NAME: SATVIK DANDALE  
DIV: TY E  
BATCH: B1

GR NO: 1710797

#include<bits/stdc++.h>

using namespace std;

class Matrix

{

int \*\*row;

int r,c;

public:

Matrix()

{

r = 0;

c = 0;

}

Matrix(int nr,int nc)

{

r = nr;

c =nc;

row = (int \*\*)malloc(sizeof(int \*)\*r);

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

{

row[i] = (int \*)malloc(sizeof(int)\*c);

}

}

int get\_element(int r,int c)

{

return \*(\*(row+r)+c);

}

void set\_element(int r,int c,int value)

{

\*(\*(row+r)+c) = value;

}

void set\_size\_matrix(int &nr,int &nc)

{

nr = r;

nc = c;

}

void get\_data()

{

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

{

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

{

cin>>\*(\*(row+i)+j);

}

}

}

Matrix operator +(Matrix B)

{

int row1,col,value,e;

set\_size\_matrix(row1,col);

Matrix C(row1,col);

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

{

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

{

e = \*(\*(row+i)+j);

value = e +(B.get\_element(i,j));

C.set\_element(i,j,value);

}

}

return C;

}

Matrix operator -(Matrix B)

{

int row1,col,value,e;

set\_size\_matrix(row1,col);

Matrix C(row1,col);

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

{

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

{

e = \*(\*(row+i)+j);

value = e -(B.get\_element(i,j));

C.set\_element(i,j,value);

}

}

return C;

}

int IsSparse()

{

int count = 0,e;

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

{

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

{

e = \*(\*(row+i)+j);

if(e==0)

count++;

}

}

int n = r\*c;

if(count<(n/2))//NON-ZERO < HALF

{

return 0;

}

else

return 1;

}

Matrix ToCompact()

{

int count = 0,e;

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

{

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

{

e = \*(\*(row+i)+j);

if(e!=0)

count++;

}

}

Matrix B(3,count);

int i = 0,j = 0;

int k=0;

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

{

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

{

if(\*(\*(row+i)+j)!=0)

{

B.set\_element(0,k,i);

B.set\_element(1,k,j);

B.set\_element(2,k,\*(\*(row+i)+j));

k++;

}

}

}

return B;

}

void Display()

{

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

{

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

{

cout<<\*(\*(row+i)+j)<<" ";

}

cout<<"\n";

}

}

};

int main(){

char choice2;

int m,n;

printf("Enter 1st Matrix dimensions : 1. Rows 2. Columns");

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

Matrix Aa(m,n);

Aa.get\_data();

printf("Enter 2nd Matrix dimensions : 1. Rows 2. Columns");

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

Matrix Ba(m,n);

Ba.get\_data();

Matrix Ca = Aa+Ba;

Matrix C\_b = Aa-Ba;

do

{

printf("\nMATRIX : ");

printf("a. Addition\nb. Subtraction\nc. Issparse\nd. Exit");

scanf("%c",&choice2);

switch(choice2)

{

case 'a':

Ca.Display();

break;

case 'b':

//int m,n;

C\_b.Display();

break;

case 'c':

//int m,n;

if(Aa.IsSparse())

{

cout<<"Is sparse";

}

else

printf("Not a Sparse Matrix!!");

case 'd':

break;

}

}while(choice2!='d');

return 0;

}

