2D Array Practice Questions

1. WAP to input a 2D array of size M\*N and display it in tabular form.

#include ‹stdio.h> int main()

int i,j,m,n;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&a[i][j]);

printf(”Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

### WAP to input a 2D array of size M\*N and find the sum and average of all the elements.

#include ‹stdio.h> int main()

int i,j,m,n,s=0;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

s c and( "id " , &a [ 1] [j ] ) ; s=s+a [ 1] [ j ] ;

printf(”Sum of All Elements: %d\n”,s); printf(”Average: %f\n”,(float)s/(m\*n));

1. **WAP to input a 2D array of size M\*N and find the sum of individual rows and individual columns.**

#include ‹stdio.h> int main()

int i,j,m,n,r=0,c=0;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&a[i][j]);

printf(”Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,a[i][j]);

printf(”\n”); for(i=0;i<=m-1;i++)

r=0,c=0;

for(j=0;j<=n-1;j++)

r=r+a[i][j];

c=c+a[j][i];

printf(”Sum of Row No %d: %d\n”,i+1,r); printf(”Sum of Column No %d: %d\n”,i+1,c);

## WAP to input a 2D array of size M\*N and display boundary elements in matrix form.

#include ‹stdio.h>

int main()

int i,j,m,n;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&a[i][j]);

printf(”Boundary Elements:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

if(i==0||j==0||i==(m-1)||j==(n-1)) printf(”%d\t”,a[i][j]);

else

printf(”\t”);

printf(”\n”);

## WAP to input a 2D array of size M\*N and display the transpose of it.

#include ‹stdio.h> int main()

int i,j,m,n;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&a[i][j]);

printf(”Transpose:\n”); for(i=0;i<=n-1;i++)

for(j=0;j<=m-1;j++)

printf(”%d\t”,a[j][i]);

printf(”\n”);

### WAP to input a 2D array and find the sum of its diagonal elements.

#include ‹stdio.h>

int main()

int i,j,m,n,s=0;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

if(m==n)

int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

scanf(”%d”,&a[i][j]); if(i==j||(i+j)==(m-1))

s=s+a[i][j];

printf(”Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

printf(”%d\t”,a[i][j]);

printf(”\n”);

printf(”Sum of Diagonal Elements = %d\n”,s);

else

printf(”Not A Square Matrix Therfore No Diagonal Sum\n”);

### WAP to input a 2D array and display diagonal elements in matrix form.

#include <stdio.h> int main()

int i,j,m,n,s=0;

printf(”Enter no. Rows and columns\n”); scanf("%d%d",&m,&n);

if(m==n)

### int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n));

#### for(i=0;i‹=m-1;i++)

for(j=0;j‹=n-l;j++)

scanf(”%d",&a[i][j]);

printf(”Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

printf(”Diagonal Elements\n”); for(i=0;i<=m-1;i++)

for(j=8;j‹=n-1;j++)

if(i==j||(i+j)==(m-1)) printf(”%d\t”,a[i][j]);

else

printf(”\t”);

printf(”\n");

else

printf(”Not A Square Natrix Therfore No Diagonal Print\n”);

1. WAP to input 2 matrices from the user and add them.

#include <stdio.h>

int main()

int i,j,m,n,p,q;

printf(”Enter no. Rows and columns of Matrix A\n”); scanf(”%d%d”,&m,&n);

printf(”Enter no. Rows and columns of Matrix B\n”); scanf(”%d%d”,&p,&q);

if(m==p&&n==q)

int a[m][n],b[m][n],c[m][n];

printf(”Enter %d elements of Matrix A:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&a[i][j]);

printf(”Enter %d elements of Matrix B:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&b[i][j]);

for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) c[i][j]=a[i][j]+b[i][j];

printf(”Matrix A is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

printf(”Matrix B is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

printf(”Matrix C is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

printf(”%d\t”,c[i][j]); printf(”\n”);

else

printf(”Cannot Add as order of 2 matrices if diffrent\n”);

1. WAP to input a matrix and check if its identity matrix or not.

#include <stdio.h> int main()

int i,j,m,n,d=0,ot=0;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

if(m==n)

