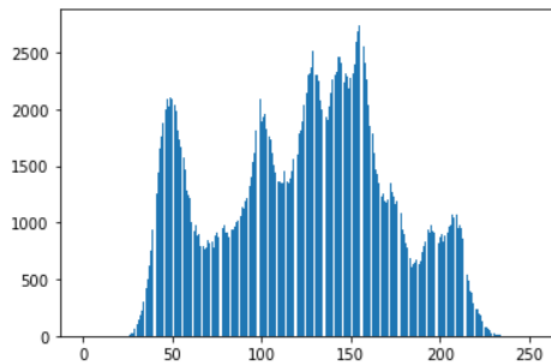


I. Environment Setup**Language: Python 3 (on jupyter)****Library: numpy, PIL, matplotlib**

```
1 import numpy as np
2 from PIL import Image, ImageDraw
3 import matplotlib.pyplot as plt
```

II. Q1: original image and its histogram**Step 1: Read image using Image.open****Step 2: traverse all the elements and calculate the number of pixels in different intensity.****Step 3: use plt draw the histogram.****output image: original lena.bmp****code:**

```
1 # (a) original image and its histogram
2
3 img = Image.open("lena.bmp")
4 width, height = img.size
5
6 x = np.arange(256)
7 y = np.zeros(256)
8 for c in range(width):
9     for r in range(height):
10         y[img.getpixel((c,r))] +=1
11 plt.bar(x,y)
12 plt.show()
```

III. Q2: image with intensity divided by 3 and its histogram

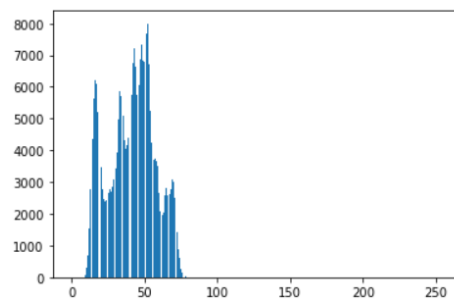
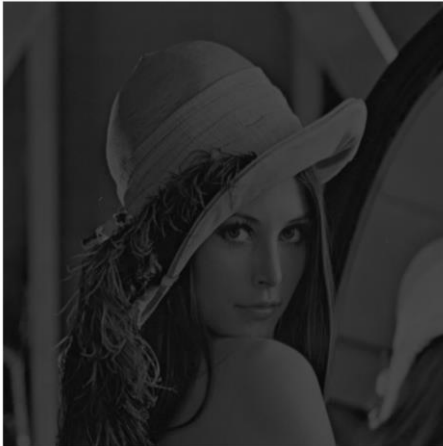
Step 1: read image using Image.open()

Step 2: traverse all the pixels, get pixel value, divided by 3 and calculate the number of pixels in different intensity.

Step 3: create a new image to store the new value divided by 3 from Step 2.

Step 3: use pyplot to draw histogram

output image



```
1 # (b) image with intensity divided by 3 and its histogram
2
3 original = Image.open("lena.bmp")
4 width, height = img.size
5
6 x = np.arange(256)
7 dividedHistogram = np.zeros(256)
8
9 for c in range(width):
10     for r in range(height):
11         index = round(img.getpixel((c,r))/3)
12         dividedHistogram[index] +=1
13
14 dividedImage = Image.new("L", original.size)
15 dividedImageArray = dividedImage.load()
16
17 for c in range(width):
18     for r in range(height):
19         dividedImageArray[c,r] = round(img.getpixel((c,r))/3)
20
21 dividedImage.save("DividedImage.bmp")
22 dividedImage.show()
23 plt.bar(x,y)
24 plt.show()
```

IV. **Q3: image after applying histogram equalization to (b) and its histogram**

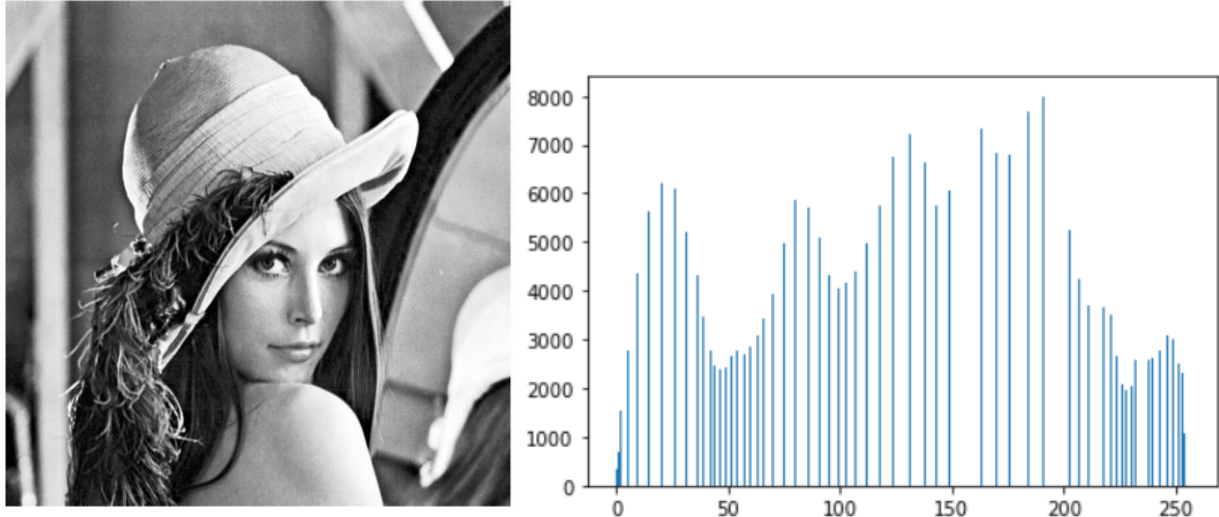
Step 1: read image from (b) using Image.open()

Step 2: get divided image histogram count

Step 3: calculate the equalization histogram

Step 3: make new equalized Image

output image:



```
1 # (c) image after applying histogram equalization to (b) and its histogram
2
3 # Load basic parameter
4 dividedImage = Image.open("DividedImage.bmp")
5 width, height = dividedImage.size
6 totalNumberOfPixels = width * height
7
8
9 s = np.zeros(256)
10 dividedImageHis = np.zeros(256)
11
12 # get divided image histogram count
13 for c in range(width):
14     for r in range(height):
15         dividedImageHis[dividedImage.getpixel((c,r))] += 1
16
17 # calculate the equalization histogram
18 for i in range(256):
19     accumulation[i] = np.sum(dividedImageHis[0:i+1])
20     s[i] = accumulation[i] * 255 / totalNumberOfPixels
21
22 # make new equalizedImage
23 equalizedImage = Image.new("L", dividedImage.size)
24 for c in range(width):
25     for r in range(height):
26         equalizedImage.putpixel((c,r), int(s[dividedImage.getpixel((c,r))]))
27 equalizedImage.save("EqualizedImage.bmp")
28 equalizedImage.show()
29
30 # make the histogram of equalization image
31 equalizedHis = np.zeros(256)
32 for c in range(width):
33     for r in range(height):
34         equalizedHis[equalizedImage.getpixel((c,r))] += 1
35
36 plt.bar(x,equalizedHis)
37 plt.show()
```