

RBU, Nagpur
CSE III Sem
PRACTICAL NO. 7 CP-Task

Name:	Kishan Saini
Sec-Batch-Rollno:	A4-B4-55

Aim: Competitive Programming Code Submission

Q) Given an undirected graph with n vertices and m edges, your task is to determine if a Hamiltonian path exists in the graph.

A Hamiltonian path is a path in an undirected graph that visits each vertex exactly once.

You are provided the following:

- n : The number of vertices in the graph.
- m : The number of edges in the graph.
- $edges[i][j]$: A 2D list where each element $edges[i]$ represents an edge between two vertices $edges[i][0]$ and $edges[i][1]$.

Code: class Solution:

```
def check(self, n, m, edges):  
    g = [[] for _ in range(n)]  
    for u, v in edges:  
        g[u - 1].append(v - 1)  
        g[v - 1].append(u - 1)  
  
    def dfs(u, visited, count):
```

```

if count == n: return
    True
for v in g[u]:
    if not visited[v]:
        visited[v] = True
        if dfs(v, visited, count + 1): return
            True
        visited[v] = False
return False

for i in range(n): visited
    = [False] * n
    visited[i] = True
    if dfs(i, visited, 1): return
        1
return 0

```

Submission:

Problem Solved Successfully 

[Suggest Feedback](#)

Test Cases Passed

52 / 52

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored 

4 / 4

Your Total Score: 4 

Time Taken

0.04