
Calculator

P40155_en

Segon Concurs de Programació de la UPC - Primera Semifinal (2004-09-14)

Your little sister has lost her pocket calculator just before an important exam. She asks you to lend her your portable computer and to write her a program to evaluate single expressions. Your fraternal love is so big that you immediately agree on both things.

The calculator that your sister wishes is very simple: expressions can only contain one-digit numbers and the four basic arithmetic operations (where divisions should be understood as integer divisions). Parentheses are also allowed. The calculator must honor the usual priority rules. All operators have left-to-right associativity (just like in C++). All expressions strictly adhere to the following syntax:

$$\begin{aligned} \text{expression} &\rightarrow \text{expression} + \text{term} \mid \text{expression} - \text{term} \mid \text{term} \\ \text{term} &\rightarrow \text{term} * \text{factor} \mid \text{term} / \text{factor} \mid \text{factor} \\ \text{factor} &\rightarrow \text{number} \mid (\text{expression}) \\ \text{number} &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

Input

Input consists of several expressions following the grammar above, with no spaces at all. No expression has more than 2000 characters. You have the guarantee that no division by 0 will ever occur, and that no final nor intermediate result will overflow in a 32-bit `int`.

Output

For every expression, print the result of its evaluation.

Sample input

```
4
4+4
4-2-2
(8+7+9-1-2)*(1-2)*((5-1)+2)+1
3+3*2
9/3/3
5/2
((8))
(1-9)-(3-4)
```

Sample output

```
4
8
0
-125
9
1
2
8
-7
```

Problem information

Author : Jordi Petit
Generation : 2013-09-03 14:07:37