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

#define V 4 // Number of vertices (cells)

// Function to check if the current color assignment is safe
bool isSafe(int v, bool graph[V][V], int color[], int c) {
    for (int i = 0; i < V; i++)
        if (graph[v][i] && color[i] == c)
            return false;
    return true;
}

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

    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;
}

// Function to solve the graph coloring problem
void graphColoring(bool graph[V][V], int m) {
    int color[V] = {0};

    if (!graphColoringUtil(graph, m, color, 0)) {
```

```
        printf("Solution does not exist.\n");
        return;
    }

    printf("Assigned frequencies (colors):\n");
    for (int i = 0; i < V; i++)
        printf("Cell %d ---> Frequency %d\n", i, color[i]);
}

int main() {
    // Example adjacency matrix for 4 cells
    bool graph[V][V] = {
        {0, 1, 1, 1},
        {1, 0, 1, 0},
        {1, 1, 0, 1},
        {1, 0, 1, 0}
    };

    int m = 3; // Number of frequencies (colors)
    graphColoring(graph, m);

    return 0;
}
```

OUTPUT

```
Assigned frequencies (colors):
Cell 0 ---> Frequency 1
Cell 1 ---> Frequency 2
Cell 2 ---> Frequency 3
Cell 3 ---> Frequency 2
The program 'C:\Users\Dell\javascript\PRACT8.exe' has exited with code 0 (0x00000000).
```