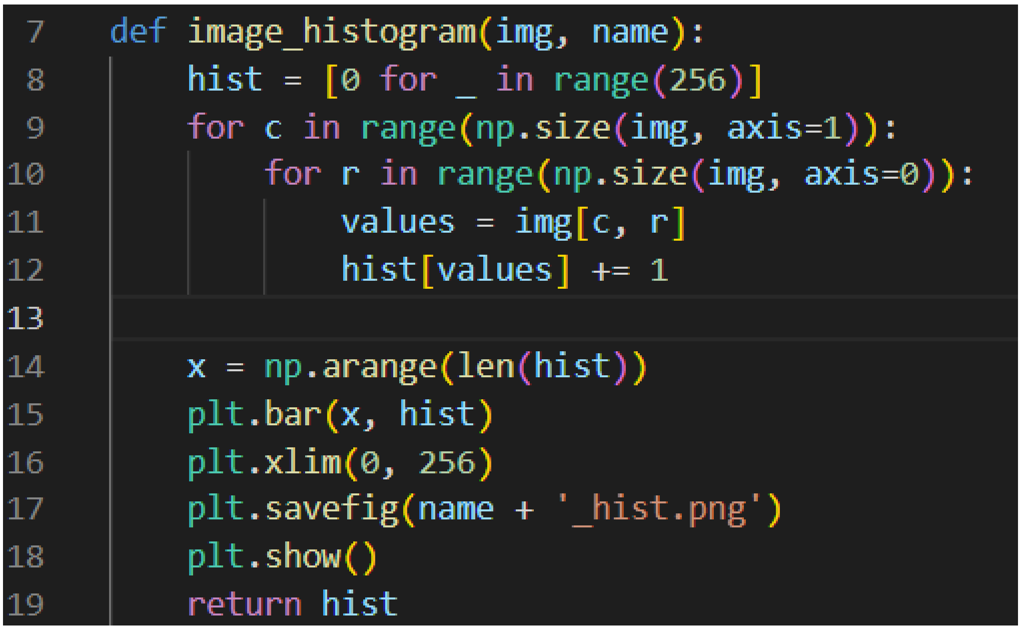
資工所碩一　R10922123 周昱豪

Code:

I use the python cv2 library to read the image，so the parameter img is numpy array, so I can do some array operation.

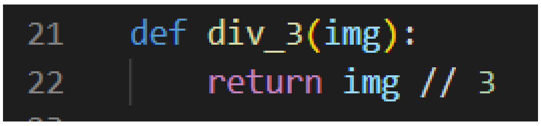
Part(a): Same as the HW2



```python
def image_histogram(img, name):
    hist = [0 for _ in range(256)]
    for c in range(np.size(img, axis=1)):
        for r in range(np.size(img, axis=0)):
            values = img[c, r]
            hist[values] += 1

    x = np.arange(len(hist))
    plt.bar(x, hist)
    plt.xlim(0, 256)
    plt.savefig(name + '_hist.png')
    plt.show()
    return hist
```

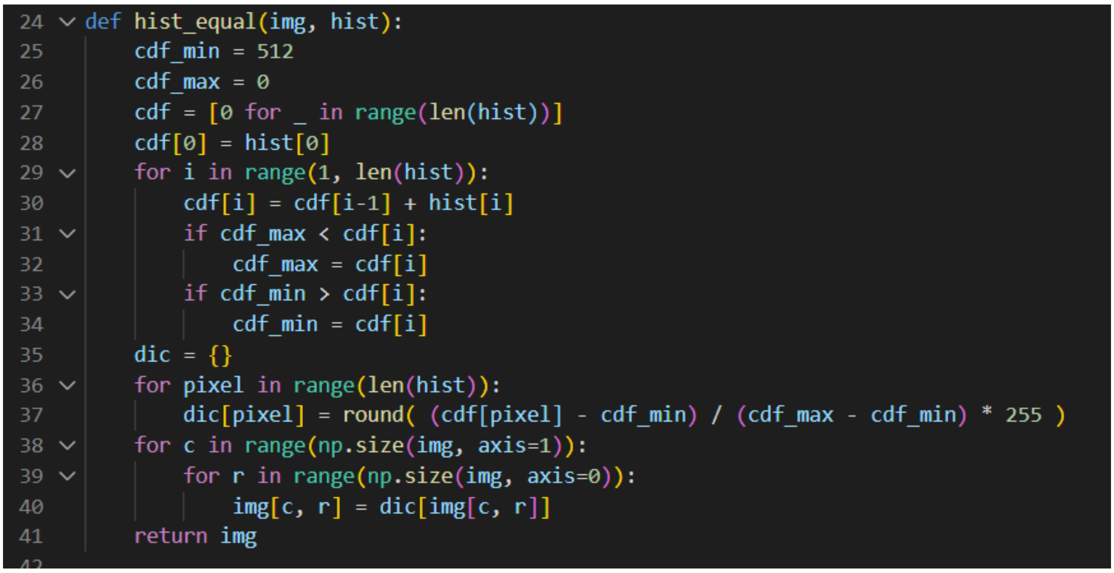
Part(b): Simply divided by 3.



```python
def div_3(img):
    return img // 3
```

Part(c):

