

# Conner Um

Completed

Original



```
#Name: Conner Um  
#Date: 10/19/2021  
#Project 2
```

```
def collage():  
    setMediaPath()  
    ogPic = makePicture("Rocky.jpg")  
    signaturePic = makePicture("signature.jpg")  
    startX = 53; endX = 175  
    startY = 23; endY = 233  
    grayPic = grayScale(ogPic) #function call 1  
    negativePic = negPic(ogPic) #function call 2  
    reduceRed = redReduce(ogPic, startX, endX, startY, endY) #function call 3  
    cropLeftPic = cropLeft(grayPic) #function call 4  
    cropRightPic = cropRight(grayPic) #function call 5  
    mirrorPic = mirrorBottom(ogPic) #function call 6  
    #Add & Copy all images into canvas collage  
    canvas = makeEmptyPicture(846, 703, blue) #Create Canvas  
    rotatePic(mirrorPic, canvas, 0, 0) #1st copy  
    rotatePic(mirrorPic, canvas, 282, 0) #2nd copy  
    rotatePic(mirrorPic, canvas, 564, 0) #3rd copy  
    rotatePic(mirrorPic, canvas, 0, 492) #4th copy  
    rotatePic(mirrorPic, canvas, 282, 492) #5th copy  
    rotatePic(mirrorPic, canvas, 564, 492) #6th copy  
    copyInto(ogPic, canvas, 318, 210) #7th copy  
    copyInto(negativePic, canvas, 108, 210) #8th copy  
    copyInto(reduceRed, canvas, 528, 210) #9th copy
```

```

copyInto(cropLeftPic, canvas, 0, 210) #10th copy
copyInto(cropRightPic, canvas, 738, 210) #11th copy
chromakeySig(signaturePic, canvas, 100, 192) #12th copy
show(canvas)

def chromakeySig(signaturePic, canvas, targetX, targetY): #creates a chromakey function to add signature
    for sX in range(0, getWidth(signaturePic)):
        for sY in range(0, getHeight(signaturePic)):
            sPx = getPixelAt(signaturePic, sX, sY)
            sColor = getColor(sPx)
            targetPx = getPixelAt(canvas, sX + targetX, sY + targetY)
            if distance (black, sColor) < 150:
                setColor(targetPx, red)

def cropRight(grayPic): #crops the left side of the grayscale picture leaving the right side
    width = 108; height = getHeight(grayPic)
    newPic = makeEmptyPicture(width, height)
    targetX = 0
    for sourceX in range(102, 210):
        targetY = 0
        for sourceY in range(0, height):
            srcPx = getPixelAt(grayPic, sourceX, sourceY)
            srcColor = getColor(srcPx)
            targetPx = getPixelAt(newPic, targetX, targetY)
            setColor(targetPx, srcColor)
            targetY = targetY + 1
        targetX = targetX + 1
    return newPic

def cropLeft(grayPic): #crops the right side of the grayscale picture leaving the left side
    width = 108; height = getHeight(grayPic)
    newPic = makeEmptyPicture(width, height)
    targetX = 0
    for sourceX in range(0, width):
        targetY = 0
        for sourceY in range(0, height):
            srcPx = getPixelAt(grayPic, sourceX, sourceY)
            srcColor = getColor(srcPx)
            targetPx = getPixelAt(newPic, targetX, targetY)
            setColor(targetPx, srcColor)
            targetY = targetY + 1
        targetX = targetX + 1
    return newPic

```

```

def rotatePic(mirrorPic, canvas, startX, startY): #rotates the mirrored picture to the right
    targetX = 0
    height = getHeight(mirrorPic)
    for sourceX in range(0, getWidth(mirrorPic)):
        targetY = 0
        for sourceY in range(0, getHeight(mirrorPic)):
            sourcePx = getPixelAt(mirrorPic, sourceX, sourceY)
            sourceColor = getColor(sourcePx)
            targetPx = getPixelAt(canvas, (height-1-targetY) + startX, targetX + startY)
            setColor(targetPx, sourceColor)
            targetY = targetY + 1
        targetX = targetX + 1

def mirrorBottom(picture): #mirrors the bottom of the original picture to the top
    newPic=duplicatePicture(picture)
    mirrorPoint = getHeight(newPic) / 2
    height = getHeight(newPic)
    for x in range(0,getWidth(newPic)):
        for y in range(0,mirrorPoint):
            topPixel = getPixel(newPic,x,y)
            bottomPixel = getPixel(newPic,x,height -y -1)
            color = getColor(bottomPixel)
            setColor(topPixel,color)
    return newPic

def redReduce(picture, startX, endX, startY, endY): #reduces red within a range of the original picture
    newPic = duplicatePicture(picture)
    for x in range(startX, endX):
        for y in range(startY, endY):
            currentPx = getPixelAt(newPic, x, y)
            currentColor = getColor(currentPx)
            if (distance(currentColor, red)<300):
                redValue = getRed(currentPx)
                setRed(currentPx, redValue * 0.45)
    return(newPic)

def grayScale(picture): #creates a grayscale copy of the original picture
    newPic = duplicatePicture(picture)
    for px in getAllPixels(newPic):
        r = getRed(px)
        g = getGreen(px)
        b = getBlue(px)
        greyColor = makeColor((r + g + b)/3)
        setColor(px, greyColor)

```

```
return newPic
```

```
def negPic(picture): #creates a negative copy of the original picture
    newPic = duplicatePicture(picture)
    for px in getAllPixels(newPic):
        r = getRed(px)
        g = getGreen(px)
        b = getBlue(px)
        negative = makeColor(255-r, 255-g, 255-b)
        setColor(px, negative)
    return newPic
```