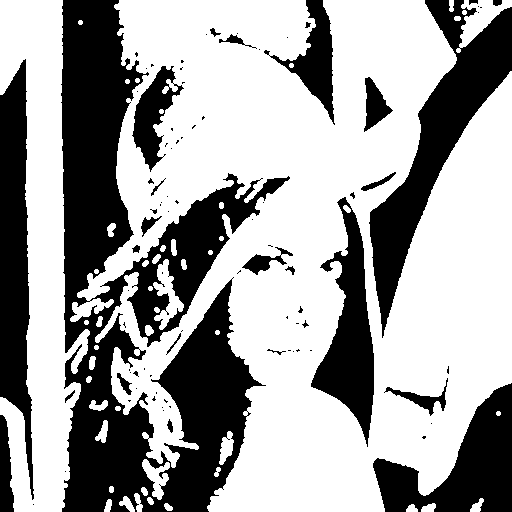
# CV Homework 4

R02922124 葉信良

Part 1 Dilation (using lena.bmp)

$ gcc –o hw4\_1\_dilation hw4\_1\_dilation.cpp

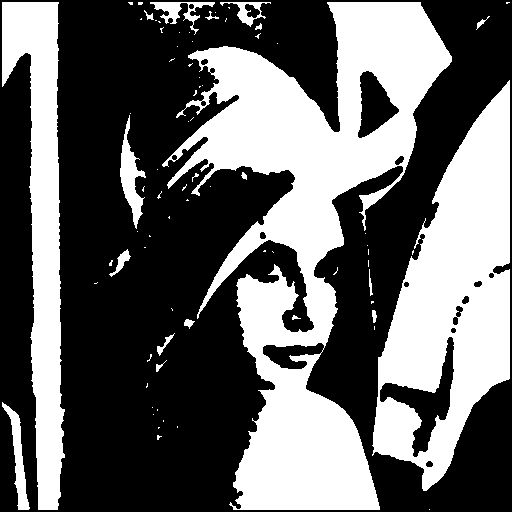
$ ./hw4\_1\_dilation

我直接使用c++讀入bmp file lena.bmp，並將header 1078個字元直接寫入到output。接著將大於127的數值都改成255，小於等於127的數值都改成0，並存到BBMap中，接著模仿ppt中去check每個pixel周遭點的值來決定該是0還是1，接著輸出圖片。圖片檔名是hw4\_1\_dilation.bmp。  


Part 2 Erosion (using lena.bmp)

$ gcc –o hw4\_2\_erosion hw4\_2\_erosion.cpp

$ ./hw4\_2\_erosion

我直接使用c++讀入bmp file lena.bmp，並將header 1078個字元直接寫入到output。接著將大於127的數值都改成255，小於等於127的數值都改成0，並存到BBMap中，接著模仿ppt中去check每個pixel周遭點的值來決定該是0還是1，接著輸出圖片。圖片檔名是hw4\_2\_erosion.bmp。  


Part 3 Opening (using lena.bmp)

$ gcc –o hw4\_3\_opening hw4\_3\_opening.cpp

$ ./hw4\_3\_opening

我直接使用c++讀入bmp file lena.bmp，並將header 1078個字元直接寫入到output。接著將大於127的數值都改成255，小於等於127的數值都改成0，並存到BBMap中，接著模仿ppt中去check每個pixel周遭點的值來決定該是0還是1，即做一次erosion，輸出到bool型別的TMap陣列，再做一次dilation，接著輸出圖片。圖片檔名是hw4\_3\_opening.bmp。  


Part 4 Closing (using lena.bmp)

$ gcc –o hw4\_4\_closing hw4\_4\_closing.cpp

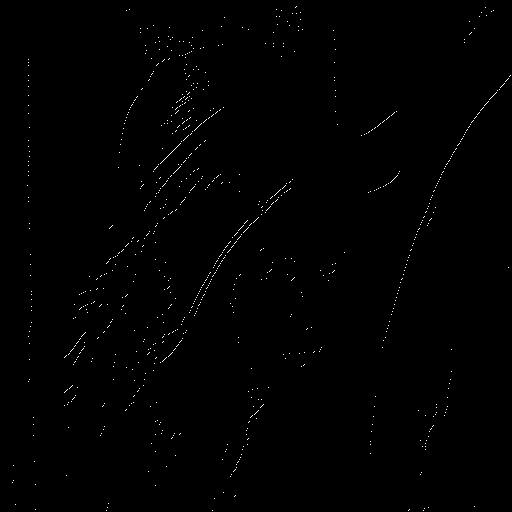
$ ./hw4\_4\_closing

我直接使用c++讀入bmp file lena.bmp，並將header 1078個字元直接寫入到output。接著將大於127的數值都改成255，小於等於127的數值都改成0，並存到BBMap中，接著模仿ppt中去check每個pixel周遭點的值來決定該是0還是1，即做一次dilation，輸出到bool型別的TMap陣列，再做一次erosion，接著輸出圖片。圖片檔名是hw4\_4\_closing.bmp。  


Part 5 Hit-and-Miss (using lena.bmp)

$ gcc –o hw4\_5\_HandM hw4\_5\_HandM.cpp

$ ./hw4\_5\_HandM

我直接使用c++讀入bmp file lena.bmp，並將header 1078個字元直接寫入到output。接著將大於127的數值都改成255，小於等於127的數值都改成0，並存到BBMap中，接著模仿ppt中去check每個pixel周遭點的值來決定該是0還是1，即做A erosion by J跟A complement erosion by K，但實作上我直接把這兩個合併一起做，參考的是網路上數學形態學的投影片，並改成是偵測右上角的corner，然後輸出圖片。圖片檔名是hw4\_5\_HandM.bmp。  


🡪以上的source code、執行檔、result image皆附在壓縮檔中。