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

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 i, j;

void input(int x[3][3])

{

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

{

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

{

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

}

}

}

void display(int x[3][3])

{

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

{

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

{

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

}

printf("\n");

}

}

void add(int a[3][3], int b[3][3], int c[3][3])

{

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

{

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

{

c[i][j] = (a[i][j]) + (b[i][j]);

}

}

}

int main()

{

int a[3][3];

int b[3][3];

int c[3][3];

printf("Enter the elements of the first array:\n");

input(a);

printf("Enter the elements of the second array:\n");

input(b);

printf("The first array :\n");

display(a);

printf("The second array :\n");

display(b);

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

{

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

{

c[i][j] = 0;

}

}

add(a, b, c);

printf("The final array :\n");

display(c);

return 0;

}

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

#include <stdio.h>

int i, j;

void input(int x[3][3])

{

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

{

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

{

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

}

}

}

void display(int x[3][3])

{

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

{

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

{

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

}

printf("\n");

}

}

void multiply(int a[3][3], int b[3][3], int c[3][3])

{

int k;

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

{

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

{

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

{

c[i][j] += (a[i][k]) \* (b[k][j]);

}

}

}

}

int main()

{

int a[3][3];

int b[3][3];

int c[3][3];

printf("Enter the elements of the first array:\n");

input(a);

printf("Enter the elements of the second array:\n");

input(b);

printf("The first array :\n");

display(a);

printf("The second array :\n");

display(b);

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

{

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

{

c[i][j] = 0;

}

}

multiply(a, b, c);

printf("The final array :\n");

display(c);

return 0;

}

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

#include <stdio.h>

void input(int n, int m, int arr[n][m]);

void display(int n, int m, int xrr[n][m]);

void transpose(int n, int m, int yrr[n][m], int trr[m][n]);

int main()

{

printf("Enter the dimensions of the matrix: ");

int n, m;

scanf("%d %d", &n, &m);

int arr[n][m];

printf("Enter the elements of the matrix:\n");

input(n, m, arr);

printf("Original matrix:\n");

display(n, m, arr);

int trr[m][n];

transpose(n, m, arr, trr);

printf("Transpose of the matrix:\n");

display(m, n, trr);

return 0;

}

void input(int n, int m, int arr[n][m])

{

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

{

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

{

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

}

}

}

void display(int n, int m, int xrr[n][m])

{

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

{

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

{

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

}

printf("\n");

}

}

void transpose(int n, int m, int yrr[n][m], int trr[m][n])

{

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

{

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

{

trr[j][i] = yrr[i][j];

}

}

}

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

#include <stdio.h>

int i, j, n, m;

void input(int n, int m, int arr[n][m])

{

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

{

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

{

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

}

}

}

void display(int n, int m, int xrr[n][m])

{

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

{

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

{

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

}

printf("\n");

}

}

int sumrow(int n, int m, int arr[n][m])

{

int s = 0;

for (i = 0, j = m - 1; i < n, j >= 0; i++, j--)

{

s += arr[i][j];

}

printf("Sum of all right diagonal Elements is %d \n", s);

}

int main()

{

printf("Enter the dimension of the matrix:\n");

scanf("%d %d", &n, &m);

int arr[n][m];

printf("Enter the elements of the matrix:\n");

input(n, m, arr);

display(n, m, arr);

sumrow(n, m, arr);

}

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

#include<stdio.h>

int i,j,n,m;

void input(int n, int m, int arr[n][m])

{

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

{

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

{

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

}

}

}

void display(int n, int m, int xrr[n][m])

{

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

{

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

{

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

}

printf("\n");

}

}

int sumleft(int n,int m,int arr[n][m])

{

int s=0;

for(i=0,j=0;i<n,j<m;i++,j++)

{

s+=arr[i][j];

}

printf("Sum of all left diagonal Elements is %d \n",s);

}

int main()

{

printf("Enter the dimension of the matrix:\n");

scanf("%d %d",&n,&m);

int arr[n][m];

printf("Enter the elements of the matrix:\n");

input(n,m,arr);

display(n,m,arr);

sumleft(n,m,arr);

}

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

#include <stdio.h>

int i, j, n, m;

void input(int n, int m, int arr[n][m])

{

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

{

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

{

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

}

}

}

void display(int n, int m, int xrr[n][m])

{

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

{

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

{

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

}

printf("\n");

