EXPERIMENT NO. 10

Roll No.: 24141001
Batch: I1

Title: Graph Colouring Problem Using Backtracking.

Objectives:

- To understand how to colour a graph such that no two adjacent vertices have the same colour.
- To learn and apply backtracking to solve constraint satisfaction problems.

Theory:

The Graph Colouring Problem assigns colours to all vertices of a graph such that:

No two adjacent vertices share the same colour.

Only M colours can be used.

This is known as the M-Colouring Problem.

Why Backtracking?

Graph colouring involves trying different colour assignments.
Backtracking helps by:

1. Assigning a colour to a vertex.
2. Checking if it's safe (i.e., no neighbour has the same colour).
3. If safe → continue.
4. If not → undo the choice and try the next colour.

This systematically explores only valid colour combinations.

Program:

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

```
#define MAX 20
int graph[MAX][MAX];
int color[MAX];
int n, m;
int isSafe(int v, int c) {
    for (int i = 0; i < n; i++)
        if (graph[v][i] == 1 && color[i] == c)
            return 0;
    return 1;
}
int graphColoring(int v) {
    if (v == n)
        return 1;
    for (int c = 1; c <= m; c++) {
        if (isSafe(v, c)) {
            color[v] = c;
            if (graphColoring(v + 1))
                return 1;
            color[v] = 0;  // backtrack
        }
    }
    return 0;
}
int main() {
    printf("Enter number of vertices: ");
```

```
    scanf("%d", &n);
    printf("Enter adjacency matrix:\n");
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &graph[i][j]);
    printf("Enter number of colours: ");
    scanf("%d", &m);
    if (graphColoring(0)) {
        printf("\nColours assigned:\n");
        for (int i = 0; i < n; i++)
            printf("Vertex %d → Colour %d\n", i, color[i]);
    } else {
        printf("\nNo solution exists using %d colours.\n", m);
    }
    return 0;
}
```

OUTPUT:

Applications of Graph Colouring:

- Register allocation in compilers
- Map colouring
- Scheduling problems
- Frequency assignment in wireless networks
- Sudoku solving

Conclusion:

The graph colouring problem can be efficiently solved using backtracking, which explores all valid colour assignments while avoiding conflicts. This method provides a systematic way to colour a graph using a limited set of colours such that adjacent vertices never share the same colour.