def collage():
    setMediaPath()
    ogimg = makePicture(getMediaPath("seahorse.jpg"))
    sig = makePicture(getMediaPath("signature.jpg"))
    canvas = makeEmptyPicture(2000,2000,black)

    #scaling original image (and creating copies)
    scaleFactor = 10
    widS = getWidth(ogimg)/scaleFactor
    heiS = getHeight(ogimg)/scaleFactor
    # I will need three versions of this picture that will be edited seperately, thus these steps are repeated
    scaled = makeEmptyPicture(widS,heiS)
    scaledbig1 = makeEmptyPicture(widS,heiS)
    scaledbig2 = makeEmptyPicture(widS,heiS)
    scaleDown(ogimg,widS,heiS,scaled,scaleFactor)
    scaleDown(ogimg,widS,heiS,scaledbig1,scaleFactor)
    scaleDown(ogimg,widS,heiS,scaledbig2,scaleFactor)
    ogimg = scaled

    #creating a flipped version of the original image (and creating a copy)
    #double the width to flip on the "halfway line", which will just be the far edge of the original picture
    flipimg = makeEmptyPicture(690,heiS,black)
    copy(ogimg,flipimg,0,0)
    mirror(flipimg)
    #crop the picture so that it is just the mirrored version, do it twice
    mirrorimg = makeEmptyPicture(getWidth(ogimg),getHeight(ogimg),black)
    colorCrop(flipimg,mirrorimg,getWidth(ogimg),getWidth(flipimg),0,getHeight(flipimg),0,0)
mirrorimgbig = makeEmptyPicture(getWidth(ogimg), getHeight(ogimg), black)
colorCrop(flipimg, mirrorimgbig, getWidth(ogimg), getWidth(flipimg), 0, getHeight(flipimg), 0, 0)

# creating a cleaner version of the flipped images and original images
cromaback = makeEmptyPicture((getWidth(mirrorimg)+1), (getHeight(mirrorimg)+1), black)
cromakeyblack(mirrorimg, chromaback)
cromakeyblack(ogimg, chromaback)
cromakeyblack(mirrorimgbig, chromaback)
cromakeyblack(scaledbig1, chromaback)
cromakeyblack(scaledbig2, chromaback)

# creating a slightly bigger versions of the original images
scaleFactor = .55
wid = getWidth(scaledbig1)/scaleFactor
hei = getHeight(scaledbig1)/scaleFactor
wid = int(wid)
hei = int(hei)
scaledup1 = makeEmptyPicture(wid, hei)
scale(scaled1, wid, hei, scaledup1, scaleFactor)

# the scaleFactor is different in each because the seahorses need to be different sizes
scaleFactor = .6
wid = getWidth(scaledbig2)/scaleFactor
hei = getHeight(scaledbig2)/scaleFactor
wid = int(wid)
hei = int(hei)
scaledup2 = makeEmptyPicture(wid, hei)
scale(scaled2, wid, hei, scaledup2, scaleFactor)

scaleFactor = .65
wid = getWidth(mirrorimgbig)/scaleFactor
hei = getHeight(mirrorimgbig)/scaleFactor
wid = int(wid)
hei = int(hei)
mirrorback = makeEmptyPicture(wid, hei)
scale(mirrorimgbig, wid, hei, mirrorback, scaleFactor)

# creating the collage
# big, poster, dark, flip version
scaledupposter = scaledup1
posterize(scaledupposter)
darken(scaledupposter)

← means the line is continued on the next line.
darken(scaledupposter)
darken(scaledupposter)
darken(scaledupposter)
crop(scaledupposter, canvas, 0, getWidth(scaledupposter), 0, getHeight(scaledupposter), 0, 375)

# big, cyan, dark, og version
cyanotype(mirrorback)
darken(mirrorback)
darken(mirrorback)
darken(mirrorback)
darken(mirrorback)
darken(mirrorback)
colorCrop(mirrorback, canvas, 0, getWidth(mirrorback), 0, getHeight(mirrorback), 200, 370)

# big, red, dark, flip version
scaledupred = scaledup2
redotype(scaledupred)
darken(scaledupred)
darken(scaledupred)
darken(scaledupred)
darken(scaledupred)
darken(scaledupred)
darken(scaledupred)
colorCrop(scaledupred, canvas, 0, getWidth(scaledupred), 0, getHeight(scaledupred), 380, 425)

"normal" seahorse
colorCrop(mirrorimg, canvas, 0, getWidth(mirrorimg), 0, getHeight(mirrorimg), 200, 500)

greyscaled tail
grayScale(mirrorimg)
colorCrop(mirrorimg, canvas, 0, (getWidth(mirrorimg)/3), 0, getHeight(mirrorimg), 200, 500)

# make the canvas the sized needed so i can get an A on the project
croppedcanvas = makeEmptyPicture(600, 700, black)
crop(canvas, croppedcanvas, 210, 810, 525, 1225, 0, 0)

