//To implement graph coloring problem solution

#include<iostream>

using namespace std;

int x[10];//solution vector

int g[10][10];//adjacency graph

int n=0;//the number of vertices

int m=0;//colors

void nextValue(int k)

{ int j;

while(1)

{

x[k] = (x[k]+1)%(m+1);//next highest color

if(x[k]==0)

return;

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

{

//check if this color is distinct from adjacent colors

if(g[k][j]==1&&x[k]==x[j])//if (k,j) is an edge and if adjacent vertices have the same color

break;

}

if(j==(n))//new color found

{

return;

}

}

}

void mcoloring(int k)

{

while(1)

{

//generate all legal assignments for x[k]

nextValue(k);

if(x[k]==0)

return;

if(k==n)

{

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

{

cout<<x[i]<<" ";

}

cout<<"\n";

}

else

mcoloring(k+1);

}

}

int main()

{

int i,j;

cout<<"Enter the number of vertices"<<endl;

cin>>n;

cout<<"Enter the adjacency matrix "<<endl;

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

{

cout<<"Enter the adjacency list of vertix "<<i+1<<endl;

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

{

cin>>g[i][j];

}

}

cout<<"Enter the m color"<<endl;

cin>>m;

i=0;

while(i<n)

{

x[i++] = 0;

}

mcoloring(0);

return 0;

}

OUTPUT

