

10610CS235101

Data Structures

Homework 1

2017/10/17 10:10

~

2017/10/31 23:59

Target

1. Check the format of parentheses that is valid or not
2. If true,
 output “True” and
 evaluate the value of expression
 If false,
 output “False” and
 transform the expression to the correct form and
 evaluate the value of expression

I/O

- Input format:
 - $0 < |\text{input}| \leq 1000000$
 - **1st character**一定是 (或 { 或 [
 - Operands range from **0 to 9**.
 - The set of operators contains **() { } [] + - * /**
 - / : integer division, e.g. $5/4=1$, $5/8=0$
 - No space character in expression

I/O

- Output format:
 - If true,
 - True
 - the value of expression
 - If false,
 - False
 - transform the expression to the correct form
 - the value of expression

Input	Output
{() }	True 0
(7/[5+1]	False (7/[5+1]) 1

Please follow the format on the above to prevent the error on Online Judge

The definition of input data

A. 有運算式要計算,其運算式為infix expression e.g. $[(2+3)*8/3]$

1. 單一括號內必定會是合法的infix expression
 - e.g. 不會有 $5(+)3$, $(* 5)$, $(- 5)$
 2. 不會有兩數之間沒有arithmetic operators(根據infix expression)
 - e.g. 不會有 $(3)(5)$
 3. 不會有/0的情況
 4. 在運算式中,不會有空括號計算的情況
 - e.g. 不會有 $[4+5/()]*9$, $(() + 2)$
 5. 在運算式中,不會有括號的外側直接連接數字
 - e.g. 不會有 $(3*5)5$, $3(4/3)$
- > 若在運算式之外,則數字可與空括號連結
e.g. $\{ \} \{4+5\} * 6 \{ \}$

運算式第一個數字出現的左邊和
運算式最後一個數字的右邊,就可
以出現空括號

e.g. $\{ \} \{4+5\} * 6 \{ \}$

結論是運算式從開始到結束前,不
會出現5.此種可能,但運算式之前
或之後可以出現

前面和後面出現的括號規則就等
同於B類型的input

除了括號錯,不會有其他錯
不會出現的input大家就不需要考慮囉~

B. 沒有運算式,只有單純括號排列 e.g. $(((\{ \} \{ \} () \{ \})) [[[[]]]]$

- value=0

The definition of parentheses that is valid or not

the priority of operator is “left to right”

- 不管三種括號的優先權,(小括號),[中括號},{大括號} 都是等價的
- 運算式中即使括號不必要,只要相互對應合法就好 -> () [] {}
 - Valid input: ({ [2] }), { [3+5] }, (), ([])
- 相對應的右括號沒合法,要修改成對應合法的括號
 - Invalid input: (} { (}] , ((}]
 - Valid output: () { () } , (())
- 相對應的位置沒有右括號要補滿並且要為對應合法的括號
 - Invalid input: { (, [[({
 - Valid output: { () } , [[({ })]]
- 沒有和它相對應的左括號,即為多餘的右括號,要刪掉
 - Invalid input: ()] , { { } }))
 - Valid output: () , { { } }

顏色一樣的就是相對應的位置

Output

If false,

the process of devising and evaluating the value

- $[() \{2+3*5\}]]]$
→ $[() \{2+3*5\}] = 17$
- $\{[1+3+3])) / 8] + 5 ($
→ $\{[1+3+3] \} / 8 + 5 () = 5$
- $() [] \{ () \}$
→ $() [] \{ () \} = 0$
- $(((6-8) / 2$
→ $(((6-8) / 2)) = -1$

Resulting value (int)

- Must be positive or negative or zero
- No overflow situation

R: 多餘

G: 對應錯誤

B: 補齊

Input	Output
$\{ \{ (2+3) * 5 \} \}$	False $\{ \{ (2+3) * 5 \}$ 25
$(\{ [() 7 / [5 + 1]$	False $(\{ [() 7 / [5 + 1] \})$ 1
$\{ () \} [] []$	True 0
$() [] \{ () \}$	False $() [] \{ () \}$ 0
$(3 + 4 * 5) () [\{ \}$	True 23
$\{ (5 + 9 * 1 + 2) / 2 + [5 - 4] \}$	True 9

Be reminded & tips

“ left to right ”

- 一定要照著規則走, 其實就是跟著**stack規則**走
- 一個input只會得到唯一的output, 一筆測資內可能包含多個input
- 同一組對應的括號不合法的話, 更正成合法的右括號去對應左括號
- 多餘的左括號, 補齊相對應合法的右括號
- 多餘的右括號, 刪掉

左括號必定保留, 右括號去對應
錯誤修正的priority: 更正 > 補齊 = 刪掉

STL is not allowed

`<stack>` `<queue>` `<list>` `<vector>` are not allowed

- `main.cpp` & `function.h` 會給你們,不能更改
- 因此不會有`<stack>``<queue>`被宣告

Submission

- OJ : #11595 - DS_2017fall_HW1
 - Submit your code
- ILMS :
 - 遵照之前公告的作業繳交模式
 - 以Homework1為例,請將所有檔案放到名為hw1的資料夾之下,並且將hw1打包成一個.zip檔

Zip檔裡要包含 code.cpp, function.h, main.cpp

Zip檔的命名格式

- 學號_作業_版本.zip
 - 例: 101062122_hw1_v1.zip
 - 檔案版本用以識別避免爭議
 - 所有英文檔案名皆須為小寫
- 勿使用zip以外的格式, 如tar,rar,7z等

上傳到ILMS 和 OJ 的code必須一致,否則不計分,一律0分