Neha Dattatray Bhoite

Assignment No 1

1) DFS

```
class Graph:
    def __init__(self):
        self.graph = {}

    def add_edge(self, u, v):
        if u not in self.graph:
            self.graph[u] = []
        self.graph[u].append(v)

    def dfs_util(self, node, visited):
        visited.add(node)
        print(node, end=' ')

        for neighbor in self.graph.get(node, []):
            if neighbor not in visited:
                self.dfs_util(neighbor, visited)

    def dfs(self, start_node):
        visited = set()
        self.dfs_util(start_node, visited)

# Example usage:
g = Graph()
g.add_edge(0, 1)
g.add_edge(0, 2)
g.add_edge(1, 2)
g.add_edge(2, 0)
g.add_edge(2, 3)
g.add_edge(3, 3)

print("DFS starting from node 2:")
g.dfs(2)
```

Output-

```
DFS starting from node 2:
2 0 1 3
=== Code Execution Successful ===
```

2)BFS

```
from collections import deque

class Graph:
    def __init__(self):
        self.graph = {}

    def add_edge(self, u, v):
        if u not in self.graph:
            self.graph[u] = []
        self.graph[u].append(v)

    def bfs(self, start_node):
        visited = set()
        queue = deque([start_node])

        while queue:
            node = queue.popleft()
            if node not in visited:
                print(node, end=' ')
                visited.add(node)

                for neighbor in self.graph.get(node, []):
                    if neighbor not in visited:
                        queue.append(neighbor)

# Example usage:
g = Graph()
g.add_edge(0, 1)
g.add_edge(0, 2)
g.add_edge(1, 2)
g.add_edge(2, 0)
g.add_edge(2, 3)
g.add_edge(3, 3)
print("BFS starting from node 2:")
g.bfs(2)
```

Output-

```
BFS starting from node 2:
2 0 3 1
=== Code Execution Successful ===
```