Program 12

Design and implement in java to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking Principle.

import java.util.Scanner;

public class P12{

static int [] x = new int [25];

static void Next\_Vertex(int G[][],int n,int k)

{

int j;

while(true)

{

x[k]=(x[k]+1)%(n+1);

if(x[k]==0)

return;

if(G[x[k-1]][x[k]]!=0)

{

for(j=1;j<=k-1;j++)

{

if(x[j]==x[k])

break;

}

if(j==k)

{

if((k<n)||((k==n)&&(G[x[n]][x[1]]!=0)))

return;

}

}

}

}

static void H\_Cycle(int G[][],int n,int k)

{

int i;

while(true)

{

Next\_Vertex(G,n,k);

if(x[k]==0)

return;

if(k==n)

{

System.out.println("\n");

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

System.out.print(x[i] +"-->");

System.out.print(x[1]);

}

else

H\_Cycle(G,n,k+1);

}

}

public static void main(String[] args) {

int i,j,n;

int [][] G = new int [25][25];

Scanner read = new Scanner(System.in);

System.out.println("Enter the number of vertices of the graph");

n=read.nextInt();

System.out.println("Enter the Path adjacency matrix");

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

{

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

{

G[i][j]=read.nextInt();

x[i]=0;

}

}

x[1]=1;

System.out.println(" \n Hamiltonian Cycles are");

H\_Cycle(G,n,2);

}

}