**2D Assignment**

Name :- Khushal Agarwal

**Q1. WAP to input a 2D array of size M\*N and display it in tabular form.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows : ");

scanf("%d",&r);

printf("Enetr no. of column : ");

scanf("%d",&c);

int a[r][c];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

printf("\n");

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

{

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

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

printf("\n");

}

}

**Q2. WAP to input a 2D array of size M\*N and find the sum and average of all the elements.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of columns = ");

scanf("%d",&c);

int a[r][c];

float sum=0,avg;

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

sum += a[i][j];

}

}

avg = sum/(r\*c);

printf("Sum = %.0f\n",sum);

printf("Average = %.2f",avg);

}

**Q3. WAP to input a 2D array of size M\*N and find the sum of individual rows and individual columns.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of columns = ");

scanf("%d",&c);

int a[r][c],rsum,csum;

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

printf("\n");

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

{

rsum =0;

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

rsum += a[i][j];

printf("Row %d sum = %d\n",i+1,rsum);

}

printf("\n");

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

{

csum =0;

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

csum += a[i][j];

printf("Column %d sum = %d\n",j+1,csum);

}

}

**Q4. WAP to input a 2D array of size M\*N and display boundary elements in matrix form.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of column = ");

scanf("%d",&c);

int a[r][c];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

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

{

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

{

if(i==0||j==0||i==r-1||j==c-1)

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

else

printf(" ");

}

printf("\n");

}

}

**Q5. WAP to input a 2D array of size M\*N and display the transpose of it.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of column = ");

scanf("%d",&c);

int a[r][c];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

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

{

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

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

printf("\n");

}

}

**Q6. WAP to input a 2D array and find the sum of its diagonal elements.**

**Soln.** #include<stdio.h>

int main()

{

int n;

printf("Enter no. of rows/column = ");

scanf("%d",&n);

int a[n][n],ld=0,rd=0;

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

if(i==j)

ld+=a[i][j];

if(i+j==n)

rd+=a[i][j];

}

}

printf("Sum of Left diagonal = %d\n",ld);

printf("Sum of Right diagonal = %d",rd);

}

**Q7. WAP to input a 2D array and display diagonal elements in matrix form.**

**Soln.** #include<stdio.h>

int main()

{

int n;

printf("Enter no. of rows/column = ");

scanf("%d",&n);

int a[n][n];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

printf("\n");

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

{

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

{

if(i==j||i+j==n-1)

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

else

printf(" ");

}

printf("\n");

}

}

**Q8. WAP to input 2 matrices from the user and add them.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of column = ");

scanf("%d",&c);

int a[r][c],b[r][c],s[r][c];

printf("For First Matrix\n");

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

printf("\nFor Second Matrix\n");

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

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

}

}

printf("\nMatrix after Sum\n");

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

{

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

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

printf("\n");

}

}

**Q9. WAP to input a matrix and check if its identity matrix or not.**

**Soln.** #include<stdio.h>

int main()

{

int n,f=0;

printf("Enter no. of rows/column = ");

scanf("%d",&n);

int a[n][n];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

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

{

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

{

if(i==j){

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

f=1;

}

if(i!=j){

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

f=1;

}

}

if(f==1)

break;

}

if(f==0)

printf("Identity Matrix");

else

printf("Not an Identity Matrix");

}

**Q10. WAP to input a matrix of order M\*N and check if it’s sparse or dense matrix.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of column = ");

scanf("%d",&c);

int a[r][c],zero=0,ele=0;

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

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

zero++;

else

ele++;

}

}

if(ele>zero)

printf("Dense");

else if(zero>ele)

printf("Sparse");

else

printf("Neither Dense Nor Sparse");

}

**Q11. Write a program in C to calculate determinant of a 3 x 3 matrix.**

**Soln.** #include<stdio.h>

int main()

{

int a[3][3];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

int det=0;

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

det += a[0][i]\*(a[1][(i+1)%3]\*a[2][(i+2)%3]-a[1][(i+2)%3]\*a[2][(i+1)%3]);

printf("Determinant = %d",det);

}

**Q12. 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 term 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.**

**Soln.** #include<stdio.h>

#include<limits.h>

int main()

{

int n;

printf("Enter no. of students = ");

scanf("%d",&n);

int r[n],q[3][n];

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

{

printf("Enter roll no. of student %d = ",i+1);

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

}

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

{

printf("Quiz %d =>\n",i+1);

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

{

printf("Score of rollno. %d = ",r[j]);

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

}

}

**//1.High & low score**

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

{

int max=INT\_MIN,min=INT\_MAX,maxpos,minpos;

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

{

if(q[i][j]>max)

{

maxpos=j;

max=q[i][j];

}

if(q[i][j]<min)

{

minpos=j;

min=q[i][j];

