

學號: 41147046S 姓名: 楊子萱

使用語言: C++

編譯: 把資料夾解壓縮後在終端機(Linux)下 make 後會出現 hw3_41147046S 的執行檔

執行: 在終端機輸入 ./hw3_41147046S

執行步驟:

輸入運算式後會根據運算式中有出現的字母順序出現，請輸入一個數字，不須先空格，輸完最後一個字母代表的數字後會跳出 level-order tree、postfix expression、prefix expression 的結果，請再按下一個 '='，會跳出運算式的結果

```
Please enter an infix expression and press enter:a+b*(c-d)+e
a = ?2
b = ?3
c = ?5
d = ?1
e = ?4
The level-order of the expression tree:
+
+e
a*
b-
cd
The postfix expression: abcd-*+e+
The prefix expression: ++a*b-cde
=
18
```

若要離開，請按下 esc 鍵並按下 enter

```
Please enter an infix expression and press enter:^[
```

如果運算式中誤打了空格(space)，會出現錯誤訊息並讓使用者重新輸入

```
nieves@nieves-VirtualBox:~/data structure$ ./hw3_41147046S
Please enter an infix expression and press enter:a +b*(c-d)
Input shouldn't have any space
```

若運算式不符合“含 5~20 個變數”的規定，也會出現錯誤訊息，並讓使用者再次輸入

```
nieves@nieves-VirtualBox:~/data structure$ ./hw3_41147046S
Please enter an infix expression and press enter:a+b*(c-d)
there should be 5~20 operand in operational expression
```

各 function 介紹:

bool isOperator(char c): 判斷字符是否是+*/中其中一個

int precedence(char op): 設定運算符的優先序，*/比+-高

Node* constructTree(const string& infix): 根據運算式建一棵 tree

void levelOrder(Node* root): 輸出 tree 的 levelOrder，每一層都會換一次行

void postOrder(Node* root, string& expression): 輸出運算式的 postfix，並且把結果記進 string expression 裡

void preOrder(Node* root): 輸出運算式的 prefix

int evaluate(const string& expression, map<char, int>& num): 計算運算式在給定的數字下的運算結果