Date: Ex No: 3.2	**Title of the Lab** Graph colouring	**Name:** Yuvraj Singh Chauhan **Registration Number:** RA1911027010058 **Section:** N1 **Lab Batch:** 1 **Day Order:** 3

AIM:

To implement the graph colouring problem in python.

Description of the Concept or Problem given:

Graph colouring is the procedure of assignment of colours to each vertex of a graph G such that no adjacent vertices get same colour.The objective is to minimise the number of colours while colouring agraph. The smallest number of colours required to colour a graph Gis called its chromatic number of that graph.

Manual Solution:

The steps required to colour a graph G with n number of verticesare as follows −

Arrange the vertices of the graph in some order.

Choose the first vertex and colour it with the first colour. Choose the next vertex and colour it with the lowest numbered colour that has not been coloured on any vertices adjacent to it.

If all the adjacent vertices are coloured with this colour, assign a new colour to it. Repeat this step until all the vertices are coloured.

Program Implementation [Coding]

```
import matplotlib.pyplot as plt
import networkx as nx
from matplotlib.patches import Polygon
import numpy as np
G = nx.Graph()

colors = {0:"red", 1:"green", 2:"blue", 3:"yellow"}
G.add_nodes_from([1,2,3,4,5])
G.add_edges_from([(1,2), (1,3), (2,4), (3,5), (4,5)])
nodes = list(G.nodes)
edges = list(G.edges)
color_lists = []
color_of_edge = []
some_colors = ['red','green','blue','yellow']
for i in range(len(nodes) + 1):
    color_lists.append([])
    color_of_edge.append(-1)
```

```python
def getSmallestColor(ls1,ls2):
    i = 1
    while(i in ls1 or i in ls2):
        i = i + 1
    return i

#iterate over edges
i = 0
for ed in edges:
    newColor = getSmallestColor(color_lists[ed[0]],color_lists[ed[1]])
    color_lists[ed[0]].append(newColor)
    color_lists[ed[1]].append(newColor)
    color_of_edge[i] = newColor
    i = i + 1

# Makin graph again G = nx.Graph()
for i in range(len(edges)):
    G.add_edge(edges[i][0],edges[i][1],color=some_colors[color_of_edge[i]-1])
colors = nx.get_edge_attributes(G,'color').values()
nx.draw(G, edge_color=colors, with_labels=True, width=2)
plt.show()
```

Screenshots of the Outputs:
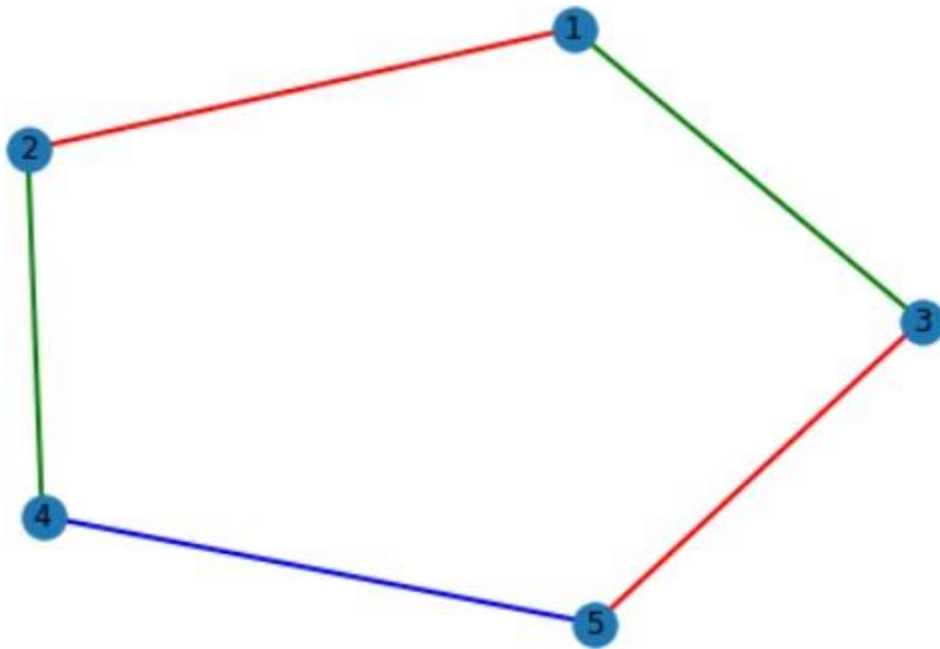


Signature of the Student

[YUVRAJ SINGH CHAUHAN]