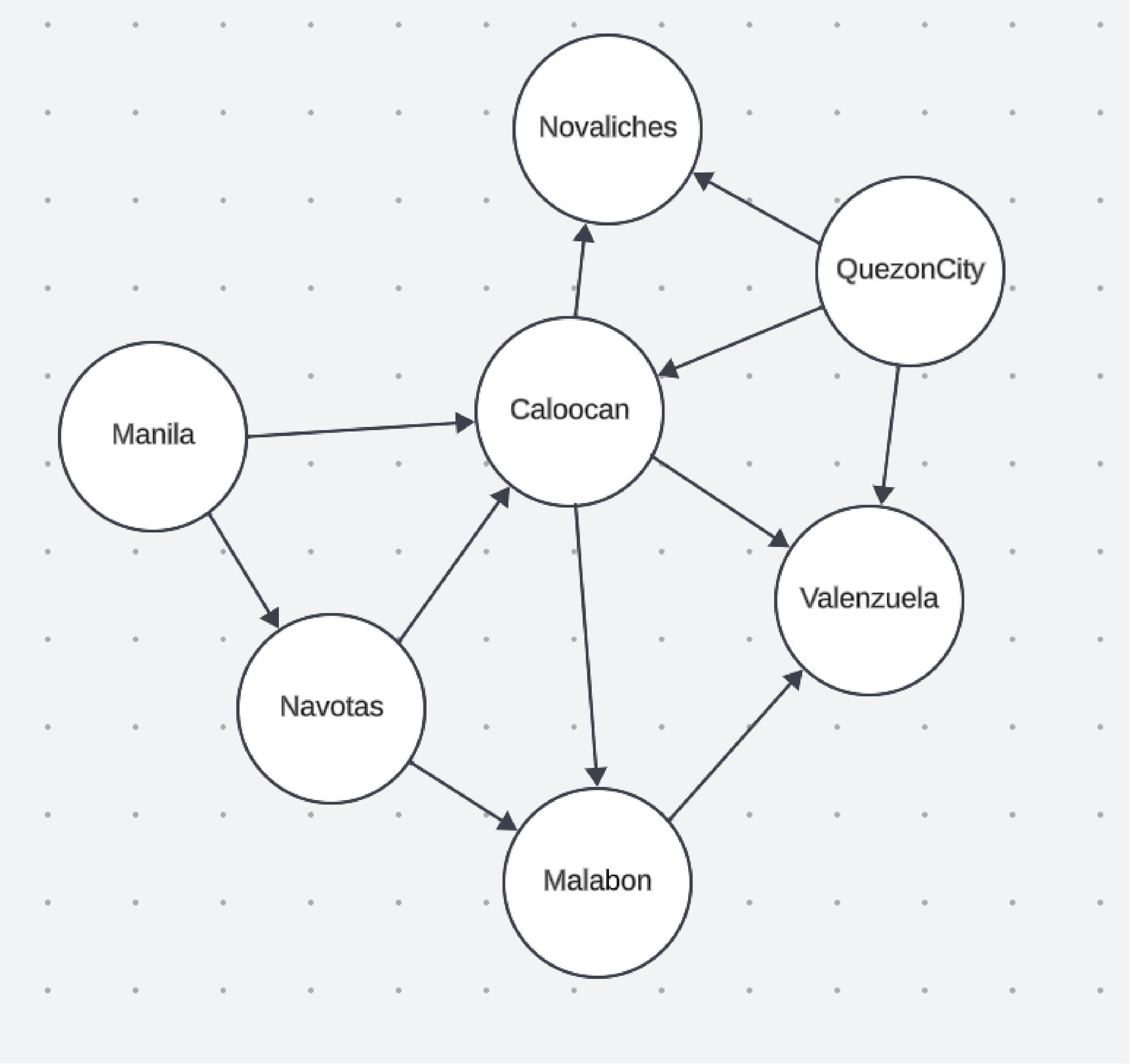
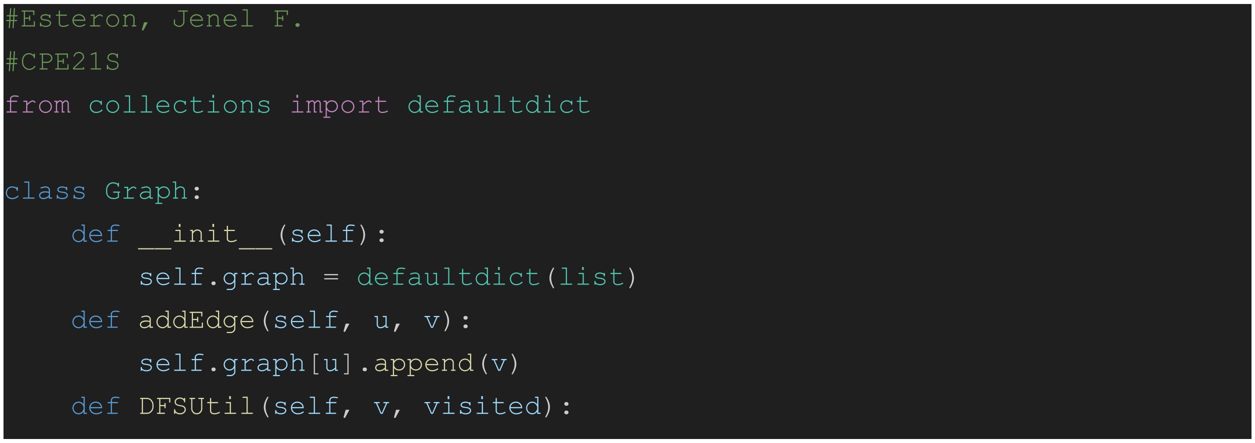
