: 55

Sum of non-principle diagonal: 76

Menue

Enter choice: 5

Enter elements of matrix

78 90 0 0

Result → sum of rows: 168, 0

sum of column: 78, 90

Menue

Enter your choice: 6

Enter elements of matrix

1 43 56 7

Tranpose is

1 56

43 7

Menue

Enter choice 7

Enter elements of matrix

15 87 5 3

The matrix isn't symmetric.

```
Enter the size of the square matrices: 2 2
Enter the elements of Matrix 1:
4 7 8 3
Enter the elements of Matrix 2:
12 98 55 23

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: Invalid choice.

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 1
Matrix 1:
2 4
7 8

Matrix 2:
3 12
98 55
```

```
Adding matrices:
5 18
106 63

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 2
Matrix 1:
2 4
7 8

Matrix 2:
3 12
98 55

Subtracting matrices:
-1 -8
-91 -47

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
```

```
Enter your choice: 4
Matrix:
2 4
7 8

Sum of Principal Diagonal: 10

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 5
Matrix:
2 4
7 8

Sum of Non-Principal Diagonal: 11

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 6
Matrix:
2 4
```

```
Enter your choice: 3
Matrix 1:
2 4
7 8

Matrix 2:
3 12
98 55

Multiplying matrices:
398 244
805 524

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 4
Matrix:
2 4
7 8

Sum of Principal Diagonal: 10

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
```

```
Enter your choice: 4
Matrix:
2 4
7 8

Sum of Principal Diagonal: 10

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 5
Matrix:
2 4
7 8

Sum of Non-Principal Diagonal: 11

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 6
Matrix:
2 4
```

```
Enter your choice: 6
Matrix:
2 4
7 8

Sum of Rows:
Sum of elements in row 1: 6
Sum of elements in row 2: 15

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 7
Matrix:
2 4
7 8

Sum of Columns:
Sum of elements in column 1: 9
Sum of elements in column 2: 12

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
```

```
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 8
Matrix:
2 4
7 8

Transpose of Matrix:
2 7
4 8

Matrix Operations
1. Add Matrices
2. Subtract Matrices
3. Multiply Matrices
4. Sum of Principal Diagonal
5. Sum of Non-Principal Diagonal
6. Sum of Rows
7. Sum of Columns
8. Print Transpose
9. Check Symmetry
0. Exit
Enter your choice: 9
Matrix:
2 7
4 8

The matrix is not symmetric.
```