#Collage
#Coded by Billie Wellman
#Submitted 9 March 2020

#FOR SEPIA IMG

greyscale prgm
def grayScale(picture):
    for px in getPixels(picture):
        newRed = getRed(px) * 0.299
        newGreen = getGreen(px) * 0.587
        newBlue = getBlue(px) * 0.114
        luminance = newRed + newGreen + newBlue
        setColor(px,makeColor(luminance,luminance,luminance))

sepia prgm
def sepiaTint(picture):
    grayScale(picture)
    for p in getPixels(picture):
        red = getRed(p)
        blue = getBlue(p)
        if (red < 63):
            red = red*1.1
            blue = blue*0.9
        if (red > 62 and red < 192):
            red = red*1.15
            blue = blue*0.85
        if (red > 191):
            red = red*1.08
        if (red > 255):
            red = 255
            blue = blue*0.93
        setBlue(p,blue)
        setRed(p,red)

#USED IN POSTERIZE, EDGE DETECT

luminance calculator
def luminance(pixel):
    r = getRed(pixel)
    g = getGreen(pixel)
b = getBlue(pixel)
return (r+g+b)/3

#FOR POSTERIZED IMG

def blackWhiteGrey(picture):
    for p in getPixels(picture):
        luminancepx = luminance(p)
        if (luminancepx < 80):
            setColor(p,black)
        elif (luminancepx > 120):
            setColor(p,white)
        else:
            color = makeColor(128,128,128)
            setColor(p,color)

#FOR LINE DRAWING IMG

def edgeDetect(picture):
    for px in getPixels(picture):
        x = getX(px)
        y = getY(px)
        if y < getHeight(picture)-1 and x < getWidth(picture)-1:
            botrt = getPixel(picture,x+1,y+1)
            thislum = luminance(px)
            brlum = luminance(botrt)
            if abs(brlum-thislum) > 10:
                setColor(px,black)
            if abs(brlum-thislum) <= 10:
                setColor(px,white)

#FOR RAINBOW LINES IMG

def horizontalLines(picture,start,count,color):
    for x in range(start,getHeight(picture),count):
        for y in range(0,getWidth(picture)):
            setColor(getPixel(picture,y,x),color)

def verticalLines(picture,start,count,color):
    for x in range(start,getWidth(picture),count):
        for y in range(0,getHeight(picture)):
            setColor(getPixel(picture,x,y),color)

def rainbowLines(picture):
    indigo = makeColor(29,0,51)
    violet = makeColor(106,13,173)
    verticalLines(picture,0,35,red)
    verticalLines(picture,5,35,orange)
    verticalLines(picture,10,35,yellow)
    verticalLines(picture,15,35,green)
    verticalLines(picture,20,35,blue)
    verticalLines(picture,25,35,indigo)
verticalLines(picture, 30, 35, violet)
horizontalLines(picture, 0, 35, red)
horizontalLines(picture, 5, 35, orange)
horizontalLines(picture, 10, 35, yellow)
horizontalLines(picture, 15, 35, green)
horizontalLines(picture, 20, 35, blue)
horizontalLines(picture, 25, 35, indigo)
horizontalLines(picture, 30, 35, violet)
return picture

# GENERAL USE

# image scaling
def scale(picture_in, picture_out, factor):
    inX = 0
    for outX in range(0, (getWidth(picture_in) * (1.0 * factor))):
        inY = 0
        for outY in range(0, (getHeight(picture_in) * (1.0 * factor))):
            inpx = getPixel(picture_in, int(inX), int(inY))
            color = getColor(inpx)
            setColor(getPixel(picture_out, outX, outY), color)
            inY = inY + (1.0 / factor)
        inX = inX + (1.0 / factor)

# image copying
def copyPicture(picture_in, picture_out, targ_x, targ_y):
    targetX = targ_x
    for sourceX in range(0, getWidth(picture_in)):
        targetY = targ_y
        for sourceY in range(0, getHeight(picture_in)):
            color = getColor(getPixel(picture_in, sourceX, sourceY))
            setColor(getPixel(picture_out, targetX, targetY), color)
            targetY = targetY + 1
        targetX = targetX + 1

# COPY SIGNATURE

# signature chromakey
def copyWithChromakey(picture_in, picture_out, targ_x, targ_y):
    targetX = targ_x
    for sourceX in range(0, getWidth(picture_in)):
        targetY = targ_y
        for sourceY in range(0, getHeight(picture_in)):
            pixel = getPixel(picture_in, sourceX, sourceY)
            lum = luminance(pixel)
            if (lum < 150):
                setColor(getPixel(picture_out, targetX, targetY), cyan)
        targetY = targetY + 1
    targetX = targetX + 1

# MAIN FUNCTION
def collage():
    picture = makePicture(getMediaPath("honolulu.jpg"))
    width = getWidth(picture)
    height = getHeight(picture)
    canvas = makeEmptyPicture(width, height)
    copyPicture(picture, canvas, 0, 0)
picture2 = makeEmptyPicture(int(width * 0.8), int(height * 0.8))
scale(picture, picture2, 0.8)
rainbowLines(picture2)
copyPicture(picture2, canvas, int(width * 0.1), int(height * 0.1))
picture3 = makeEmptyPicture(int(width * 0.6), int(height * 0.6))
scale(picture, picture3, 0.6)
sepiaTint(picture3)
copyPicture(picture3, canvas, int(width * 0.2), int(height * 0.2))
picture4 = makeEmptyPicture(int(width * 0.4), int(height * 0.4))
scale(picture, picture4, 0.4)
blackWhiteGrey(picture4)
copyPicture(picture4, canvas, int(width * 0.3), int(height * 0.3))
picture5 = makeEmptyPicture(int(width * 0.2), int(height * 0.2))
scale(picture, picture5, 0.2)
edgeDetect(picture5)
copyPicture(picture5, canvas, int(width * 0.4), int(height * 0.4))
signature = makePicture(getMediaPath("signature.jpeg"))
copyWithChromakey(signature, canvas, 750, 440)
show(canvas)