Solve a **Map Coloring problem** using constraint satisfaction approach by applying following constraints

Aim: To Solve a Map Coloring problem using constraint satisfaction approach using Graphonline and visualago online simulator

Algorithm:

Step 1: Confirm whether it is valid to color the current vertex worth the current color (by checking whether any of its adjacent vertices are colored with the same color)

Step 2: If yes then color it and otherwise try a different color

Step 3: check if all vertices are colored or not

Step 4: If not then move to the next adjacent uncolored vertex

Step 5: Here backtracking means to stop further recursive calls on adjacent vertices.

Program:

```
class Graph:
    def _init_(self, vertices):
        self.v = vertices
        self.graph = [[0 for column in range(vertices)] for row in range(vertices)]

    # A utility function to check if the current color assignment is safe for vertex v
    def is_safe(self, v, color, c):
        for i in range(self.v):
            if self.graph[v][i] == 1 and color[i] == c:
                return False
        return True

    # A recursive utility function to solve m-coloring problem
    def graph_color_util(self, m, color, v):
        if v == self.v:
            return True

        for c in range(1, m+1):
            if self.is_safe(v, color, c):
                color[v] = c
```

```python
            if self.graph_color_util(m, color, v+1):
                return True
            color[v] = 0


    def graph_coloring(self, m):
        color = [0] * self.v
        if not self.graph_color_util(m, color, 0):
            return False


        # Print the solution
        print("Solution exists and following are the assigned colors:")
        for c in color:
            print(c, end=" ")


# Driver Code
if __name__ == '__main__':
    g = Graph(4)
    g.graph = [[0, 1, 1, 1], [1, 0, 1, 0], [1, 1, 0, 1], [1, 0, 1, 0]]
    m = 3
    # Function call
    g.graph_coloring(m)
```
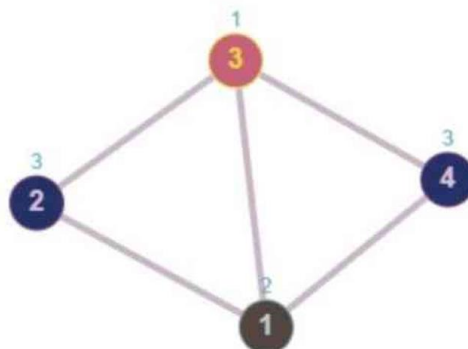
Output:

```
Solution exists and following are the assigned colors:
1 2 3 2
```


Color number is 3

Result:

Thus Solving a Map Coloring problem using constraint satisfaction approach using Graphonline and visulago online simulator was successfully executed and output was verified.