

EXPERIMENT - 2

AIM

To develop agent programs for real world problems

M COLORING PROBLEM

ALGORITHM

- 1) Create a recursive function that takes current index, number of vertices and output color array.
- 2) If the current index is equal to number of vertices. check if the output color configuration is safe. i.e., check if the adjacent do not have same color. If the condition are met, print the configuration and break.
- 3) Assign a color to a vertex (1 to m)
- 4) For every assigned color recursively call the function with next index and number of vertices.
- 5) If any recursive function returns true break the loop and return true.

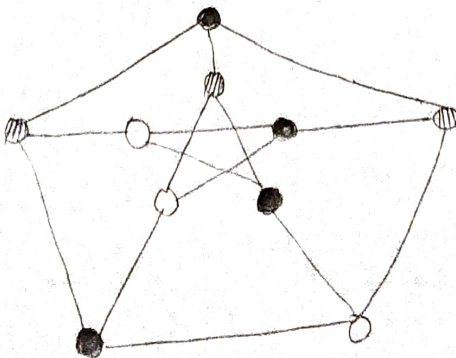
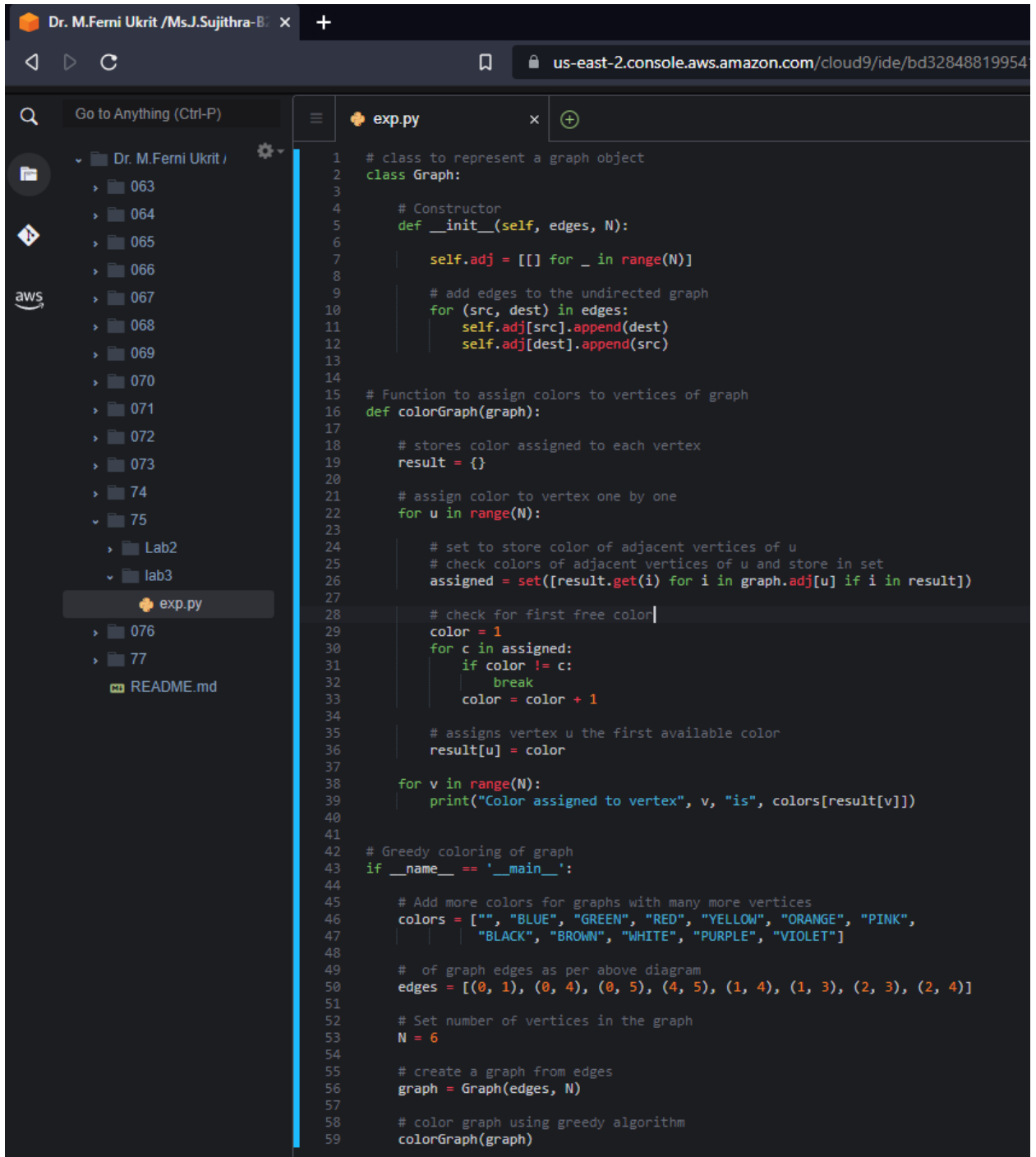


