

資結 練習三 報告

(一) 簡介:

任務一: 合法運算式檢測

先將使用者輸入的一串字串 `getline` 讀進來，接著丟進寫好的 `class` 裡面的 `missionone function`，移除字串間的所有空白後，先判斷字串是否為空(假使使用者沒有輸入任何東西則會出現 `error3`)，然後進入迴圈從字串第一個字元開始判斷，如果是數字就用 `append` 繼續加到暫存字串 `temp` 直到跑到不為數字的字元，然後將它 `insert` 進去預先寫好的鏈結串列 `expLIST`，另外還有一個條件是下一個字元如果是 '(' 會出現 `error3`，因為數字跟 (之間要有四則運算；如果是四則運算，則會判斷(1)下一個是否也是四則運算(2)目前這個四則運算是否是整個字串的第一位或最後一位(3)下一個是否為 ')' 會出現 `error3`，如果都沒問題就會加到暫存字串 `temp` 並且將它 `insert` 進去預先寫好的鏈結串列 `expLIST`；如果是括號，判斷如果是右括號，就會將預設 `int` 的 `closepar` 加一(用來判斷括號出錯的部分)，接著判斷(1)如果目前左括號是小於右括號(2)如果下一個字元是數字(')'跟數字之間要有四則運算) (3) 如果下一個字元是左括號(')'跟 '(' 之間要有四則運算) 會出現 `error2` 或 `error3`，判斷如果是左括號，就會將預設 `int` 的 `openpar` 加一(用來判斷括號出錯的部分)，接著判斷(1) 如果下一個字元是右括號 ('('跟 ')' 之間不能連在一起) (2)如果下一個字元是四則運算(')'後面不能直接接四則運算)，會出現 `error3`，如果都沒問題就會加到暫存字串 `temp` 並且將它 `insert`

進去預先寫好的鏈結串列 `expLIST`；如果都不是則會出現 `error1`，最後，如果都沒問題就會顯示這是一個合法的中序運算式字樣，並接著後面的任務二-中序式轉為後序式的部分。

任務二：將中序式轉為後序式

前面任務一的地方判斷無誤後，便會接著任務二，將前面設定好的 `list` 帶進入 `missiontwo` 的 `function` 後，一個一個去找原來存放在 `list` 的 `value`，並先判斷如果是數字則直接將字串存入後來預設的 `postfixlist`；如果是四則運算，則先判斷堆疊裡的 `top` 是否為四則運算和 `top` 的前一個是否也為四則運算(以便判斷要跟堆疊裡的四則運算比較幾次)，如果裡面在遇到括號或到底之前有兩個四則運算堆疊則比較兩次，如果只剩一個則只需要比較一次(而我們的比較是:如果目前這個是乘號或除號而堆疊中 `top` 是加號或減號則直接返回，並把四則運算存進 `postfixlist`，如果是其它條件則一律先刪掉存放目前值進堆疊中，並把刪掉的值存進 `postfixlist`，如果堆疊裡面的 `top` 不是四則運算則將此值直接放進堆疊裡；如果是括號，則先判斷此值是右括號還是左括號，如果是左括號，也是直接放進堆疊裡，如果是右括號，把到 `openpar` 以前的 `op` 都放出來，並把在堆疊要被刪掉的 `op` 存進去 `postfixlist`，最後，把堆疊裡的 `openpar` 刪掉。利用鏈結串列指標一個一個取得目前 `list` 的值做判斷直到遇到空值，最後印出全部存在 `postfixlist` 的值。

任務三：以後序式求解

一樣是前面任務一的地方判斷無誤後，便會接著任務二和三，直接將 postfix 帶進寫好的 missionthree function，一個一個去找原來存放在 postfixlist 的 value，預先新增一個用來計算結果的堆疊，接著判斷如果是數字就直接 push 進堆疊裡；如果是四則運算，就從堆疊裡 pop 出最上面兩個值，並將字串轉換成數字，接著依照目前此運算子判斷此兩數做四則運算(另外除號部分:如果分母是 0 則會出現錯誤訊息)，接著將計算結果 push 回堆疊讓之後找到的 postfix 值能接著計算，最後跑完到鏈結串列的最後一個後停止並判斷如果沒有錯誤則將答案印出。

心得:

在寫任務一時，找 bug 花了一點時間，忘記自己是用 for 迴圈還以為是用 while 迴圈導致習慣寫 i++，因此序號都會不停往後多一個，不少字元都沒被存到，找了很久的 bug 後來才知道是自己犯蠢。

任務二比起任務一更需要清晰的頭腦，才能去準確的判斷每個元素的先後順序以及該放到哪個指標陣列中，當時在元素的比較上花了一些時間，因為比起*/或括號，+-在屬於後面的序位，我們一直在思考還沒被丟入堆疊的+-符號該怎麼去做比較，還有如果 stack 那裏放了很多個運算子的話是不是要一直比較它們的

level 先後順序，所幸後來再回去把後序式的影片看一遍後，研究了一下各種預想的數字和四則運算，終於了解了整個後序式 op 的優先順序跟判斷依據還有比較次數的概念。

這次流程圖遇到一個問題，因為 for 迴圈跑到最後會回到前面並且累加 i 的值，所以有跨線的問題，我一直糾結流程圖的線是否可以交叉，交叉是不是就代表我畫錯，所以在這次的圖上花了比較多時間，後來在網路上看到跨線的畫法，也理解流程圖跨線並不代表畫錯。

