PROGRAM

def addColor(R, province, color):

ans = []

for rr in R:

res = checkRestriction(rr, province, color)

if res == False:

return False

elif res == None:

continue

else:

ans.append(res)

return ans

def checkRestriction(rr, province, color):

index = -1

other = -1

if rr[0] == province:

index = 0

other = 1

elif rr[1] == province:

index = 1

other = 0

else:

return rr

if isinstance(rr[other], int):

if (color != rr[other]):

return None

else:

return False

else:

return [rr[other], color]

# solving the CSP by variable elimination

# recursive structure: ci is the province index to be colored (0 = bc, 1 = ab, etc)

# n is the number of colors

# provinces is a list of provinces

# if coloring is possible returns the province-> color map, otherwise False

def solveCSP(provinces, n, R, ci):

if (ci == 0):

newR = addColor(R, provinces[0], 1)

if (newR == False):

return False

ans = {provinces[0]:1}

res = solveCSP(provinces, n, newR, 1)

if (res == False):

return False

ans.update(res)

return ans

elif (ci == len(provinces)):

return {}

for color in range (1,n+1):

ans = {provinces[ci]:color}

newR = addColor(R, provinces[ci], color)

if (newR == False):

continue

res = solveCSP(provinces, n, newR, ci+1)

if (res == False):

continue

print(ans)

print(res)

print("============")

ans.update(res)

return ans

return False

n=5

colors=[]

for i in range(1,n+1):

colors.append(i)

cmap = {}

cmap["ab"] = ["bc","nt","sk"]

cmap["bc"] = ["yt", "nt", "ab"]

cmap["mb"] = ["sk","nu","on"]

cmap["nb"] = ["qc", "ns", "pe"]

cmap["ns"] = ["nb", "pe"]

cmap["nl"] = ["qc"]

cmap["nt"] = ["bc", "yt", "ab", "sk", "nu"]

cmap["nu"] = ["nt", "mb"]

cmap["on"] = ["mb", "qc"]

cmap["pe"] = ["nb", "ns"]

cmap["qc"] = ["on", "nb", "nl"]

cmap["sk"] = ["ab", "mb", "nt"]

cmap["yt"] = ["bc", "nt"]

# CSP restrictions

# each restriction is modeled as a pair [a,b] which means the province a's

# color is not equal to b, where b is either a color (a number 1 to n) or

# another province. Examples ['bc', 'ab'] means the color of bc should

# not be equal to ab -- ["bc",4] means the color of bc should not be 4

# R is the list of restrictions

R = []

for x in cmap:

for y in cmap[x]:

R.append([x,y])

provinces = []

for p in cmap:

provinces.append(p)

print(solveCSP(provinces, 3, R, 0))

while(1):

num=int(input("Enter number of the color? "))

print(solveCSP(provinces, num, R, 0))

print(R)

print(" ================= ")

print(checkRestriction(["ab",4],"ab",4))

R = addColor(R, 'bc', 4)

print(R)

print(" ================= ")

print(checkRestriction(["ab",4],"ab",4))

R = addColor(R, 'ab', 4)

print(R)

OUTPUT

