## Image range reflect

def reflect(filename):

"""Implement reflect(), described above."""

image = SimpleImage(filename)

out = SimpleImage.blank(image.width, image.height\*2)

for y in range(image.height):

for x in range(image.width):

# Copy over top half of image

pixel = image.get\_pixel(x, y)

pixel\_out = out.get\_pixel(x, y) # the key line

pixel\_out.red = pixel.red

pixel\_out.green = pixel.green

pixel\_out.blue = pixel.blue

# Create the bottom mirror image

pixel = image.get\_pixel(x, y)

pixel\_out = out.get\_pixel(x, out.height - y - 1) # the key line

pixel\_out.red = pixel.red

pixel\_out.green = pixel.green

pixel\_out.blue = pixel.blue

return out

## Image range purple

def purple(filename, margin):

"""Implement purple(), described above."""

image = SimpleImage(filename)

image = SimpleImage(filename)

# Create out image, n pixels wider than original

out = SimpleImage.blank(image.width + 2\*margin, image.height)

# Create the margin-pixel aqua stripe

for y in range(image.height):

for x in range(out.width):

pixel\_out = out.get\_pixel(x, y)

pixel\_out.green = 0

# Copy the original over - make drawing to guide code here

for y in range(image.height):

for x in range(image.width):

pixel = image.get\_pixel(x, y)

pixel\_out = out.get\_pixel(x + margin, y) # the key line

pixel\_out.red = pixel.red

pixel\_out.green = pixel.green

pixel\_out.blue = pixel.blue

return out

## Image range squeeze height

def squeeze\_height(filename, n):

"""Implement squeeze\_height(), described above."""

image = SimpleImage(filename)

image = SimpleImage(filename)

out = SimpleImage.blank(image.width, image.height // n)

# Here looping x,y over out, not original

for y in range(out.height):

for x in range(out.width):

pixel\_out = out.get\_pixel(x, y)

# your code here - compute pixel\_in based on x,y

pixel\_in = image.get\_pixel( x, y\*n )

pixel\_out.red = pixel\_in.red

pixel\_out.green = pixel\_in.green

pixel\_out.blue = pixel\_in.blue

return out