Programming Assignment 3

Infix Expression Calculation

In this programming assignment, you will implement a program calculating an infix arithmetic expression given as a string. For simplicity, we only consider three operators (`+`, `-`, `*`), parentheses, and single digit integers (1~9). The parentheses can be nested. We do not consider plus and minus signs. So, in a given expression, one character always corresponds to one symbol. Two functions need to be implemented: `in2post()` that converts an infix expression to a postfix expression, and `postcalc()` that calculate a post expression. Please freely declare and define variables, user data types, and local functions. However, all of the implementation codes should be in `main.c`. Your implementation will be tested by given testing codes and data files. If you successfully pass the tests, you will see the screen like the below.

```
[Test1: postfix expression validity check]
   1234+*5678+*-9*++ is valid and correctly calcualted: -750
   1234+*+5678+*-9*+ is valid and correctly calcualted: -750
   1234+*+5678+*-9* is not valid
   234+*+5678+*-9*+ is not valid
   1234+*+678+*-9*+ is not valid
[Test2: postfix expression calculation]
   100000 correct answers for 100000 expressions: 100.00
[Test3: infix to postfix conversion]
   1+2*3+(4*5+6)*7 --> 123*+45*6+7*+
   1+2*3+((4*5+6)*7) --> 123*+45*6+7*+
   (1+2*3)+(4*5+6)*7 --> 123*+45*6+7*+
   (((1)+(2*3))+((4*5+6)*7)) --> 123*+45*6+7*+
   1+2*3+((4*5+6)*7 is not valid
   1+2*3+(4*5+6)*7) is not valid
   1++2*3+((4*5+6)*7) is not valid
   1+2*3+((4*5+6)*7+) is not valid
   1+2*3+((4*5+6)*7)+ is not valid
   1+2*3+(9(4*5+6)*7) is not valid
   1+2*3+((4*5+6)*7)9 is not valid
[Test4: infix expression calculation]
   100000 correct answers for 100000 expressions: 100.00
```

Please note that the whole program can be run within 2 seconds on my laptop (i7, 16GRAM). If you feel your implementation takes too much time, probably it indicates that your implementation has a problem.

You should implement from the bottom with only C standard library. Any code from the Internet

and third-party libraries are not allowed. If you cannot pass all tests, please do your best. You can have partial scores.

With your implementation, please answer the following question.

[Test 1-2] Answer the following questions with the below postfix expressions.

(1)	`1234+*5678+*-9*++`
(2)	`1234+*+5678+*-9*+`
(3)	`1234+*+5678+*-9*`
(4)	`234+*+5678+*-9*+`
(5)	`1234+*+678+*-9*+`

1. What is the infix expression of (1)? What is the calculated value? What is the infix expression of (2)? Is it possible that the same infix expression has the different postfix expression?

2. Expression (3) ~ (5) are not valid. Explain why.

3. Implement `postcalc()` function. This function takes a postfix expression as a string and returns the calculated value. If the expression is not valid, it should return 0. (Of course, it should return 0 if the calculated value is 0). Explain clearly about (A) your stack structure and implementation, (B) validity checking procedure, and (C) the overall calculation procedure. Please provide your code and the screen shot of Test 1. Your code should include all data types, variables and local functions used in `postcalc()`.

4. Run Test 2 and provide the screen shot. Report your score.

[Test 3-4] Answer the following questions with the below infix expressions.

(1)	`1+2*3+(4*5+6)*7`
(2)	`1+2*3+((4*5+6)*7)`
(3)	`(1+2*3)+(4*5+6)*7`
(4)	`(((1)+(2*3))+((4*5+6)*7))`
(5)	`1+2*3+((4*5+6)*7`
(6)	`1+2*3+(4*5+6)*7)`
(7)	`1++2*3+((4*5+6)*7)`
(8)	`1+2*3+((4*5+6)*7+)`
(9)	`1+2*3+((4*5+6)*7)+`
(10)	`1+2*3+(9(4*5+6)*7)`
(11)	`1+2*3+((4*5+6)*7)9`

5. What are the postfix expressions of (1) ~ (4)? Do they have the same expression? If they do not, please explain why.

6. Expression (5)~(11) are not valid. Explain why.

7. Implement `in2post()` function. The function takes an infix expression as a string. The other argument is a character array where the converted postfix expression will be stored. Here, we assume that the array is long enough so that we do not worry about overflow. The converted expression should be given as a string, which means it ends with a null character. It returns 1 if the conversion is successful. Otherwise, it returns 0. Explain clearly about (A) your stack structure and implementation, (B) validity checking procedure, and (C) the overall conversion procedure. Please provide your code for `in2post()` and the screen shot of Test 3. Your code should include all data types, variables and local functions used in `in2post()`.

8. Run Test 4 and provide the screen shot. Report your score.

Please submit two files:
(1) Report: a single electronic file including answers of the questions, screen shots, source code in the format of doc, hwp or pdf.
(2) main.c: a plain-text source code file that can generate the result of the report. Ideally, the screen shot on page 1 can be reproduced with this file.

Note that your assignment will be graded with only the report. So, if you do not include source codes in the report, you will not get scores. The plain-text source code will be considered when you have problems, for example, suspicion of the copy&paste or non-sense results. If the plain-text source code is different from codes on the report or it cannot reproduce the results on the report, then your grade will be 0. Please submit the above files through the blackboard system.

If you have any question for this project assignment, please contact me or TA.