11. Write the python program for Map Coloring to implement CSP.

# Map Coloring Problem using CSP (Backtracking)

# Define the map (graph) as adjacency list

neighbors = {

"WA": ["NT", "SA"],

"NT": ["WA", "SA", "Q"],

"SA": ["WA", "NT", "Q", "NSW", "V"],

"Q": ["NT", "SA", "NSW"],

"NSW":["Q", "SA", "V"],

"V": ["SA", "NSW", "T"],

"T": ["V"]

}

# Available colors

colors = ["Red", "Green", "Blue"]

def is\_valid(state, node, color):

"""Check if assigning 'color' to 'node' is valid under CSP constraints"""

for neighbor in neighbors[node]:

if neighbor in state and state[neighbor] == color:

return False

return True

def backtrack(state):

"""Backtracking search for coloring"""

# If all nodes are assigned, return solution

if len(state) == len(neighbors):

return state

# Select an unassigned variable

unassigned = [node for node in neighbors if node not in state]

node = unassigned[0]

# Try each color

for color in colors:

if is\_valid(state, node, color):

state[node] = color

result = backtrack(state)

if result:

return result

state.pop(node) # Backtrack

return None

# Run CSP

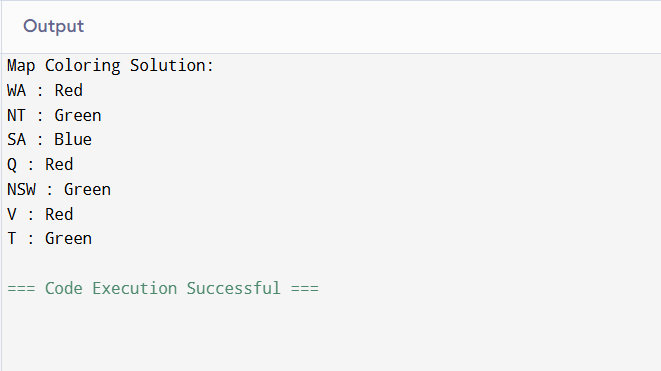
solution = backtrack({})

print("Map Coloring Solution:")

for region, color in solution.items():

print(region, ":", color)

**OUTPUT:**

****