這次的作業我們學到很多：

(1)如果不是以三個任務的方式出現，我應該會寫更久，這種分段式的練習引導我們寫出一支完整的程式，因此讓我相信在這堂課不只能學到寫程式，還能學習分析能力。

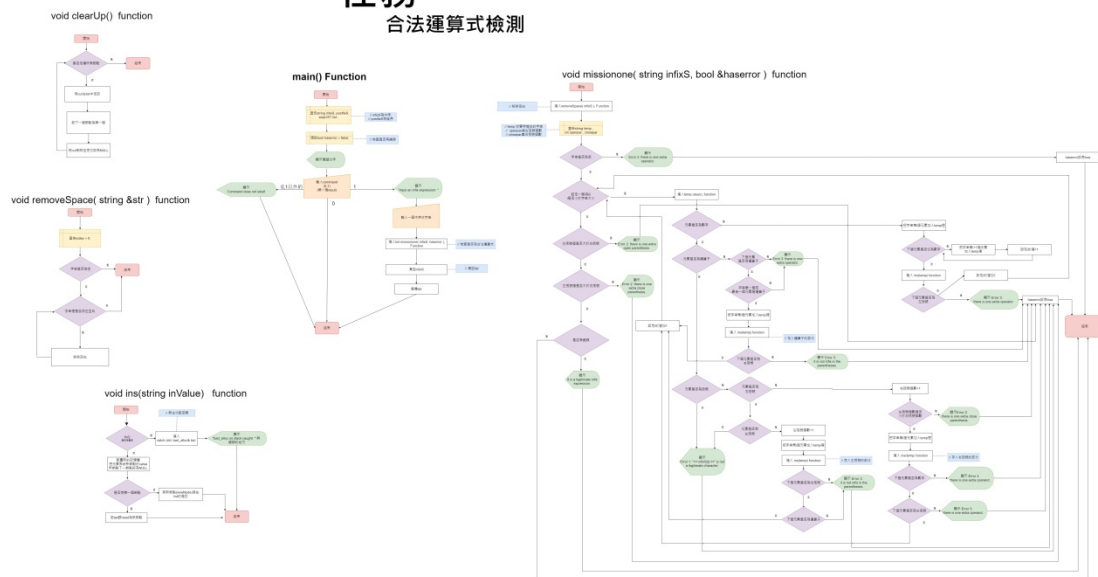
(2)寫作業二時沒看清楚要用 class 練習做作業，交了用結構做的作業，非常遺憾也很後悔，因此我們練習三時，就試著用 class 來寫，

因為對於 class 還沒有用結構來的熟悉，所以我們還有先去研究一番，花了一點時間跟 class 相處，對於 class 的用法也更加熟練。

(二) 圖示:

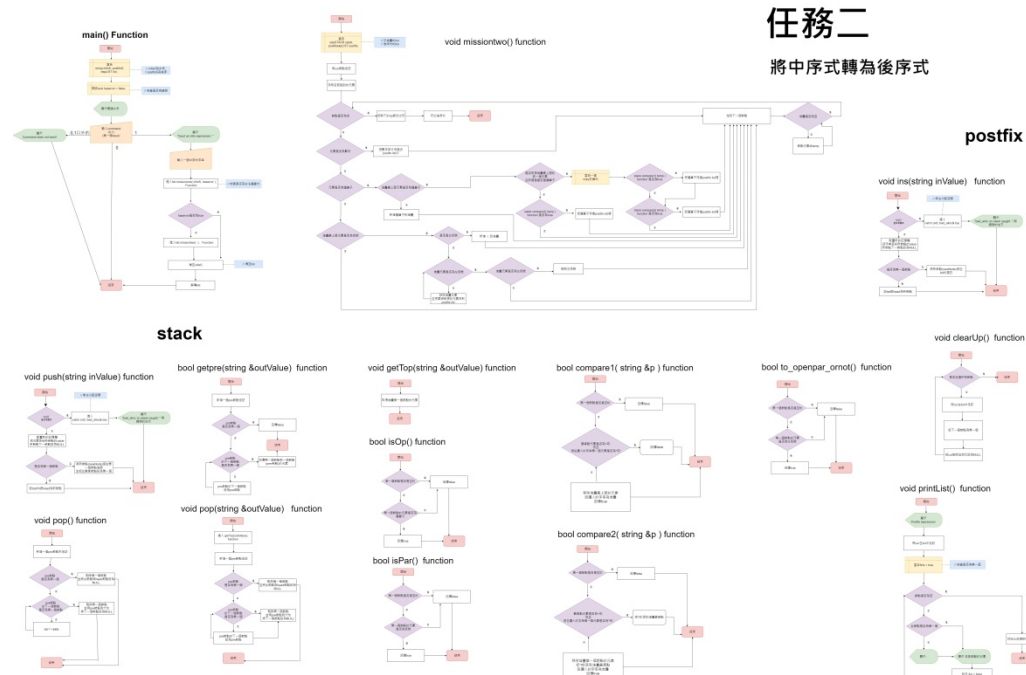
任務一:

任務一 合法運算式檢測



任務二:

任務二 將中序式轉為後序式



(三) 解說

影片連結: [資料結構 第 27 組 講解影片 - YouTube](#)