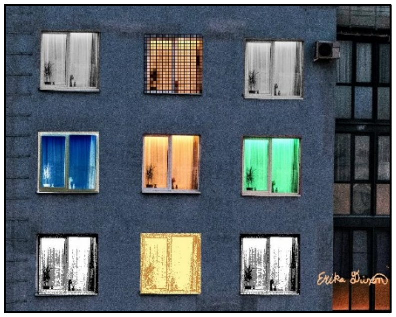
Erika Dixon

Completed



Originals

```
#Erika Dixon 3-15-2021

#main function to create collage
def collage():
  #set up background
  canvas = makeEmptyPicture(560, 640)
  originalBackground = makePicture(getMediaPath("building.jpg"))
  background = makeEmptyPicture(int(getWidth(originalBackground)/5), int(getHeight(originalBackground)/5))
  scale(originalBackground, background,int(getWidth(originalBackground)/5),int(getHeight(originalBackground)/5),5)
  #add pixelated picture to collage
  pixelatedPic = pixelateWithLines(getWindow())
  copy(pixelatedPic, background, 268, 55)
  #add grayscale pictures to collage
  grayPic = grayScale(getWindow())
  copy(grayPic, background, 65, 46)
  copy(grayPic, background, 462, 63)
  #add edge detection picture to collage
  edgePic = edgeDetect(getWindow())
  copy(edgePic, background, 262, 441)
  #add negative picture to collage
  negativePic = negative(getWindow())
  copy(negativePic, background, 63, 244)
  #add color swapped picture to collage
  swapPic = colorSwap(getWindow())
  copy(swapPic, background, 457, 251)
  #add posterized picture to collage
  posterizePic = posterize(getWindow())
  copy(posterizePic, background, 62, 442)
  copy(posterizePic, background, 454, 442)
  #add signature to collage and display collage
  finalCollage=(chromakey(background))
  explore(finalCollage)

#function that pixelates an image and adds lines between pixel "blocks"
def pixelateWithLines(picture):
  smallPic = makeEmptyPicture(int(getWidth(picture)/6), int(getHeight(picture)/6))
  scale(picture, smallPic, int(getWidth(picture)/6), int(getHeight(picture)/6), 6)
  bigPic = makeEmptyPicture(getWidth(smallPic)*6, getHeight(smallPic)*6)
  scale(smallPic, bigPic, getWidth(smallPic)*6, getHeight(smallPic)*6, 1.0/6)
  for x in range(0, getWidth(bigPic), 6):
    for y in range(0, getHeight(bigPic)):
      p = getPixel(bigPic, x, y)
      setColor(p, black)
  for y in range(0, getHeight(bigPic), 6):
    for x in range(0, getWidth(bigPic)):
      p = getPixel(bigPic, x, y)
      setColor(p, black)
  return bigPic

#function that uses grayscale effect on 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

#function that uses edge detection on a picture
def edgeDetect(picture):
  for p in getPixels(picture):
    x = getX(p)
    y = getY(p)
    lightColor = makeColor(255, 245, 165)
    darkColor = makeColor(205, 130, 60)
    if (y < getHeight(picture)-1) and (x < getWidth(picture)-1):
      botrt = getPixel(picture, x+1, y+1)
      thislum = (getRed(p) + getGreen(p) + getBlue(p))/3
      brlum = (getRed(botrt) + getGreen(botrt) + getBlue(botrt))/3
      if abs(brlum-thislum) > 24:
        setColor(p, darkColor)
      if abs(brlum-thislum) <= 24:
        setColor(p, lightColor)
  return picture
```

```
#function that produces the negative version of a picture
def negative(picture):
  for p in getPixels(picture):
    negativeColor = makeColor(255 - getRed(p), 255 - getGreen(p), 255 - getBlue(p))
    setColor(p, negativeColor)
  return picture

#function that swaps the colors of picture
def colorSwap(picture):
  for p in getPixels(picture):
    redValue = getRed(p)
    blueValue = getBlue(p)
    greenValue = getGreen(p)
    setRed(p, blueValue)
    setBlue(p, greenValue)
    setGreen(p, redValue)
  return picture

#function that posterizes a picture
def posterize(picture):
  for p in getPixels(picture):
    luminance = (getRed(p) + getGreen(p) + getBlue(p))/3
    if luminance < 105:
      setColor(p, black)
    if 105 <= luminance <= 175:
      setColor(p, gray)
    if luminance > 175:
      setColor(p, white)
  return picture

#function that scales down and copies a window
def getWindow():
  picture = makePicture(getMediaPath("window.jpg"))
  canvas = makeEmptyPicture(int(getWidth(picture)/5), int(getHeight(picture)/5))
  scaledPicture = scale(picture, canvas, int(getWidth(picture)/5), int(getHeight(picture)/5), 5)
  return scaledPicture

#function that recolors signature and places it onto collage
def chromakey(background):
  name = makePicture(getMediaPath("name.jpg"))
  scaledName = makeEmptyPicture(int(getWidth(name)/12), int(getHeight(name)/12))
  scale(name, scaledName, int(getWidth(name)/12), int(getHeight(name)/12), 12)
  newBackground = makeEmptyPicture(getWidth(background), getHeight(background))
  for p in getPixels(scaledName):
    if getRed(p) + getGreen(p) < getBlue(p):
      setColor(p, makeColor(255, 185, 120))
  copy(scaledName, newBackground, 600, 500)
  for p in getPixels(newBackground):
    if getRed(p)>240 and getGreen(p)>240 and getBlue(p)>240:
      bgpx = getPixel(background, getX(p), getY(p))
      bgcol = getColor(bgpx)
      setColor(p, bgcol)
  return newBackground

#function that copies a picture
def copy(picture, canvas, targetX, targetY):
  tarX = targetX
  for picX in range(0, getWidth(picture)):
    tarY = targetY
    for picY in range(0, getHeight(picture)):
      color = getColor(getPixel(picture, picX, picY))
      setColor(getPixel(canvas, tarX, tarY), color)
      tarY = tarY + 1
    tarX = tarX + 1
  return canvas

#function that scales a picture
def scale(picture_in, picture_out, width, height, add):
  tarX = 0
  for x in range(0, width):
    tarY = 0
    for y in range(0, height):
      color = getColor(getPixel(picture_in, int(tarX), int(tarY)))
      setColor(getPixel(picture_out, x, y), color)
      tarY = tarY + add
    tarX = tarX + add
  return picture_out
```