Fig. Example of graph that can be coloured with three different colours.

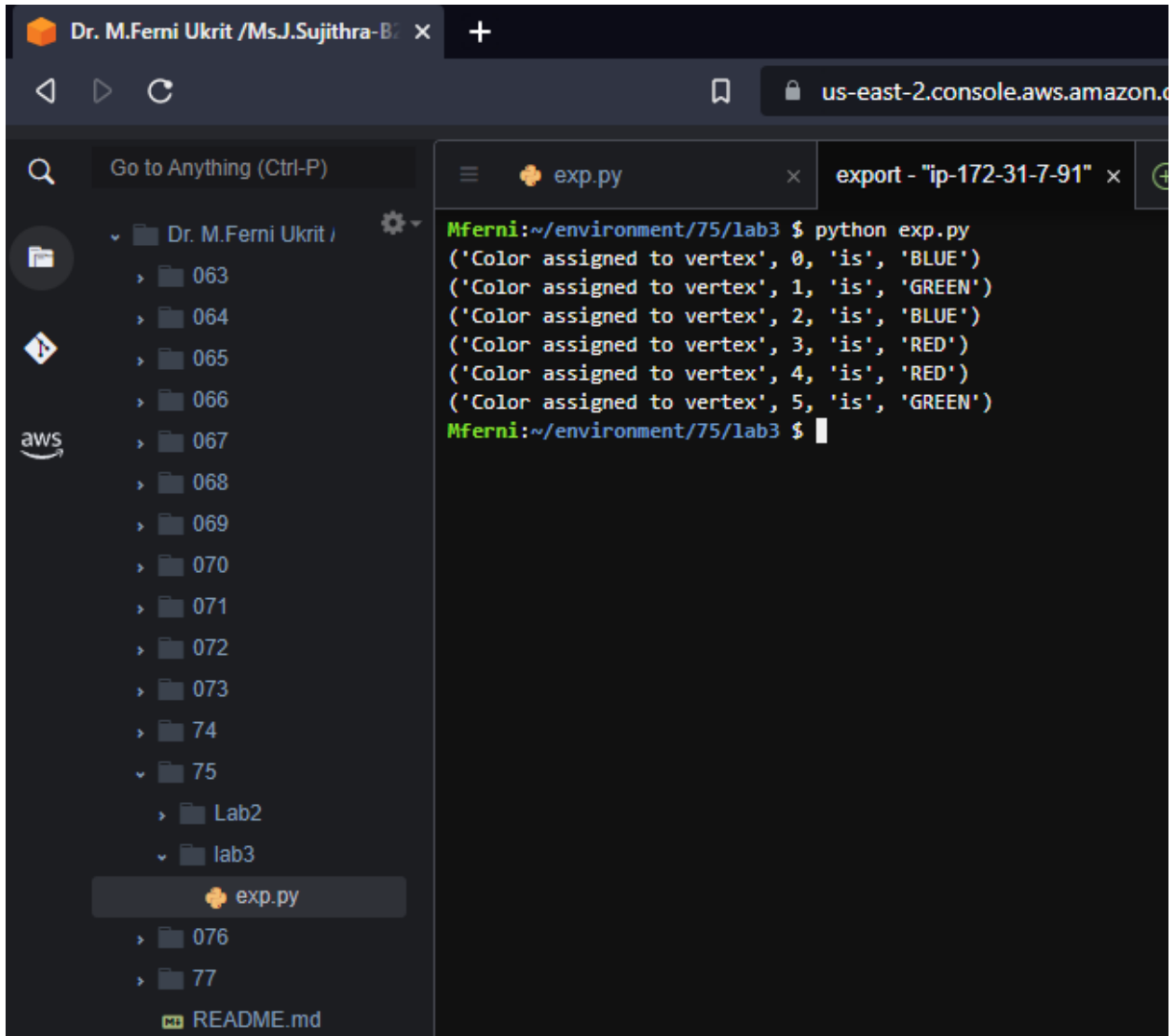
PROGRAM :



