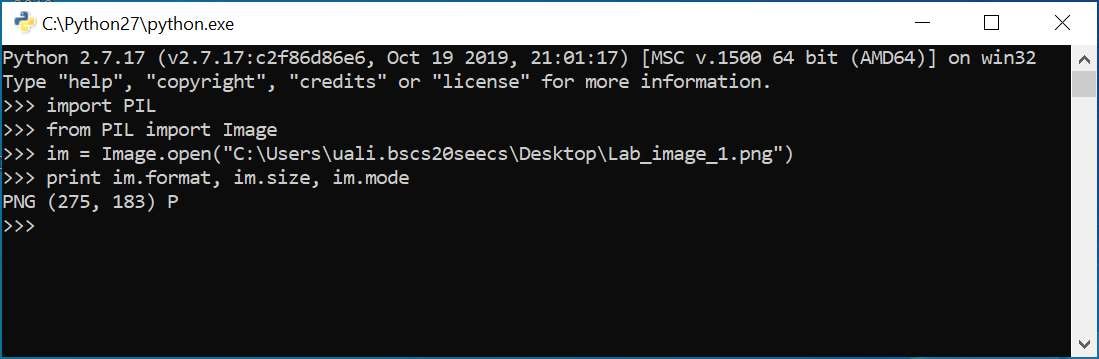


EE433 – Digital Image Processing Lab 2 – Intro. To Python Submitted

Task 1:

# Part 1 – Reading images using Python Image Library (PIL)



# Part 2 – Writing images (converting to JPEG) using Python Image Library (PIL)

|  |
| --- |
| import os, sys from PIL import Image  for infile in sys.argv[1:]:  f, e = os.path.splitext(infile) outfile = f + ".jpg" if infile != outfile: try:  Image.open(infile).save(outfile) except IOError:  print ("cannot convert", infile) |

Task 3

# Part (a) – Converting an RGB image to Grayscale using PIL

Code:

from PIL import Image

im = Image.open("Lab\_image\_2.jpeg") print(im.mode) im\_grayscale = im.convert('L') print(im\_grayscale.mode) im\_grayscale.save("Lab\_image\_2\_grayscale.jpeg") im\_grayscale.show()

Output:

Original Grayscaled

# Part (b) – Applying smoothening filter to an image

Code:

from PIL import Image, ImageFilter

im = Image.open("Lab\_image\_2.jpeg") im\_sharpened = im.filter(ImageFilter.SMOOTH) im\_sharpened.save("Lab\_image\_2\_sharpened.jpeg") im\_sharpened.show() im.show()

Output:

Original Smoothened

# Part (c) – Applying sharpening filter to an image

Code:

from PIL import Image, ImageFilter

im = Image.open("Lab\_image\_2.jpeg") im\_sharpened = im.filter(ImageFilter.SHARPEN) im\_sharpened.save("Lab\_image\_2\_sharpened.jpeg") im\_sharpened.show() im.show()

Output:

Original Sharpened

# Part (d) – Grayscaling an image without using PIL’s built-in function

Code:

|  |
| --- |
| from PIL import Image import numpy  im = Image.open("Lab\_image\_2.jpeg") im\_numpy = numpy.array(im) new\_im\_numpy = numpy.zeros((im.size[0], im.size[1])) print(len(new\_im\_numpy)) print(len(new\_im\_numpy[0]))    #Using formula gray\_pixel = 0.2989 \* R + 0.5870 \* G + 0.1140 \* B i = 0 for column in im\_numpy:  j = 0 for pixel in column:  pixel[0] = pixel[0] \* 0.2989 pixel[1] = pixel[1] \* 0.5870 pixel[2] = pixel[2] \* 0.1140 new\_pixel = pixel[0] + pixel[1] + pixel[2] if (i < im.size[0]) and (j < im.size[1]):  new\_im\_numpy[i, j] = new\_pixel j += 1 i += 1  im\_grayscaled = Image.fromarray(new\_im\_numpy) if im\_grayscaled.mode != 'RGB':  im\_grayscaled = im\_grayscaled.convert('RGB') im\_grayscaled.show() im\_grayscaled.save("Lab\_image\_2\_manual\_grayscale.jpeg") |

Output:

Original Manually Grayscaled