

## PROBLEM J

## SNAGGLE

30 POINTS

Professor McSnort has invented a new programming language, *Snaggle*. An expression in *Snaggle* may be a positive or negative integer, or may be of the form  $(p\ e_1\ e_2)$ , where  $p$  is a floating point number between 0.0 and 1.0 (inclusive) and  $e_1$  and  $e_2$  are *Snaggle* expressions.

The value of a Snaggle expression is defined as follows:

- The value of an integer is the integer value itself.
- The value of the expression  $(p\ e_1\ e_2)$  is  $x+y$  with probability  $p$  and  $x-y$  otherwise, where  $x$  and  $y$  are the values of the *Snaggle* expressions  $e_1$  and  $e_2$  respectively.



## Input

Input consists of up to 25 Snaggle expressions, one per line, followed by a line containing `()`, which should not be processed.

- Integers are sequences of at most 10 digits, optionally preceded by '-'.
- Real numbers are in standard floating point format (i.e. not E format) without a sign.
- Snaggle expressions of the form  $(p\ e_1\ e_2)$  have a single space separating the three elements and no spaces elsewhere.
- Snaggle expressions are at most 300 characters in length.

*Expected value*

The expected value of a random variable is the weighted average over all possible outcomes. For example, if a variable has the value  $n_1$  with some probability  $p$  and the value  $n_2$  otherwise, the expected value is  $p \times n_1 + (1 - p) \times n_2$ .

## Output

Output is a single line for each Snaggle expression in the input giving the expected value of the expression to two decimal places.

## Sample Input

```
7
(0.5 3 9)
(0.125 (0.5 100 200) -1000)
()
```

## Output for Sample Input

```
7.00
3.00
850.00
```