

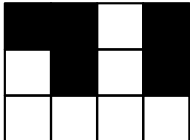
Name: _____ Andrew Id: _____

15-112 Spring 2021 Quiz 06

20 minutes. Do not use dictionaries, sets, try/except, or recursion on this quiz.

1. Reasoning Over Code [8 pts] For the following code, draw an image which will cause `roc1(myPic)` to return `True`.

In your picture, you should make clear the width and height of the image and the color of each pixel in the image. You may wish to draw a grid with each box representing a color. For example, here is a representation of a picture with a height of 3, width of 4, with twelve pixels: five of them black and seven of them white:



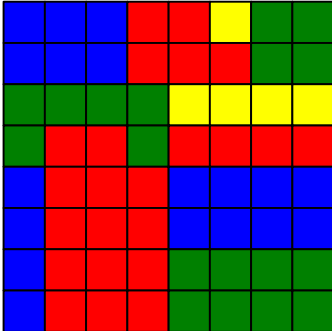
```
def helper(myPic, c, r):
    b = 0
    t = 0
    for i in range(c[0],c[1]):
        for j in range(r[0],r[1]):
            if ImageWriter.getColor(myPic,i,j) == [0,0,0]:
                b += 1
            t += 1
    return 100*b//t

def roc1(myPic):
    h = ImageWriter.getHeight(myPic)
    w = ImageWriter.getWidth(myPic)
    assert h == 8 and w == 4

    L = [((w//2,w),(h//2,h)),
          ((0,w//2),(h//2,h)),
          ((0,w//2),(0,h//2)),
          ((w//2,w),(0,h//2))]
    r = []
    for item in L:
        v = helper(myPic,item[0],item[1])
        r.append(v)
    return r == [25, 50, 75, 100]
```

2. Free Response – Widest Red Line [12 pts] Making use of the ImageWriter library presented in class, write the function `widestRedLine(filename)` which analyzes an image contained in `filename` and returns the width of the widest red line in the picture.

For example, consider the following 8x8 picture (represented as a grid of pixels to make things easier to see) stored in `sample.png`:



Calling `widestRedLine("sample.png")` will return 4.