Name: Sriram Gopalkrishnan Iyer
Section: A3
Batch: B3
Roll No: 38

Shri Ramdeobaba College of Engineering and Management, Nagpur
Department of Computer Science and Engineering
Session: 2023-2024

Design and Analysis of Algorithms Lab **V Semester**

PRACTICAL NO. 8

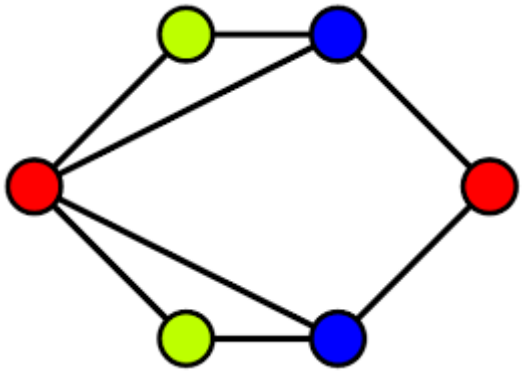
Aim: Implement Graph Colouring algorithm use Graph colouring concept.

Problem Statement:

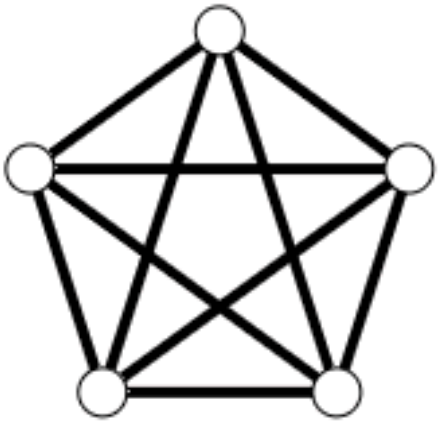
A GSM is a cellular network with its entire geographical range divided into hexadecimal cells. Each cell has a communication tower which connects with mobile phones within cell. Assume this GSM network operates in different frequency ranges. Allot frequencies to each cell such that no adjacent cells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

```
#include <stdio.h>
#include <stdbool.h>

#define MAX 20

int graph[MAX][MAX];
int color[MAX];
int V;
int m;

bool isSafe(int v, int c) {
   for (int i = 0; i < V; i++) {
       if (graph[v][i] && color[i] == c)
           return false;
   }
```

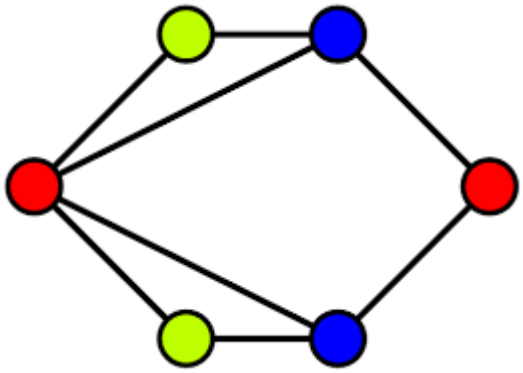
```
    return true;
}

bool graphColoringUtil(int v) {
    if (v == V)
        return true;

    for (int c = 1; c <= m; c++) {
        if (isSafe(v, c)) {
            color[v] = c;

            if (graphColoringUtil(v + 1))
                return true;

            color[v] = 0;
        }
    }
    return false;
}

bool graphColoring() {
    for (int i = 0; i < V; i++)
        color[i] = 0;

    if (!graphColoringUtil(0)) {
```

```
        printf("\nSolution does Not Exist.\n");
        return false;
    }

    printf("\nColor Assigned to Each Vertex:\n");
    for (int i = 0; i < V; i++)
        printf("Vertex %d ---> Colour %d\n", i + 1, color[i]);
    return true;
}

int main() {
    printf("Enter the Number of Vertices: ");
    scanf("%d", &V);

    printf("Enter the Adjacency Matrix (%d x %d):\n", V, V);
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            scanf("%d", &graph[i][j]);
        }
    }

    printf("Enter the Number of Colors Available: ");
    scanf("%d", &m);

    graphColoring();
```

```
    return 0;

}
```

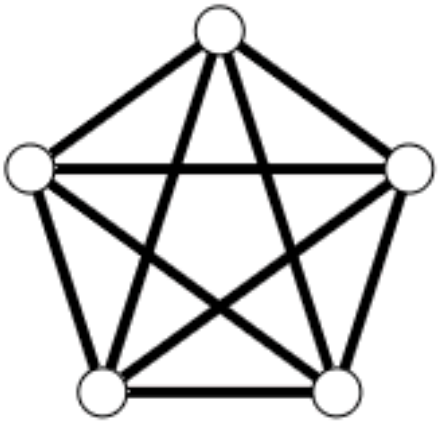
Output:

Graph 1:



```
Enter the Number of Vertices: 6
Enter the Adjacency Matrix (6 x 6):
0 1 1 0 1 1
1 0 1 0 0 0
1 1 0 1 0 0
0 0 1 0 1 0
1 0 0 1 0 1
1 0 0 0 1 0
Enter the Number of Colors Available: 3

Color Assigned to Each Vertex:
Vertex 1 ---> Colour 1
Vertex 2 ---> Colour 2
Vertex 3 ---> Colour 3
Vertex 4 ---> Colour 1
Vertex 5 ---> Colour 2
Vertex 6 ---> Colour 3
```

Graph 2:



```
Enter the Adjacency Matrix (5 x 5):
0 1 1 1 1
1 0 1 1 1
1 1 0 1 1
1 1 1 0 1
1 1 1 1 0
Enter the Number of Colors Available: 5

Color Assigned to Each Vertex:
Vertex 1 ---> Colour 1
Vertex 2 ---> Colour 2
Vertex 3 ---> Colour 3
Vertex 4 ---> Colour 4
Vertex 5 ---> Colour 5
```

GFG Output: