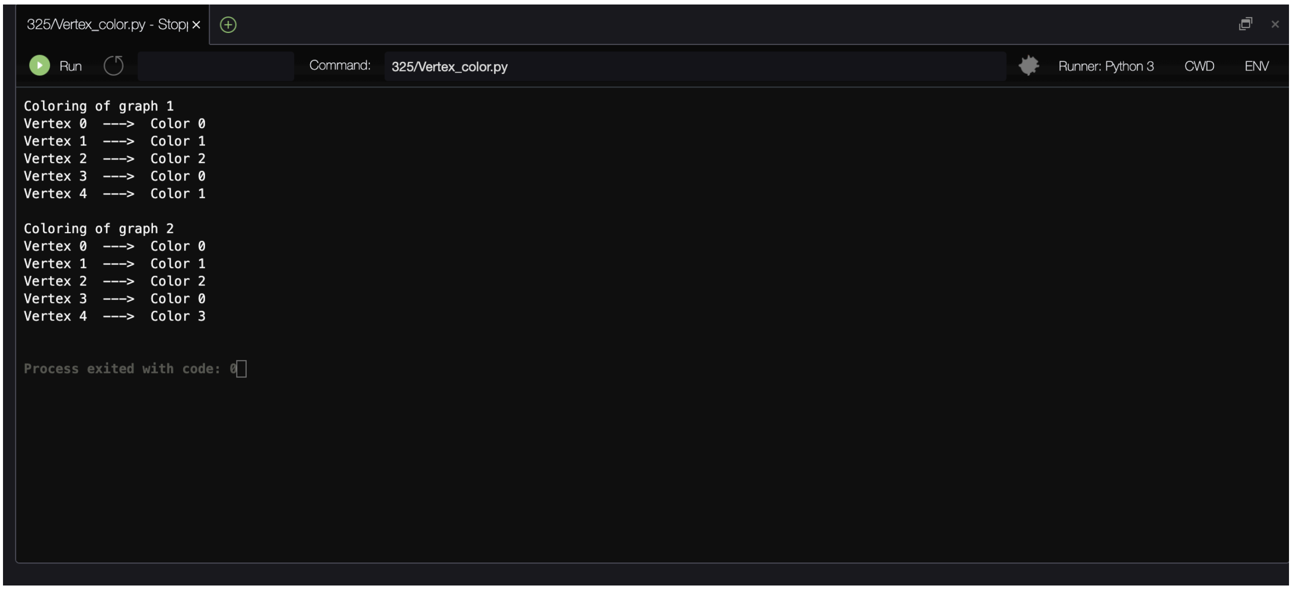
SOURCE CODE:-

```
def addEdge(adj, v, w):

    adj[v].append(w)

    # Note: the graph is undirected
    adj[w].append(v)
    return adj

# Assigns colors (starting from 0) to all
# vertices and prints the assignment of colors
def greedyColoring(adj, V):

    result = [-1] * V

    # Assign the first color to first vertex
    result[0] = 0;


    # A temporary array to store the available colors.
    # True value of available[cr] would mean that the
    # color cr is assigned to one of its adjacent vertices
    available = [False] * V

    # Assign colors to remaining V-1 vertices
    for u in range(1, V):

        # Process all adjacent vertices and
        # flag their colors as unavailable
        for i in adj[u]:
            if (result[i] != -1):
                available[result[i]] = True

        # Find the first available color
        cr = 0
        while cr < V:
            if (available[cr] == False):
                break

            cr += 1

        # Assign the found color
        result[u] = cr

        # Reset the values back to false
        # for the next iteration
        for i in adj[u]:
            if (result[i] != -1):
                available[result[i]] = False

    # Print the result
    for u in range(V):
        print("Vertex", u, " --->  Color", result[u])

# Driver Code
if __name__ == '__main__':

    g1 = [[] for i in range(5)]
    g1 = addEdge(g1, 0, 1)
    g1 = addEdge(g1, 0, 2)
    g1 = addEdge(g1, 1, 2)
    g1 = addEdge(g1, 1, 3)
    g1 = addEdge(g1, 2, 3)
    g1 = addEdge(g1, 3, 4)
    print("Coloring of graph 1 ")
    greedyColoring(g1, 5)

    g2 = [[] for i in range(5)]
    g2 = addEdge(g2, 0, 1)
    g2 = addEdge(g2, 0, 2)
    g2 = addEdge(g2, 1, 2)
    g2 = addEdge(g2, 1, 4)
    g2 = addEdge(g2, 2, 4)
    g2 = addEdge(g2, 4, 3)
    print("\nColoring of graph 2")
    greedyColoring(g2, 5)
```



```
Coloring of graph 1
Vertex 0  --->  Color 0
Vertex 1  --->  Color 1
Vertex 2  --->  Color 2
Vertex 3  --->  Color 0
Vertex 4  --->  Color 1

Coloring of graph 2
Vertex 0  --->  Color 0
Vertex 1  --->  Color 1
Vertex 2  --->  Color 2
Vertex 3  --->  Color 0
Vertex 4  --->  Color 3


Process exited with code: 0
```