Write a program that performs symbolic derivation $f'(x) = df(x)/dx$ of a given function $f(x)$. The function $f(x)$ is defined by an expression which may contain the following operations: '`+`' (addition), '`-`' (subtraction), '`*`' (multiplication), '`/`' (division), and '`ln`' (natural logarithm). Besides, the operands may be the variable x and numerical constants. The expression may contain arbitrarily nested sub-expressions in parentheses `( )`. The expression is given in a usual, infix form, such as:

```
(2*ln(x+1.7)-x*x)/((-7)+3.2*x*x)+(x+3*x)*x
```

Numerical constants have the form 'd.d', with an optional sign (`+` or `-`), where the number of digits both in integer and decimal parts are arbitrary.

Input

The input is a textual file which has one $f(x)$ definition per line. The input lines do not have blanks. The input expression is guaranteed to be correct (no syntax error can occur).

Output

The output should contain lines with corresponding symbolic derivations $f' = df/dx$, one line for each f. The strings representing $f(x)$ and $f'(x)$ are guaranteed to have no more than 100 characters.

The output expression should be written in infix form. It *should not* be optimized for human reading. This means, it can contain redundancies, such as '`0*x`', '`1*x`', '`0+x`', etc. The derivation should be performed using the following rules:

1. The operators '`*`' and '`/`' are of higher priority than the operators '`+`' and '`-`'. Parentheses may change the priorities as usually.

2. The operators '`+`', '`-`', '`*`', and '`/`' are left-associative, meaning that they group from left to right:

   `a*b*c = (a*b)*c, a/b/c = (a/b)/c, a/b*c = (a/b)*c`, etc.

3. The rules for derivation are:

   $(a + b)' = a' + b'$

   $(a - b)' = a' - b'$

   $(a * b)' = (a' * b + a * b')$

   $(a/b)' = (a' * b - a * b')/b^2$. **Note:** use b^2 and not $(b * b)$ for presentation

   $\ln(a)' = (a')/(a)$

   $x' = 1$

   $const' = 0$

4. While producing the symbolic derivation, use parentheses for output strictly as stated in the previous rule. Do not perform presentation optimizations, such as '`0*a = 0`', '`1*a = a`', etc.

Sample Input

```
x*x/x
-45.78*x+x
-2.45*x*x+ln(x-3)
```

Sample Output

```
((1*x+x*1)*x-x*x*1)/x^2
(0*x-45.78*1)+1
((0*x-2.45*1)*x-2.45*x*1)+(1-0)/(x-3)
```