Postfix Evaluation

1 second, 128 MB

A mathematical expression we use everyday is written in *Infix notation* that requires parentheses, e.g.,

(3 + 5) / (5 - 2)

However, there are many other notations. In this problem, we consider the postfix notation. The above expression can be written as

3 5 + 5 2 - /

Read more about postfix notation at: https://en.wikipedia.org/wiki/Reverse_Polish_notation

To evaluate expressions in postfix notation using a stack, we iteratively consider each token. If it is an operand (a number), push it on top of the stak. If it is an operator, take the top two operands on the stack, perform the operation, and push the result back on top of the stack.

Write a program that reads an expression in the postfix notation and evaluate it. You can start with the template provided at http://theory.cpe.ku.ac.th/~jittat/courses/algo-lab/postfix/ .

Input

The input consists of many lines. Each line contains one token of the expression. There are 3 types of token:

- Operands, i.e., an integer.
- Operators, one of + - * /
- Last token: =

The last token is always '='

Output

You program should output a single line containing the result of the evaluation. The number should be printed with 4 decimal digits.

Examples

Input	Output
3 5 + 5 2 - / =	2.6667

Input	Output
10 20 + 30 * 10 5 / + =	902.0000