**I**.作者簡介：

系級 : 測量112

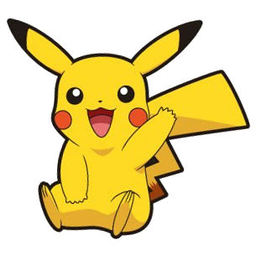
學號 : F64099552

姓名 : 陳英翔

e-mail : [0307eito@gmail.com](mailto:0307eito@gmail.com)

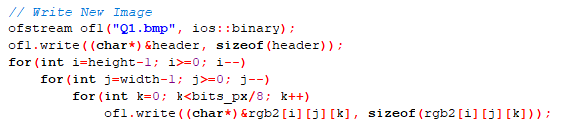
**II**.程式簡介：

* 程式的部分，我主要是針對老師提供的範例程式碼做修改與添加
* 每一題練習題都為一個程式碼檔案。第五題的部分將「調亮」「調暗」部分各分開為兩個程式碼檔案。
* 圖檔方面，為避免額外處理padding的問題，採用256x256 24位元點陣圖



256x256

**1.將圖片旋轉180度並輸出**

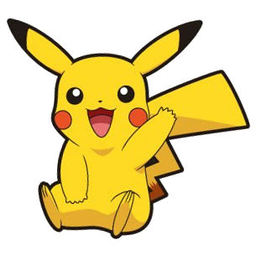
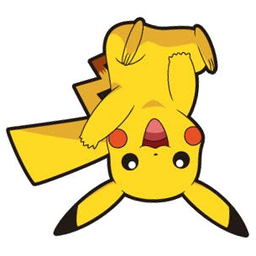


v

v

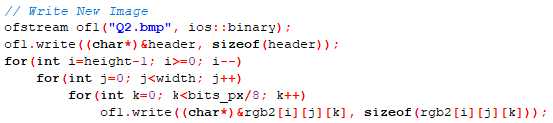
我將原本範例程式碼中，輸出新圖片的部分做修改。原本height和width

是從0開始，低到高去堆疊，但是我將height和width這裡改為從高到低堆疊，讓最後輸出的圖片為左右顛倒、上下顛倒，符合題意。

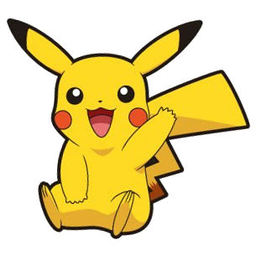
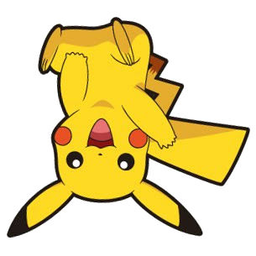
輸入(256x256) 輸出(256x256)

**2.將圖片垂直翻轉並輸出**



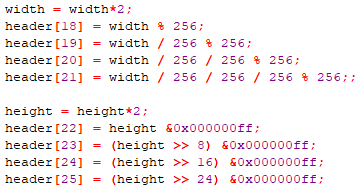
v

題目要求垂直翻轉，也就代表要讓圖片以上下顛倒輸出，所以只需要修改height的部分，將原本從低到高堆疊改為高到低堆疊，就可以得到垂直翻轉的結果。

輸入(256x256) 輸出(256x256)

