**作者簡介**：

姓名：李展瑋

系級：測量113

學號：F64096198

E-mail：[brucelee097527297@gmail.com](mailto:brucelee097527297@gmail.com)

**程式簡介與輸出**

前情提要：

我選了一張鳥人圖檔檔名是nlnlOeO.bmp，之後的輸出nlnlOeO\_n.bmp，n分別對應題號，鳥人圖如下：

nlnlOeO.bmp

1. Ｑ：請將圖片翻轉１８０度並輸出

這題的要求只是把圖片翻轉１８０度且大小並沒有改變，所以我們可以想像只要把上下左右都顛倒，且**沒有改變圖片大小**，因此實施的方法很簡單，只要把寫入rgb2三維陣列的第29行statement稍微做更改就行：

rgb2[i][j][k] = rgb[i][j][k]改成rgb2[i][j][k] = rgb[height-i-1][j][k]

如此一來上下就可以顛倒

rgb2[i][j][k] = rgb[i][j][k]改成rgb2[i][j][k] = rgb[i][width-j-1][k]

如此一來左右就可以顛倒

rgb2[i][j][k] = rgb[i][j][k]改成rgb2[i][j][k] = rgb[height-i-1][width-j-1][k]

如此一來上下左右就都會顛倒，就達成旋轉１８０度的題目要求了

輸出圖片如下：  
　　　　　　　　　nlnlOeO\_1.bmp

1. Ｑ：請將圖片上下顛倒並輸出：

這題其實是第一題的限縮版，因為只要上下顛倒就好了，所以只要

rgb2[i][j][k] = rgb[i][j][k]改成rgb2[i][j][k] = rgb[height-i-1][j][k]

如此一來上下就可以顛倒

輸出圖片如下：

nlnlOeO\_2.bmp

1. Ｑ：請將圖片放大兩倍並輸出：

這題有兩個要點，第一的要點是rgb2的陣列大小height以及width要改成兩倍，因為圖片要放大兩倍，第二個要點是要把長寬兩倍創造出來的空間都填入原本那個空間的rgb資料，實施方法如下：

第25行

unsigned char rgb2[height\*2][width\*2][bits\_px/8];//這行是為了把圖片放大

第30~33行

rgb2[i\*2][j\*2][k] = rgb[i][j][k];

rgb2[i\*2+1][j\*2][k] = rgb[i][j][k];

rgb2[i\*2][j\*2+1][k] = rgb[i][j][k];

rgb2[i\*2+1][j\*2+1][k] = rgb[i][j][k];

//這四行是為了把多出來的空間都填入原本的rgb資料

如此一來就能達成圖片放大兩倍的目標

輸出圖片如下：

nlnlOeO\_3.bmp

註：放大兩倍的bmp檔在我的word無法直接顯示，所以我用原圖代替放在這邊展示，放大兩倍的圖檔在壓縮檔裡面有，直接執行也可以輸出放大兩倍的圖檔

1. Ｑ：將圖片保留中間1/4並輸出

這題有兩個要點，第一個要點是rgb2的height跟width要改成原本的一半，第二個要點是控制三層迴圈的height跟width從圖中開始跑，實施方法如下：

第25行

將資料的長寬改為原本的一半

unsigned char rgb2[height/2][width/2][bits\_px/8];

第26~29行

讓複製資料的開始與結束處由原長寬的1/4跑到3/4

for(int i=0; i<height/2; i++)

for(int j=0; j<width/2; j++)

for(int k=0; k<bits\_px/8; k++)

rgb2[i][j][k] = rgb[i+height/4][j+width/4][k];

如此一來就能輸出中間1/4的圖片部分並輸出

輸出圖片如下：

nlnlOeO\_4.bmp

（５－１）Ｑ：調整圖片調暗或調亮並輸出

　　　這題的思維邏輯比較簡單，單位像素有２４位元ＲＧＢ值，每８單位的

　　　ＲＧＢ值越大則代表越鮮豔，所以這題只要把單位像素的鮮豔度做線性

　　　調整就可以達到題目要求，我選擇的是把圖片調暗，實施方法如下：

　　　第２９行

　　　rgb2[i][j][k] = rgb[i][j][k]\*0.5

　　　將寫到輸出圖片像素資料的這行敘述，把每個單位像素的大小乘以二分

　　　之一，就可以將鮮豔度降低，達成把圖片調暗的目的

　　　輸出圖片如下：

nlnlOeO\_5-1.bmp

（５－２）Ｑ：調整圖片亮度並輸出，依照老師所給的參考資料

前情提要：

　　我選擇利用維基百科參考資料的灰階圖來做這題的輸入輸出，因為灰階圖所呈現出來的效果比較好，圖片如下

view.bmp

這題照著老師所提供的參考資料的公式



利用到的參數有，h(v)該像素值v的調整後數值，cdf(v)該像素數值的累積出現次數，cdfmin最少的累積像素出現次數，cdfmax最多的累積像素出現次數，Ｌ像素深度，我在程式碼裡面同樣宣告了max以及min以及利用自訂陣列color來輔助我計算這些參數，且同樣都有輸出在小黑窗供我進行修正，最後輸出完全符合維基百科介紹的圖片，非常的有成就感，實施方法如下：

第26~36行

宣告無號整數陣列color[256]來儲存每個像素值的累積出現次數

並利用此迴圈，當跑到該個像素點時，將對應的color++

unsigned int color[256]={0};

for(int i=0; i<height; i++)

{

for(int j=0; j<width; j++)

{

for(int k=0; k<bits\_px/8; k++)

{

color[(int)rgb[i][j][k]]++;

}

}

}

第37~38行

這邊是為了將數據作累積，因為方法的參數是要用累積出現次數

可以特別注意到ｉ是從１開始，因為陣列開頭沒有前一個數值給他累積

for(int i=1;i<256;i++)

color[i]+=color[i-1];

第60行

這邊就是整個程式最重要的statement，也就是將參考的公式實現的地方

rgb2[i][j][k] = ((round)(color[rgb[i][j][k]]-min)/(max-min)\*255);

輸出圖片如下：

view\_2.bmp

可以見到圖片的對比度有著巨幅的上升，且與參考資料的輸出圖片完全一致，用老師提供的參考資料完成這題相當的有成就感

**謝謝老師以及助教～**