Name:-Suryakant Upadhyay

PRN:-20220802043

Roll No.:-A20

Practical Assignment No. 5

1. Write a c to implement graph colouring problem using backtracking method

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

#define V 4  // Number of vertices in the graph

// Function to check if it's safe to assign a color to a vertex
bool 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 false;
      }
   }
   return true;
}

// Recursive function to solve the graph coloring problem
bool graphColoringUtil(int graph[V][V], int m, int color[], int v) {
   if (v == V) {
      return true; // All vertices are colored
   }

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

```
            if (graphColoringUtil(graph, m, color, v + 1)) {
                return true;
            }

            color[v] = 0; // Backtrack
        }
    }

    return false;
}

// Main function to solve the graph coloring problem
bool graphColoring(int graph[V][V], int m) {
    int color[V];
    for (int i = 0; i < V; i++) {
        color[i] = 0;
    }

    if (!graphColoringUtil(graph, m, color, 0)) {
        printf("Solution does not exist");
        return false;
    }

    printf("Solution exists: Following are the assigned colors:\n");
    for (int i = 0; i < V; i++) {
        printf("Vertex %d --> Color %d\n", i, color[i]);
    }

    return true;
}

int main() {
    int graph[V][V] = {
        {0, 1, 1, 1},
        {1, 0, 1, 0},
```

```
        {1, 1, 0, 1},
        {1, 0, 1, 0}
    };
    int m = 3; // Number of colors

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

Output:

```
/tmp/R4yXNOKj3E.o
Solution exists: Following are the assigned
    colors:
Vertex 0 --> Color 1
Vertex 1 --> Color 2
Vertex 2 --> Color 3
Vertex 3 --> Color 2
```