### int a[m][n];

printf("Enter %d elements of 2D array:\n”,(m\*n));

#### for(i=0;i‹=m-1;i++)

for(j=8;j‹=n-1;j++)

scanf(”%d”,&a[i][j]);

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

d++;

else

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

printf("Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n");

if(d==m&&ot==(m\*m-m)) printf(”Identity Natrix\n”);

else

printf(”Non Identity Natrix\n”);

else

printf(”Not A Square Natrix Therfore Non Identity Natrix\n”);

1. WAP to input a matrix of order M”N and check if it’s sparse or dense matrix.

#include <stdio.h> int main()

int i,j,m,n,z=0,nz=0;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

### int a[m][n];

printf(”Enter %d elements of 2D array:\n",(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++)

scanf(”%d”,&a[i][j]);

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

else

nz++;

printf(”Entered Natrix is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

#### if(z>nz)

printf(”Sparse Natrix\n”);

#### else if(z‹nz)

printf(”Dense Natrix\n”);

else

printf(”Not Dense ot Sparse Natrix\n”);

# Write a program in C to calculate determinant of a 3 x 3 matrix.

#include ‹stdio.h> int main()

int i,j,d=0; int a[3][3];

printf(”Enter 9 elements of 3\*3 Matrix :\n”); for(i=0;i<=2;i++)

for(j=0;j<=2;j++)

scanf(”%d”,&a[i][j]);

printf(”Entered Matrix is:\n”); for(i=0;i<=2;i++)

for(j=0;j<=2;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

for(i=0;i<=2;i++)

d=d+a [0] [ 1] \*( (a [ 1] [ (1+1)¥3] \*a [ 2] [ (1+2)¥3] ) -

(a [ 1] [ (1+2)¥3] \*a [ 2] [ ( 1+1)¥3] ) ) ;

printf(”Determinant = %d\n”,d);

# Write a program to keep records and perform statistical analysis for a class of students. The class may have up to 10 students. There are three quizzes during the semester for computer programming. Each student is identified by a four digit roll no. The program will print the student scores and calculate and print the following statistics

* 1. High score & Low score for each quiz along with the roll number of that student.
  2. Average of Each Quiz and overall average of all the 3 quizzes
  3. Highest Average and Lowest average of all the three quizzes along with quiz no.
  4. Highest marks & lowest marks of each student in the 3 quizzes along with the quiz no.

#include ‹stdio.h>

int main()

1nt

i,j,d=0,s=0,max1,max2,max3,min1,min2,min3,rmax1,rmax2,rmax3,rmin1,rmin2,rmin3;

### int a[10][4];

#### int avg[3]=(0};

int marks[10][2]=(0};

int lmarks[10][2]=(0};

for(i=0;i<=9;i++)

printf("Enter Roll no. of student %d and his marks in quiz 1 2

#### 3:\n”,i+1);

for(j=0;j‹=3;j++)

scanf(”%d",&a[i][j]);

s=s+a[i][j];

### avg[0]=avg[0]+a[i][1];

if(j==2)

### avg[1]=avg[1]+a[i][2];

if(j==3)

#### avg[3]=avg[3]+a[i][3];

if(a[i][j]>marks[i][0])

marks[i][0]=a[i][j]; //highest marks in each quiz marks[i][1]=j+1; //quiz number of highest marks

if(a[i][j]<lmarks[i][0])

lmarks[i][0]=a[i][j]; //lowest marks in each quiz lmarks[i][1]=j+1; //quiz number of lowest marks

maxl=a[0][1],max2=a[0][2],max3=a[0][3];

rmaxl=a[0][0],rmax2=a[0][0],rmax3=a[0][0];

minl=a[0][1],min2=a[0][2],min3=a[0][3];

rminl=a[0][0],rmin2=a[0][0],rmin3=a[0][0];

### for ( i=0; ’i‹=9; i++)

for(j=0;j‹=4;j++)

### if(a[i][1]>max1)

maxl=a[i][1];

rmaxl=a[i][0];

### if(a[i][2]>max2)

max2=a[i][2];

rmax2=a[i][0];

