Assignment - 16 A Job Ready Bootcamp in C++, DSA and IOT MySirG

Multi-Dimensional Array in C Language

1. Write a program to calculate the sum of two matrices each of order 3x3.

#include <stdio.h>

int main()

{

int a[3][3], b[3][3], sumMatrix[3][3];

printf("Enter Element in First 3\*3 Matrix: \n");

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

scanf("%d", &a[i][j]);

printf("Enter Element in Second 3\*3 Matrix: \n");

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

scanf("%d", &b[i][j]);

// Sum of Matrix

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

sumMatrix[i][j] = a[i][j] + b[i][j];

// print First Matrix

printf("First Matrix:\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

printf("%d ", a[i][j]);

printf("\n");

}

// print Second Matrix

printf("Second Matrix:\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

printf("%d ", b[i][j]);

printf("\n");

}

// print Sum Of Upper Two Matrix

printf("Sum of Twoa Matrix:\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

printf("%d ", sumMatrix[i][j]);

printf("\n");

}

return 0;

}

2. Write a program to calculate the product of two matrices each of order 3x3.

#include <stdio.h>

int main()

{

int a[3][3], b[3][3], productMatrix[3][3] = {0};

printf("Enter Element in First 3\*3 Matrix: \n");

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

scanf("%d", &a[i][j]);

printf("Enter Element in Second 3\*3 Matrix: \n");

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

scanf("%d", &b[i][j]);

// Product of Matrix

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

for (int k = 0; k < 3; k++)

{

productMatrix[i][j] += a[i][k] \* b[k][j];

}

}

}

// print First Matrix

printf("First Matrix:\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

printf("%d ", a[i][j]);

printf("\n");

}

// print Second Matrix

printf("Second Matrix:\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

printf("%d ", b[i][j]);

printf("\n");

}

// print product Of Upper Two Matrix

printf("Sum of Twoa Matrix:\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

printf("%d ", productMatrix[i][j]);

printf("\n");

}

return 0;

}

3. Write a program in C to find the transpose of a given matrix.

#include <stdio.h>

int main()

{

int row, col;

printf("Enter Row and Column of Matrix: ");

scanf("%d%d", &row, &col);

int matrix[row][col];

printf("Enter Element in %d\*%d Matrix: \n", row, col);

for (int i = 0; i < row; i++)

for (int j = 0; j < col; j++)

scanf("%d", &matrix[i][j]);

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

// find Transpose, here we only find Transpose not assign according to question

printf("Transpose of Matrix: \n");

for (int i = 0; i < col; i++)

{

for (int j = 0; j < row; j++)

printf("%d ", matrix[j][i]);

printf("\n");

}

return 0;

}

4. Write a program in C to find the sum of right diagonals of a matrix.

#include <stdio.h>

int main()

{

int row, rightDiagoSum=0;

printf("Enter Row of Matrix: ");

scanf("%d", &row);

int matrix[row][row];

printf("Enter Element in %d\*%d Matrix: \n", row, row);

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

{

scanf("%d", &matrix[i][j]);

if (i == j)

rightDiagoSum += matrix[i][j];

}

}

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

// right Diagonalls sum

printf("Right Diagonals of Matrix Sum is: %d\n", rightDiagoSum);

return 0;

}

5. Write a program in C to find the sum of left diagonals of a matrix.

#include <stdio.h>

int main()

{

int row, leftDiagoSum = 0;

printf("Enter Row of Matrix: ");

scanf("%d", &row);

int matrix[row][row];

printf("Enter Element in %d\*%d Matrix: \n", row, row);

for (int i = 0; i < row; i++)

for (int j = 0; j < row; j++)

scanf("%d", &matrix[i][j]);

// find left Diagonal Sum

for (int i = 0, j = row - 1 - i; i < row; i++, j--)

leftDiagoSum += matrix[i][j];

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

// right Diagonalls sum

printf("left Diagonals of Matrix Sum is: %d\n", leftDiagoSum);

return 0;

