The two properties of arithmetic operators which determine the order in which they are executed in an expression are their precedence levels and their associativities. Precedence levels determine the order in which operators with different precedence levels are executed. Associativities determine the order in which a repeated operator is executed. Left associative operators are evaluated left to right while right associative operators are evaluated right to left. Different operators with equal precedence levels are evaluated left to right. Examples:

- If the precedence level of * is higher than that of +, the expression 5*3+4 would evaluate to 19.

- If the precedence level of + is higher than that of *, the expression 5*3+4 would evaluate to 35.

- Left-associativity of − would cause the expression 3-2-1 to be interpreted as (3-2)-1 which evaluates to 0.

- Right-associativity of − would cause 3-2-1 to be interpreted as 3-(2-1) which evaluates to 2.

In the Southeastern region of the planet ACM, a strange collection of dialects has resulted in the use of non-standard operators for the integer operations of plus, minus, times, and integer divide. The operations are the usual binary operations, but the symbols used, their precedences and their associativities, vary widely across the region. Your team is to write a program that will implement a simple integer calculator with operations "+", "−", "*", and "/" where "/" denotes integer division. The catch is, your calculator has to deal with all of the local dialects.

Input

The input file will contain 4 lines that describe the local symbol set followed by 1 or more lines each containing an expression using the local symbol set to be evaluated. The first four lines each contain a four character string $c_1c_2c_3c_4$ beginning in column one, where:

c_1 denotes the standard operator that is being described

c_2 denotes the local symbol being used for that operator

c_3 is a digit denoting the local precedence of the operator (higher digit means higher precedence)

c_4 is a single letter denoting the local associativity of the operator ('L' for left associativity, 'R' for right).

The line

−@1R

means that the symbol '@' will be used to denote minus which will be right associative and have precedence 1. The expression '5@3@1' under these circumstances will evaluate to 3.

Output

For each input expression, your program must print one line containing the expression with standard operators followed by a space, an equal sign, and the result.

Note:

The last expression of the Sample Input below, parenthesized to show you the order of execution is (2+((3*12)/(6/(5-3)))).

Sample input

```
+@1L
−+3R
*−2R
//2R
1@1
5@5+4
2@3−12/6/5+3
```

Sample output

```
1 +1 = 2
5+5−4 = 6
2+3*12/6/5−3= 14
```