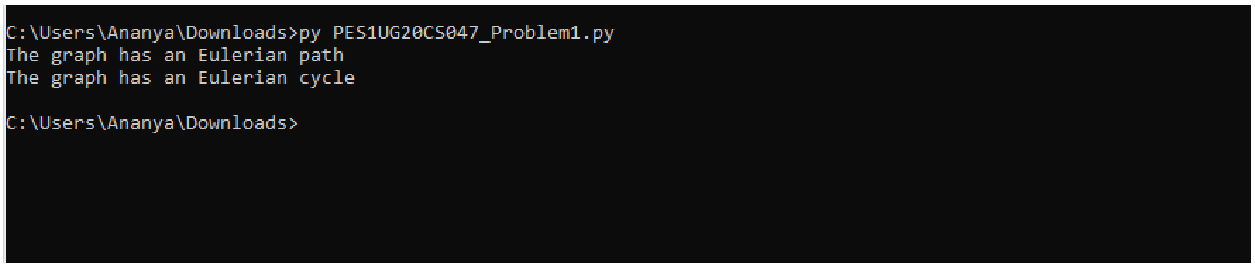
LAB 1

NAME : ANANYA SHASHISHEKAR
SRN : PES1UG20CS047
SECTION : A
PROBLEM STATEMENT : To find if the graph is Eulerian

CODE:

```
class Graph:
    def __init__(self, n, edges=[]):

        self.adjList = [[] for _ in range(n)]


        for edge in edges:
            self.addEdge(edge[0], edge[1])


    def addEdge(self, u, v):
        self.adjList[u].append(v)
        self.adjList[v].append(u)



def DFS(graph, v, discovered):
    discovered[v] = True


    for u in graph.adjList[v]:
        if not discovered[u]:
            DFS(graph, u, discovered)



def isConnected(graph, n):


    discovered = [False] * n


    for i in range(n):
        if len(graph.adjList[i]):
            DFS(graph, i, discovered)
            break


    for i in range(n):
        if not discovered[i] and len(graph.adjList[i]):
            return False

    return True



def countOddVertices(graph):
    count = 0
    for list in graph.adjList:
        if len(list) & 1:
            count += 1
    return count


if __name__ == '__main__':


    edges = [(0, 1), (0, 3), (1, 2), (1, 3), (1, 4), (2, 3), (3, 4)]


    n = 5

    graph = Graph(n, edges)


    is_connected = isConnected(graph, n)


    odd = countOddVertices(graph)


    if is_connected and (odd == 0 or odd == 2):

        print('The graph has an Eulerian path')


        if odd == 0:
            print('The graph has an Eulerian cycle')
        else:
            print('The Graph is Semi–Eulerian')
    else:
        print('The Graph is not Eulerian')
```

OUTPUT:



```
C:\Users\Ananya\Downloads>py PES1UG20CS047_Problem1.py
The graph has an Eulerian path
The graph has an Eulerian cycle

C:\Users\Ananya\Downloads>
```