

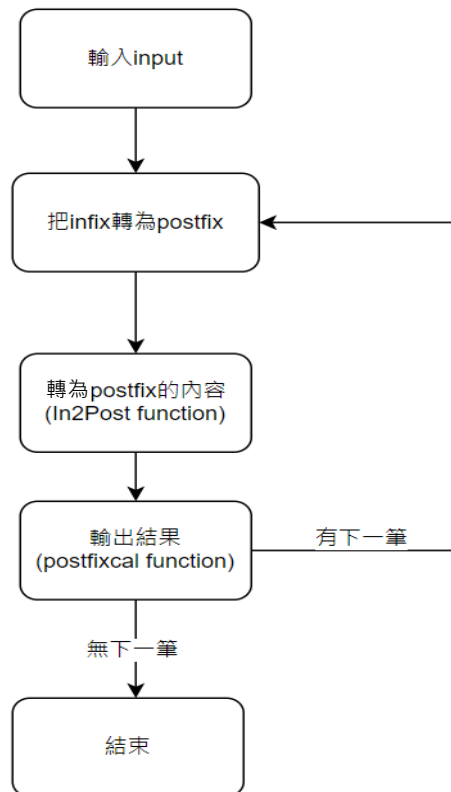
姓名：郭庭維

系級：會計 112

學號：H14086030

HW2-1 四則計算機

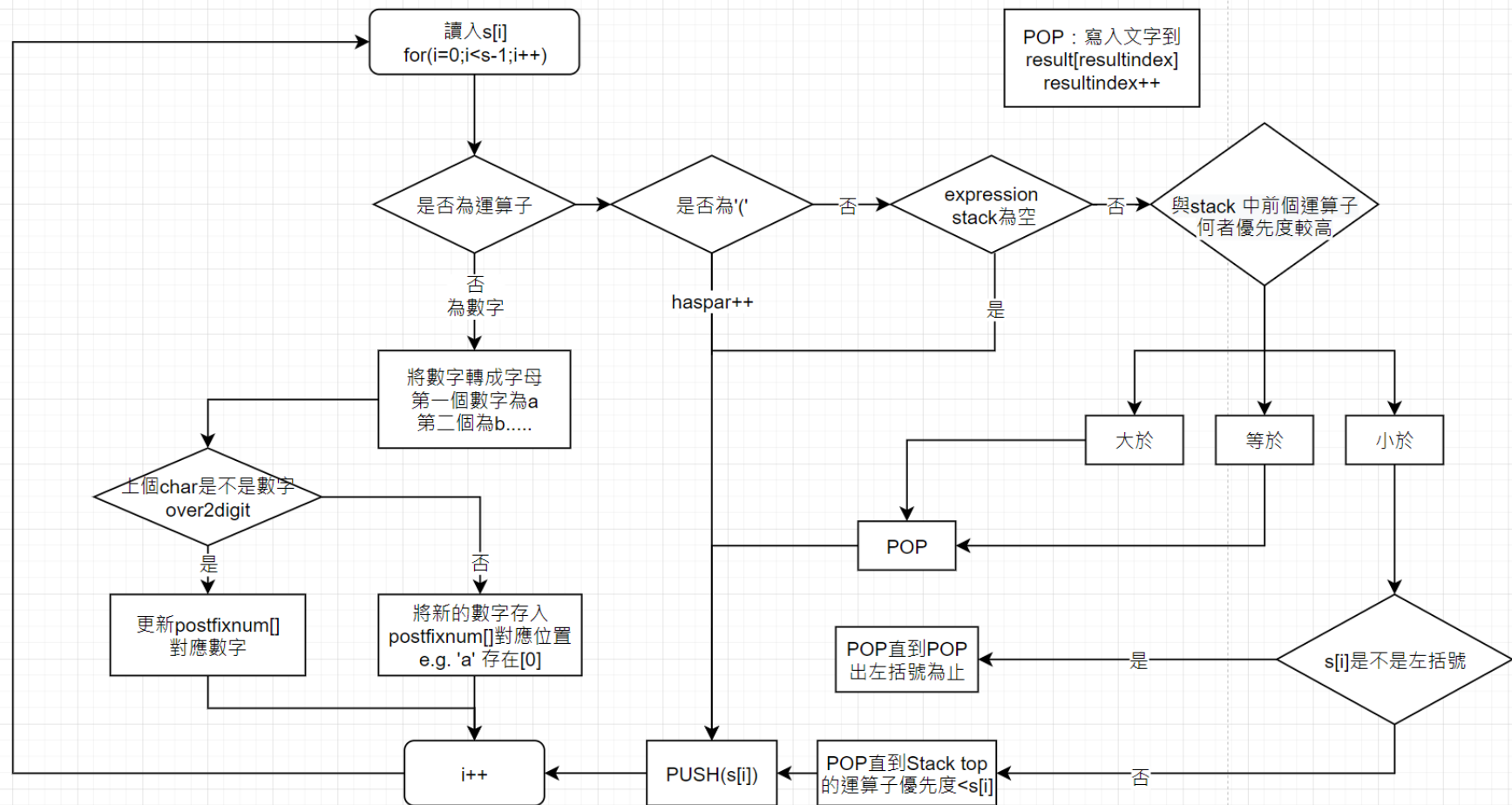
一、流程圖：



整體程式流程圖

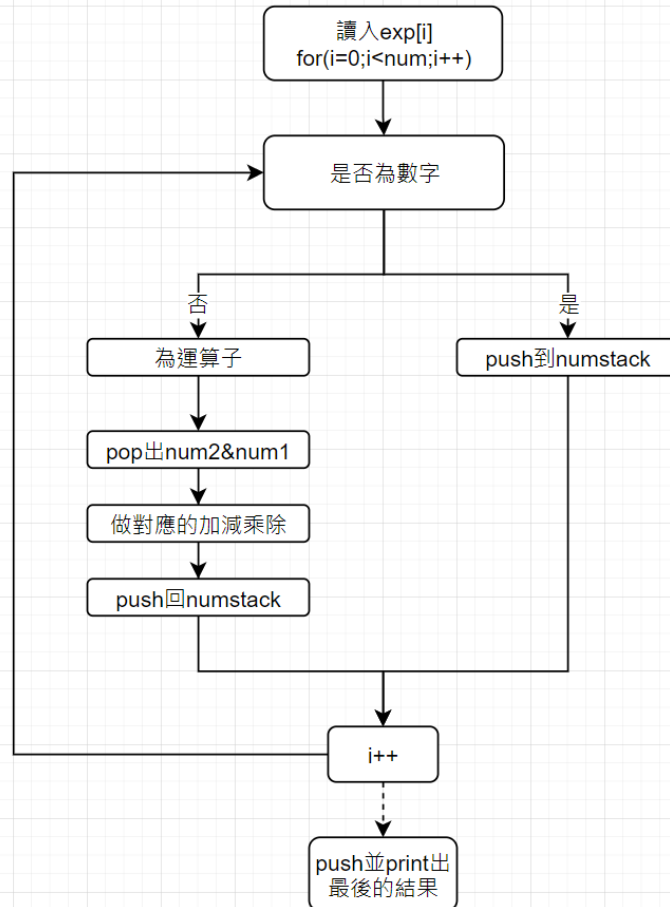
In2Post function

In2Post function
 從main得到
 input陣列s、s長度size、結果寫入位置result
 return result 長度(resultindex)



Postfixcal function

postfixcal function
從main得到
postfix expression exp[]
exp長度num
exp中數字a,b,c參照用的numindex[]