### if(a[i][3]>max3)

max3=a[i][3];

rmax3=a[i][0];

### if(a[i][1]‹min1)

minl=a[i][1];

rminl=a[i][0];

### if(a[i][2]‹min2)

min2=a[i][2];

rmin2=a[i][0];

### if(a[i][3]‹min3)

min3=a[i][3];

rmin3=a[i][0];

printf(”Naximum Score of Quiz 1 and Roll no. %d = %d\n",rmax1,max1); printf(”Naximum Score of Quiz 2 and Roll no. %d = %d\n”,rmax2,max2); printf(”Naximum Score of Quiz 3 and Roll no. %d = %d\n”,rmax3,max3); printf(”Ninimum Score of Quiz 1 and Roll no. %d = %d\n”,rmin1,min1); printf(”Ninimum Score of Quiz 2 and Roll no. %d = %d\n”,rmin2,min2); printf(”Minimum Score of Quiz 3 and Roll no. %d = %d\n”,rmin3,min3); printf(”Average of all Quizs = %f”,(float)s/30);

printf(”Average of 1st Quiz = %f”,(float)avg[0]/10); printf(”Average of 2nd Quiz = %f”,(float)avg[l]/l0); printf(”Average of 3rd Quiz = %f",(float)avg[2]/10); float max=avg[0],min=avg[0];

int maxr=0,minr=0; for(i=0;i<=2;i++)

#### if(avg[i]>max)

maxr=i; max=avg[i];

### if(avg[i]‹min)

minr=i; min=avg[i];

printf(”Highest Average is of Quiz %d in all Quizs = %f”,maxr+1,max); printf(”Lowest Average is of Quiz %d in all Quizs = %f”,minr+1,min); for(i=0;i<=9;i++)

printf(”Roll Number %d\n”,a[i][0]); printf(”Highest Narks are %d and in Quiz No.

%d\n”,marks[i][0],marks[i][1]);

printf(”Lowest Narks are %d and in Quiz No.

%d\n”,lmarks[i][0],lmarks[i][1]);

1. WAP to input 2 matrixes and **multiply** them.

#include <stdio.h> int main()

int i,j,p,q,m,n,k;

printf(”Enter no. Rows and columnsof Natrix A\n”); scanf(”%d%d”,&m,&n);

printf(”Enter no. Rows and columnsof Natrix B\n”); scanf(”%d%d",&p,&q);

if(q==m)

### int a[m][n],b[p][q],c[m][q];

printf("Enter %d elements Natrix A:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++) scanf(”%d”,&a[i][j]);

printf("Enter %d elements Natrix B:\n”,(m\*n));

for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) scanf(”%d”,&b[i][j]);

printf(”Matrix A is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

printf(”Matrix B is:\n”); for(i=0;i<=p-1;i++)

for(j=0;j<=q-1;j++) printf(”%d\t”,b[i][j]);

printf(”\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=q-1;j++)

c[i][j]=0; for(k=0;k<=n-1;k++)

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

printf(”Multiplication of A anb B is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=q-1;j++) printf(”%d\t”,c[i][j]);

printf(”\n”);

else

printf(”Cannot Multiply as Column of A is not equal to Row of B\n”);

# WAP to input a matrix and print its upper triangular matrix.

#include ‹stdio.h>

int main()

int i,j,m,n;

printf("Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

if(m==n)

### int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=8;j‹=n-1;j++)

scanf(”%d”,&a[i][j]);

printf("Entered 2D Array is:\n”);

#### for(i=0;i‹=m-1;i++)

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

#### printf(”Upper Triangular Natrix is:\n”); for(i=0;i‹=m-1;i++)

for(j=8;j‹=n-1;j++)

else

printf(”%d\t",a[i][j]); printf(”\t”);

printf(”\n”);

else

printf(”Not A Square Natrix Therefore No Upper Triangular Natrix”);

1. WAP to input a matrix and print its Lower triangular matrix.

#include <stdio.h> int main()

int i,j,m,n;

printf(”Enter no. Rows and columns\n”); scanf("%d%d",&m,&n);

if(m==n)

