1. The algorithm

- 先將 lena 各 pixel 除以 3,在利用, $s_k = 255 \sum_{j=0}^{k} \frac{n_j}{n}$ 算出各點之值，及可求出 Hsitogram Equalization 之圖,在利用前次作業的功能,求出其 Histogram

2. Parameters

- Mat src;//輸入之圖
- int i,j;//for 迴圈使用
- int row = src.rows;//輸入圖片 row
- int col = src.cols; //輸入圖片 col
- Mat divi(row, col, CV_8U);//輸出圖片
- double nj = 0;//計算 nj
- int sk[256] = {0};//紀錄 sk

3. Principal code fragment

```
1 #include <iostream>
2 #include <opencv2/core/core.hpp>
3 #include <opencv2/highgui/highgui.hpp>
4 #include <cv.h>
5 using namespace std;
6 using namespace cv;
7
8 int main(int argc,char** argv )
9 {
10     Mat src;
11     src = imread("lena.bmp",CV_LOAD_IMAGE_GRAYSCALE);
12
13     int i,j;
14     int row = src.rows;
15     int col = src.cols;
16
17     Mat divi(row, col, CV_8U);
18     int pixel[256] = {0};
19
20
21     for(i=0; i<col; i++)
```

```
    {
        for(j=0; j<row; j++)
        {
            divi.at<unsigned char>(i,j) = (src.at<unsigned char>(i,j))/3;
            pixel[divi.at<unsigned char>(i,j)]++;
            //cout << pixel[divi.at<unsigned char>(i,j)] << endl;

        }
    }


    namedWindow( "divi3", WINDOW_AUTOSIZE  );
    imshow( "divi3", divi  );
    imwrite("divi3.bmp", divi );




    double nj = 0;
    int sk[256] = {0};
    for(i=0; i<256; i++)
    {
        nj += pixel[i];
        sk[i] = 255*(nj/262144);
    } 48


    for(i=0; i<col; i++)
    {
        for(j=0; j<row; j++)
        {
```

```
56            divi.at<unsigned char>(i,j) = sk[divi.at<unsigned char>(i,j)];
57        }
58    }
59
60
61    namedWindow( "answer", WINDOW_AUTOSIZE   );
62    imshow( "answer", divi   );
63    imwrite("answer.bmp", divi );
64
65
66    int count[256] = {0};
67    for(i = 0; i<512; i++)
68    {
69        for(j=0; j<512; j++)
70        {
71            count[divi.at<unsigned char>(i,j)]++;
72        }
73    }
74
75
76    int maxCount = 0;
77    for(i=0; i<256; i++)
78    {
79        if(count[i]>maxCount)
80        {
81            maxCount = count[i];
82        }
83    }
84    cout <<"maxC:"<< maxCount;
85
86    Mat app(803, 256, CV_8U);
87
88    for(i=0; i<256; i++)
```

```
89    {
90       for(j=0; j<803; j++)
91       {
92          app.at<unsigned char>(j,i) = 0;
93       }
94    }
95
96    //namedWindow( "temp", WINDOW_AUTOSIZE);
97    //imshow( "temp", app);
98    //imwrite("temp.bmp", dst );
99
100
101    for(i=0; i<256; i++)
102    {
103       for(j=0; j<(count[i]/10); j++)
104       {
105          app.at<unsigned char>(803-j,i) = 255;
106       }
107
108    }
109
110    namedWindow( "finalHistogram", WINDOW_AUTOSIZE    );
111    imshow( "finalHistogram", app);
112    imwrite("finalHistogram.bmp", app );
113    waitKey(0);
114
115    return 0;
116
117
118 }
```

4. Resulting images