

```

g1={
    0:[1,2],
    1:[2,3,0],
    2:[3,1,0],
    3:[1,2,4],
    4:[3]
}

```

```

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

```

```

    result = [-1] * V
    print(result)

```

```

    # 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
    print(available)

```

```

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

```

```

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

```

```

        # 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])

print("Coloring of graph 1 ")
greedyColoring(g1, 5)
```