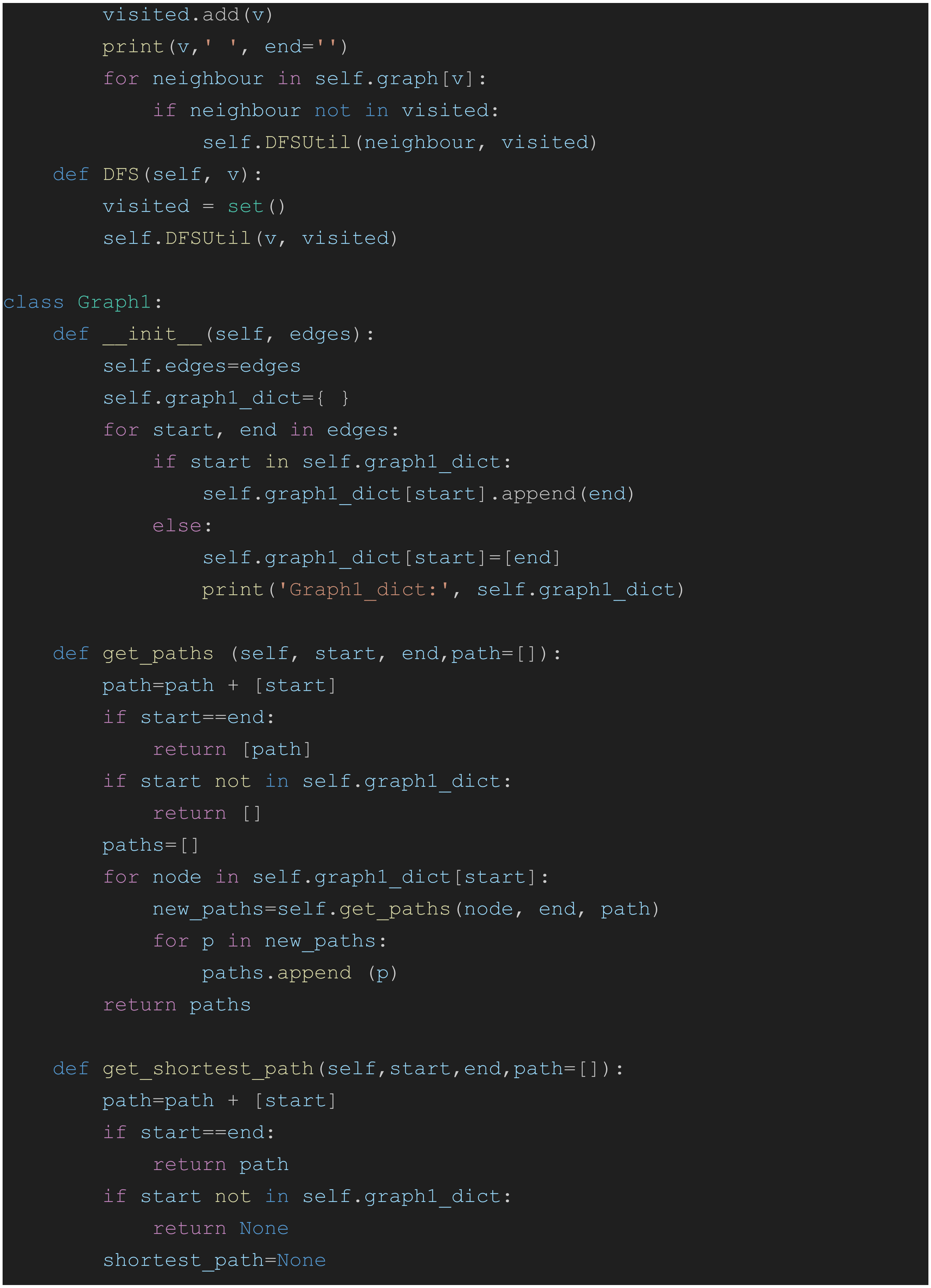
Esteron, Jenel F.