}

}

int sumrow(int n, int m, int arr[n][m])

{

int s = 0;

printf("Enter the Row number:\n");

scanf("%d", &i);

for (j = 0; j < m; j++)

{

s += arr[i - 1][j];

}

printf("Sum of all Elements in %d row is %d \n", i, s);

}

int sumcol(int n, int m, int arr[n][m])

{

int s = 0;

printf("Enter the column number:\n");

scanf("%d", &j);

for (i = 0; i < n; i++)

{

s += arr[i][j - 1];

}

printf("Sum of all elements in %d column is %d", j, s);

}

int main()

{

printf("Enter the dimension of the matrix:\n");

scanf("%d %d", &n, &m);

int arr[n][m];

printf("Enter the elements of the matrix:\n");

input(n, m, arr);

display(n, m, arr);

sumrow(n, m, arr);

sumcol(n, m, arr);

}

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

#include <stdio.h>

int i, j, n, m;

void input(int n, int m, int arr[n][m])

{

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

{

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

{

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

}

}

}

void display(int n, int m, int xrr[n][m])

{

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

{

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

{

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

}

printf("\n");

}

}

void lowtri(int n, int m, int arr[n][m])

{

int lt[n][m];

for (i = 0; i < n; i++)

{

for (j = 0; j < m; j++)

{

if (i >= j)

lt[i][j] = arr[i][j];

else

lt[i][j] = 0;

}

}

printf("Lower triangular matrix is:\n");

display(n, m, lt);

}

int main()

{

printf("Enter the dimension of the matrix:\n");

scanf("%d %d", &n, &m);

int arr[n][m];

printf("Enter the elements of the matrix:\n");

input(n, m, arr);

printf("Your matrix:\n");

display(n, m, arr);

lowtri(n, m, arr);

}

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

#include <stdio.h>

int i, j, n, m;

void input(int n, int m, int arr[n][m])

{

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

{

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

{

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

}

}

}

void display(int n, int m, int xrr[n][m])

{

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

{

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

{

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

}

printf("\n");

}

}

void uptri(int n, int m, int arr[n][m])

{

int lt[n][m];

for (i = 0; i < n; i++)

{

for (j = 0; j < m; j++)

{

if (i <= j)

lt[i][j] = arr[i][j];

else

lt[i][j] = 0;

}

}

printf("Uper triangular matrix is :\n");

display(n, m, lt);

}

int main()

{

printf("Enter the dimension of the matrix:\n");

scanf("%d %d", &n, &m);

int arr[n][m];

printf("Enter the elements of the matrix:\n");

input(n, m, arr);

printf("Your matrix:\n");

display(n, m, arr);

uptri(n, m, arr);

}

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

#include <stdio.h>

int isSparse(int m, int n, int a[m][n])

{

int i, j, count = 0;

for (i = 0; i < m; i++)

{

for (j = 0; j < n; j++)

{

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

{

count++;

}

}

}

if (count > (m \* n) / 2)

{

return 1;

}

else

{

return 0;

}

}

int main()

{

int m, n, i, j;

printf("Enter the number of rows and columns of the matrix: ");

scanf("%d %d", &m, &n);

int a[m][n];

printf("Enter the elements of the matrix: \n");

for (i = 0; i < m; i++)

{

for (j = 0; j < n; j++)

{

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

}

}

if (isSparse(m, n, a))

{

printf("The matrix is sparse.\n");

}

else

{

printf("The matrix is not sparse.\n");

}

return 0;

}

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

#include <stdio.h>

int rowWithMaxOnes(int m, int n, int a[m][n])

{

int i, j, maxRow = 0, maxCount = 0, count;

for (i = 0; i < m; i++)

{

count = 0;

for (j = 0; j < n; j++)

{

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

{

count++;

}

}

if (count > maxCount)

{

maxCount = count;

maxRow = i ;

}

}

return maxRow;

}

int main()

{

int m, n, i, j;

printf("Enter the number of rows and columns of the matrix: ");

scanf("%d %d", &m, &n);

int a[m][n];

printf("Enter the elements of the matrix: \n");

for (i = 0; i < m; i++)

{

for (j = 0; j < n; j++)

{

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

}

}

int row = rowWithMaxOnes(m, n, a);

printf("The row with maximum number of ones is: %d\n", row+1);

return 0;

}