

# 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.

## CODE:

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

#define MAX 10

int graph[MAX][MAX];
int color[MAX];
int n, m; // n = number of vertices, m = number of colors

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

```
}
```

```
bool graphColoring(int v) {  
    if (v == n)  
        return true;  
  
    for (int c = 1; c <= m; c++) {  
        if (isSafe(v, c)) {  
            color[v] = c;  
            if (graphColoring(v + 1))  
                return true;  
            color[v] = 0;  
        }  
    }  
  
    return false;  
}
```

```
void inputGraph1() {  
    n = 4;  
    int g1[4][4] = {  
        {0,1,1,1},  
        {1,0,1,0},  
        {1,1,0,1},  
        {1,0,1,0}  
    };  
    for(int i=0;i<n;i++)  
        for(int j=0;j<n;j++)  
            graph[i][j] = g1[i][j];  
}
```

```

void inputGraph2() {
    n = 5;
    int g2[5][5] = {
        {0,1,1,0,1},
        {1,0,1,1,0},
        {1,1,0,1,0},
        {0,1,1,0,1},
        {1,0,0,1,0}
    };
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            graph[i][j] = g2[i][j];
}

int main() {
    int choice;
    printf("Choose Graph:\n1. Graph 1\n2. Graph 2\nEnter choice: ");
    scanf("%d", &choice);

    if(choice == 1)
        inputGraph1();
    else if(choice == 2)
        inputGraph2();
    else {
        printf("Invalid choice!");
        return 0;
    }

    printf("Enter number of colors: ");
    scanf("%d", &m);
}

```

```

for (int i = 0; i < n; i++)
    color[i] = 0;

if (graphColoring(0)) {
    printf("\nSolution exists:\n");
    for (int i = 0; i < n; i++)
        printf("Vertex %d ---> Color %d\n", i + 1, color[i]);
} else {
    printf("No solution exists using %d colors.\n", m);
}

return 0;
}

```

## OUTPUT:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Python Lab Project> cd "d:\Python Lab Project\" ; if ($?) { gcc new.c -o new } ; if ($?) { .\new }

Choose Graph:
1. Graph 1
2. Graph 2
Enter choice: 2
Enter number of colors: 3

Solution exists:
Vertex 1 ---> Color 1
Vertex 2 ---> Color 2
Vertex 3 ---> Color 3
Vertex 4 ---> Color 1
Vertex 5 ---> Color 2
PS D:\Python Lab Project>

```