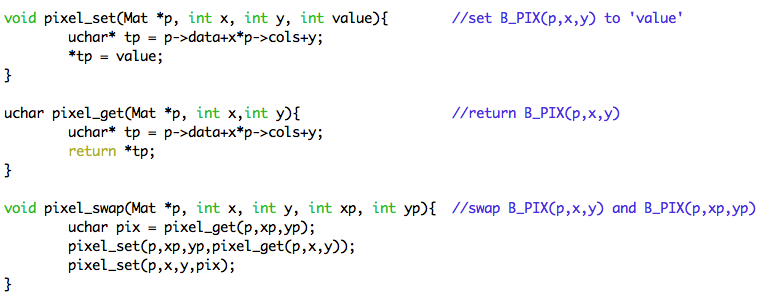
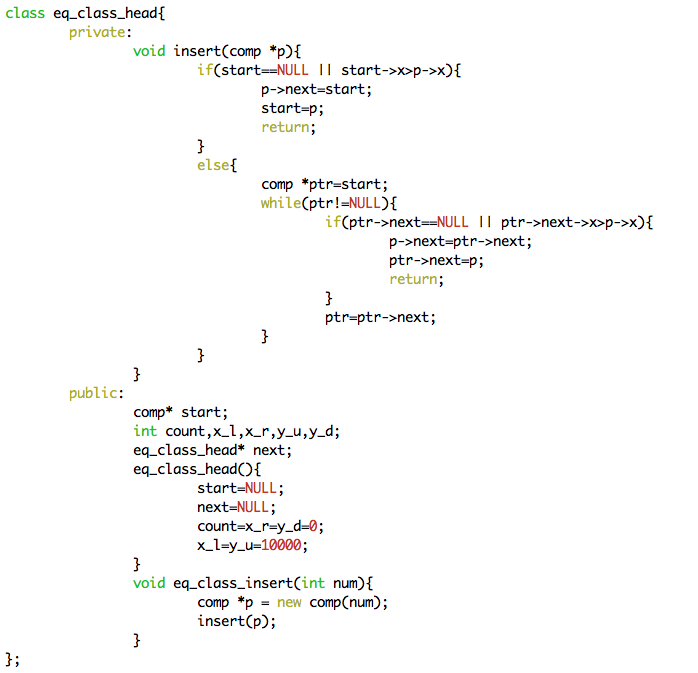
Computer Vision hw\_2

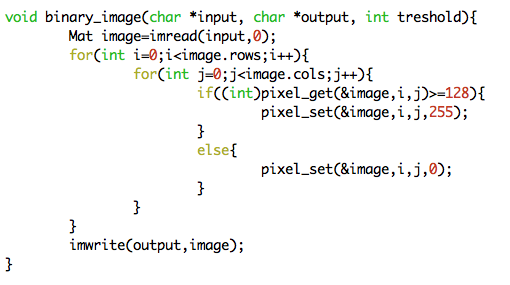
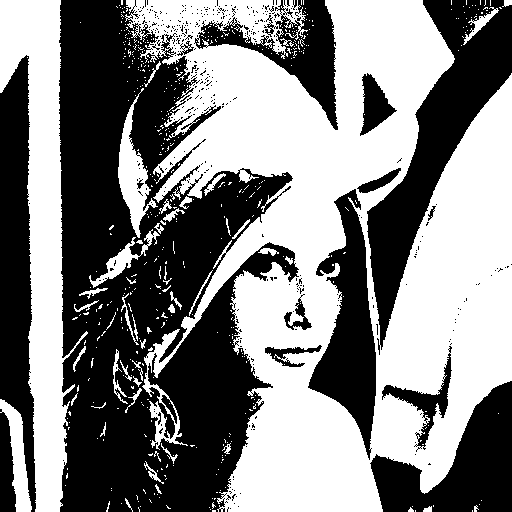
By R01922124 許彥彬

In this part the OpenCV-2.4.2 I/O function was included.

