

上機練習二

11127115 陳姿安

一、簡介：

任務一：

用 string 的型別存取輸入的 infix expression，接著依序判斷此前序式是否合法：

1. 若出現除了括弧或加減乘除之外的符號(e.g. ^, %) → 非法前序式(運算子錯誤)
2. 若括弧沒有成對出現(e.g. (9*(9+2))) → 非法前序式(括弧不對稱)
3. 若出現其他非法情況(e.g. 9(5+6), (+3*9)) → 非法前序式(其餘狀況)

若皆無上述情況，即該前序式合法。

任務二：

一個一個判斷前序式中每一個字元：

1. 若該字元為數字 → 存進 operand 的 string
2. 若該字元為運算子 → 將 operand 加入存放 postfix 的 linked list
 - (1) 若該運算子為右括弧 → 直接存進 stack
 - (2) 若該運算子為左括弧 → 將在右括弧之後存進 stack 的運算子存進 postfix
 - (3) 若該運算子為 * 或 /
 - i. 若前一個存放進 stack 的運算子為 * 或 /
→ 先將前一個運算子存入 postfix 並刪除，再將目前的運算子加入 stack
 - ii. 若前一個存放進 stack 的運算子為括弧或 +, -
→ 直接將目前的運算子加入 stack
 - (4) 若該運算子為 + 或 -
→ 將之前的存進的運算子存入 postfix 直到遇到左括弧或 stack 為空
3. 若 operand 中仍有東西或 stack 中仍有運算符 → 存進 postfix

最後輸出轉換後的後序式。

發現：

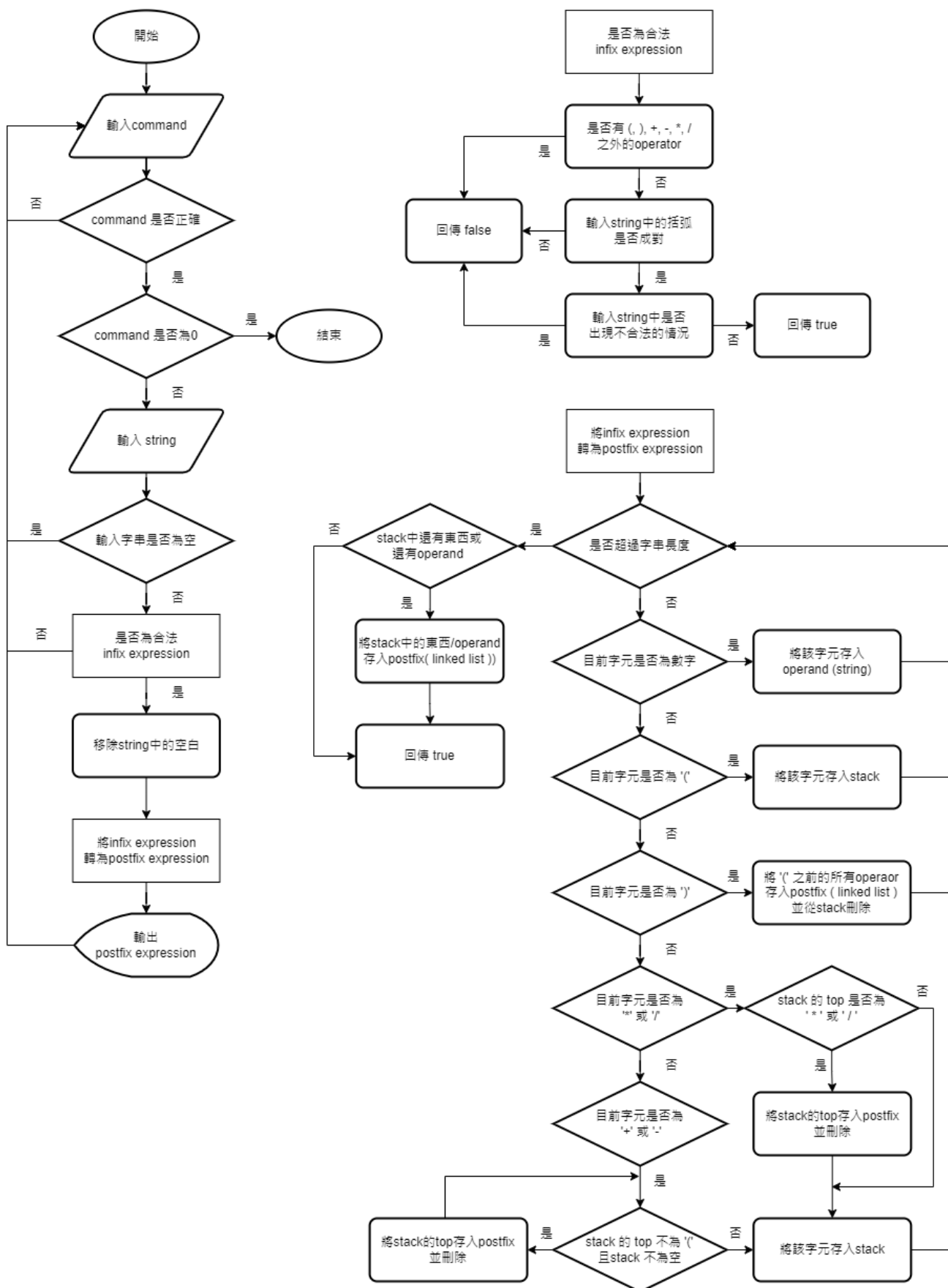
在寫中序式轉為後序式的 function 時，處理 + 及 - 這兩個運算符時想了很多該怎麼處理的想法，寫了很多判斷式，但停下來仔細思考之後發現，其實這兩個符號跟遇到左括弧的意義並沒有差很多，因為只要之前存進 stack 的符號不是括弧，那就都要先把之前的丟出來，以這個角度想的話程式碼就會簡潔了很多。

心得：

自己在測試程式有沒有錯的時候，本來很開心好像沒問題了，結果測到最後一個測資的時候發現原本應該是加號的地方卻變成了乘號，讓我很迷惑消失的家後到底跑去哪裡了，把整個程式重新檢查一次，試了很多地方，結果最後把兩行程式碼交換了之後加號就出現了……，突然覺得自己蠻蠢的。

二、流程圖：

任務一、任務二



三、解説：