

```
def Hamiltonian(adj, N):
    Dp = [[False for i in range(1 << N)]
           for j in range(N)]
    for i in range(N):
        Dp[i][1 << i] = True
    for i in range(1 << N):
        for j in range(N):
            if ((i & (1 << j)) != 0):
                for k in range(N):
                    if ((i & (1 << k)) != 0 and adj[k][j] == 1 and j != k and Dp[k][i ^ (1 << j)]):
                        Dp[j][i] = True
                        break
    for i in range(N):
        if (Dp[i][(1 << N) - 1]):
            return True
    return False

adj = [[0, 1, 1, 1, 0],
        [1, 0, 1, 0, 1],
        [1, 1, 0, 1, 1],
        [1, 0, 1, 0, 0]]

N = len(adj)
if (Hamiltonian(adj, N)):
    print("hamiltonian is possible")
else:
    else
```

Project ▾

- fact non recursive.py
- fact recursive.py
- fib non recersive.py
- fib recursive.py
- floyds.py
- gcd non recursive.py
- gcd recursive.py
- hamiltonian.py
- knapsack.py
- lcm non recursive.py
- lcm recursive.py
- max and min.py
- max non recursive.py
- max recursive.py
- mergesort.py

```
15         return True
16     return False
17     adj=[[0,1,1,1,0],
18         [1,0,1,0,1],
19         [1,1,0,1,1],
20         [1,0,1,0,0]]
21     N=len(adj)
22     if (Hamiltonian(adj,N)):
23         print("hamiltonian is possible")
24     else:
25         print("hamiltonian is not possible")
26
else
```

hamiltonian (1) ×

```
C:\Users\kativ\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/kativ/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/hamil
hamiltonian is possible

Process finished with exit code 0
```