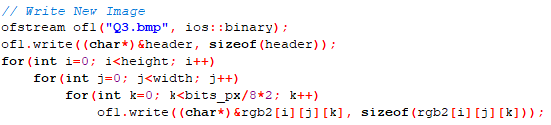
**3.將圖片長寬各放大一倍並輸出**



v

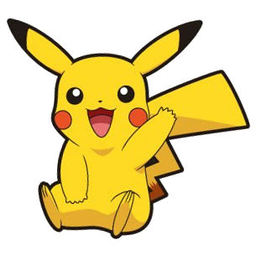
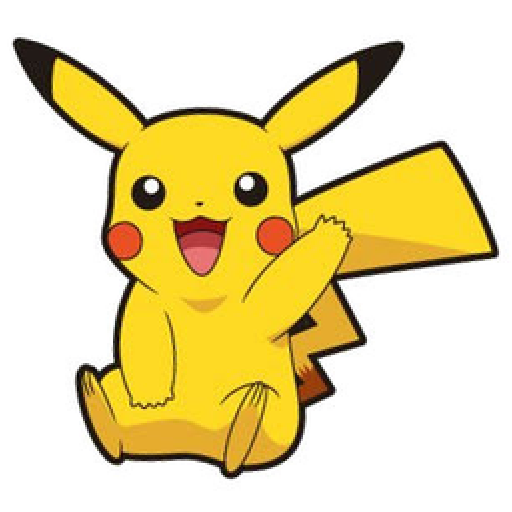
v

圖片的長寬要放大一倍，因此將輸出圖檔的header重新設定，將width和height乘2。



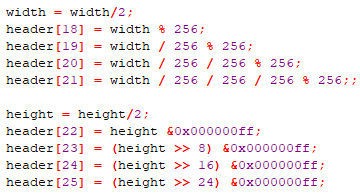
v

輸出圖檔的部分，由於width和height在上述已經乘2，因此這邊不須再做修改。 每一個Pixel的bytes大小必須乘2，因為不管是垂直堆疊還是橫向堆疊的時候，width和height已被拉長為兩倍，必須要用兩倍的bytes數才能順利完成Pixel的擴充。

輸入(256x256) 輸出(512x512)

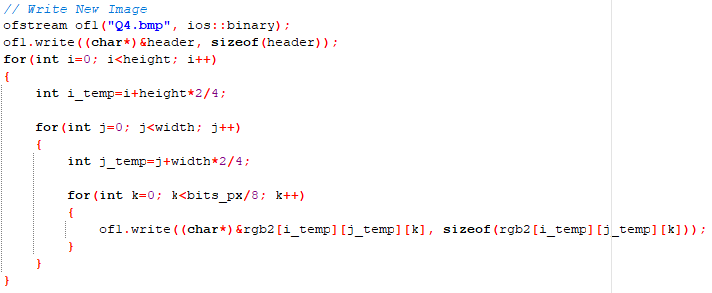
**4.將圖片上下兩邊裁切1/4高、左右兩邊裁切1/4寬，留下中心1/4面積的畫面部分並輸出**



v

v

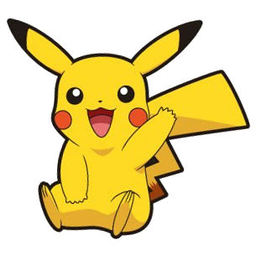
輸出的圖檔面積是原本的四分之一，因此在輸出檔的header中，把width和height大小都設為一半。



v

v

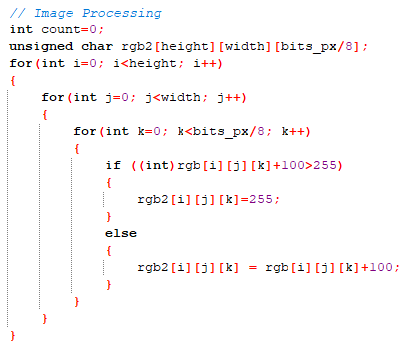
巢狀迴圈中，我設了變數i\_temp和j\_temp，每次執行迴圈的時候，i\_temp代表i值加上「輸入圖檔height值/4」、j\_temp代表j值加上「輸入圖檔width值/4」。 之所以這麼做是因為題目要求的是圖檔中心1/4的面積，因此pixel不能從(0,0)的位置開始取，要從(height/4, width值/4)的位置開始才對。

輸入(256x256) 輸出(128x128)

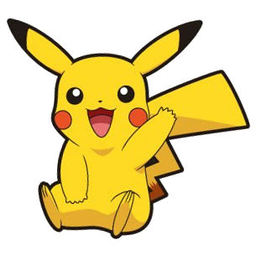
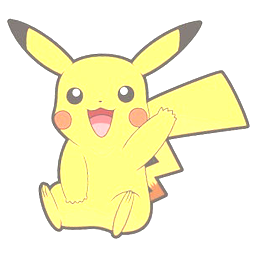
**5.調整原圖亮度(依每位同學所選的圖片，自行決定調亮或調暗)**

