def collage():
    # starting images
    picture = getMediaPath('butterfly1.jpg')
    source1 = scaleDown1(makePicture(picture), 2)
    source2 = scaleDown1(makePicture(picture), 2)
    source3 = scaleDown1(makePicture(picture), 2)
    source4 = scaleDown1(makePicture(picture), 2)
    source5 = scaleDown1(makePicture(picture), 2)
    emptypic = makeEmptyPicture(4 * getWidth(source1), 3 * getHeight(source1) - 8)
    h = getHeight(source1)
    w = getWidth(source1)
    H = getHeight(emptypic)
    W = getWidth(emptypic)
    # creates altered images
    image1 = source1
    image2 = invertColor(source2)
    image3 = grayscale(source3)
    image4 = edgeDetect(source4)
    image5 = sepia(source5)
    # copying of altered images onto canvas
    tarX = 0
    tarY = H - h - 5
    copy(image1, emptypic, tarX, tarY)
    copy(image2, emptypic, tarX + w, tarY)
    copy(image3, emptypic, tarX + 2 * w, tarY)
    copy(image5, emptypic, tarX + 3 * w, tarY)
    copy(image4, emptypic, 0, getHeight(source1))
    copy(image4, emptypic, getWidth(source1), getHeight(source1))
    copy(image4, emptypic, 2 * getWidth(source1), getHeight(source1))
    copy(image4, emptypic, 3 * getWidth(source1), getHeight(source1))
    newpic = mirrorHorizontal(emptypic)
    repaint(newpic)
    return newpic

def copy(source, target, tarX, tarY):
    targetX = tarX
    for sourceX in range(0, getWidth(source)):
targetY=tarY
for sourceY in range(0, getHeight(source)):
    px=getPixel(source, sourceX, sourceY)
    tx=getPixel(target, targetX, targetY)
    setColor(tx, getColor(px))
    targetY=targetY+1
    targetX=targetX+1

#programs to edit pictures

def sepia(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)
    return picture

def grayscale(picture):
    for p in getPixels(picture):
        intensity=(getRed(p)+getGreen(p)+getBlue(p))/3
        setColor(p, makeColor(intensity, intensity, intensity))
    return picture

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

def edgeDetect(source):
    for px in getPixels(source):
        x=getX(px)
        y=getY(px)
        if y< getHeight(source)-1 and x< getWidth(source)-1:
            botrt=getPixel(source, 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)
    return source
def invertColor(source):
    for px in getPixels(source):
        red = getRed(px)
        green = getGreen(px)
        blue = getBlue(px)
        negColor = makeColor(255-red, 255-green, 255-blue)
        setColor(px, negColor)
    return source

def scaleDown1(picture,f):
    small_picture = makeEmptyPicture(int(getWidth(picture)/f), int(getHeight(picture)/f))
    scaleDown(picture, small_picture, f)
    return small_picture

def scaleDown(picture_in, picture_out,f):
    sourceX = 0
    for targetX in range(0, int(getWidth(picture_in)/f)):
        sourceY = 0
        for targetY in range(0, int(getHeight(picture_in)/f)):
            color = getColor(getPixel(picture_in, sourceX, sourceY))
            setColor(getPixel(picture_out, targetX, targetY), color)
            sourceY = sourceY+f
        sourceX = sourceX+f
    return picture_out

#mirrors collage horizontally
def mirrorHorizontal(source):
    mirrPoint = getHeight(source)/2
    height = getHeight(source)
    for x in range(0, getWidth(source)):
        for y in range(0, mirrPoint):
            topPixel = getPixel(source, x, y)
            bottomPixel = getPixel(source, x, height-y-1)
            bottomColor = getColor(bottomPixel)
            setColor(topPixel, bottomColor)
    show(source)
    return source

##I tried to chromakey my signature, but couldn't successfully complete it
def chromakey():
    picture = pickAFile()
    pic = makePicture(picture)
    canvas = makeEmptyPicture(getWidth(pic), getHeight(pic))
    newColor = makeColor(75,74,79)
    targetx = 570
    for x in range(60,210):
        targety = 950
        for y in range(0, 55):
            color = getColor(getPixel(pic, x, y))
            if distance(color, newColor)<160:
                setColor(getPixel(canvas, targetx, targety), color)
        targety = targety+1
        targetx = targetx+1
    show (pic)

# means the line is continued on the next line.