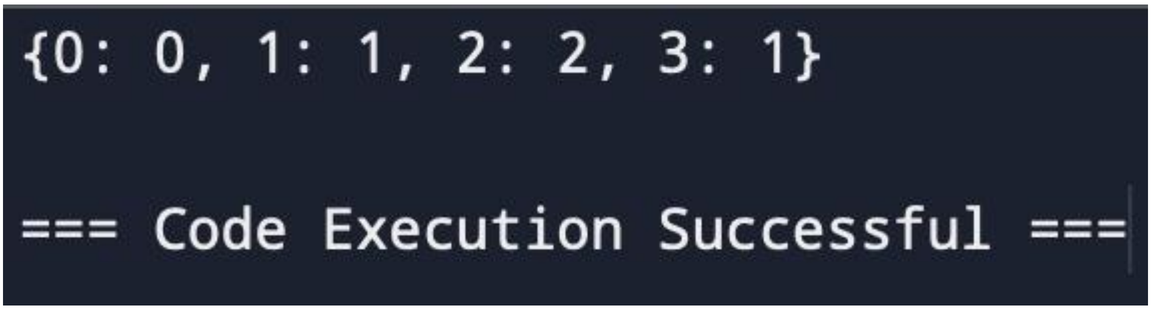
217. You and your friends are tasked with coloring a map using a limited set of colors, with the following rules: At each step, you can choose any region of the map that hasn't been colored yet and color it with any available color. Your friend Alice will then color the next region using the same strategy, followed by your friend Bob. You aim to maximize the number of regions you color. Given a map represented as a list of regions and their adjacency relationships, write a function to determine the maximum number of regions you can color. Write a program to implement the Graph coloring technique for an undirected graph. Implement an algorithm with minimum number of colors. edges = [(0, 1), (1, 2), (2, 3), (3, 0), (0, 2)] No. of vertices, n = 4, k = 3

PROGRAM:-

```
def graph_coloring(edges, n, k):
    graph = {}
    colors = {}

    for u, v in edges:
        if u not in graph:
            graph[u] = []
        if v not in graph:
            graph[v] = []
        graph[u].append(v)
        graph[v].append(u)

    for node in graph:
        neighbor_colors = set()
        for neighbor in graph[node]:
            if neighbor in colors:
                neighbor_colors.add(colors[neighbor])

        for color in range(k):
            if color not in neighbor_colors:
                colors[node] = color
                break

    return colors

# Example Usage
edges = [(0, 1), (1, 2), (2, 3), (3, 0), (0, 2)]
n = 4
k = 3
result = graph_coloring(edges, n, k)
print(result)
```

OUTPUT:-



```
{0: 0, 1: 1, 2: 2, 3: 1}

=== Code Execution Successful ===
```

TIME COMPLEXITY:- $O(n^2)$