

## 一、規格要求，違反者以零分計！

- (a) 以 Dev-C++ 或 Code::Blocks 編譯與成功執行的 C/C++ 程式碼，必須有充分的註解。
- (b) 任何一部分程式碼都不得被偵測為抄襲，程式碼的第一列要註解學號和姓名。
- (c) 檔名限以「DS1101ex03\_組別\_學號\_學號」開頭，兩人一組只限繳交一份。

## 二、作業內容

整合下列任務於同一個程式及單一操作選單下，未整合、無法連續執行或沒有輸入防呆措施，都各扣 5 分。若導致任務無法正常運作，該任務不計分。

### 合法運算式簡述：

- 1. 運算元限合理範圍內的正整數，求解時碰到除數為零的狀況，程式必須能繼續正常執行。
- 2. 運算子只限使用整數的四則運算  $+$   $-$   $*$   $/$  及左右括弧  $()$ ，由左而右計算，內層括弧優先，其次是乘除  $*$   $/$ ，最後才是加減  $+$   $-$ 。
- 3. 範例： $(29+101)*33/25$ ,  $24 * 7770 / (55 + 30 * 2)$ ,  $(90 + (70 * (68 - 55 / 10)))$ ,  $69/3+30*5-24/8$ 。

### 必須遵守的規範：(每個任務違反一項就各扣 5 分)

- 1. 必須採用動態配置空間的鏈結串列儲存後序運算式的每一項。
- 2. 鏈結串列不再需要被使用時，必須立即釋放佔用的所有空間。
- 3. 必須自行實作堆疊類別，並依照教材介紹的運用方式來求解。

#### (任務一) 合法運算式檢測

輸入：一個允許空白字元的字串。

步驟：檢測字串是否為合法的中序運算式，至少包括(1)限用的文數字符號、(2)成對括弧及(3)中序運算式，發現一項錯誤即可停止檢測。

輸出：於螢幕上顯示檢測的結果訊息，若發現錯誤，必須明確指出錯誤類型，例如：(1)非合法字元、(2)缺少或多餘的括弧、(3)非中序的運算式。

#### (任務二) 將中序式轉為後序式

輸入：(任務一)確定是中序運算式的一個字串。

步驟：自訂一個【堆疊】類別的資料結構，藉此將輸入字串轉為後序運算式，並以鏈結串列儲存，後序運算式的每個運算元和運算子各自記錄成一個節點。

輸出：將鏈結串列儲存的後序運算式由左而右依序顯示於螢幕上，以逗號隔開每個運算元和運算子。

程式碼：上機前一晚(週一)上傳原始碼至/評量區/作業/作業#3 程式碼/，缺交者於隔日補繳。

流程圖：上機結束前(週二)上傳流程圖至/評量區/同儕互評/作業#3 流程圖/，不接受補繳。

## 三、參考範例，格式自訂，內容不可欠缺！

### (任務一) 合法運算式檢測

Input:  $(29+101)*33/25$

It is a legitimate infix expression.

Input:  $24 * 7770 ^ (55 + 30 * 2)$

Error 1: ^ is not a legitimate character.

Input:  $(90 + (70 * (68 - 55 / 10)))$

Error 2: there is one extra open parenthesis.

Input:  $(29+101)*33/25)$

Error 2: there is one extra close parenthesis.

Input:  $69/3++30*5-24/8$

Error 3: there is one extra operator.

Input:  $(24 * 7770) 19 / (55 + 30 * 2)$

Error 3: there is one extra operand.

### (任務二) 將中序式轉為後序式

Input:  $(29+101)*33/25$

Postfix expression: 29 , 101, +, 33, \*, 25, /

Input:  $24 * 7770 / (55 + 30 * 2)$

Postfix expression: 24, 7770, \*, 55, 30, 2, \*, +, /

Input:  $(90 + (70 * (68 - 55 / 10)))$

Postfix expression: 90, 70, 68, 55, 10, /, -, \*, +

Input:  $69/3+30*5-24/8$

Postfix expression: 69, 3, /, 30, 5, \*, +, 24, 8, /, -