

TASK:6

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
```

```
            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:

Color number is 3

