Problem 4: Given an undirected graph and a number m, determine ifthe graph can be colored with at most m colors such that no two adjacent vertices of the graph are colored with the same color.

```
def graphColoringUtil(graph, m, colors, v):
    if v == len(graph):
        return True
    for c in range(1, m + 1):
        if isSafe(graph, colors, v, c):
            colors[v] = c
            if graphColoringUtil(graph, m, colors, v + 1):
                return True
            colors[v] = -1

    return False

def isSafe(graph, colors, v, c):
    for u in graph[v]:
        if colors[u] == c:
            return False
    return True

def graphColoring(N, M, Edges):
    graph = [[] for _ in range(N)]
    for u, v in Edges:
        graph[u].append(v)
        graph[v].append(u)

    colors = [-1] * N
    if graphColoringUtil(graph, M, colors, 0):
        return 1
    return 0

N = 4
```

M = 3

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

print(graphColoring(N, M, Edges))