PROGRAM TITLE 11

MAP COLORING CSP

AIM:

To Write the python program for Map Coloring to implement CSP.

PROCEDURE:

1. **Define Variables:**
    - Identify the regions on the map that need to be colored. Each region is a variable.
2. **Define Domains:**
    - Assign a set of colors as the domain for each variable. The domain represents the possible values that a variable can take. In this case, the colors are the possible values.
3. **Define Constraints:**
    - Specify the constraints that must be satisfied. In the map coloring problem, the constraint is that no two adjacent regions can have the same color.
4. **Initialize:**
    - Start with an initial assignment of colors to regions. This can be a partial assignment or an empty assignment.
5. **Backtracking:**
    - Use a backtracking algorithm to systematically explore possible assignments.
        - Pick an unassigned variable.
        - Try assigning each color from its domain.
        - Check if the assignment violates any constraints. If not, move to the next variable.

- If a violation is found, backtrack to the previous variable and try a different color.

6. **Termination:**

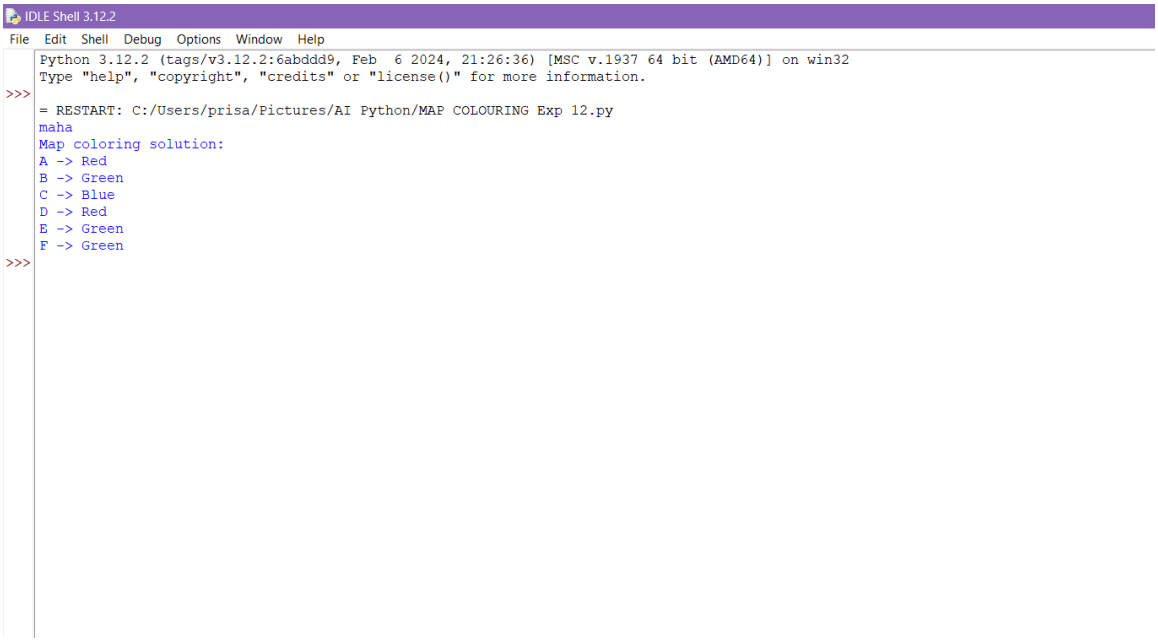
   - The algorithm terminates when all variables are assigned and the constraints are satisfied, or when it determines that no valid assignment exists.

CODING:

```
class MapColoring:
    def __init__(self, graph, colors):
        self.graph = graph
self.colors = colors
self.solution = {}

    def is_safe(self, vertex, color):        for neighbor in
self.graph[vertex]:            if neighbor in self.solution and
self.solution[neighbor] == color:
                return False
return True

    def solve(self, vertex):
if vertex not in self.graph:
            return True

        for color in self.colors:            if
self.is_safe(vertex, color):
self.solution[vertex] = color                if
self.solve(next_vertex(vertex)):
```

```
                return True
self.solution.pop(vertex)

        return False

def next_vertex(vertex):
return vertex + 1

if __name__ == "__main__":
    graph = {
        0: [1, 2, 3],
        1: [0, 2],
        2: [0, 1, 3],
        3: [0, 2]
    }     colors = ['Red', 'Green', 'Blue',
'Yellow']

    map_coloring = MapColoring(graph, colors)
if map_coloring.solve(0):
        print("Map coloring solution:")         for region,
color in map_coloring.solution.items():
            print(f"Region {region} -> {color}")
else:
        print("No solution found.")
```

OUTPUT:



RESULT:

Hence the program been successfully executed and verified.