10910EECS204001  
Data Structure HW2-Elementary arithmetic

NTHU OJ 12925

Description:

In this homework, you are asked to implement Stack, Queue and certain arithmetic operations (addition, subtraction, multiplication, division and brackets) to evaluate elementary arithmetic expressions, and update the expressions with Dequeue, Enqueue operations.

Functions:

1. Implement Queue to store the expression elements

Eg: 1+2\*3 front back

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | + | 2 | \* | 3 |

Queue

2. Dequeue **N**: Delete N elements from the Queue’s front (where N is an integer and 0<N<= *the number of elements in current expression*).

3. Enqueue **E**: Add the element E at the back of Queue. (E is an operator or operand).

4. Implement Stack that transforms expressions from infix to postfix and evaluates the expressions.

5. Implement the function to transform the expressions from infix to postfix.

6. Implement the function to evaluate the expressions.

7. Check if the expression is legal.

Example:

(1) Binary operation:

Legal: a\*b (one operator with two operands)

Illegal: a\*b\* (the first operator \* can perform binary operation with two operands. However, the second operator \* cannott operate with only one operand)

(2) Unmatched parenthesis:

Legal: (1a+b)2/(3(4c-d)5-e)6

(1 is matched with )2, (3 is matched with )6, (4 is matched with )5.

Illegal: (1(2a+b)3

(1 has no parenthesis to match.

(1a+b)2)3

)3 has no parenthesis to match.

8. Print: Retrieve all elements in queue as an expression, and determine if the expression is legal (8.1) or not (8.2)

p.s. After retrieving all the elements, you should put it in a queue again for the upcoming commands.

8.1. If the expression is *illegal*, output the current expression (1) and the string “Invalid” (2).

8.2. If the expression is *legal*, output the current expression (1), expression postfix (2), and the result of the expression (3).

Input:

1. **N** lines of commands where **N**<=100.

1.1. The first line is an expression.

1.2. The other lines are the commands, including Dequeue (followed by an integer), Enqueue (followed by an operator or operand) or Print.

2. The number of operators ( +, -, \*, / ) in the expression (including the first line and any moment we perform an Enqueue or Dequeue operation) is smaller than 20; The number of operators ( ( , ) ) is smaller than 20 ;The number of operands is smaller than 20.

3. The expression does not contain any non-divisible condition.

Eg: 8/3 does not appear in the expression.

4. All operands in the expressions are integers within [0,9].

5. The result of the expression (including the first line, at any moment we perform an Enqueue or Dequeue operation, and any intermediate result of the process of calculating) is an integer in range -2147483648 to 2147483647.

Output:

For output command “Print”, if the current expression is illegal, print out the current expression, shift to newline and print out the string “Invalid”;

If the expression is legal, print out the current expression, shift to newline; Then, print its postfix and shift to newline; Finally, the result of the expression.

Sample I/O:

|  |  |  |
| --- | --- | --- |
| Input | 1+2\*3  Print  Dequeue 2  Print  Enqueue /  Enqueue 6  Print | ((1+2)\*9-2  Print  Dequeue 1  Print  Enqueue \*  Enqueue \*  Enqueue 3  Print |
| Output | 1+2\*3  123\*+  7  2\*3  23\*  6  2\*3/6  23\*6/  1 | ((1+2)\*9-2  Invalid  (1+2)\*9-2  12+9\*2-  25  (1+2)\*9-2\*\*3  Invalid |

Notice:

Any kind of STL is not allowed in this homework. (string is allowed)

Your code needs to be submitted to NTHU OJ and iLMS before the deadline.

Please zip your code as studentID.zip and submit to iLMS HW2.(eg, 108062568.zip)