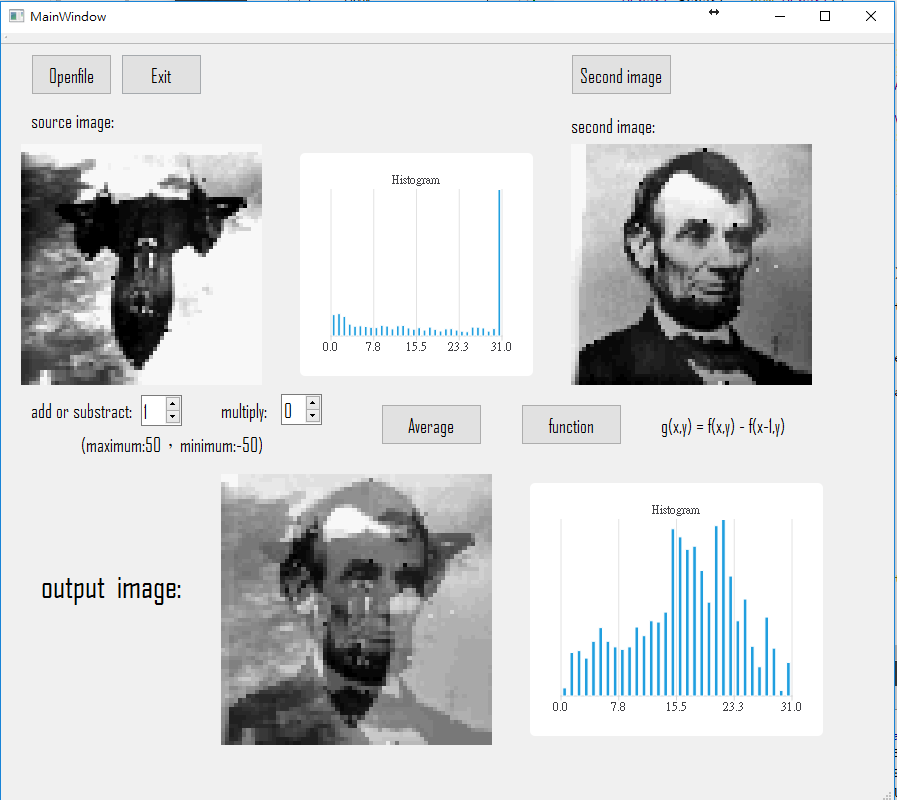
Image Processing Report

下方為整個程式大致上的介面，每個鍵的功能都像上面的文字一樣很好懂。



而整個程式主要的部份就是:

開檔 -> 讀檔 ->將(.64)檔案輸入矩陣中，再將其換成RGB\_32的格式，輸出到介面程式中。

然後算出圖的histogram

for(int i=0; i<64; i++){

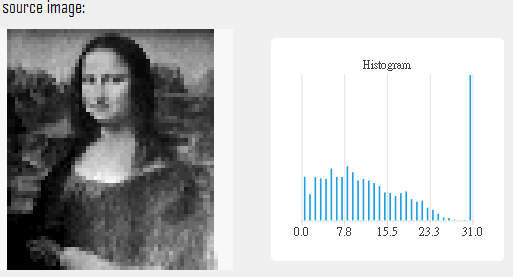
for(int j=0; j<64; j++){

histo[ image[i][j] ]++;

}

}

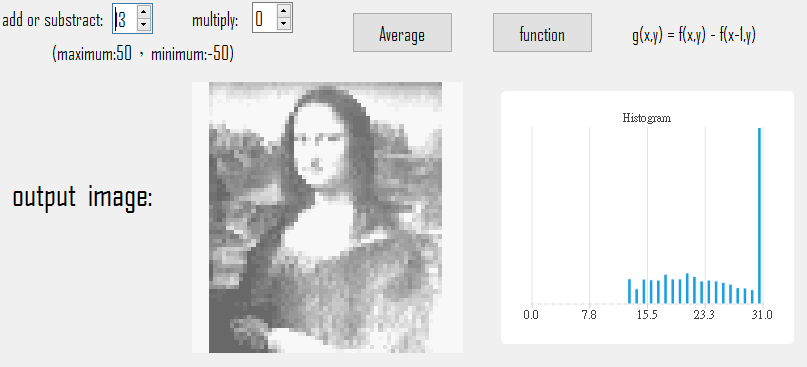
最後就是依照第二題的各種要求來做按鈕。



1.

第一題的結果如上圖所示，可以看到蒙娜麗莎這張圖中，灰階值是31的點有非常多，因為這張圖左右有兩條白白的地方。

2.

1. Add or subtract a constant value to each pixel in the image.

第一題加減的地方，我有設定上限和下限值，可以手動調整要加減的數值，所以加減會有變淡變亮的情況產生，從直方圖中也可以看到黑色的點都沒有了，如果不設定上下限值的話，圖片在加超過32以後會循環回來，所以當add的值為32時，又會得到原圖。

1. Multiply a constant to each pixel in the image.



乘的話，也是會變亮，也可以手動調整乘的數值，從直方圖中可以看到他的間隔會變大，因為中間的值都因為乘了之後就不見了。

1. Create a new image which is the average image of two input images.



平均的功能就會讓這兩個圖疊在一起，然後直方圖也變成兩張的平均。可以看到蒙娜麗莎的白邊在和林肯這張圖疊合的時候就很明顯摟。

for(int i=0;i<64;i++){

for(int j=0;j<64;j++){

image\_aft[i][j]=(image[i][j]+image\_2[i][j])/2;

if(image\_aft[i][j]>31){

image\_aft[i][j]=31;

}

if(image\_aft[i][j]<0){

image\_aft[i][j]=0;

}

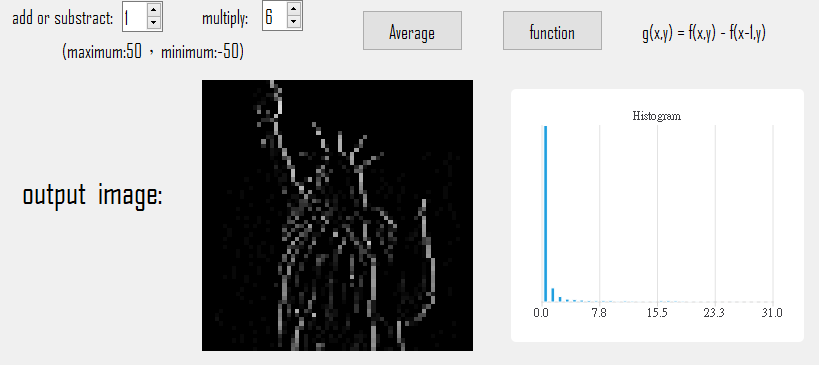
}

}

核心的程式碼如上，將兩張圖平均，然後設定上下限的值，讓他不會超過上下限。

1. Create a new image *g*(*x*,*y*) in which the value of each pixel is determined by calculating the pixel values of the input image *f*(*x*,*y*) using the following equation:

*g*(*x*,*y*) = *f*(*x*,*y*) - *f*(*x-*1,*y*)



最後的這個功能，以這個function來看，是把原本的點和左邊的點相減，可以發現，這樣會看到右亮左暗的輪廓。