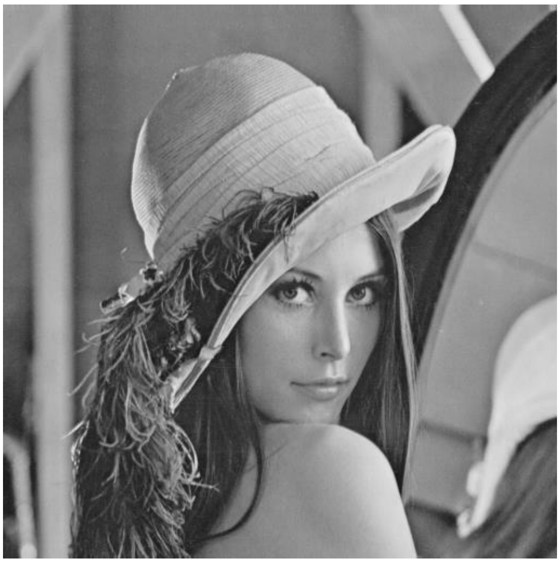
1. Calculate CDF with each pixel.
2. $h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{cdf_{max} - cdf_{min}} \times (L-1)\right)$, v is the original pixel value, and L is the number of grey levels used.

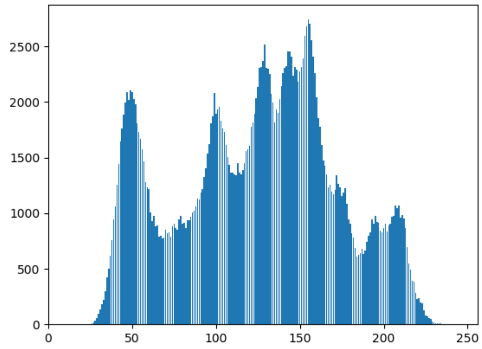


```
def hist_equal(img, hist):
    cdf_min = 512
    cdf_max = 0
    cdf = [0 for _ in range(len(hist))]
    cdf[0] = hist[0]
    for i in range(1, len(hist)):
        cdf[i] = cdf[i-1] + hist[i]
        if cdf_max < cdf[i]:
            cdf_max = cdf[i]
        if cdf_min > cdf[i]:
            cdf_min = cdf[i]
    dic = {}
    for pixel in range(len(hist)):
        dic[pixel] = round( (cdf[pixel] - cdf_min) / (cdf_max - cdf_min) * 255 )
    for c in range(np.size(img, axis=1)):
        for r in range(np.size(img, axis=0)):
            img[c, r] = dic[img[c, r]]
    return img
```

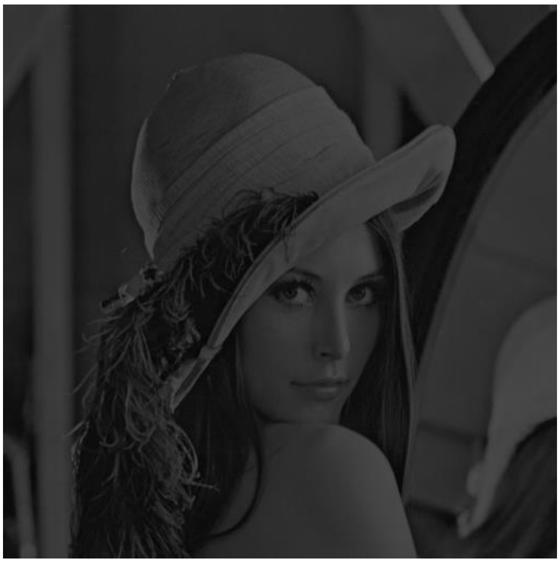
Result

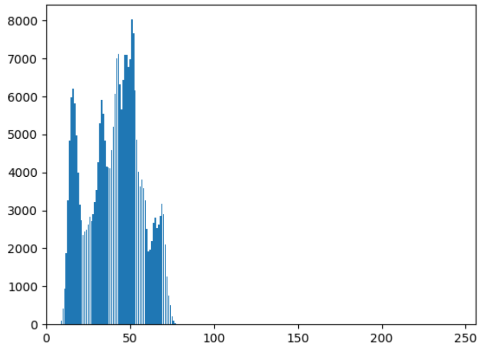
Part(a)



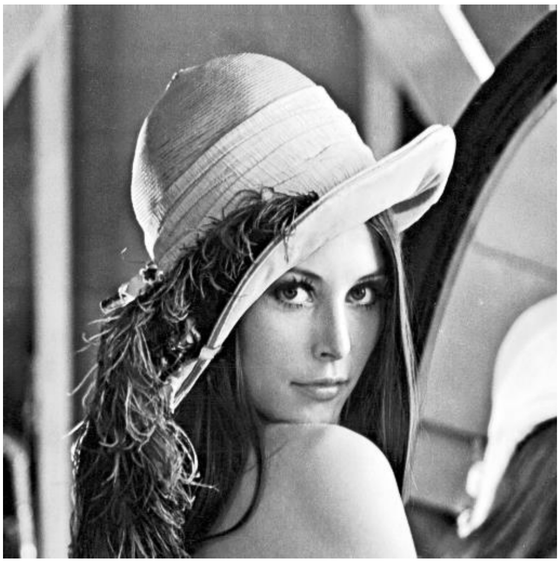


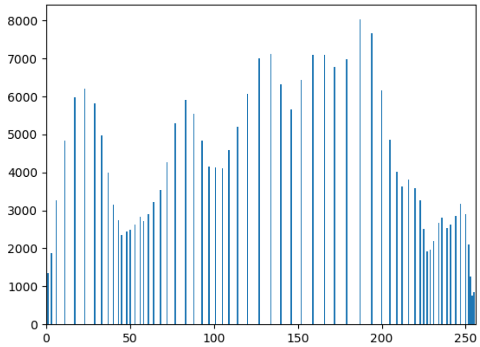
Part(b)




8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

Part(c)




8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250