Jammed Carry Bar

"Drat!" cursed Charles. "This stupid carry bar is not working in my Arithmetic Engine! I just tried to calculate the square of a number, but it's wrong; all of the carries are lost."
"Hmm," mused Ada, "arithmetic without carries! I wonder if I can figure out what your original input was, based on the result I see on the Engine."

Carryless addition, denoted by $\oplus$, is the same as normal addition in base 10, except any carries are ignored. Thus, $37 \oplus 48$ is 75, not 85.

Carryless multiplication, denoted by $\otimes$, is defined as follows.

- For two digits a and b ($0 \le a, b < 10$), it is the same as normal multiplication in base 10, except the carry is ignored. Thus, $9 \otimes 2$ is 8 and $6 \otimes 7$ is 2.
- Otherwise, for two positive integers a and b, it performed using the schoolboy algorithm for multiplication, column by column, but multiplication of digits is calculated using carryless multiplication, and all intermediate additions are calculated using carryless addition. More formally, let $a_m a_{m-1} \ldots a_1 a_0$ be the digits of a, where a_0 is its least significant digit. Similarly define $b_n b_{n-1} \ldots b_1 b_0$ be the digits of b. The digits of $c = a \otimes b$ are given by the following equation:

  $$c_k = (a_0 \otimes b_k) \oplus (a_1 \otimes b_{k-1}) \oplus \ldots \oplus (a_{k-1} \otimes b_1) \oplus (a_k \otimes b_0),$$

  where any a_i or b_j is considered zero if $i > m$ or $j > n$.

For example, $9 \otimes 1\,234$ is $9\,876$, $90 \otimes 1\,234$ is $98\,760$, and $99 \otimes 1\,234$ is $97\,536$.

Given a positive integer n, find the smallest positive integer a such that $a \otimes a = n$.

Input

The input consists of a single line with a positive integer n, with at most 25 digits and no leading zeros.

Output

Print, on a single line, the least positive number a such that $a \otimes a = n$. If there is no such a, print '`-1`' instead.

Examples

input	output
6	4
149	17
123476544	11112
15	-1