Programming Part

Problem 4 - Evil Eval (100 pts)

Problem Description

In this problem, you "simply" have to implement a calculator like what has been taught in the classes. The calculator should support the following operator on double-precision floating points with correct precedence:

- arithmetic operators: `+`, `-`, `*`, `/`
- parentheses: `(`, `)`

It is guaranteed that the divisor would not be zero during the process of the calculation.

Please do not try to solve this problem by calling other program in your source code. If (and only if) you patiently, wholeheartedly code the homework out, you will gain a better coding skill and a deeper understanding of the data structure!

Input

The input contains T lines, each representing an arithmetic expression.

Output

For each test case print a double-precision floating point number in one line, which indicates the answer of the expression. Your solution will be considered correct if the absolute or relative error between the answer and your output is less than 10^{-12}. Formally, let your answer be a, and the jury's answer be b, your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-12}$

Constraints

- $0 <$ the length of each line $L < 10^6$
- $0 < a_i < 10^8$ for each number a_i in the expression
- $L \cdot T \leq 10^6$
- every numbers in the input will be an integer containing only of digits. We expect the final output to be a floating point number, though.

Subtask 1 (25 pts)

- the operator include only `+`, `-`

Subtask 2 (25 pts)

- the operators include only `+`, `-`, `*`, `/`

Subtask 3 (50 pts)

- all operators are possible

Sample Cases

Sample Input 1

```
1+3-2
```

Sample Output 1

```
2.000000000000000
```

Sample Input 2

```
1+2*3
1-2*3
```

Sample Output 2

```
7.000000000000000
-5.000000000000000
```

Sample Input 3

```
(1+2)*3
```

Sample Output 3

```
9.000000000000000
```