

## 5) M - COLORING PROBLEM.

**LINK FOR EXPLANATION:** [📺 L16. M-Coloring Problem | Backtracking](#)

**MY CODE WHICH SHOWED ERROR:**

```
vector<int> isPossible(bool graph[101][101],int vertex,int m,vector<int> &vertexColors){
    vector<int> colorsOccupied(m+1);
    for(int i=0;i<vertexColors.size()-1;i++){
        if(graph[vertex-1][i]==1 && vertexColors[i+1]!=0){
            colorsOccupied[vertexColors[i+1]]=1;
        }
    }
    vector<int> colorsLeft;
    for(int i=1;i<m+1;i++){
        if(colorsOccupied[i]==0){
            colorsLeft.push_back(i);
        }
    }
    return colorsLeft;
}

bool fun(bool graph[101][101],int vertex,int m,vector<int> &vertexColors){
    if(vertex==vertexColors.size()){
        return true;
    }
    else{
        vector<int> colorsLeft=isPossible(graph,vertex,m,vertexColors);
        if(colorsLeft.size()==0){
            return false;
        }
        else{
            for(int i=0;i<colorsLeft.size();i++){
                vertexColors[vertex]=colorsLeft[i];
                bool status=fun(graph,vertex+1,m,vertexColors);
                if(status==true){
                    return true;
                }
                else{
                    vertexColors[vertex]=0;
                }
            }
        }
    }
}

bool graphColoring(bool graph[101][101], int m, int v)
{
    vector<int> vertexColors(v+1);
    return fun(graph,1,m,vertexColors);
}
```

## CORRECT CODE AFTER EXPLANATION:

```
bool isSafe(bool graph[101][101],vector<int>vertexColors,int current,int color,int v){
    for(int i=0;i<v;i++){
        if(graph[current-1][i]==1 && vertexColors[i+1]==color){
            return false;
        }
    }
    return true;
}

bool func(bool graph[101][101],vector<int>vertexColors,int current,int v,int m){
    if(current==v+1){
        return true;
    }
    else{
        for(int i=1;i<=m;i++){
            if(isSafe(graph,vertexColors,current,i,v)==true){
                vertexColors[current]=i;
                bool status=func(graph,vertexColors,current+1,v,m);
                if(status==true){
                    return true;
                }
            }
            else{
                vertexColors[current]=0;
            }
        }
    }
    return false;
}

bool graphColoring(bool graph[101][101], int m, int v){
    vector<int> vertexColors(v+1);
    return func(graph,vertexColors,1,v,m);
}
```