DSA LAB 3

**//Q.1Write a program to add two matrices and display it using function.**

#include <stdio.h>

#include <stdio.h>

void readmatrix(int m[10][10], int row, int col)

{

int i, j;

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

{

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

{

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

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

}

}

}

void printmatrix(int m[10][10], int row, int col)

{

int i, j;

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

{

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

{

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

}

printf("\n");

}

}

int main()

{

int a[10][10], b[10][10], result[10][10];

int i, j, rows1, cols1, rows2, cols2;

printf("Number of Rows of Matrix 1: ");

scanf("%d", &rows1);

printf("Number of Columns of Matrix 1: ");

scanf("%d", &cols1);

printf("\nEnter Elements of matrix 1:\n");

readmatrix(a, rows1, cols1);

printf("Number of Rows of Matrix 2: ");

scanf("%d", &rows2);

printf("Number of Columns of Matrix 2: ");

scanf("%d", &cols2);

printf("\nEnter Elements of Matrix 2: \n");

readmatrix(b, rows2, cols2);

if (rows1 == rows2 && cols1 == cols2)

{

for (i = 0; i < rows1; i++)

{

for (j = 0; j < cols1; j++)

{

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

}

}

printf("\nMatrix after adding:\n");

printmatrix(result, rows1, cols1);

}

else

{

printf("\nMatrix can not be added, Number of Rows & Columns are Different");

}

return 0;

}

**OUTPUT**

**Number of Rows of Matrix 1: 2**

**Number of Columns of Matrix 1: 2**

**Enter Elements of matrix 1:**

**Enter element [1,1] : 4**

**Enter element [1,2] : 6**

**Enter element [2,1] : 7**

**Matrix after adding:**

**12 6**

**15 17**

**//Q2. Write a program to multiply two matrices and display it using function.**

#include<stdio.h>

void multiply(int mat1[10][10],int mat2[10][10],int ,int ,int );

void main()

{

int mat1[10][10],mat2[10][10];

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

printf("Enter the number of rows and columns for 1st matrix\n");

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

printf("Enter the elements of the 1st matrix\n");

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

{

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

{

scanf("%d",&mat1[i][j]);

}

}

printf("Enter the number of columns for 2nd matrix\n");

scanf("%d",&p);

printf("Enter the elements of the 2nd matrix\n");

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

{

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

{

scanf("%d",&mat2[i][j]);

}

}

printf("The 1st matrix\n");

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

{

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

{

printf("%d\t",mat1[i][j]);

}

printf("\n");

}

printf("The 2nd matrix\n");

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

{

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

{

printf("%d\t",mat2[i][j]);

}

printf("\n");

}

multiply(mat1,mat2,m,n,p);

}

void multiply(int mat1[10][10],int mat2[10][10],int m,int n,int p)

{

int mul[10][10],i,j,k;

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

{

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

{

mul[i][j]=0;

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

{

mul[i][j]=mul[i][j]+mat1[i][k]\*mat2[k][j];

}

}

}

printf("The resultant matrix formed on multiplying the two matrices\n");

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

{

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

{

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

}

printf("\n");

}

}

**OUTPUT**

Enter the number of rows and columns for 1st matrix

2

2

2

3

4

5

The 1st matrix

4 5

6 7

The 2nd matrix

2 3

4 5

The resultant matrix formed on multiplying the two matrices

28 37

40 53

**//Q3. WAP to find the Trace(sum of the diagonal element) of a given mxn matrix** using function

#include <stdio.h>

#include <stdlib.h>

int findTrace( int d, int\*\* M){

int trace = 0;

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

trace += M[i][i];

return trace;

}

int main(){

int d;

printf("Enter row/col count of matrix: ");

scanf("%d", &d);

int \*\*M = (int \*\*)malloc(d \* sizeof(int \*));

for (int i = 0; i < d; i++) {

M[i] = (int \*)malloc(d \* sizeof(int));

}

// Input

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

for( int i = 0; i < d; i++ ){

printf("Row %d: ", i+1);

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

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

}

printf("Trace value of matrix is %d.\n", findTrace(d, M));

return 0;

}

**OUTPUT**

Enter row/col count of matrix: 2

Enter the matrix:

Row 1: 2

3

Row 2: 4

5

Trace value of matrix is 7.

