But, while he was shouting, he saw with surprise
That the moon of the evening was starting to rise
Up over his head in the darkening skies.

From the top of his tower of turtles, King Yertle sees the moon rising. As everyone knows, the moon is made of green cheese and the best cheese in the world is produced in France. France has also produced many great mathematicians including Descartes, Laplace and Fermat. Fermat's famous last theorem states that the equation

```
x^n+y^n=z^n
```

has no non-zero integer solutions for `x`, `y` and `z` when `n` ¿ 2. For readability, this equation might be better rendered

```
 n    n    n
x  + y  = z
```

While it would be easy to render this equation and other mathematical expressions graphically using TEX or LATEX, sometimes we want to include the expressions in text. Providing the TEX or LATEX source for rendering the math is one possibility, but it isn't always easy to read, especially for low-TEX people.

Handling all possible expressions would render this question intractable, but unary minus (`-`) and the binary operators for addition (`+`), subtraction (`-`), multiplication (`*`), division (`/`), exponentiation (`^`) and equality (`=`) are sufficient for most math work. Remember that `^` associates right to left while all of the other binary operators associate left to right.

The order of precedence for the operations is

1. unary `-` *(highest)*
2. `^`
3. `*` and `/`
4. `+` and `-`
5. `=`

The grouping operators `()` (explicit) and {} (implicit) serve to usurp the usual order of operations.

Input

Input will consist of expressions using the above operators, real numbers (12, 1.2, .35, etc.) and single-letter variables. All expressions will be syntactically valid; each expression will appear on a line by itself. The format for a real number is a string of one or more decimal digits with an optional decimal point occurring at any position. Support for exponents is not required.

Output

Output will be complicated. The sample output will be useful for understanding the output specification.

Each partial expression fills a rectangular box and has a logical vertical centre (LVC). Operators glue text characters and boxes together in various ways. Simple numbers and variables occupy boxes with the same width as their string representation naturally requires (four characters wide for 44.4, for example) and are one character high. Since they are only one character high, their LVC is on the line containing their text.

The unary minus operator followed by an expression E is one character wider than E. The LVC will be the same as for E and the character '`-`' will appear to the left of E at the LVC.

The binary operators other than `/` and `^` join two expressions E_1 and E_2, with the result separating the two expressions by three characters positions. The character for the operator will appear as the middle character at the LVCs of the two expressions. This specifies their vertical alignment and the new LVC.

The binary operator `/` joins two expressions E_1 and E_2. The result will be a vertical arrangement of E_1, a line of hyphens (`-`) and E_2. The line will be as wide as the widest of the two expressions. The skinnier one should be centered horizontally; if it can't be centered exactly it should be placed one space to the right. The line forms the LVC of the result.

The binary operator `^` joins two expressions, a base and an exponent. The result will be the exponent positioned above and to the right of the base with no extra space added. The result's LVC will be in line with the base's LVC.

The implicit grouping operator {} does not change the rendering or LVC of the expression that it surrounds. The explicit one (`()`) is rendered by adding a column of `(`'s to the left of the grouped expression and another column of `)`'s on the right. The resulting LVC will be halfway down; if this cannot be done exactly then the LVC should be placed slightly further downwards.

There should be a blank line between each expression that is printed.

Sample Input

```
x^n+y^n=z^n
1/{1+1/{1+1/{1+x}}}
123/1/12
123/{-1/12}
a^b^c+4/(1+x/{1-x})^{x-y}
```

Sample Output

```
 n    n    n
x  + y  = z

      1
-------------
        1
1 + ---------
          1
    1 + -----
        1 + x

123
---
 1
---
 12

123
---
 -1
 --
 12

  c
 b            4
a   + ----------------
                 x - y
      (      x  )
      (1 + -----)
      (    1 - x)
```