### int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n));

#### for(i=0;i‹=m-1;i++)

for(j=0;j‹=n-l;j++)

scanf(”%d”,&a[i][j]);

printf(”Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

printf("Lower Triangular Natrix is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++)

else

printf(”%d\t”,a[i][j]); printf(”\t”);

printf(”\n");

else

printf(”Not A Square Natrix Therefore No Lower Triangular Natrix”);

1. WAP to input a matrix and print it in zig zack form. Example :- if matrix is 1 2 3 4

5 6 7 8

9 1 2 3

4 5 9 1

Then output matrix will be 1 2 3 4

8 7 6 5

9 1 2 3

1 9 5 4

#include <stdio.h> int main()

#### int i,j,m,n;

printf("Enter no. Rows and columns\n”); scanf("%d%d",&m,&n);

### int a[m][n];

printf(”Enter %d elements of 2D array:\n”,(m\*n));

#### for(i=0;i‹=m-1;i++)

for(j=0;j‹=n-l;j++)

scanf("%d”,&a[i][j]);

printf(”Entered 2D Array is:\n”); for(i=0;i<=m-1;i++)

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

#### printf(”Zick Zack Natrix is:\n”);

for(i=0;i<=m-1;i++)

### if(i%2==1)

for(j=n-l;j>=0;j--) printf(”%d\t”,a[i][j]);

else

for(j=0;j‹=n-l;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

1. Suppose there is a game known as “MATCH THE TABLES”, in which the player picks up two tables(each having 10 rows and 10 columns) and matches them. If out of 100 entries at least 90 corresponding entries match then the tables are said to be identical and the player is declared the winner. Wap in ’C’ to implement the above game.

#include <stdio.h>

int ma in ( )

int i,j,m,n,c=0;

printf(”Enter no. Rows and columns\n”); scanf(”%d%d”,&m,&n);

int a[m][n],b[m][n];

printf(”Enter %d elements of 1st Table:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

s c and( "id " , &a [ i ] [j ] ) ;

printf(”Enter %d elements of 2nd Table:\n”,(m\*n)); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

s c and( "id " , &b [ i ] [j ] ) ;

printf(”First Table is: \n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,a[i][j]);

printf(”\n”);

printf(”2nd Table is: \n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++) printf(”%d\t”,b[i][j]);

printf(”\n”); for(i=0;i<=m-1;i++)

for(j=0;j<=n-1;j++)

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

c++;

if(c›=90)

printf(”winner”);

else

printf(”Loser”);

# Which of the following initializations of a 2d array are valid?

(i) int abc[2][2] = {1, 2, 3 ,4 } (ii) int abc[] [] = {1, 2, 3 ,4 }

(iii) int abc[][2] = {1, 2, 3,4 } (iv) int abc[2][] = {1, 2, 3,4 ) ANSWER: i and iii are valid

1. An array =X [-15..........10, 15...............40] requires one byte of storage. If beginning location is 1500 determine the location of X [5][20]for data stored as

(i) Column major wise (ii) Row major wise.

ANSWER:

i)No. of Rows = 10-(-15) + 1 = 26

Address of X[5][20] = 1500 + 1\*[(20-15)\*26 + (5--15)] =1650

ii) No. of Columns = 40-15 + 1 = 26

Address of X[5][20] = 1500 + 1\* [(5-(-15))\*26 + (20-15)] =2025

1. Consider the following declaration of a ’two-dimensional array in C: char a[100][100];

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0. Find the address of a[40][50] using row and column major.

ANSWER: Row Wise Address of a[40][50] = 0 + 1\* [(40-0)) \*50 + (50-0)] =2050

Column Wise Address of a[40][50] = 0 + 1\* [(50-0))” 40 + (40-0)] =2040

1. Let A be a square matrix of size n x n. Consider the following program. What is the expected output?

C = 100;

for(i=0;i<n;i++) for(j=0;j<n;j*++\*

Temp = A[i][j] + C;

A[i][j] = A[j][i];

A[j][i] = Temp — C;

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

### for(j=0;j<n;j*++\* printf(“%d ”,A[i][j]);

OUTPUT: Will Print Transpose Of Matrix