The screenshot shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for 'Dr. M.Ferni Ukrit / Ms.J.Sujithra-B', with folders 063 through 077, and subfolders 'Lab2' and 'lab3'. The file 'exp.py' is selected. The code editor shows the following Python code:

```
1 # class to represent a graph object
2 class Graph:
3
4     # Constructor
5     def __init__(self, edges, N):
6
7         self.adj = [[] for _ in range(N)]
8
9         # add edges to the undirected graph
10        for (src, dest) in edges:
11            self.adj[src].append(dest)
12            self.adj[dest].append(src)
13
14
15    # Function to assign colors to vertices of graph
16    def colorGraph(graph):
17
18        # stores color assigned to each vertex
19        result = {}
20
21        # assign color to vertex one by one
22        for u in range(N):
23
24            # set to store color of adjacent vertices of u
25            # check colors of adjacent vertices of u and store in set
26            assigned = set([result.get(i) for i in graph.adj[u] if i in result])
27
28            # check for first free color
29            color = 1
30            for c in assigned:
31                if color != c:
32                    break
33                color = color + 1
34
35            # assigns vertex u the first available color
36            result[u] = color
37
38        for v in range(N):
39            print("Color assigned to vertex", v, "is", colors[result[v]])
40
41
42    # Greedy coloring of graph
43    if __name__ == '__main__':
44
45        # Add more colors for graphs with many more vertices
46        colors = ["", "BLUE", "GREEN", "RED", "YELLOW", "ORANGE", "PINK",
47                "BLACK", "BROWN", "WHITE", "PURPLE", "VIOLET"]
48
49        # of graph edges as per above diagram
50        edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
51
52        # Set number of vertices in the graph
53        N = 6
54
55        # create a graph from edges
56        graph = Graph(edges, N)
57
58        # color graph using greedy algorithm
59        colorGraph(graph)
```

OUTPUT :



The screenshot shows a VS Code editor window with a dark theme. The top bar displays the user 'Dr. M.Ferni Ukrit /Ms.J.Sujithra-B' and the browser address 'us-east-2.console.aws.amazon.com'. The left sidebar shows a file explorer with a tree view of folders: 'Dr. M.Ferni Ukrit /', '063', '064', '065', '066', '067', '068', '069', '070', '071', '072', '073', '74', '75', 'Lab2', 'lab3', '076', '77', and 'README.md'. The 'exp.py' file is selected. The main editor area shows the output of the command 'python exp.py' in a terminal window. The output is as follows:

```
Mfern timer:~/environment/75/lab3 $ python exp.py
('Color assigned to vertex', 0, 'is', 'BLUE')
('Color assigned to vertex', 1, 'is', 'GREEN')
('Color assigned to vertex', 2, 'is', 'BLUE')
('Color assigned to vertex', 3, 'is', 'RED')
('Color assigned to vertex', 4, 'is', 'RED')
('Color assigned to vertex', 5, 'is', 'GREEN')
Mfern timer:~/environment/75/lab3 $
```