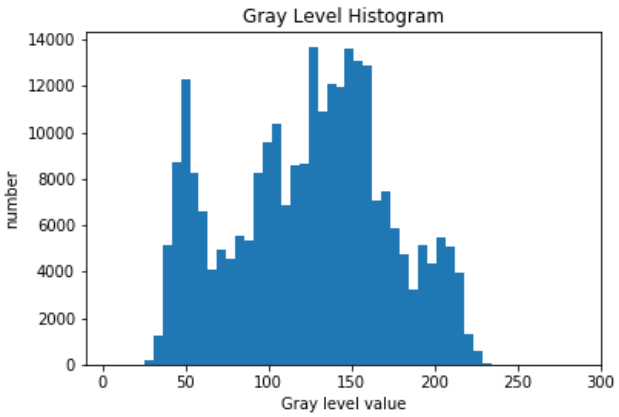
(a) original image and its histogram

```
#Histogram
def draw_hist(myList,Title,Xlabel,Ylabel):
    plt.hist(myList, 40)
    plt.xlabel(Xlabel)
    plt.ylabel(Ylabel)
    plt.title(Title)
    plt.xlim(xmin = -10)
    plt.xlim(xmax = 300)
```

```
#a.original image and its histogram

img = cv2.imread('lena.bmp')
data = []
for i in range(512):
    for j in range(512):
        data.append(int(img[i][j][0]))
hist = draw_hist(data,'Gray Level Histogram','Gray level value','number')
plt.savefig("original.png")
```

- 使用迴圈讀取lena.bmp的array，將每個vector的第0個value取出並存入data中。完成存取個value後，使用一開始用matplotlib.pyplot套件組和成的draw_hist() function將data中的value以長條圖的方式呈現並儲存。

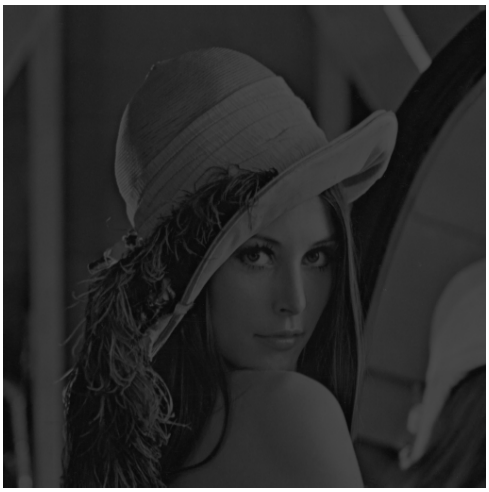


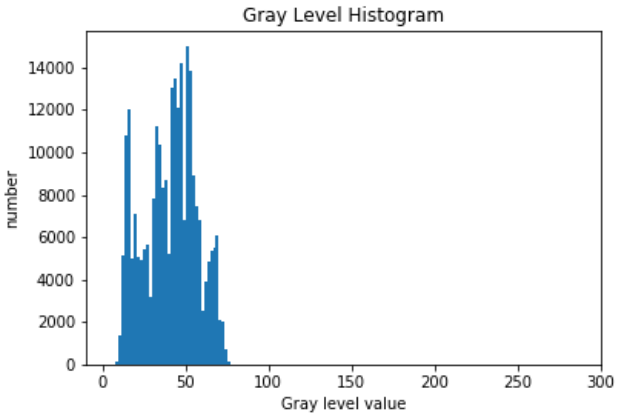
(b) image with intensity divided by 3 and its histogram

```
#b.image with intensity divided by 3 and its histogram

img2 = img/3
data2 = []
for i in range(512):
    for j in range(512):
        n = int(img2[i][j][0])
        data2.append(n)
        img2[i][j] = [n, n, n]
data2 = np.array(data2)
draw_hist(data2,'Gray Level Histogram','Gray level value','number')
plt.savefig("divided-by-3.png")
cv2.imwrite('intensity-divided-by-3.bmp', img2)
```

- 將img中的所有value除以3後存入img2中（此時的values不一定都為整數），再透過雙層迴圈將img2的所有value進行四捨五入，同時也將四捨五入後的value存入data2中作為畫直方圖時使用。跟(a)一樣使用draw_hist()將data2中的values以直方圖的方式呈現並儲存。使用cv2.write將img2中的資料儲存為"intensity-divided-by-3.bmp"。





(c) image after applying histogram equalization to (b) and its histogram

```
#c.image after applying histogram equalization to (b) and its histogram

n = img.shape[0]*img.shape[1] #total pixel number
gl_list = np.unique(data2) #all unique gray level value
eq_list = np.zeros(len(gl_list)) #new gray level value

cumul = 0
for i in range(len(gl_list)):
    num = len(data2[data2==gl_list[i]])
    cumul += num
    eq_list[i] = 255.*cumul/n
```

- 將img的shape[0]和shape[1]相乘以計算照片中的總pixel數量並儲存至n中。
- 使用numpy.unique尋找所有data2裡出現過的values並儲存至gl_list中。
- 先開啟一個一維的陣列eq_list(長度和gl_list相同)並初始化所有值為0，作為儲存gl_list中所有gray level value在equalization後對照轉換的新gray level value。
- 將變數cumul初始化為0，依序尋找每個gl_list裡的value出現在data2的次數，累加至cumul中，最後將255.*cumul再除以n(255. * 累積次數 / 總pixel數)作為equalization後的更新值依序並存入eq_list中。

```
for i in range(len(gl_list)):
    img2 = np.where(img2 == gl_list[i], eq_list[i], img2)
cv2.imwrite('after-equalization.bmp', img2)
```

- 使用更新完的eq_list和gl_list對img2進行equalization，在img2上依序尋找gl_list中的數值，將找到的數值更新成eq_list中相對應的數值(使用index作為對應)。

```
data3 = []
for i in range(512):
    for j in range(512):
        data3.append(img2[i][j][0])
draw_hist(data3,'Gray Level Histogram','Gray level value','number')
plt.savefig("after-equalization.png")
```

- 將img2以array的方式讀取，把每個vector中第0個value儲存到data3中。完成後，和(a)、(b)一樣使用draw_hist()將data2中的values以直方圖的方式呈現並儲存。
- 使用cv2.write將更新完的img2中的資料儲存為"after-equalization.png"。