Problem – 4 Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not. (using BFS)

```
from collections import defaultdict, deque


class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = defaultdict(list)

    def add_edge(self, u, v):
        self.graph[u].append(v)
        self.graph[v].append(u)

    def is_cyclic_util(self, v, visited, parent):
        queue = deque([(v, parent)])
        visited[v] = True

        while queue:
            current_node, parent = queue.popleft()

            for neighbor in self.graph[current_node]:
                if not visited[neighbor]:
                    queue.append((neighbor, current_node))
                    visited[neighbor] = True
                elif parent != neighbor:
                    return True

        return False

    def contains_cycle(self):
        visited = [False] * self.V
```

```
        for v in range(self.V):
            if not visited[v]:
                if self.is_cyclic_util(v, visited, -1):
                    return True

        return False

V = 4
E = 4
g = Graph(V)

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

for u, v in edges:
    g.add_edge(u, v)

if g.contains_cycle():
    print("The graph contains a cycle.")
else:
    print("The graph does not contain any cycle.")
```