

map colouring.py - C:/Users/DRKR/Desktop/map colouring.py (3.10.5)

File Edit Format Run Options Window Help

```
class Graph():

def __init__(self, vertices):
    self.V = vertices
    self.graph = [[0 for column in range(vertices)]\
                  for row in range(vertices)]

def isSafe(self, v, colour, c):
    for i in range(self.V):
        if self.graph[v][i] == 1 and colour[i] == c:
            return False
    return True

# A recursive utility function to solve m
# coloring problem
def graphColourUtil(self, m, colour, v):
    if v == self.V:
        return True

    for c in range(1, m + 1):
        if self.isSafe(v, colour, c) == True:
            colour[v] = c
            if self.graphColourUtil(m, colour, v + 1) == True:
                return True
            colour[v] = 0

def graphColouring(self, m):
    colour = [0] * self.V
    if self.graphColourUtil(m, colour, 0) == None:
        return False

# Print the solution
print ("Solution exist and Following are the assigned c
for c in colour:
    print (c,end=' ')
```

IDLE Shell 3.10.5

File Edit Shell Debug Options Window Help

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/DRKR/Desktop/map colouring.py =====
Solution exist and Following are the assigned colours:
1 2 3 2
>>>
|
```

map colouring.py - C:/Users/DRKR/Desktop/map colouring.py (3.10.5)

File Edit Format Run Options Window Help

```
def isSafe(self, v, colour, c):
    for i in range(self.V):
        if self.graph[v][i] == 1 and colour[i] == c:
            return False
    return True

# A recursive utility function to solve m
# coloring problem
def graphColourUtil(self, m, colour, v):
    if v == self.V:
        return True

    for c in range(1, m + 1):
        if self.isSafe(v, colour, c) == True:
            colour[v] = c
            if self.graphColourUtil(m, colour, v + 1) == True:
                return True
            colour[v] = 0

def graphColouring(self, m):
    colour = [0] * self.V
    if self.graphColourUtil(m, colour, 0) == None:
        return False

    # Print the solution
    print ("Solution exist and Following are the assigned colours")
    for c in colour:
        print (c,end=' ')
    return True

# Driver Code
g = Graph(4)
g.graph = [[0, 1, 1, 1], [1, 0, 1, 0], [1, 1, 0, 1], [1, 1, 1, 0]]
n = 3
g.graphColouring(m)
```

Python Shell 3.10.5

File Edit Shell Debug Options Window Help

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/DRKR/Desktop/map colouring.py =====
Solution exist and Following are the assigned colours:
1 2 3 2
>>>
```