**Name : Prashant Raghuwanshi**

**Roll No : A4\_B2\_34**

**Code :**

class GraphColoring:

    def \_\_init\_\_(self, graph, m):

        self.graph = graph

        self.V = len(graph)

        self.m = m

        self.color = [0] \* self.V

    def isSafe(self, v, c):

        for i in range(self.V):

            if self.graph[v][i] == 1 and self.color[i] == c:

                return False

        return True

    def solve(self, v):

        if v == self.V:

            return True

        for c in range(1, self.m + 1):

            if self.isSafe(v, c):

                self.color[v] = c

                if self.solve(v + 1):

                    return True

                self.color[v] = 0  # Backtrack

        return False

    def graphColoring(self):

        if self.solve(0):

            print("✅ Solution exists: Following are the assigned colors:")

            for i in range(self.V):

                print(f"Vertex {i} ---> Color {self.color[i]}")

        else:

            print("❌ No solution exists with the given number of colors.")

print("--------------------------------------------------")

print("🔶 GRAPH 1: Coloring as per the image (given)")

print("--------------------------------------------------")

graph1\_colors = {

    0: "Red",

    1: "Yellow-Green",

    2: "Blue",

    3: "Red",

    4: "Yellow-Green",

    5: "Blue"

}

for v, c in graph1\_colors.items():

    print(f"Vertex {v} ---> {c}")

print("\n--------------------------------------------------")

print("🔷 GRAPH 2: Coloring using Backtracking Algorithm")

print("--------------------------------------------------")

graph2 = [

    [0, 1, 1, 1, 1],

    [1, 0, 1, 1, 1],

    [1, 1, 0, 1, 1],

    [1, 1, 1, 0, 1],

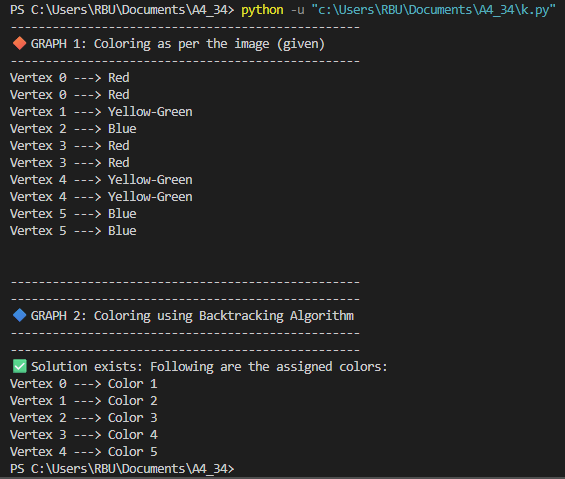
    [1, 1, 1, 1, 0]

]

g2 = GraphColoring(graph2, m=5)

g2.graphColoring()

**Output :**

****