Weird Numbers

`numbers.c | numbers.C | numbers.java | numbers.p`

Binary numbers form the principal basis of computer science. Most of you have heard of other systems, such as ternary, octal, or hexadecimal. You probably know how to use these systems and how to convert numbers between them. But did you know that the system base (radix) could also be negative? One assistant professor at the Czech Technical University has recently met *negabinary* numbers and other systems with a negative base. Will you help him to convert numbers to and from these systems?

A number N written in the system with a positive base R will always appear as a string of digits between 0 and $R-1$, inclusive. A digit at the position P (positions are counted from right to left and starting with zero) represents a value of R^P. This means the value of the digit is multiplied by R^P and values of all positions are summed together. For example, if we use the octal system (radix $R = 8$), a number written as 17024 has the following value:

$$1.8^4 + 7.8^3 + 0.8^2 + 2.8^1 + 4.8^0 = 1.4096 + 7.512 + 2.8 + 4.1 = 7700$$

With a negative radix $-R$, the principle remains the same: each digit will have a value of $(-R)^P$. For example, a negaoctal (radix $R = -8$) number 17024 counts as:

$$1.(-8)^4 + 7.(-8)^3 + 0.(-8)^2 + 2.(-8)^1 + 4.(-8)^0 = 1.4096 - 7.512 - 2.8 + 4.1 = 500$$

One big advantage of systems with a negative base is that we do not need a minus sign to express negative numbers. A couple of examples for the negabinary system ($R = -2$):

decimal	negabinary	decimal	negabinary	decimal	negabinary
-10	1010	-3	1101	4	100
-9	1011	-2	10	5	101
-8	1000	-1	11	6	11010
-7	1001	0	0	7	11011
-6	1110	1	1	8	11000
-5	1111	2	110	9	11001
-4	1100	3	111	10	11110

You may notice that the negabinary representation of any integer number is unique, if no “leading zeros” are allowed. The only number that can start with the digit “`0`”, is the zero itself.

Input Specification

The input will contain several conversions, each of them specified on one line. A conversion from the decimal system to some negative-base system will start with a lowercase word "`to`" followed by a minus sign (with no space before it), the requested base (radix) R, one space, and a decimal number N.

A conversion to the decimal system will start with a lowercase word "`from`", followed by a minus sign, radix R, one space, and a number written in the system with a base of $-R$.

The input will be terminated by a line containing a lowercase word "`end`". All numbers will satisfy the following conditions: $2 \leq R \leq 10$, $-1\,000\,000 \leq N \leq 1\,000\,000$ (decimal).

Output Specification

For each conversion, print one number on a separate line. If the input used a decimal format, output the same number written in the system with a base $-R$. If the input contained such a number, output its decimal value.

Both input and output numbers must not contain any leading zeros. The minus sign "`-`" may only be present with negative numbers written in the decimal system. Any non-negative number or a number written in a negative-base system must not start with it.

Sample Input

```
to-2 10
from-2 1010
to-10 10
to-10 -10
from-10 10
end
```

Output for Sample Input

```
11110
-10
190
10
-10
```