

Homework2

STACK exercise

DUE: 4월 6일

HW2-1: Parentheses Checker

Program: Write a program to read a text file and print whether or not the parentheses are balanced.

Data file should contain the following 10 data

$(A * B) + (C * D)$
 $(A * (B + (C * D + E)))$
 $\{ [A + B] - [(C - D)] \}$
 $(A + B) - \{ C + D \} - [F + G]$
 $((A))$
 $((A + B)$
 $) A + B (- C$
 $(A + B)) - (C + D$
 $(A + B))$
 $\{ [A + B] - [(C - D)]$

Output:

$(A * B) + (C * D)$	balanced
$(A * (B + (C * D + E)))$	balanced
$\{ [A + B] - [(C - D)] \}$	balanced
$(A + B) - \{ C + D \} - [F + G]$	balanced
$((A))$	balanced
$((A + B)$	unbalanced
$) A + B (- C$	unbalanced
$(A + B)) - (C + D$	unbalanced
$(A + B))$	unbalanced
$\{ [A + B] - [(C - D)]$	unbalanced

[주의] (, [, {, },],) 등의 개수를 세는 방법은 안되며 반드시 Stack을 활용해야 함

HW2-2: Infix to Postfix Conversion & Postfix Evaluation

Program:

1. Convert the **infix data** into **postfix data** by using infix-to-postfix algorithm.
2. And then, **Evaluate the Postfix data**
3. 전체 프로젝트는 다음과 같이 구성한다.
 - 1) infix.cpp: main program file
 - 2) infix.h: header file (ex: #include file, constants, variables,...)
 - 3) infixADT.cpp: ADT file (push, pop, create, ...)

Note:

1. 수식 데이터는 Keyboard 에서 입력 받는다. (input data from keyboard)
2. 입력 받은 수식을 화면에 다시 한번 출력한다. (Echo data)
3. infix-to-postfix algorithm 에 의해 conversion 한 후 postfix form 을 출력한다.
(print the data as POSTFIX form)
4. Postfix evaluation 알고리즘을 사용하여 최종값을 출력한다. (print the final result)
 - Postfix form 은 9 미만의 십진수를 postfix 형태로 표현한것을 이용한다.
(the number must be below 9)
 - 사칙연산만을 사용한다. (use the operator only "+ - * /")

<Output example>

- 0) Enter Data: 3 + 2
- 1) Echo data (infix form) : 3 + 2
- 2) Conversion (postfix form): 3 2 +
- 3) Result : 5