}

6. Write a program in C to find the sum of rows and columns of a Matrix.

#include <stdio.h>

int main()

{

int row, col, sumOfMatrix = 0;

printf("Enter Row and Column of Matrix: ");

scanf("%d%d", &row, &col);

int matrix[row][col];

printf("Enter Element in %d\*%d Matrix: \n", row, col);

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

scanf("%d", &matrix[i][j]);

sumOfMatrix += matrix[i][j];

}

}

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

// matrix sum

printf("Matrix All Element Sum is: %d\n", sumOfMatrix);

return 0;

}

7. Write a program in C to print or display the lower triangular of a given matrix.

#include <stdio.h>

int main()

{

int row;

printf("Enter Row of Matrix: ");

scanf("%d", &row);

int matrix[row][row];

printf("Enter Element in %d\*%d Matrix: \n", row, row);

for (int i = 0; i < row; i++)

for (int j = 0; j < row; j++)

scanf("%d", &matrix[i][j]);

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

// print Lower triangular Matrix

printf("lower triangular Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j <= i; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

return 0;

}

8. Write a program in C to print or display an upper triangular matrix.

#include <stdio.h>

int main()

{

int row;

printf("Enter Row of Matrix: ");

scanf("%d", &row);

int matrix[row][row];

printf("Enter Element in %d\*%d Matrix: \n", row, row);

for (int i = 0; i < row; i++)

for (int j = 0; j < row; j++)

scanf("%d", &matrix[i][j]);

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

printf("%d ", matrix[i][j]);

printf("\n");

}

// print uppper triangular Matrix

printf("lower triangular Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < row; j++)

{

if (j >= row - 1 - i)

printf("%d ", matrix[i][j]);

else

printf(" ");

}

printf("\n");

}

return 0;

}

9. Write a program in C to accept a matrix and determine whether it is a sparse matrix.

// the sparse matrix can be defined as the matrix that has a greater number of zero elements than the non-zero elements.

#include <stdio.h>

int main()

{

int row, col, countZeros = 0;

printf("Enter Row and column of Matrix: ");

scanf("%d%d", &row, &col);

int matrix[row][col];

printf("Enter Element in %d\*%d Matrix: \n", row, col);

for (int i = 0; i < row; i++)

for (int j = 0; j < col; j++)

scanf("%d", &matrix[i][j]);

// print Matrix

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

printf("%d ", matrix[i][j]);

if (matrix[i][j] == 0)

countZeros++;

}

printf("\n");

}

// Check Sparsh Matrix

if (countZeros > ((row \* col) - countZeros))

printf("\nYup!, Given Matrix is Sparse Matrix\nTotal Zeros Contain is : %d", countZeros);

else

printf("\nNo!, Given Matrix is Not Sparse Matrix\nTotal Zeros contain is: %d", countZeros);

return 0;

}

10. Write a program in C to find the row with maximum number of 1s.

// the sparse matrix can be defined as the matrix that has a greater number of zero elements than the non-zero elements.

#include <stdio.h>

int main()

{

int row, col;

printf("Enter Row and column of Matrix: ");

scanf("%d%d", &row, &col);

int matrix[row][col], countOnes, findRow = 0, maxOnes = 0;

printf("Enter Element in %d\*%d Matrix: \n", row, col);

for (int i = 0; i < row; i++)

for (int j = 0; j < col; j++)

scanf("%d", &matrix[i][j]);

// print Matrix and find row with max 1

printf("Given Matrix: \n");

for (int i = 0; i < row; i++)

{

countOnes = 0;

for (int j = 0; j < col; j++)

{

printf("%d ", matrix[i][j]);

if (matrix[i][j] == 1)

countOnes++;

}

if (countOnes > maxOnes)

{

maxOnes = countOnes;

findRow = i + 1;

}

printf("\n");

}

if (findRow > 0)

printf("\nMax Ones Contain Row is: %d", findRow);

else

printf("\nNo, Row Contain 1!");

return 0;

}