***Assignment No. 2B***

Represent matrix using two dimensional arrays and perform following operations pointers:

I. Addition II. Multiplication

III. Transpose IV. Saddle Point

ROLL NO.66

Batch: - S3

#include<stdio.h>

#include<stdlib.h>

int\*\* create(int m ,int n);

void create1(int a[][10],int m , int n);

void print(int \*\*a,int m ,int n);

void print1(int a[][10], int m , int n);

void transpose(int \*\*a,int m ,int n);

int\*\* addmat(int \*\*a,int m1,int n1 ,int \*\*b,int m2,int n2);

void multmat(int a[][10],int m1,int n1 ,int b[][10],int m2,int n2,int c[][10]);

int saddle(int a[][10],int m,int n);

void main()

{

int \*\*a,\*\*b,\*\*c,m1,n1,m2,n2,m3,n3;

int a1[10][10],b1[10][10],c1[10][10];

int ch;

do

{

printf("\n\*\*\*\*\*\*menu\*\*\*\*\*\*");

printf("\n1)Transpose of the Matrix");

printf("\n2)Add two matrices");

printf("\n3)Multiply two matrices");

printf("\n4)Saddle point");

printf("\n5)Quit");

printf("\nEnter Your Choice : ");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("Enter the size of the matrix :");

scanf("%d%d",&m1,&n1);

a=create(m1,n1);

if(m1==n1)

{

transpose(a,m1,n1);

printf("\nResult=\n");

print(a,m1,n1);

}

else

printf("Not a square matrix :");

break;

case 2: printf("Enter the size of the 1st matrix:");

scanf("%d%d",&m1,&n1);

a=create(m1,n1);

printf("Enter the size of the 2nd matrix:");

scanf("%d%d",&m2,&n2);

b=create(m2,n2);

if(m1==m2 & n1==n2)

{

c=addmat(a,m1,n1,b,m2,n2);

printf("Result=");

print(c,m1,n1);

}

else

printf(" Can not be added ");

break;

case 3: printf(" Enter the size of the 1st matrix:");

scanf("%d%d",&m1,&n1);

create1(a1,m1,n1);

printf(" Enter the size of the 2nd matrix:");

scanf("%d%d",&m2,&n2);

create1(b1,m2,n2);

if(n1==m2)

{ multmat(a1,m1,n1,b1,m2,n2,c1);

printf(" Result= ");

print1(c1,m1,n2);

}

else

printf(" Can not multiply");

break;

case 4: printf(" Enter the size of the 1st matrix:");

scanf("%d%d",&m1,&n1);

create1(a1,m1,n1);

saddle(a1,m1,n1);

break;

case 5: exit(0);

break;

default:

printf("Invalid choice");

break;

}

}while(ch!=5);

}

int \*\* create(int m ,int n)

{

int i,j; int \*\*a;

a=(int\*\*)malloc(m\*sizeof(int\*));

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

\*(a+i)=(int\*)malloc(n\*sizeof(int));

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

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

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

scanf("%d",(\*(a+i)+j));

return(a);

}

void create1(int a[][10],int m , int n)

{ int i,j;

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

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

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

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

}

void print(int \*\*a,int m ,int n)

{ int i,j;

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

{ printf("\n");

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

printf("%5d",\*(\*(a+i)+j));

}

}

void print1(int a[][10],int m ,int n)

{ int i,j;

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

{

printf("\n");

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

printf("%5d",a[i][j]);

}

}

void transpose(int \*\*a,int m ,int n)

{ int i,j,temp;

if(m==n)

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

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

{

temp=\*(\*(a+i)+j);

\*(\*(a+i)+j)=\*(\*(a+j)+i);

\*(\*(a+j)+i)=temp;

}

}

}

int \*\* addmat(int \*\*a,int m1,int n1 ,int \*\*b,int m2,int n2)

{ int i,j; int \*\*c;

c=(int\*\*)malloc(m1\*sizeof(int\*));

for(i=0;i<m1;i++)

\*(c+i)=(int\*)malloc(n1\*sizeof(int));

if(m1==m2 && n1==n2)

for(i=0;i<m1;i++)

for(j=0;j<n1;j++)

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

return(c);

}

void multmat(int a[][10],int m1,int n1 ,int b[][10],int m2,int n2,int c[][10])

{ int i,j,k,temp;

if(n1==m2)

{ for(i=0;i<m1;i++)

for(j=0;j<n2;j++)

{

temp=0;

for(k=0;k<n1;k++)

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

c[i][j]=temp;

}

}

}

int saddle(int a[][10],int m,int n)

{ int i,j,small,large,col\_of\_small,row\_of\_large;

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

{ small=a[i][0];

col\_of\_small=0;

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

if(a[i][j] < small)

{

small=a[i][j];

col\_of\_small=j;

}

large= a[0][col\_of\_small];

row\_of\_large=0;

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

if(a[j][col\_of\_small]>large)

{ large=a[j][col\_of\_small];

row\_of\_large=j;

}

if(i==row\_of\_large)

{

printf(" Saddle point exist at (%d,%d) with value as %d",i,col\_of\_small,a[i][col\_of\_small]);

return(1);

}

}

printf("Saddle point does not exist ");

return(0);

}

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

\*\*\*\*\*\*menu\*\*\*\*\*\*

1)Transpose of the Matrix

2)Add two matrices

3)Multiply two matrices

4)Saddle point

5)Quit

Enter Your Choice : 1

Enter the size of the matrix :2

2

Enter the data:5

1

3

4

Result=

5 3

1 4

\*\*\*\*\*\*menu\*\*\*\*\*\*

1)Transpose of the Matrix

2)Add two matrices

3)Multiply two matrices

4)Saddle point

5)Quit

Enter Your Choice : 2

Enter the size of the 1st matrix:2

2

Enter the data:3

1

5

6

Enter the size of the 2nd matrix:2

2

Enter the data:6

4

5

2

Result=

9 5

10 8

\*\*\*\*\*\*menu\*\*\*\*\*\*

1)Transpose of the Matrix

2)Add two matrices

3)Multiply two matrices

4)Saddle point

5)Quit

Enter Your Choice : 3

Enter the size of the 1st matrix:2

2

Enter the data:3

6

4

1

Enter the size of the 2nd matrix:2

2

Enter the data:5

3

8

1

Result=

63 15

28 13

\*\*\*\*\*\*menu\*\*\*\*\*\*

1)Transpose of the Matrix

2)Add two matrices

3)Multiply two matrices

4)Saddle point

5)Quit

Enter Your Choice : 4

Enter the size of the 1st matrix:2

2

Enter the data:3

5

4

6

Saddle point exist at (1,0) with value as 4

\*\*\*\*\*\*menu\*\*\*\*\*\*

1)Transpose of the Matrix

2)Add two matrices

3)Multiply two matrices

4)Saddle point

5)Quit

Enter Your Choice : 5