Find String Roots

In mathematics, the N-th root of a number M, is a number K such that $K^N = M$, i.e. KKK . . . K = M where K is multiplied N times.
We can translate this into strings. In string notation, the yuxtaposition is concatenation instead of multiplication. So, the N-th root of a string S is another string T such that $T^N = S$, where T N = T T T . . . T is the string T concatenated N times. For instance, if S = “abcabcabcabc”, for N = 2 the string T = “abcabc” is the N-th root of S, while for N = 4 its N-th root is T = “abc”. Note that for N = 1 any string S is the N-th root
of S itself.
Given a string S you have to find the maximum N such that the N-th root of S exists. In the above example the answer would be 4, because there is no N-th root of S = “abcabcabcabc” for N > 4.

Input

The input contains several test cases, each one described in a single line. The line contains a non-empty string S of at most 10^5 characters, entirely formed of digits and lowercase letters. The last line of the input contains a single asterisk (“*”) and should not be processed as a test case.

Output

For each test case output a single line with the greatest integer N such that there exists a string T that concatenated N times is equal to S.

Example

Input:

```
abcabcabcabc
abcdefgh012
aaaaaaaaaa
*
```

Output:

```
4
1
10
```