Experiment No: 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.*;
public class ham {
	static int n,c=0;
	static int x[] = new int[10];
	static int g[][]= new int[10][10];
	public static void main(String[] args) {
	Scanner sc= new Scanner(System.in);
	int i,j;
	System.out.println("enter the number of vertices");
	n= sc.nextInt();
	System.out.println("enter the adjacency matrix");
	for(i=1;i<=n;i++)
	{
		x[i]=0;
		for(j=1;j<=n;j++)
			g[i][j]=sc.nextInt();
	}
	x[1]=1;
	hcycle(2);
	if(c==0)
		System.out.println("Hamiltonian Cycle doesnot exists");
	}
public static void  next_vertex(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]]==1)
		{
			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]]==1))
					return;
			}
		}


	}

}
public static void hcycle(int k)
{
	int i;
	while(true)
	{
		next_vertex(k);
		if(x[k]==0)return;
		if(k==n)
		{
			c++;
```

```
                        for(i=1;i<=n;i++)
                        System.out.print(x[i]+ "->");
                        System.out.println(x[1]);
                }
                else
                        hcycle(k+1);
                }
        }
}

/* OUTPUT

enter the number of vertices
4
enter the adjacency matrix
0 1 1 0
1 0 1 1
1 1 0 0
0 1 0 0
Hamiltonian Cycle does not exists


enter the number of vertices
4
enter the adjacency matrix
0 1 1 1
1 0 1 1
1 1 0 0
1 1 0 0
1->3->2->4->1
1->4->2->3->1
```