#Main Collage function
#adds crops of edited photos of Big Ben on top of each other to creat cascading effect

def collage():
    original = makePicture(getMediaPath("bigben.jpg"))
    canvas = duplicatePicture(original)
    signature = scaleAny(makePicture(getMediaPath("signature.jpeg")), 0.15)
    crop(grayPosterize(original), 20), canvas, int((getWidth(canvas)-(getWidth(original)-40))/2), int((getHeight(canvas)-(getHeight(original)-40))/2)
    copyPicture(crop(negative(original), 40), canvas, int((getWidth(canvas)-(getWidth(original)-80))/2), int((getHeight(canvas)-(getHeight(original)-80))/2))
    copyPicture(crop(redYellowShift(original, 3), 60), canvas, int((getWidth(canvas)-(getWidth(original)-120))/2), int((getHeight(canvas)-(getHeight(original)-120))/2))
    copyPicture(crop(blurMore(original), 80), canvas, int((getWidth(canvas)-(getWidth(original)-160))/2), int((getHeight(canvas)-(getHeight(original)-160))/2))
    explore(chromakey(signature, canvas))

#copies one pic on top of another
def copyPicture(picture_in, picture_out, targetX, targetY):
    for sourceX in range(0, getWidth(picture_in)):
        target_y = targetY
        for sourceY in range(0, getHeight(picture_in)):
            color = getColor(getPixel(picture_in, sourceX, sourceY))
            setColor(getPixel(picture_out, targetX, target_y), color)
            target_y = target_y + 1
            targetX = targetX + 1

#blurring function

Statement continues on the next line.
def blurMore(source):
    target = duplicatePicture(source)
    for x in range(2, getWidth(source) - 2):
        for y in range(2, getHeight(source) - 2):
            top = getPixel(source, x, y - 1)
            left = getPixel(source, x - 1, y)
            bottom = getPixel(source, x, y + 1)
            right = getPixel(source, x + 1, y)
            center = getPixel(target, x, y)
            topl = getPixel(source, x, y - 2)
            leftl = getPixel(source, x - 2, y)
            bottoml = getPixel(source, x, y + 2)
            rightl = getPixel(source, x + 2, y)

            newRed = (getRed(top) + getRed(left) + getRed(bottom) + getRed(right) + getRed(center) + getRed(topl) +
                       getRed(leftl) + getRed(bottoml) + getRed(rightl)) / 9
            newGreen = (getGreen(top) + getGreen(left) + getGreen(bottom) + getGreen(right) + getGreen(center) +
                       getGreen(topl) + getGreen(leftl) + getGreen(bottoml) + getGreen(rightl)) / 9
            newBlue = (getBlue(top) + getBlue(left) + getBlue(bottom) + getBlue(right) + getBlue(center) + getBlue(topl) +
                       getBlue(leftl) + getBlue(bottoml) + getBlue(rightl)) / 9

            setColor(center, makeColor(newRed, newGreen, newBlue))
    return target

# posterized the pic to only include white and black
def grayPosterize(pic):
    grayCanvas = makeEmptyPicture(getWidth(pic), getHeight(pic))
    for p in getPixels(pic):
        r = getRed(p)
        g = getGreen(p)
        b = getBlue(p)
        luminance = (r + g + b) / 3
        if luminance < 92:
            setColor(getPixel(grayCanvas, getX(p), getY(p)), black)
        if luminance >= 92:
            setColor(getPixel(grayCanvas, getX(p), getY(p)), white)
    return grayCanvas

# returns a negative of given pic
def negative(pic):
    negativeCanvas = makeEmptyPicture(getWidth(pic), getHeight(pic))
    for px in getPixels(pic):
        red = getRed(px)
        green = getGreen(px)
        blue = getBlue(px)

        newRed = 255 - red
        newGreen = 255 - green
        newBlue = 255 - blue

        setColor(getPixel(negativeCanvas, getX(px), getY(px)), makeColor(newRed, newGreen, newBlue))
    return negativeCanvas
blue=getBlue(px)
negColor=makeColor(255-red, 255-green, 255-blue)
setColor(getPixel(negativeCanvas, getX(px), getY(px)), negColor)
return(negativeCanvas)

# changes colors to red and yellow, as well as shifts every other line to different direction by given amount
def redYellowShift(pic, shift):
crazyCanvas = makeEmptyPicture(getWidth(pic), getHeight(pic))
towerBrown = makeColor(140, 100, 80)
skyBlue = makeColor(70, 120, 150)
targetX = 0
for sourceX in range(0, getWidth(pic)):
    targetY = 0
    for sourceY in range(0, getHeight(pic)):
        color = getColor(getPixel(pic, sourceX, sourceY))
        if distance(color, towerBrown) < 60:
            setColor(getPixel(crazyCanvas, targetX, targetY), red)
        elif distance(color, skyBlue) < 50:
            setColor(getPixel(crazyCanvas, targetX, targetY), yellow)
        targetY = targetY + 1
    targetX = targetX + 1
final = duplicatePicture(crazyCanvas)
for x in range(0, getWidth(crazyCanvas)-shift):
    for y in range(0, getHeight(crazyCanvas), 2):
        color = getColor(getPixel(crazyCanvas, x, y))
        setColor(getPixel(final, x + shift, y), color)
for x in range(shift, getWidth(crazyCanvas)):
    for y in range(1, getHeight(crazyCanvas), 2):
        color = getColor(getPixel(crazyCanvas, x, y))
        setColor(getPixel(final, x - shift, y), color)
return(final)

# crops given pixel amount off of each side of border
def crop(pic, borderCut):
cropCanvas = makeEmptyPicture(getWidth(pic)-(2*borderCut), getHeight(pic)-(2*borderCut))
targetX = 0
for sourceX in range(borderCut, getWidth(pic)-borderCut):
    targetY = 0
    for sourceY in range(borderCut, getHeight(pic)-borderCut):
        color = getColor(getPixel(pic, sourceX, sourceY))
        setColor(getPixel(cropCanvas, targetX, targetY), color)
        targetY = targetY + 1
    targetX = targetX + 1

# Statement continues on the next line.
return(cropCanvas)

# copies the signature onto the final canvas

def chromakey(source, canvas):
    black = makeColor(0, 0, 0)
    for sourceX in range(0, getWidth(source)):
        for sourceY in range(0, getHeight(source)):
            color = getColor(getPixel(source, sourceX, sourceY))
            if distance(color, black) < 190.0:
                setColor(getPixel(canvas, sourceX, sourceY), green)
    return canvas

# used to scale down signature pic

def scaleAny(picIn, scaleFactor):
    picOut = makeEmptyPicture(int(getWidth(picIn)*scaleFactor), int(getHeight(picIn)*scaleFactor))
    sourceX = 0
    for targetX in range(0, int((getWidth(picIn))*scaleFactor)):
        sourceY = 0
        for targetY in range(0, int((getHeight(picIn)*scaleFactor))):
            color = getColor(getPixel(picIn, int(sourceX), int(sourceY)))
            setColor(getPixel(picOut, targetX, targetY), color)
            sourceY = sourceY + 1.0/scaleFactor
            sourceX = sourceX + 1.0/scaleFactor
    return picOut