CPE21S1

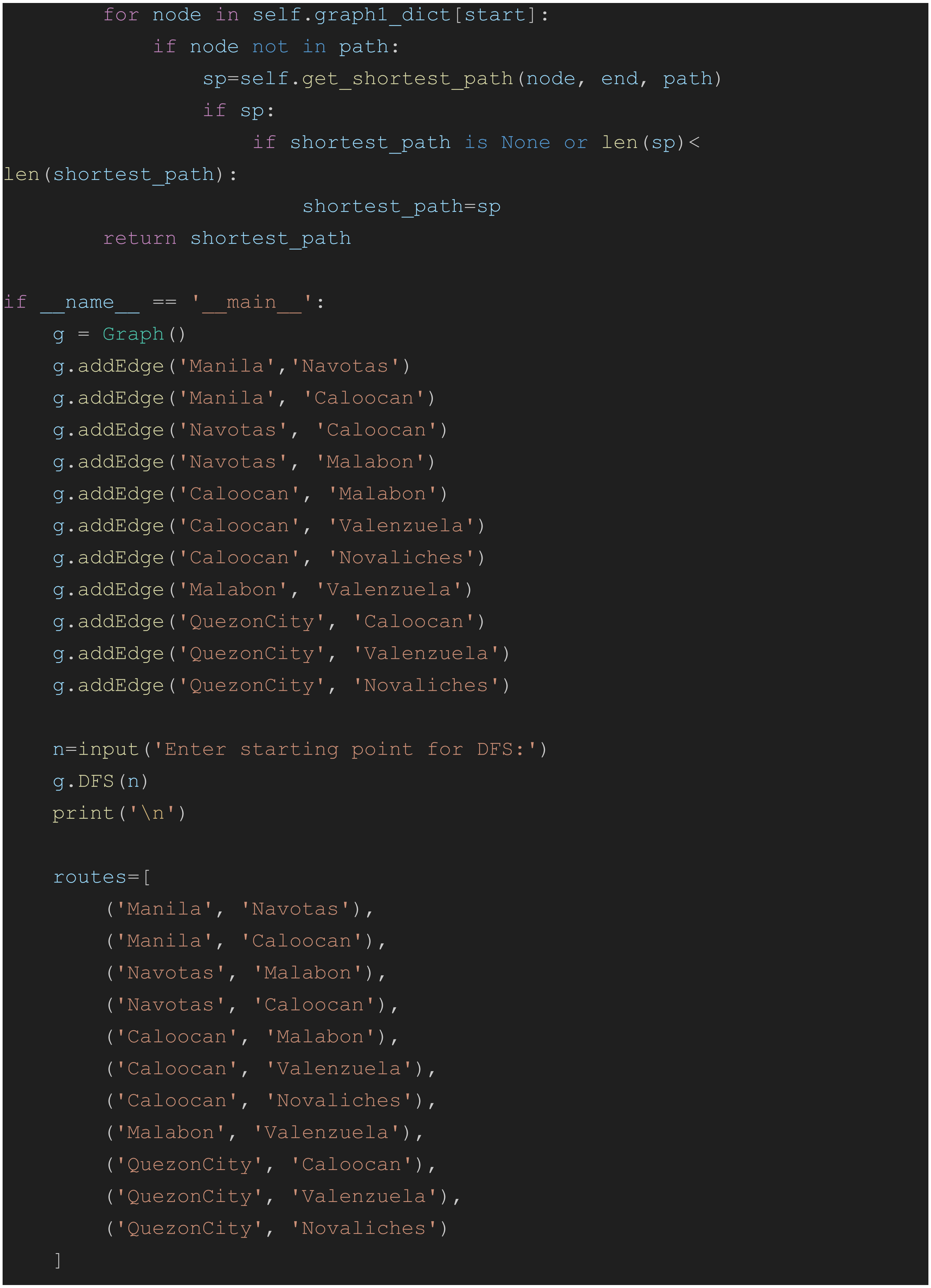




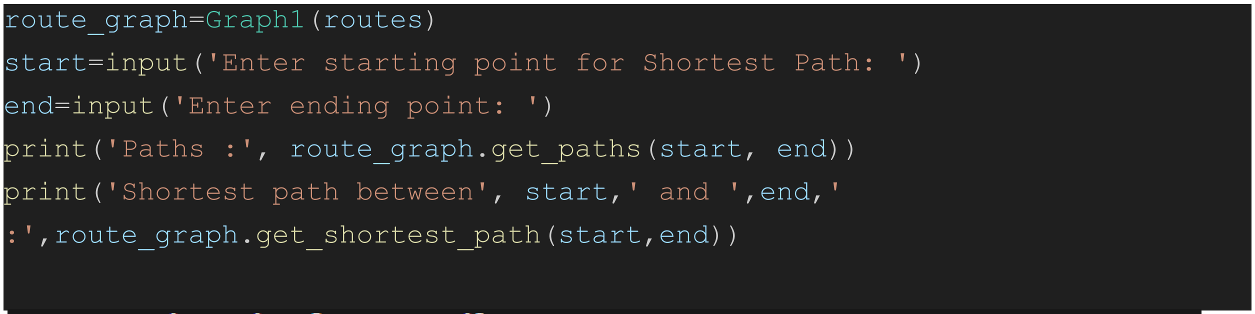
```
#Esteron, Jenel F.
#CPE21S
from collections import defaultdict

class Graph:
    def __init__(self):
        self.graph = defaultdict(list)
    def addEdge(self, u, v):
        self.graph[u].append(v)
    def DFSUtil(self, v, visited):
```



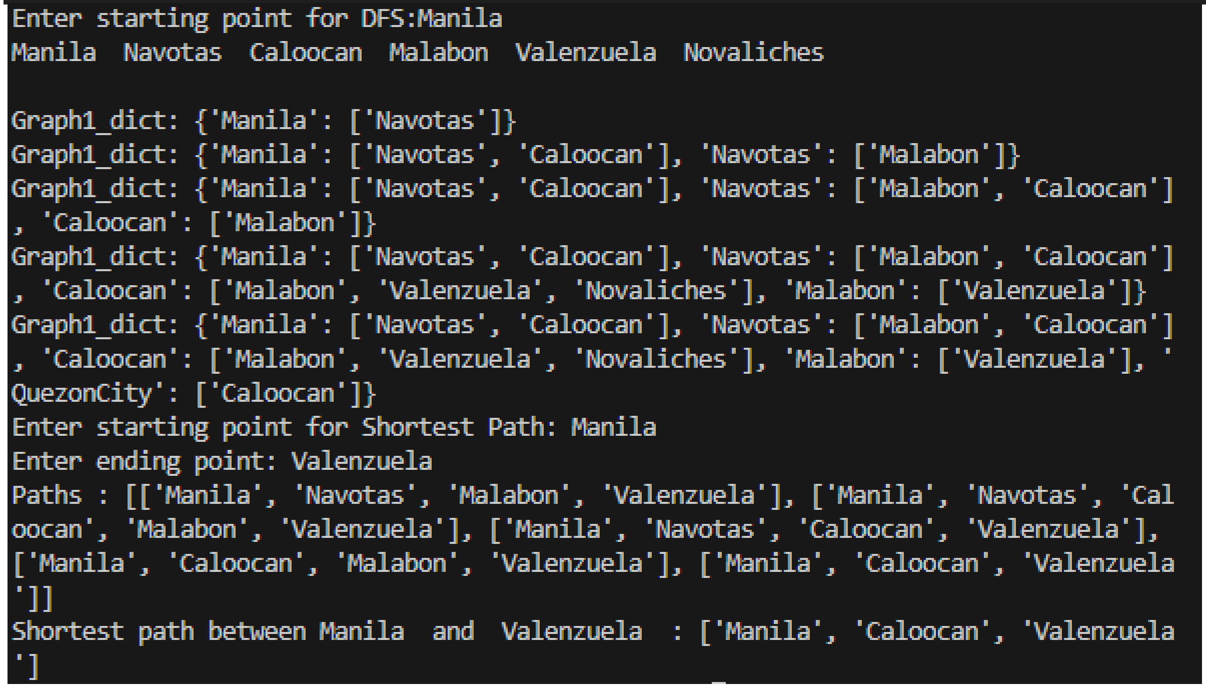
```python
        visited.add(v)
        print(v,' ', end='')
        for neighbour in self.graph[v]:
            if neighbour not in visited:
                self.DFSUtil(neighbour, visited)
    def DFS(self, v):
        visited = set()
        self.DFSUtil(v, visited)

class Graph1:
    def __init__(self, edges):
        self.edges=edges
        self.graph1_dict={ }
        for start, end in edges:
            if start in self.graph1_dict:
                self.graph1_dict[start].append(end)
            else:
                self.graph1_dict[start]=[end]
                print('Graph1_dict:', self.graph1_dict)

    def get_paths (self, start, end,path=[]):
        path=path + [start]
        if start==end:
            return [path]
        if start not in self.graph1_dict:
            return []
        paths=[]
        for node in self.graph1_dict[start]:
            new_paths=self.get_paths(node, end, path)
            for p in new_paths:
                paths.append (p)
        return paths

    def get_shortest_path(self,start,end,path=[]):
