Appendix

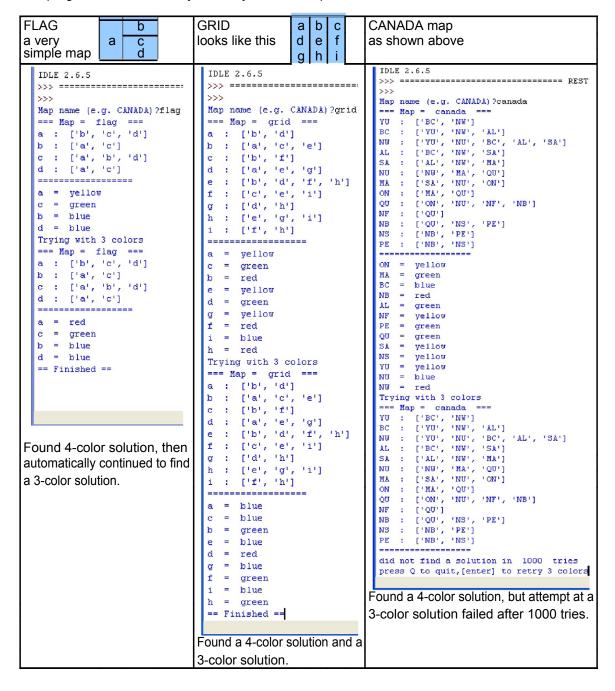
= Source code listing of colors.py =

```
import random
border = ( ) # dictionary containing ( region0 : [neighbor1, neighbor2, ...]
                                      region1 : [neighbor1, neighbor2, ...]
state = { } # dictionary containing { region0 : 0, region1 : 1 , ...}
max = 0
             # number of regions in the map
co = [ ]
            # array of random color assignment for each region in state list
colors = ("red", "green", "blue", "yellow") # fixed names of 4 colors
tries = 1000  # limited number of tries in each search
mapname = "CANADA"
                     # name of file containing borders data
def randomColors(choices):
                             # choose a random color for each region
    for n in range(0.max):
        c = random.randint(0,choices-1) # random number
                                        # store random color in colors list
        co[n] = colors[c]
                             # print list of regions and colors
def listColors():
    for s in state:
       print s , " = " , co[state[s]]
def checkColors(): # check all pairs of regions and neighbors for same color
    okay = True
                                 # result (success so far)
    for x in border:
                                  # each region in border dictionary
       for c in border[x]:
                                 # each in neighbors list
            a = state[x]
                                 # get color of x
           b = state[c]
                                 # get color of c
           if co[a] == co[b]:
                                 # if colors match, then success = false
               okay = False
                                  # return success or failure
    return okay
def readborders(): # read data from text-file and store in border dictionary
    count = 0
                                    # counting regions
    infile = open(mapname,'rU')
    print "=== Map = ", mapname, " ==="
                                    # loop through entire file
    while(1):
        indata = infile.readline()  # read data: region, neighbor, neighbor...
       num = len(indata)
        if indata[num-1:num] == '\n': # need to remove
           indata = indata[0:num-1]
                                    # stop at end of file
        if len(indata)<1:</pre>
           break
        items = indata.split(",") # change String to array
        num = len(items)
       border[items[0]] = items[1:num]
                                              # add data to border dictionary
        print items[0] , " : " , border[items[0]]  # region : [neighbors]
       state[items[0]] = count
                                                    # index of this region
        co.append("")
                                    # makes co an array of matching lenght
        count = count + 1
    infile.close()
    print "========"
                                    # number of regions
    return count
```

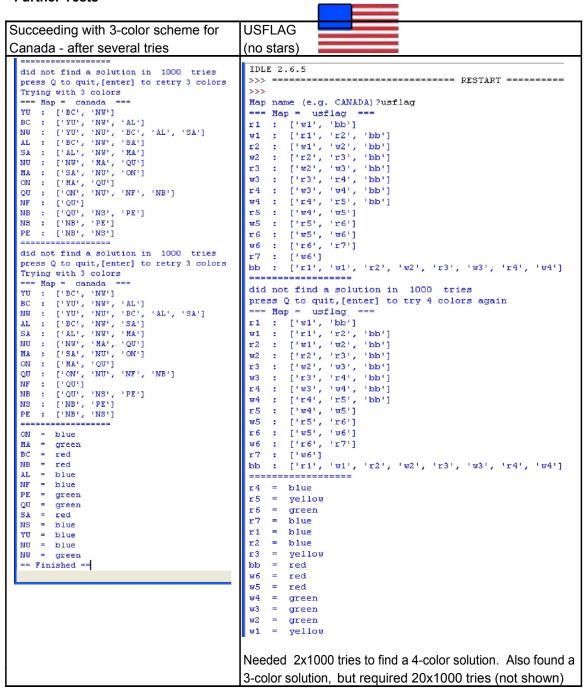
```
def go(choices):
                    # main loop - tries a set of random colors, then another
                    # repeating until success or quitting after 1000 tries
    global border, state, co, max, tries
    border = { }
    state = { }
    co = [ ]
    max = readborders()
                              # load data from file
    done=False
                              # not yet successful
    for t in range(0,tries): # try 1000 times
        if done == False:
                                        # if still unsuccssful then
            randomColors(choices)  # choose colors
if checkColors() == True: # check whether successful
                                        # if successful then print colors
                done = True
                return done
                                        # quit early if successful
    if done==False:
        print "did not find a solution in ", tries, " tries"
    return done
                                        # return success or failure
def start():
                       # main logic - try 4 colors, then 3 if
                                       it was successful
    global mapname
    mapname = raw input("Map name (e.g. CANADA)?")
    result = False
    while (result == False):
        result = go(4)
        if result == True:
            result = False
            while(result == False):
                print "Trying with 3 colors"
                result = go(3)
                if result==False:
                    answer = raw_input("press Q to quit,[enter] to retry 3 colors")
                    if answer=="Q" or answer=="q":
                        result = True
        else:
            answer = raw_input("press Q to quit,[enter] to try 4 colors again")
            if answer == "Q" or answer == "q":
                return
start()
raw input ("== Finished ==")
```

- Sample Output -

The program ran successfully on many sets of sample data - here are a few:



- Further Tests -



Further sample runs are shown in the accompanying videos.