

Data Structure

Program Homework 3

題目：簡單運算式轉換器與計算器

- 題意說明：

1. 請撰寫一個以 C/C++ 語言實作的簡單運算式轉換器與計算器，要求使用者輸入一個 infix 表示法的數學運算式，然後構建對應的二元運算樹(Binary Expression Tree)，並進行以下功能：

- 1) 以 Level-order 輸出運算樹內容
- 2) 輸出 Prefix 運算式
- 3) 輸出 Postfix 運算式
- 4) 計算結果

- 功能及輸入與輸出格式參考：

1. 顯示 "Please enter an infix expression and press enter:"，並讀入用戶的 infix 運算式輸入。
2. 每一筆輸入資料必須為 5 至 20 個變數形成的 infix 數學運算式，由以下變數構成：英文字母 "A"至"Z"。左括弧 (與右括弧)、運算子 +,-,*,/。輸入數學運算式中不允許有空白字元，同一變數可重複出現。
3. 若輸入的運算式合法，則依次執行以下步驟：
在螢幕下一行顯示 "The postfix expression:"，並輸出對應的 Postfix 運算式。
在螢幕下一行顯示 "The prefix expression:"，並輸出對應的 Prefix 運算式。
在下一行顯示 "=", 接著輸出計算結果。
4. 重複上述步驟，直到用戶按 "Esc" 鍵，則停止程式執行。

範例：

輸入：a+b*(c-d)

輸出：

The level-order of the expression tree:

+

a *

b -

c d

The postfix expression: abcd-*+

The prefix expression: +a*b-cd

請用戶指定變數的值，例如：

a = 2, b = 3, c = 5, d = 1

輸出計算結果：

= 14

- 程式內容需求

- 運用 stack 將輸入的 infix expression 建構出對應的 binary expression tree。
需要一個 stack 用來暫存 operator node，根據 operator 的優先權，決定 infix 運算式中 operators 的運算順序。(運用 stack 將 infix 轉成 postfix expression 的過程)。再依 postfix expression output 的輸出順序，建立對應的 binary expression tree。(20 分)
- 底層運用 node with children pointers 記錄 binary expression tree 資料內容。
- 實作以下功能(50 分)
 - (1) 使用 Level-order Traversal 輸出運算樹內容，每個層級的節點輸出在單獨的一行上。
 - (2) 透過 Preorder Traversal 輸出對應的 Prefix 運算式。
 - (3) 透過 Postorder Traversal 輸出對應的 Postfix 運算式。
 - (4) 讓使用者指定運算式中變數的值，透過 Postorder Evaluation 計算並輸出最終結果。

- 程式評分標準

1. 正確性。
2. 輸出入介面設計(考慮使用及操作便利性，結果顯示方式是否清楚明瞭)。(5 分)
3. 程式撰寫模組化 (功能需採用對應模組撰寫)。(15 分)
4. 書面報告: 各主要 procedure 功能簡要說明，輸出入方式說明，檢附自行測試畫面結果。(10 分)
5. on time (-10 分 each day late)

- Turned in

從 moodle 上傳 (必須附書面報告及 source code), 若需要請配合助教在規定時間內 demo

✱ 明顯相互抄襲程式內容(e.g. 只有變數名稱不同者)、未繳交書面報告及無法執行未配合助教 demo 者，以 0 分計算。