二、詳細程式說明：

主要寫一個 `char stack` 來處理運算子的操作
預設的 `stack` 頂端為-1，`char[]`大小為 100。

1. Push：

- i. `Top++`
- ii. 將資料寫入 `char[0]+top`

2. Pop：

- i. 若 `top!=-1`，則有資料
- ii. `Top--`
- iii. `Return char[top+1]`

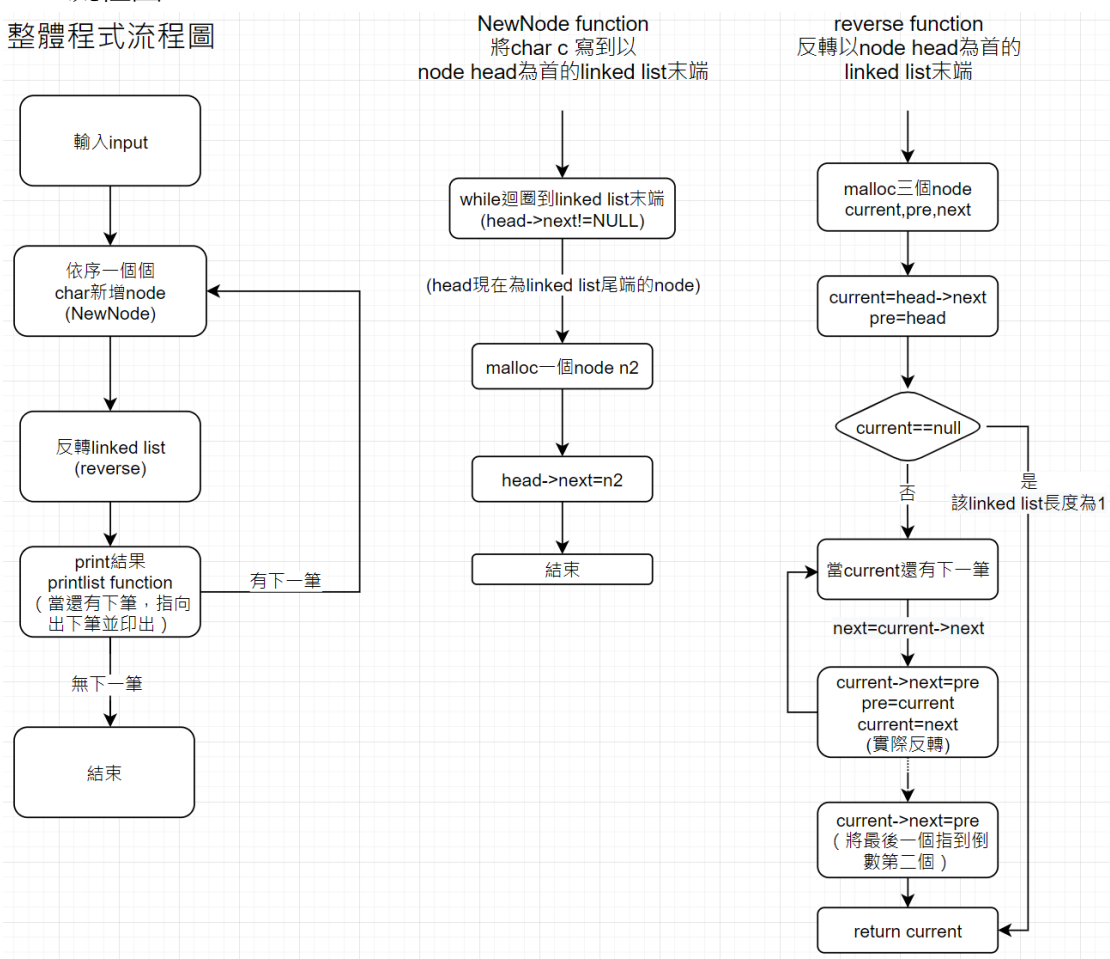
備註：詳細程式流程以及判斷方式均寫在點一流程圖中，包括何時要 `pop/push`

在 `postfixal function` 中 因為已經在實現過 `char stack` 了所以計算時在處理 `stack` 僅簡單使用一個 `float[]` 和 `int top` 來操作，沒有特別寫一個新的 `Structure`。

HW2-2 Reverse Linked List

一、流程圖：

整體程式流程圖



二、詳細程式說明：

Linked list node 形式



建立 linked list

先有一個head node



用for依次輸入資料，建立linked list

Malloc 一個新的node n2



將末端node指到 n2



反轉 linked list

Reverse linked list (以ABCD為例)



三、心得與討論：

1. W1：本來想說這應該是個蠻簡單的作業，但是後來發現要處理大於兩位數的運算，所以花了不少時間在研究要怎麼把'123'轉成(int) 123，原本要用 $1*100+2*10+3$ 來處理，可是 C 如果用 $\text{POW}(10,2)$ 會是 $\text{float}(99.999999)$ 轉成 int 會變成 99，最後用 $\text{atoi}('123')$ 才成功。
（可是後來又發現 test.txt 計算過程中會有小數，所以 number stack 也從 int 改成 float，整體來說如果沿用舊方法應該沒問題）剩下在 infix to postfix 的部分沒有太大的問題寫起來都很順。
2. 因為我自己沒有建立過 linked list 一開始也不知道要 malloc，不過後來發現後就順利寫出來了，中間只有忘了一開始的 head node 要指到 Null 導致 print 時候會不斷重複最後兩個字母花了一點時間

Time log：

W1：

日期	時間	分鐘數
4/6	11:40-12:00	20
4/6	21:10-23:00	110
4/7	09:40-12:00	140
4/7	14:15-16:30	135
4/7	20:15-22:30	135
4/8	06:15-08:00	105
4/9	00:50-01:30	40
4/9	11:30-12:05	35
4/9	14:15-16:00	105
4/9	16:40-17:40	60

共計：885 分

W2

日期	時間	分鐘數
4/12	16:40-17:00	20
4/13	9:10-10:30	90
4/14	10:30-12:20	110
4/15	11:00-12:00	60
4/15	17:00-17:30	30

共計：300 分