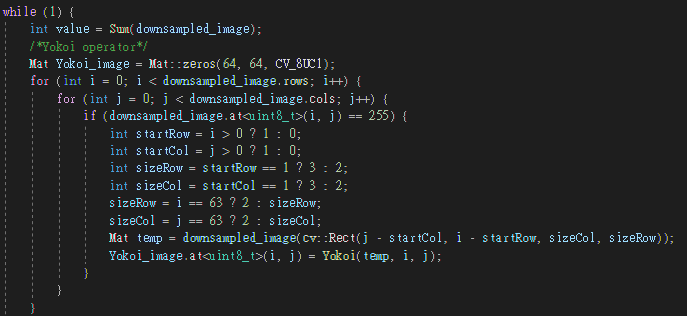
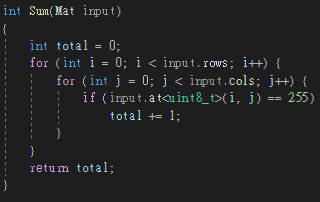
**2021/11/25 陳嘉政 r10922171 HW7**

**程式碼**

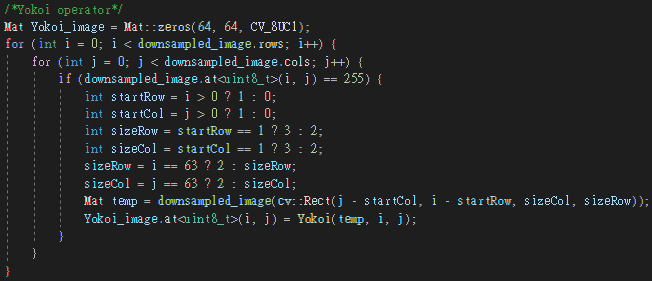
**Step1: 將原圖做binarize，接著做downsampling。**

**Step2: 進入遞迴前先計算image的sum，用來後續與結果做比對。**

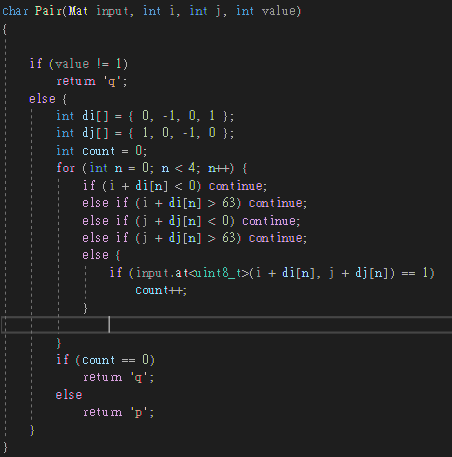




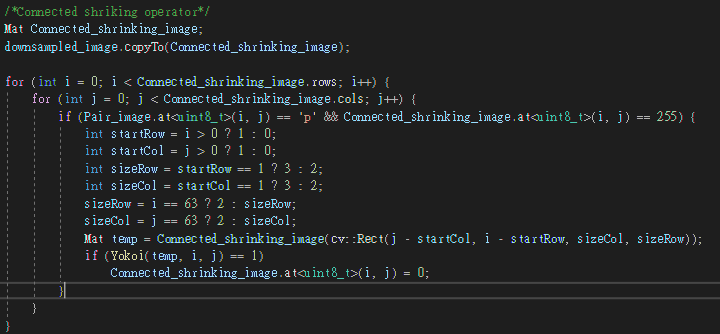
**Step3: 將downsampled image做Yokoi operator，函式與作業6相同就沒貼了。**



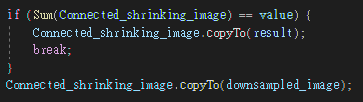
**Step4: 將Yokoi後的image拿來做pair relationship operator，input 分別為Yokoi image，row，column及此pixel的value。若value不為1直接pair image中此點assign為q，否則計算此點的四連通鄰居，若有pixel為1則count加1，最後如果count >= 1則此點assign為p，否則assign為q。**



**Step5: 宣告Connected shrinking image，將downsampled image(原圖)複製到connected shrinking image，接著每個點做，如果此點對應到的Pair image的點值為p以及此點值為255，代表我們要考慮此點要不要刪。接著與Yokoi的做法相同，以此點為中心畫出適當大小的temp矩陣，以及row，column一起當Yokoi(作業6的函式)的input，因為Yokoi的回傳值為0-5的整數，若回傳值為1則將此點的值設成0(刪掉)。**

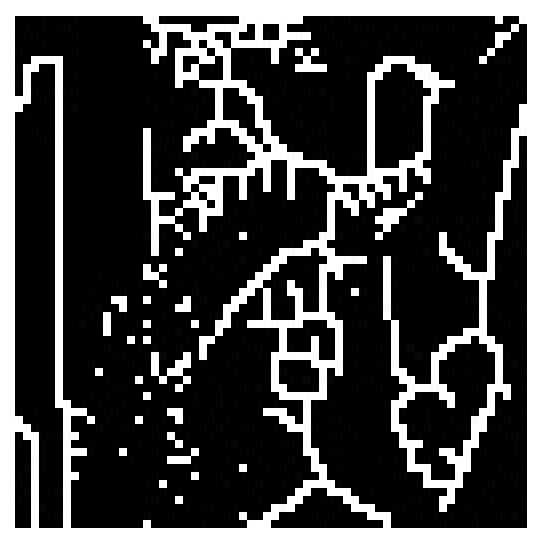


**Step6: 跑完所有點後，若結果的sum不等於原圖的sum，代表圖仍有變動，所以將結果圖當成原圖，重複做Step2,3,4,5，直到結果與原圖的sum相同就跳離迴圈，表示圖已經不變。**



**結果**

**為放大過後的64x64圖**

****