這次的過程真的是很漫長，簡單的講歷經三個大階段:建樹、編碼、排列後輸出，起初以為應該只有編碼比較過難，但事實上每個階段都是很難處理的，請待我以下分別娓娓道來。

首先，我思考著Huffman的編碼方式，他說以最小的兩個點去建新的點，然後以此類推可以做出一棵樹，於是我開始寫找未知數量參數的最小兩數函示，但是在卡了一段時間後，我才突然頓悟為何要去比較實際的數量呢?最小的兩個數其實就是找機率最小的兩個點!這算是個初期很大的突破，雖然後來看覺得只是個小改變，但是影響到了我整份code的寫法。

再來就要開始建樹了，建樹的概念以上都講過了，首先將所有輸入的點按照輸入順序建成直線的list，然後再依照以下方法進行while迴圈↓

找出最小的節點→找出第二小的節點→將最小兩個節點機率之和存入新節點→將新節點加入到串列最前面→移除兩個機率最小的節點

由於每次都會刪除節點，所以跳出while迴圈的方式就是當該份list的長度等於1也就是只剩自己時停止(代表樹建完了)。在此階段主要的瓶頸跟上次作業遇到的一樣，就是指標常常指到奇怪的地方，不是記憶體錯誤不然就是印出"看似答案"卻不是真正的答案，而會發現都是因為後面再編碼時覺得很奇怪，再回來看才發現的，這時候發現Ctrl+f10真的不錯用，而且在這邊還有遇到一個問題，不過這得先講下一個階段才能提到，這邊先暫時略過。

編碼的方式我採取的是從樹的頂端開始走訪，code上採取遞迴的方式一層一層深入，往左設0往右設1，此階段主要的問題在於編完後若當出是同樣機率的節點要怎樣分辨誰的編碼是誰的，也就是上一段所提到，所以我就在上一階段中新增一個ID作記錄(該變數新增在struct裡面)，然後之後就按照ID來印出結果，看起來好像到這邊就歡樂的結束了，但事實上在這邊是另一個噩夢的開始。

最後只剩輸出，但這邊大概卡了4天，由於我的編碼都是宣告成char\*做儲存，且因為我的編碼產生方式是採取走訪的過程生成的，所以編碼產生的順序跟當初輸入的順序並不同，因此我必須把產生的編碼存到新的陣列(因為我採取遞迴的方式，每次遞迴產生的編碼都是暫時性的，在下一次的走訪便會刷掉上一次的編碼，而產生新的下一筆)，但是我發現不能把char型態的001這種由0 1所組成的字串插到char的陣列的任一位置，簡單一點來說就是我今天想把 000插到 [0] 而 111 插到 [1]，在最後出來的答案很明顯事與願違，最初我很天真的以為把char型態的編碼用atoi轉成int型態就能解決了，而作業說明上的範例也好巧不巧剛好可以跑出一樣的答案，於是我天真的以為結束了，後來發現不對，怎麼換一下測資會造成編碼的結果非常的怪異，於是我又再次的檢查後才發現，char的001轉成int是變成1，而char的000轉成int變成0，完全不符合我所要結果，因此此時我尋找新的方法試圖突破，後來發現採取c++特有的string型態來儲存編碼不就好了嗎?但實作上發現沒這麼簡單，即使把最初的char\*宣告改成string，還有一些細部的調整仍然無法成功編譯，到這時候其實我有想過放棄，因為這就是卡了4天其中的3天，但是經過幾天的思考我後來終於突破盲點，既然要轉那何不在產生編碼的那一刻再轉就好?於是我使用了assign把\*char轉成string，解決了編碼要重新排列然後輸出的問題，這時候我以為真的結束了，但還沒。

我發現怎麼輸出的檔案只有印出一個編碼加一行解碼的結果，後來才發現原來是因為遞迴的關係，由於我的寫檔是放在遞迴函式裡，導致在 out模式開檔寫檔的情況下他只印出最後一次遞迴的結果，更精確一點來講就是前面的也有印但都被下一次的遞迴洗掉了，洗到最後當然是印出最後一次遞迴的結果，我在這邊也想了很久，最後才發現山不轉路轉，既然out模式這麼不人性那寫檔就換成其他模式吧!試過了ate、trunc、append，最後發現append模式可以解決以上的問題，而且作業也只說要輸出一次檔案，雖然輸出第二次資料會往後繼續疊，但是這符合作業需求，所以safe!當然，我以為這是最後了，但還不是。

在輸出時我發現了一個重大的問題，就是為何雖然輸入是ABCDEFG，但要印出的編碼是ABCDE時，也就是某些代碼雖然有參與編碼，但最後沒有要輸出的情況下，他的編碼是印出空的呢(輸出檔案時要印出每個代碼各別的編碼那裡)?!起初我以為是那些點(在此例就是指F、G)沒有參與到樹的走訪所以沒有編碼到，但事實上不是如此，之所以會印出空的原因是因為我在重新排列編碼順序時，編碼的來源參考陣列根本就沒有F、G，因為在該陣列我放的是待被轉換代碼串，也就是ABCDE，但這裡面根本沒有F、G，所以當然在整理時不會給F、G的編碼配到儲存最後輸出編碼的陣列的指定位置(因為我的儲存方式是判斷該陣列的代碼與在當次迴圈產生的點的ID有沒有相同)，既然沒位置可以配對到那F、G的編碼就根本不會存到，也當然會印出空的東西，所以我最後就在判斷的時候加一個ELSE當做例外判斷，F、G存到例外的陣列才解決，到這時所以問題真的結束了。

是的，一切都結束了，終於。