```
#!/usr/bin/env python

""" cclabelpy a python recursive labeling program
"""

import sys
from numpy  import *
from v4 import vx
import numpy as np
from PIL import Image

of=' '
vxif=' '
clist = vx.vxparse(sys.argv,  "if= of= -v  - ")
exec (clist )

if 'OPT' in locals():
   print ("cclabelpy program")
   print ("if= input file")
   print ("of= output file")
   print ("[-v] verbose mode")
   exit(0)

optv = 'OPTv' in locals()

def setlabel (x, y, n): # n is current label
    global im, tm
    im[y,x] = n
    # 4 connected foreground 8 connected background
    if tm[y+2,x+1] > 0 and im[y+1,x] == 0: # If pixel above current tm is foreground and not labeled in im
        setlabel(x, y+1, n)
    if tm[y,x+1] > 0 and im[y-1,x] == 0: # If pixel below current tm is foreground and not labeled in im
        setlabel(x, y-1, n)
    if tm[y+1,x] > 0 and im[y,x-1] == 0: # If pixel left of current tm is foreground and not labeled in im
        setlabel(x-1, y, n)
    if tm[y+1,x+2] > 0 and im[y,x+1] == 0: # If pixel right current tm is foreground and not labeled in im
        setlabel(x+1, y, n)

inimage = vx.Vx( vxif )
im = inimage.i
tmimage = vx.Vx( inimage )
tmimage.embedim((1,1,1,1))
tm = tmimage.i

# Clear image for output
for y in range(im.shape[0]):
    for x in range(im.shape[1]):
        im[y,x] = 0

```

```
50  n = 1 # Set initial label to be 1
51
52  for y in range(im.shape[0]):
53      for x in range(im.shape[1]):
54          if tm[y+1,x+1] > 0 and im[y,x] == 0: # If object is foreground in tm and not
            labeled in im
55              setlabel(x,y,n)
56              n = n + 1 # Increment the next label value
57
58  colored = np.zeros([im.shape[0], im.shape[1], 3], dtype=np.uint8)
59
60  # Loop to color the different labels
61  for y in range(im.shape[0]):
62      for x in range(im.shape[1]):
63          if im[y,x] > 0:
64              # RGB pixel value is based on scaled value of label
65              colored[y,x] = [(im[y,x]*100)%256, (im[y,x]*200)%256, (im[y,x]*150)%256]
                # color the label
66
67  img = Image.fromarray(colored)
68  img.save('coloredlabel.png')
69
70  if optv:
71     print (im)
72
73  inimage.write(of)
74
```