        path=path + [start]
        if start==end:
            return path
        if start not in self.graph1_dict:
            return None
        shortest_path=None
```



```
        for node in self.graph1_dict[start]:
            if node not in path:
                sp=self.get_shortest_path(node, end, path)
                if sp:
                    if shortest_path is None or len(sp)<
len(shortest_path):
                        shortest_path=sp
        return shortest_path

if __name__ == '__main__':
    g = Graph()
    g.addEdge('Manila','Navotas')
    g.addEdge('Manila', 'Caloocan')
    g.addEdge('Navotas', 'Caloocan')
    g.addEdge('Navotas', 'Malabon')
    g.addEdge('Caloocan', 'Malabon')
    g.addEdge('Caloocan', 'Valenzuela')
    g.addEdge('Caloocan', 'Novaliches')
    g.addEdge('Malabon', 'Valenzuela')
    g.addEdge('QuezonCity', 'Caloocan')
    g.addEdge('QuezonCity', 'Valenzuela')
    g.addEdge('QuezonCity', 'Novaliches')

    n=input('Enter starting point for DFS:')
    g.DFS(n)
    print('\n')

    routes=[
        ('Manila', 'Navotas'),
        ('Manila', 'Caloocan'),
        ('Navotas', 'Malabon'),
        ('Navotas', 'Caloocan'),
        ('Caloocan', 'Malabon'),
        ('Caloocan', 'Valenzuela'),
        ('Caloocan', 'Novaliches'),
        ('Malabon', 'Valenzuela'),
        ('QuezonCity', 'Caloocan'),
        ('QuezonCity', 'Valenzuela'),
