INDEX

NAME: Pulkit Shringi STD.: ______ SEC.: I1 ROLL NO.: 596 SUB.: Artificial Intelligence

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1	22/1/24	Implementation of toy problem.	10	[signature] 31/01/24
2	31/1/24	Implementation of real world problem	10	

classmate
Date 31/1/24
Page

Ex-2

Implementation of real world problem.

Graph coloring

- Aim:

Implementing a real world problem using graph coloring with graph that consists minimum 6 nodes and 3 colors.

- Algorithm:

This problem uses a backtracking algorithm to solve the graph coloring problem.

- Start with the first vertex (vertex 0).
- Try assigning a color to the current vertex.
- Check if the assigned color is safe for the current vertex.
- If the color assignment is not safe, backtrack and try a different color for the current vertex.
- Repeat steps 2 to 5 until all vertices are colored or until it's not possible to assign colors without conflicts.
- If all vertices are successfully colored, print the solution.

- Code:

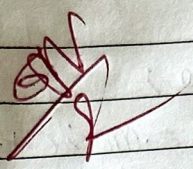
```
#include <bits/stdc++.h>
using namespace std;
#define V 6
void printSolution(int color[]);
```

```
bool isSafe ( int v, bool 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 ;
}
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) == true)
    return true ;
color[v] = 0 ;
    }
}
    return false ;
}
bool graphColoring (bool graph[V][V],
        int m) {
    int color[V] ;
    for (int i = 0 ; i < V ; i++)
        color[i] = 0 ;
if (graphColoringUtil (graph, m, color, 0)
    == false) {
    cout << "solution does not exist" ;
    return false ;
}
```

- Output :

Solution Exists : Following are the assigned colors
Vertex 0 : color 1
Vertex 1 : color 2
Vertex 2 : color 3
Vertex 3 : color 2
Vertex 4 : color 3
Vertex 5 : color 2.



```
printSolution(color);
return true;
}
void printSolution(int color[])
{
    cout << "Solution Exists:"
         << "Following are the assigned colors"
         << "\n";
for (int i = 0; i < V; i++)
    cout << " " << color[i] << " ";
cout << "\n";
int main()
{
    bool graph[V][V] = {
        {0, 1, 1, 1, 0, 1},
        {1, 0, 1, 0, 1, 0},
        {1, 1, 0, 1, 1, 0},
        {1, 1, 0, 1, 1, 0},
        {1, 0, 1, 0, 1, 0},
    };
    int m = 3;
graphColoring(graph, m);
    return 0;
}
```

Result:

Real world problem of graph coloring was implemented using backtracking.