

## Problem D. Deep Abyss

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Kieray has a hash function  $h(x)$ . She wants you to help her find a fixed point  $x_0$  of  $h$ , such that  $h(x_0) = x_0$ .

### Input

The input contains the description of  $h(x)$ .

The function  $h(x)$  is described by a procedure in Frog language. In this language, variables (for example,  $x$ ,  $y$ ,  $z$ ) are 128-bit unsigned integers. At the beginning, the variable  $x$  contains the input value to  $h$ , and the other variables are all initialized to zeroes. The result of  $h(x)$  is the value of variable  $x$  after the execution of the procedure.

The procedure contains at most 500 lines. Each line is an assignment that stores the result of an expression into a variable. Frog language supports bitwise negation ( $\sim$ ), and ( $\&$ ), or ( $\mid$ ), xor ( $\wedge$ ), and left/right shifting ( $\ll/\gg$ ). The semantics of these operations are similar to those in C/C++.

Each assignment contains at most one binary operation. (Hexa)decimal constant numbers are allowed in expressions. The bitwise negation ( $\sim$ ) can be applied to each variable or constant at most once. Every expression except bitwise xor expression ( $\wedge$ ) contains at most one variable. The shift amount (right operand) of a shifting operation is non-negative and not greater than 128.

Here is the extended BNF specification of the Frog language:

```
Newline      = "\n"
DecimalDigit = "0" ... "9"
HexDigit     = "0" ... "9" | "a" ... "f"
Letter       = "a" ... "z"
Hexadecimal  = "0" "x" HexDigit{1,32}
Variable     = Letter{1,4}
Term         = ( "~" | ) ( Variable | Hexadecimal )
ConstantTerm = ( "~" | ) Hexadecimal
ShiftAmount  = DecimalDigit
              | ( "1" ... "9" ) DecimalDigit
              | "1" ( "0" | "1" ) DecimalDigit
              | "1" "2" ( "0" ... "8" )
Expression   = Term
              | Term "_" "^" "_" Term
              | Term "_" "&" "_" ConstantTerm
              | ConstantTerm "_" "&" "_" Term
              | Term "_" "|" "_" ConstantTerm
              | ConstantTerm "_" "|" "_" Term
              | Term "_" "<<" "_" ShiftAmount
              | Term "_" ">>" "_" ShiftAmount
Assignment   = Variable "_" "=" "_" Expression
Procedure    = ( Assignment Newline )*
```

Note: “\*” means the preceding token appears zero or more times, and “{1,4}” means the preceding token appears one to four times. The symbol “\_” represents a single space character.

### Output

Print the fixed point of  $h(x)$  in hexadecimal without leading zeroes, conforming to the format of Hexadecimal in the extended BNF specification. If there are multiple fixed points, print the minimum one. If there is no fixed point, print “: (” (without quotes) instead.

### Examples

<i>standard input</i>	<i>standard output</i>
x = x   0x19260817	0x19260817
x = x ^ 0xdeadbeef	: (
y = ~x << 3 z = ~x >> 2 w = x & ~0xa k = ~x ^ 0xc k = k << 1 k = k >> 2 x = y ^ z p = k ^ w x = x ^ p	0x9a83dcd41ee6a0f73507b9a83dcd41ef
x = x   0x1 x = x << 128	0x0

### Note

This is an illustration from Kieray. It’s lovely.