        ('QuezonCity', 'Novaliches')
    ]
```



```
route_graph=Graph1(routes)
start=input('Enter starting point for Shortest Path: ')
end=input('Enter ending point: ')
print('Paths :', route_graph.get_paths(start, end))
print('Shortest path between', start,' and ',end,'
:',route_graph.get_shortest_path(start,end))
```



```
Enter starting point for DFS:Manila
Manila  Navotas  Caloocan  Malabon  Valenzuela  Novaliches

Graph1_dict: {'Manila': ['Navotas']}
Graph1_dict: {'Manila': ['Navotas', 'Caloocan'], 'Navotas': ['Malabon']}
Graph1_dict: {'Manila': ['Navotas', 'Caloocan'], 'Navotas': ['Malabon', 'Caloocan']
, 'Caloocan': ['Malabon']}
Graph1_dict: {'Manila': ['Navotas', 'Caloocan'], 'Navotas': ['Malabon', 'Caloocan']
, 'Caloocan': ['Malabon', 'Valenzuela', 'Novaliches'], 'Malabon': ['Valenzuela']}
Graph1_dict: {'Manila': ['Navotas', 'Caloocan'], 'Navotas': ['Malabon', 'Caloocan']
, 'Caloocan': ['Malabon', 'Valenzuela', 'Novaliches'], 'Malabon': ['Valenzuela'], '
QuezonCity': ['Caloocan']}
Enter starting point for Shortest Path: Manila
Enter ending point: Valenzuela
Paths : [['Manila', 'Navotas', 'Malabon', 'Valenzuela'], ['Manila', 'Navotas', 'Cal
oocan', 'Malabon', 'Valenzuela'], ['Manila', 'Navotas', 'Caloocan', 'Valenzuela'],
['Manila', 'Caloocan', 'Malabon', 'Valenzuela'], ['Manila', 'Caloocan', 'Valenzuela
']]
Shortest path between Manila  and  Valenzuela  : ['Manila', 'Caloocan', 'Valenzuela
']
```