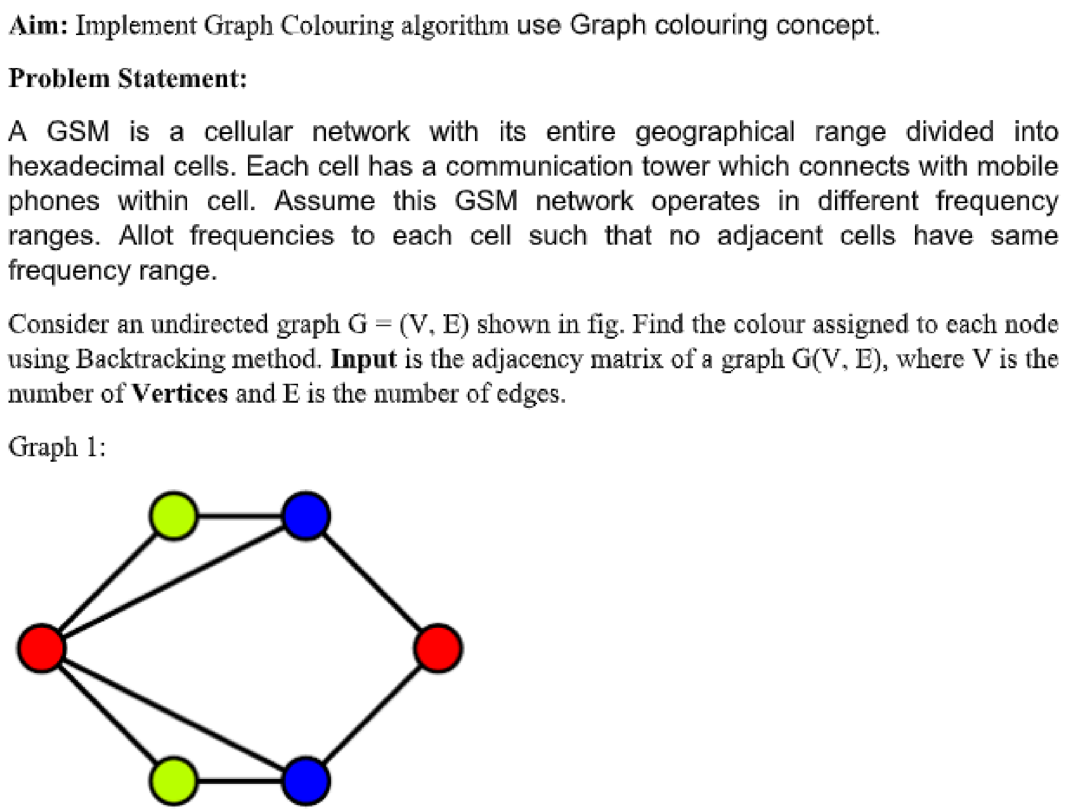
Practical 08

Name:- Mohammad Anas Ajaz

Batch : A4 - B4 - 61

DAA Practical 8

Aim:-



Aim: Implement Graph Colouring algorithm use Graph colouring concept.

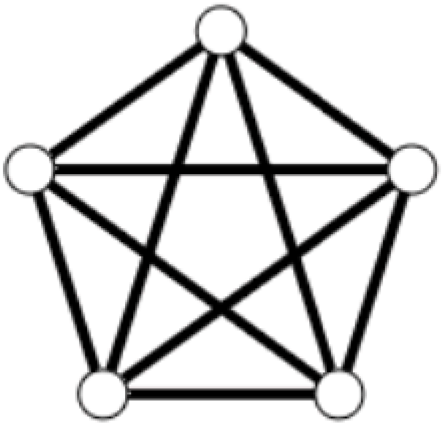
Problem Statement:

A GSM is a cellular network with its entire geographical range divided into hexadecimal cells. Each cell has a communication tower which connects with mobile phones within cell. Assume this GSM network operates in different frequency ranges. Allot frequencies to each cell such that no adjacent cells have same frequency range.

Consider an undirected graph G = (V, E) shown in fig. Find the colour assigned to each node using Backtracking method. **Input** is the adjacency matrix of a graph G(V, E), where V is the number of **Vertices** and E is the number of edges.

Graph 1:

Graph 2:

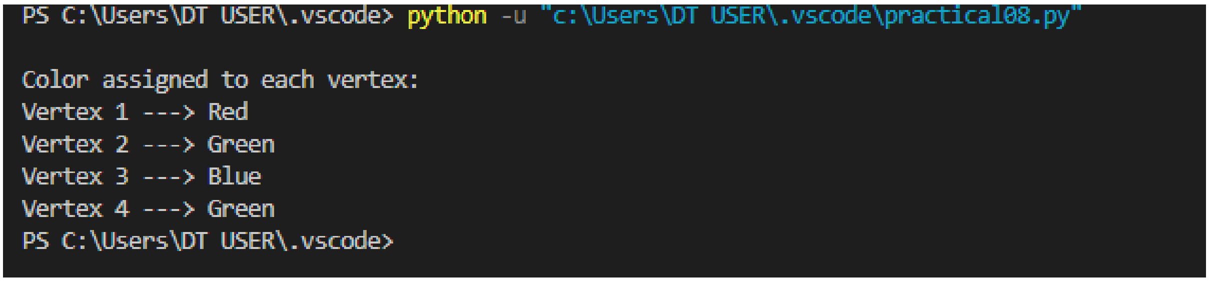


Code:-

```
def is_safe(graph, colors, v, c):
    for i in range(len(graph)):
        if graph[v][i] == 1 and colors[i] == c:
            return False
    return True
def graph_coloring(graph, color_names, colors, v):
    if v == len(graph):
        print("\nColor assigned to each vertex:")
        for i in range(len(colors)):
            print(f"Vertex {i + 1} ---> {colors[i]}")
        return True
    for c in color_names:
        if is_safe(graph, colors, v, c):
            colors[v] = c
            if graph_coloring(graph, color_names, colors, v + 1):
                return True
            colors[v] = None
    return False
graph = [
    [0, 1, 1, 1],
    [1, 0, 1, 0],
    [1, 1, 0, 1],
    [1, 0, 1, 0]
]
color_names = ["Red", "Green", "Blue", "Yellow"]
```

```
colors = [None] * len(graph)
if not graph_coloring(graph, color_names, colors, 0):
    print("No solution exists")
```

Output:-



```
PS C:\Users\DT USER\.vscode> python -u "c:\Users\DT USER\.vscode\practical08.py"

Color assigned to each vertex:
Vertex 1 ---> Red
Vertex 2 ---> Green
Vertex 3 ---> Blue
Vertex 4 ---> Green
PS C:\Users\DT USER\.vscode>
```