

La resistance!

In high-school physics we learn that to calculate the equivalent resistance of a circuit, we need to apply two simple rules:

- A circuit laid in **sequence** has a resistance equivalent to the **sum** of each subcircuit resistance.
- A circuit laid in **parallel** has a resistance equal to the **inverse of the sum of the inverse** of each subcircuit resistance.

As circuits get more complex doing these calculations by hand becomes a tedious process.

Luckily we are not in high-school anymore, we are seasoned software developers and can write a program to calculate that for us!

Input description

The circuits are coded as **s-expressions**.

Individual resistance values will be natural numbers $1 \leq R_i \leq 100$.

A sequence of resistors is expressed as $(S\ R_1\ R_2\ R_3\ \dots)$ and the aggregated resistance value is $R_1 + R_2 + R_3 + \dots$.

A parallel layout of resistors is expressed as $(P\ R_1\ R_2\ R_3\ \dots)$ and the aggregated resistance value is $1 / (1/R_1 + 1/R_2 + 1/R_3 + \dots)$.

There can be many cases per input; each case will be a single line containing an s-expression.

You can assume input is well-formed (no extraneous characters, well balanced s-expressions, regular whitespace).

Output description

A single line for each test case, with the aggregated resistance value of the circuit; expressed as a normalized (irreducible) fraction.

Samples

Input

```
42
(S 1 2 3)
(P 2 3)
(P (S 1 1) 2)
(S 1 (P 2 3) (P 4 (S 5)))
```

Output

42
6
6/5
1
199/45

Instructions

As in the previous problem, you are provided with 4 input files and only 3 output files. The output for the fourth problem will be used to validate your solution.

There's also a password-protected zip file, which (you've guessed it) can be decrypted with the *sha-1* digest of the missing output file 😊

Here are the *sha-1* digests for the provided input files:

08a6308d5673d72698dc37e5e9d971de1c4878ac	output_01.txt
2062fbcdcf39a2a1401759bc63a006908d0a5a45	output_02.txt
392b1b579e67ea1210a02aed47992385e7107b19	output_03.txt

Once you have found the password that successfully decrypts the file, please send an email to code-challenge@aircall.io with the subject **aircall-nova challenge part 2**, with the *sha-1* digest in the body and the source code for your solution as an attachment.