}

}

printf("\nQuiz %d =>\n",i+1);

printf("MAX score is %d by rollno. %d\n",max,r[maxpos]);

printf("MIN score is %d by rollno. %d\n",min,r[minpos]);

}

printf("\n");

**//2.Average**

float oavg=0,max=0,min=999999;

int maxpos,minpos;

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

{

float sum=0;

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

{

sum+=q[i][j];

}

oavg+=sum;

sum/=n;

if(sum>max)

{

maxpos=i;

max=sum;

}

if(sum<min)

{

minpos=i;

min=sum;

}

printf("Quiz %d average = %.2f\n",i+1,sum);

}

printf("Overall avg = %.2f\n\n",oavg/(3\*n));

**//3.Max & Min average**

printf("Max average : Quiz %d => %.2f\n",maxpos+1,max);

printf("Min average : Quiz %d => %.2f\n",minpos+1,min);

//4.Each student highest and lowest marks

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

{

int smax=INT\_MIN,smin=INT\_MAX;

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

{

if(q[i][j]>smax)

{

maxpos=i;

smax=q[i][j];

}

if(q[i][j]<smin)

{

minpos=i;

smin=q[i][j];

}

}

printf("\nStudent rollno. %d :\n Highest marks in Quiz %d => %d\tLowest marks in Quiz %d => %d\n",r[j],maxpos+1,smax,minpos+1,smin);

}

}

**Q13. WAP to input a matrix and print its upper triangular matrix.**

**Soln.** #include<stdio.h>

int main()

{

int n;

printf("Enter no. of rows/column = ");

scanf("%d",&n);

int a[n][n];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

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

{

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

{

if(j>i)

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

else

printf(" ");

}

printf("\n");

}

}

**Q14. WAP to input a matrix and print its lower triangular matrix.**

**Soln.** #include<stdio.h>

int main()

{

int n;

printf("Enter no. of rows/column = ");

scanf("%d",&n);

int a[n][n];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

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

{

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

{

if(j<i)

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

else

printf(" ");

}

printf("\n");

}

}

**Q15. WAP to input a matrix and print it in zig zack form.**

**Soln.** #include<stdio.h>

int main()

{

int r,c;

printf("Enter no. of rows = ");

scanf("%d",&r);

printf("Enter no. of column = ");

scanf("%d",&c);

int a[r][c];

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

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

{

if(i%2==0)

{

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

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

}

else

{

for(int j=c-1;j>=0;j--)

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

}

printf("\n");

}

}

**Q16. 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.**

**Soln.** #include<stdio.h>

int main()

{

int n=10,c=0;

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

printf("For First Matrix\n");

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

}

}

printf("\nFor Second Matrix\n");

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

{

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

{

printf("Enter [%d][%d] element = ",i+1,j+1);

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

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

c++;

}

}

if(c>=90)

printf("Tables are Identical\nPLAYER WINS.");

else

printf("Tables are UnIdentical\nPLAYER LOSES."); }

**Q17. 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[][] = {1, 2, 3 ,4 } (iv) int abc[2][] = {1, 2, 3 ,4 }**

**Soln.** (i)

**Q18. 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]);**

**Soln.** Same matrix A

**Q19. 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.**

**Soln. (i)**Column wise :- 1650

**(ii)**Row wise :- 2025

**Q20. 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.**

**Soln.**Row wise = 4050

Column wise = 5040

**Q21. WAP to input 2 matrixes and multiply them.**

**Soln.** #include <stdio.h>

int main()

{

int r1,c1,r2,c2;

printf("Enter rows for matrix 1 = ");

scanf("%d", &r1);

printf("Enter column for matrix 1 = ");

scanf("%d",&c1);

printf("\nEnter rows for matrix 2 = ");

scanf("%d",&r2);

printf("Enter column for matrix 2 = ");

scanf("%d",&c2);

if (c1 == r2)

{

int a[r1][c1],b[r2][c2],c[r1][c2];

for(int i=0;i<r1;i++)

{

for(int j=0;j<c2;j++)

c[i][j]=0;

}

printf("\nFor Matrix 1 =>\n");

for(int i=0;i<r1;i++) //input matrix 1

{

for (int j=0;j<c1;j++){

printf("Enter [%d][%d] element :- ",i+1,j+1);

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

}

}

printf("\nFor Matrix 2 =>\n");

for (int i=0;i<r2;i++) //input

{

for (int j=0;j<c2;j++)

{

printf("Enter [%d][%d] element :- ",i+1,j+1);

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

}

}

for (int i=0;i<r1;i++)

{

for (int j=0;j<c2;j++)

{

for(int k=0;k<c1;k++)

{

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

}

}

}

for (int i=0;i<r1;i++) //printing

{

for (int j=0;j<c2;j++)

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

printf("\n");

}

}

else

printf("Multiplication not possible\n");

return 0;

}