調亮



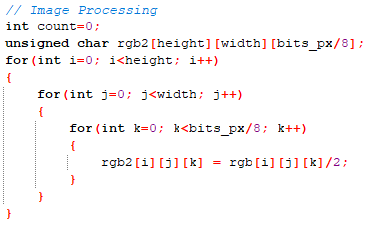
在處理讀取出來的data的rgb值的時候，順便把rgb三原色的值分都加上100，調高整體亮度。rgb三個值的最大是255，因此需要設定if敘述，將255設為上限。

v

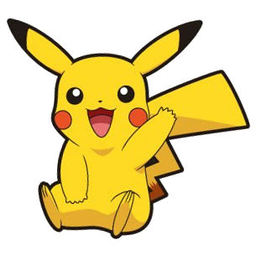
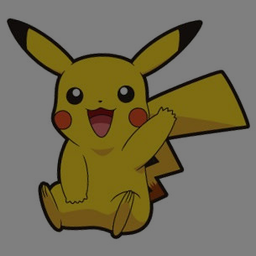
輸入(256x256) 輸出(256x256)

調暗



v

調暗的設定上，我把rgb的值乘1/2，由於rgb值的最小值是0，而乘上1/2的動作並不會使值小於0，因此不需要額外假設值小於0的情況。

輸入(256x256) 輸出(256x256)

**Histogram equalization 直方圖均衡化**

原理

目的是將原始圖像像素的色彩強度均勻地映射到整個色彩範圍內，得到一個色彩強度分佈均勻的圖像。

計算舉例

嘗試將5x5圖像的色彩範圍均衡化至更大的色彩範圍。

**假設一個5x5像素的圖像，色彩範圍設為[0,6]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 4 | 2 | 4 | 4 |
| 5 | 0 | 4 | 6 | 6 |
| 1 | 0 | 3 | 1 | 6 |
| 6 | 5 | 6 | 6 | 0 |
| 0 | 5 | 3 | 4 | 4 |

**統計每個色彩值的出現次數**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 色彩值 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 出現次數 | 4 | 3 | 1 | 2 | 6 | 3 | 6 |

**計算每個色彩值的出現機率、累積機率**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 出現機率 | 4/25 | 3/25 | 1/25 | 2/25 | 6/25 | 3/25 | 6/25 |
|  | =0.16 | =0.12 | =0.04 | =0.08 | =0.24 | =0.12 | =0.24 |
| 累積機率 | 0.16 | 0.28 | 0.32 | 0.4 | 0.64 | 0.76 | 1.0 |

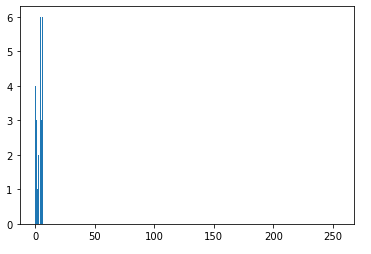
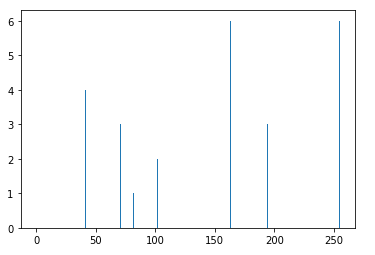
**擴大色彩範圍至[0,255]，做均衡化的動作**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 色彩值 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 出現次數 | 4 | 3 | 1 | 2 | 6 | 3 | 6 |
| 累積機率 | 0.16 | 0.28 | 0.32 | 0.4 | 0.64 | 0.76 | 1.0 |
| 新的色彩值 | 0.16x255 | 0.28x255 | 0.32x255 | 0.4x255 | 0.64x255 | 0.76x255 | 1.0x255 |
|  | ≒41 | =71 | ≒82 | =102 | ≒163 | ≒194 | =255 |

**均衡化之後的結果**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 新的色彩值 | 41 | 71 | 82 | 102 | 163 | 194 | 255 |
| 出現次數 | 4 | 3 | 1 | 2 | 6 | 3 | 6 |

**嘗試將分佈視覺化(使用matplotlib)**

均衡化前 均衡化後