PRACTICAL NO : 08

Name : Soham Adgokar

Roll No : A2-B2-28

Date : 28/10/25

Aim: Implement Graph Colouring algorithm use Graph colouring concept.

Problem Statement:

A GSM is a cellular network with its entire geo-graphical 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>
#define V 5
int graph[V][V] = {
   {0, 1, 1, 1, 0},
   {1, 0, 1, 0, 1},
   {1, 1, 0, 1, 1},
   {1, 0, 1, 0, 1},
   {0, 1, 1, 1, 0}
};
int color[V];
int isSafe(int v, int graph[V][V], int color[], int c) {
   for (int i = 0; i < V; i++)
      if (graph[v][i] && c == color[i])
         return 0;
   return 1;
}
int graphColoringUtil(int graph[V][V], int m, int color[], int v) {
   if (v == V)
      return 1;
   for (int c = 1; c <= m; c++) {
      if (isSafe(v, graph, color, c)) {
```

```
            color[v] = c;
            if (graphColoringUtil(graph, m, color, v + 1))
               return 1;
            color[v] = 0;
         }
      }
      return 0;
}
int graphColoring(int graph[V][V], int m) {
      for (int i = 0; i < V; i++)
         color[i] = 0;
      if (!graphColoringUtil(graph, m, color, 0)) {
         printf("Solution does not exist\n");
         return 0;
      }
      printf("Assigned Colours:\n");
      for (int i = 0; i < V; i++)
         printf("Vertex %d ---> Colour %d\n", i + 1, color[i]);
      return 1;
}
int main() {
      int m = 3;
```

```
    graphColoring(graph, m);
    return 0;
}
```

Output :

```
Assigned Colours:
Vertex 1 ---> Colour 1
Vertex 2 ---> Colour 2
Vertex 3 ---> Colour 3
Vertex 4 ---> Colour 2
Vertex 5 ---> Colour 1


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