NAME: Abhiram S

ROLL NO: 21003

EXP-17

.

---------------------------------CODE------------------------------------

#include<stdio.h>

#include<conio.h>

void prod(int a[10][10],int b[10][10],int m,int n,int p, int q);

void ip(int a[10][10],int\*,int\*);

void op(int a[10][10],int,int);

void trans(int [10][10],int,int);

void sob(int [10][10],int,int);

void sym();

void menu();

void main()

{

clrscr();

menu();

getch();

}

void prod(int a[10][10],int b[10][10],int m,int n,int p,int q)

{

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

if(n==p)

{

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

{

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

{

c[i][j]=0;

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

{

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

}

}

}

printf("\n Product is:");

op(c,m,q);

}

else

{

printf("\n product can't be found since no of columns in 1st matrix");

printf("\n not equal to no of rows in 2nd matrix...");

}

}

void ip(int a[10][10],int \*m,int \*n)

{

int i,j;

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

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

printf("\n Enter elements:");

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

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

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

}

void op(int a[10][10],int m,int n)

{

int i,j;

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

{

printf("\n");

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

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

}

}

void trans(int a[10][10],int m,int n)

{

int i,j;

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

if(m==n)

{

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

{

printf("\n");

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

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

}

}

else

{

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

{ printf("\n");

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

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

}

}

}

void sob(int a[10][10],int m,int n)

{

int i,j;

int sum=0;

for(i=1;i<n-1;i++)

{

sum+=a[0][i];

sum+=a[m-1][i];

}

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

{

sum+=a[i][0];

sum+=a[n-1][i];

}

sum+=(a[0][0]+a[m-1][0]+a[0][n-1]+a[m-1][n-1]);

printf("\n Sum of boundary numbers of second matrix=%d",sum);

}

void sym()

{

int a[10][10],b[10][10],m,n,f,i,j;

f=0;

printf("\n Enter details of a matrix:");

ip(a,&m,&n);

if(m==n)

{

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

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

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

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

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

{

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

f=1;

else

{

f=0;

break;

}

}

if(f==1)

{

printf("\n Entered matrix is Symmetric");

}

else

printf("\n Entered matrix is Not symmetric");

}

else

printf("\n Entered matrix is not symmetric");

}

void menu()

{

int a[10][10],m,n,b[10][10],p,op,q;

printf("\n Enter details of First matrix:");

ip(a,&m,&n);

printf("\n Enter details of Second matrix:");

ip(b,&p,&q);

do

{

printf("\n 1. find product\n 2.transpose of 1st matrix \n 3.sum of boundary elements in second matrix \n 4.input a square matrix to check if it is symmetric\n 5.exit");

scanf("%d",&op);

if(op==1)

prod(a,b,m,n,p,q);

else if(op==2)

trans(a,m,n);

else if(op==3)

sob(b,p,q);

else if(op==4)

sym();

else if(op==5)

;

else

printf("\n invalid option");

}while(op!=5);

}

--------------------------------------------------OUTPUT-------------------------------------------------------

Enter details of First matrix:

Enter order of the matrix:3 3

Enter elements:

1 2 3

4 5 6

7 8 9

Enter details of Second matrix:

Enter order of the matrix:3 2

Enter elements:

10 11

12 13

14 15

1. find product

2.transpose of 1st matrix

3.sum of boundary elements in second matrix

4.input a square matrix to check if it is symmetric

5.exit 1

Product is:

76 82

184 199

292 316

1. find product

2.transpose of 1st matrix

3.sum of boundary elements in second matrix

4.input a square matrix to check if it is symmetric

5.exit 2

Transpose of first matrix:

1 4 7

2 5 8

3 6 9

1. find product

2.transpose of 1st matrix

3.sum of boundary elements in second matrix

4.input a square matrix to check if it is symmetric

5.exit 3

Sum of boundary numbers of second matrix=75

1. find product

2.transpose of 1st matrix

3.sum of boundary elements in second matrix

4.input a square matrix to check if it is symmetric

5.exit 4

Enter details of a matrix:

Enter order of the matrix:2 2

Enter elements:1 0

0 1

Entered matrix is Symmetric

1. find product

2.transpose of 1st matrix

3.sum of boundary elements in second matrix

4.input a square matrix to check if it is symmetric

5.exit 5