# signature time
scaleFactor = 5
widS = getWidth(sig)/scaleFactor
heiS = getHeight(sig)/scaleFactor
signscale = makeEmptyPicture(widS, heiS)
scaleDown(sig, widS, heiS, signscale, scaleFactor)
sig = signscale
signCrop(sig, croppedcanvas, 0, getWidth(sig), 0, getHeight(sig), (getWidth(croppedcanvas) - getWidth(sig)), (getHeight(croppedcanvas) - getHeight(sig)))

means the line is continued on the next line.
# the final product
   explore(croppedcanvas)

```python
def chromakeyblack(source, bg):
    for px in getPixels(source):
        x = getX(px)
        y = getY(px)
        if (getRed(px) <= 75 and getGreen(px) <= 75 and getBlue(px) <= 75):
            bgpx = getPixel(bg, x, y)
            bgcol = getColor(bgpx)
            setColor(px, bgcol)
```

```python
def scaleDown(img, widS, heiS, canvas, scaleFactor):
    sourceX = 0
    for targetX in range(0, widS):
        sourceY = 0
        for targetY in range(0, heiS):
            color = getColor(getPixel(img, sourceX, sourceY))
            setColor(getPixel(canvas, targetX, targetY), color)
            sourceY = sourceY + int(scaleFactor)
            sourceX = sourceX + int(scaleFactor)
```

```python
def scaleUp(img, widB, heiB, canvas, scaleFactor):
    sourceX = 0
    for targetX in range(0, widB):
        sourceY = 0
        for targetY in range(0, heiB):
            imgpx = getPixel(img, int(sourceX), int(sourceY))
            color = getColor(imgpx)
            setColor(getPixel(canvas, targetX, targetY), color)
            sourceY = sourceY + 1.0/float(scaleFactor)
            sourceX = sourceX + 1.0/float(scaleFactor)
```

```python
def mirror(pic):
    for x in range(0, (getWidth(pic)/2)):
        for y in range(0, getHeight(pic)):
            pleft = getPixel(pic, x, y)
            pright = getPixel(pic, getWidth(pic)-x-1, y)
            setColor(pright, getColor(pleft))
```

```python
def copy(img, canvas, target_x, target_y):
```

← means the line is continued on the next line.
```python
targetX = target_x
for sourceX in range(0, getWidth(img)):
    targetY = target_y
    for sourceY in range(0, getHeight(img)):
        color = getColor(getPixel(img, sourceX, sourceY))
        setColor(getPixel(canvas, targetX, targetY), color)
        targetY = targetY + 1
    targetX = targetX + 1

def crop(img, canvas, range1x, range2x, range1y, range2y, target_x, target_y):
    targetX = target_x
    for sourceX in range(range1x, range2x):
        targetY = target_y
        for sourceY in range(range1y, range2y):
            color = getColor(getPixel(img, sourceX, sourceY))
            setColor(getPixel(canvas, targetX, targetY), color)
            targetY = targetY + 1
        targetX = targetX + 1

def colorCrop(img, canvas, range1x, range2x, range1y, range2y, target_x, target_y):
    targetX = target_x
    for sourceX in range(range1x, range2x):
        targetY = target_y
        for sourceY in range(range1y, range2y):
            color = getColor(getPixel(img, sourceX, sourceY))
            if color != black:
                setColor(getPixel(canvas, targetX, targetY), color)
            targetY = targetY + 1
        targetX = targetX + 1

def sigCrop(img, canvas, range1x, range2x, range1y, range2y, target_x, target_y):
    targetX = target_x
    for sourceX in range(range1x, range2x):
        targetY = target_y
        for sourceY in range(range1y, range2y):
            color = getColor(getPixel(img, sourceX, sourceY))
            if color != white:
                setColor(getPixel(canvas, targetX, targetY), color)
            targetY = targetY + 1
        targetX = targetX + 1

def grayScale(pic):
    for p in getPixels(pic):
```

← means the line is continued on the next line.
intensity = (getRed(p)+getGreen(p)+getBlue(p))/3
setColor(p,makeColor(intensity, intensity, intensity))

def cyanotype(pic):
    grayScale(pic)
    for p in getPixels(pic):
        blue = getBlue(p)
        green = getGreen(p)
        red = getRed(p)
        if (blue < 63):
            blue = blue*2
            green = green*.75
            red = red*.75
        if (blue > 62 and blue < 192):
            blue = blue*1.3
            green = green*.75
            red = red*.75
        if (blue > 192):
            blue = blue*1.2
            green = green*.75
            red = red*.75
        setBlue(p, blue)
        setRed(p, red)
        setGreen(p, green)

def lighten(pic):
    for x in range(0, getWidth(pic)):
        for y in range(0, getHeight(pic)):
            px = getPixel(pic,x,y)
            color = getColor(px)
            color = makeLighter(color)
            color = makeLighter(color)
            setColor(px,color)

def darken(pic):
    for x in range(0, getWidth(pic)):
        for y in range(0, getHeight(pic)):
            px = getPixel(pic,x,y)
            color = getColor(px)
            color = makeDarker(color)
            setColor(px,color)

def scale(img,wid,hei,scaled,scaleFactor):

← means the line is continued on the next line.
sourceX = 0
for targetX in range(0,wid):
    sourceY = 0
    for targetY in range(0,hei):
        color = getColor(getPixel(img,int(sourceX),int(sourceY)))
        setColor(getPixel(scaled,targetX,targetY), color)
        sourceY = sourceY + float(scaleFactor)
        sourceX = sourceX + float(scaleFactor)

def posterize(source):
    for p in getPixels(source):
        r = getRed(p)
        g = getGreen(p)
        b = getBlue(p)
        luminance = (r+g+b)/3
        if luminance < 50:
            setColor(p,black)
        if 50 <= luminance <= 165:
            setColor(p,gray)
        if luminance > 165:
            setColor(p,white)

def redotype(pic):
    grayScale(pic)
    for p in getPixels(pic):
        blue = getBlue(p)
        green = getGreen(p)
        red = getRed(p)
        if (red < 63):
            blue = blue*.75
            green = green*.75
            red = red*2
        if (red > 62 and red < 192):
            blue = blue*.75
            green = green*.75
            red = red*1.3
        if (red > 192):
            blue = blue*.75
            green = green*.75
            red = red*1.3
        setBlue(p, blue)
        setRed(p,red)
        setGreen(p,green)