Prof. Fumblemore and the Collatz Conjecture

The **Collatz function**, C(n), on positive integers is:

n/2 if n is even and 3n+1 if n is odd.

The **Collatz sequence**, CS(n), of a positive integer, n, is the sequence

CS(n) = n, C(n), C(C(n)), C(C(C(n))), …

For example, CS(12) = 12, 6, 3, 10, 5, 16, 8, 4, 2, 1. 4. 2. 1, …

The **Collatz Conjecture** (also known as the 3n+1 problem) is that CS(n) for every positive integer n eventually ends repeating the sequence 4, 2, 1. At the time this problem was created, the status of this conjecture was unknown. No proof has been given and no counter-example has been found up to very large values.

Prof. Fumblemore wants to study the problem using Collatz sequence types. The **Collatz sequence type** of an integer n, CST(n) is a sequence of letters E and O (for even and odd) which describe the parity of the values in CS(n) up to but not including the first power of 2. So

CST(12) = EEOEO

Note that

CS(908) = 908,454, 227, 682, 341, 1024, 512, 256, 128, 64, 32, 16, 8, 4, 3, 2,…

so 12 and 908 have the same CST.

Prof. Fumblemore needs a program which allows him to enter a sequence of E’s and O’s and returns the **smallest** integer n for which the given sequence is CST(n).

Note:

E’s are even numbers which are not powers of 2,

O’s are odd numbers greater than 1.

The last letter in a sequence must be an O (if C(n) is a power of 2, so is n)

There cannot be two O’s in succession (C(odd) = even)

Since, Prof. Fumblemore does not type well, you should check that the input sequence is valid before attempting to find n.

# Input

# There is one line of input containing a string of up to 50 letters E or O.

# Output

For each input there is one line of output.

The output line consists of the string INVALID if the input line is invalid, or a single decimal integer, **n**, such that **n** is the smallest integer, for which CST(**n**) is the input sequence. Inputs will be chosen so that **n** will fit in a 64-bit non-negative integer

**Sample Input**:

EEOEO

EEOOEO

**Sample Output**:

12

INVALID