**/\*Q4. Create a structure named student that has name, roll and mark as members.**

**Assume appropriate types and size of member. Write a program using structure**

**to read and display the data entered by the user by passing structure to a function.**

**\*/**

#include <stdio.h>

struct Student {

char name[100];

int roll, totalMarks;

};

void input( struct Student \*A ){

// Input details

printf("Enter name: ");

scanf("%[^\n]s", A->name);

printf("Enter roll: ");

scanf("%d", &A->roll);

printf("Enter marks: ");

scanf("%d", &A->totalMarks);

}

void display( struct Student A ){

printf("Student details are: ");

printf("\n\tName: %s\n\tRoll: %d\n\tTotal marks: %d\n", A.name, A.roll, A.totalMarks);

}

int main(){

// declare a student

struct Student A;

// Input details

input(&A);

// Display

display(A);

return 0;

}

**OUTPUT**

Enter name: RISHIKESH

Enter roll: 2105734

Enter marks: 97

Student details are:

Name: RISHIKESH

Roll: 2105734

Total marks: 97

**/\*Q5.WAP to store n employee’s data such as employee name, gender, designation, department, basic**

**pay. Calculate the gross pay of each employees as follows: Gross pay = basic pay + HR + DA HR=25% of**

**basic and DA=75% of basic by passing structure to a function.\*/**

#include <stdio.h>

struct employee

{

char name[10];

char gender[6];

char designation[10];

char department[10];

int basicpay;

};

float get\_gross\_pay(struct employee e)

{

float HRA = 0, DA = 0, gross = 0;

HRA = 0.25 \* (e.basicpay);

DA = 0.75 \* (e.basicpay);

gross = HRA + DA + e.basicpay;

return gross;

}

int main()

{

int n;

printf("Enter the number of employees = ");

scanf("%d", &n);

struct employee s[n];

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

{

printf("Enter employee name : ");

scanf("%s", s[i].name);

printf("Enter employee gender : ");

scanf("%s", s[i].gender);

printf("Enter employee designation : ");

scanf("%s", s[i].designation);

printf("Enter employee department : ");

scanf("%s", s[i].department);

printf("Enter employee basicpay : ");

scanf("%d", &s[i].basicpay);

}

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

{

float gross = get\_gross\_pay(s[i]);

printf("\nNAME : %s\nGENDER : %s\nDESIGNATION : %s\nDEPARTMENT : %s\nBASICPAY : %d\nGROSS : %.2f\n", s[i].name, s[i].gender, s[i].designation, s[i].department, s[i].basicpay, gross);

gross = 0;

}

return 0;

}

**OUTPUT**

Enter the number of employees = 1

Enter employee name : RISHIKESH

Enter employee gender : MALE

Enter employee designation : SD

Enter employee department : TECHNOLOGY

Enter employee basicpay : 1000000

NAME : RISHIKESH

GENDER : MALE

DESIGNATION : SD

DEPARTMENT : TECHNOLOGY

BASICPAY : 1000000

GROSS : 2000000.00

**/\*6. Define a structure of student having data members: name, address, marks in C**

**language, and marks in information system. Take data for n students in an array**

**and find the total marks obtained by passing structure to a function.\*/**

**#include <stdio.h>**

struct student

{

char name[10];

char addres[20];

int marks\_C, marks\_IS;

};

int get\_total\_marks(struct student s)

{

return s.marks\_C + s.marks\_IS;

}

int main()

{

int n;

printf("Enter the number of students = ");

scanf("%d", &n);

struct student s[n];

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

{

printf("\nEnter student name : ");

scanf("%s", s[i].name);

printf("Enter student address : \n");

scanf("%s", s[i].addres);

printf("Enter marks in C language : ");

scanf("%d", &s[i].marks\_C);

printf("Enter marks in Information Systems : ");

scanf("%d", &s[i].marks\_IS);

}

printf("LIST OF STUDENT DETAILS:\n");

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

{

int total = get\_total\_marks(s[i]);

printf("\nNAME : %s\nADDRES : %s\nMARKS IN C LANGUAGE : %d\nMARKS IN SYSTEMS INFORMATION : %d\nTOTAL MARKS: %d\n", s[i].name, s[i].addres, s[i].marks\_C, s[i].marks\_IS, total);

total = 0;

}

return 0;

}

**OUTPUT**

**Enter the number of students = 1**

NAME : RISHIKESH

ADDRES : RANCHI

MARKS IN C LANGUAGE : 49

MARKS IN SYSTEMS INFORMATION : 49

TOTAL MARKS: 98

**//Q7.WAP to add two distances (in km-meter) by passing structure to a function**

#include <stdio.h>

struct Distance {

int km;

int m;

};

struct Distance Add( struct Distance A, struct Distance B ){

struct Distance C;

C.km = A.km + B.km;

C.m = A.m + B.m;

// Since 1000m = 1km

if( C.m >= 1000 ){

C.km += (C.m/1000);

C.m = C.m%1000;

}

return C;

}

void display( struct Distance D ){

printf("%d km and %d m.\n", D.km, D.m);

}

int main(){

struct Distance A, B, C;

printf("Enter km and m for A: ");

scanf("%d %d", &A.km, &A.m);

printf("Enter km and m for B: ");

scanf("%d %d", &B.km, &B.m);

C = Add( A, B );

display(C);

return 0;

}

**OUTPUT**

Enter km and m for A: 7 20

Enter km and m for B: 8 40

15 km and 60 m.

**RISHIKESH**

**2105734**

**CSE 32**