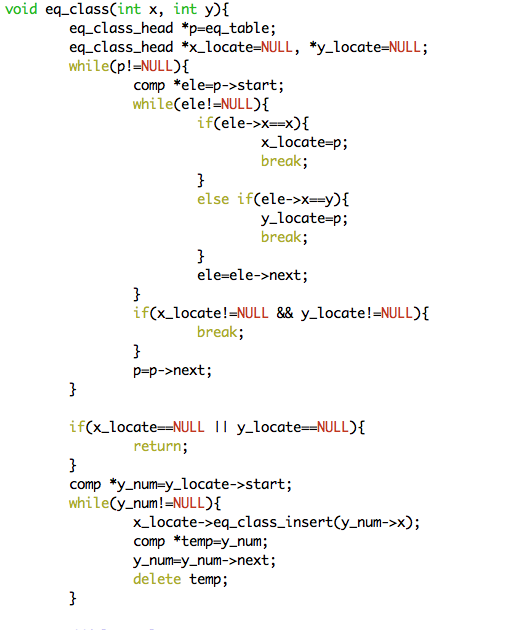
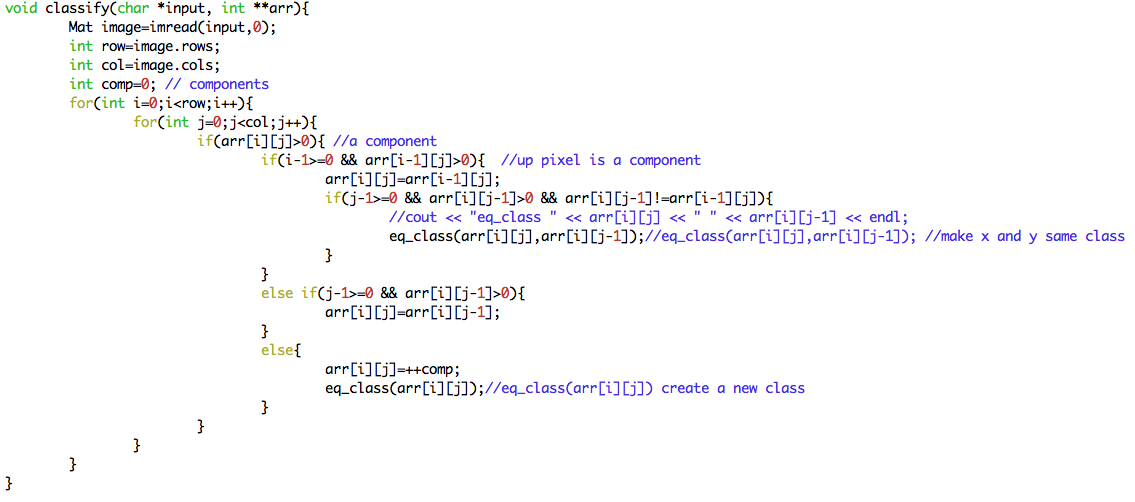
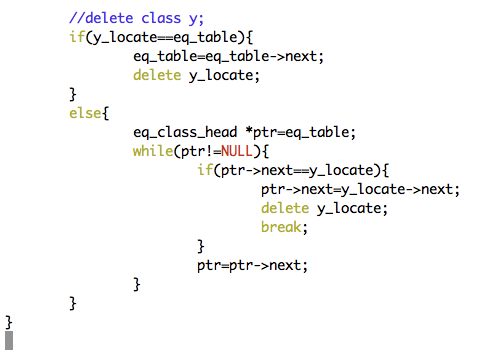
Three functions and two classes as follow will be use in the algorithms in this homework.

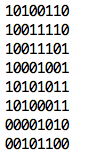
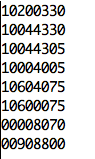


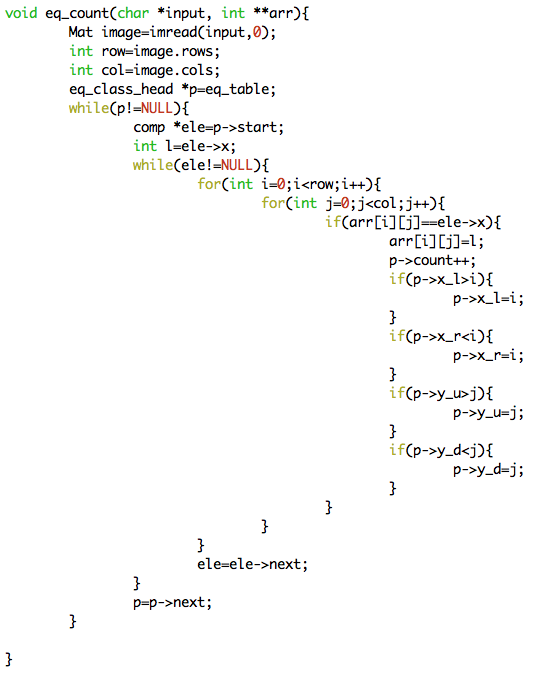
The two classes are used in making connected component. We will talk about them later in hw\_2-3.

1. A binary image with threshold 128  
   
   1. Search for the whole image lena.bmp. If the gray scale is great equal than 128, set the gray scale to 255. Otherwise set the gray scale to 0.
   2. The result is as follow.  
      
2. A histogram  
   
   1. Create an integer array sized 256 with 0 as initial.
   2. Get the gray scale for every pixel in the image lena.bmp, and add 1 to corresponding bin.  
      For example, if the gray scale of pixel(0,0) is 15, them we will add 1 to the 15th bin of array “histo”.
   3. Plot the histogram after this process done. The result is as follow.



1. Connected components (use 4 – connected)  
   In this part, I used the classical algorithm which showed in 2.3.4 of the book but adjust a little stuff. The two classes mentioned before are used in here.  
   I use “4 – connected” in this part.
   1. For the fist step, simply search the binary image and write the information into a matrix (two dimension array). Value 1 for gray scale 0 (black) and 0 for scale 255 (white).
   2. Top-down search the matrix, when matrix(a, b) is 1, if matrix(a-1, b) is k with k larger than 0. Assign matrix(a,b) as k. If matrix(a,b-1) is k’ also larger than 0. Then Label k and k’ same class. If matrix(a.b-1) is k’ and and matrix(a-1,b) is 0, assign matrix(a,b) as k’. If both matrix(a-1,b) and matrix(a,b-1) is 0, then assign matrix(a,b) as a new label.  
        
        
        
        
        
        
      An easy example using 8x8 lena.bmp is shown below.  
         
      Macintosh HD:Users:ilcic:Desktop:螢幕快照 2012-09-29 下午3.52.23.pngEquivalence classes.



* 1. Use the equivalence class information and the matrix after top – down pass to recognize the connected components and centroids. Only connected component, which has more than 500 pixels, will be recognize.  
      
  2. Result is as follow.  
     

1. Appendix
   1. build\_all.sh  
      command: “sh build\_all.sh” will automatically compile the code
   2. hw\_2.cpp  
      source code
   3. bi\_lena.bmp, histogram.bmp, bi\_lena\_4connected.bmp  
      results for this homework
   4. R01922124\_HW2.pdf