Define a *k-periodic string* as follows:

> A string s is *k-periodic* if $|s|$, the length of the string, is a multiple of k and, if you chop the string up into $|s|/k$ substrings of length k, then each of those substrings (except the first) is the same as the previous substring, but with its last character moved to the front.

For example, the following string is *3-periodic*:

```
abccabbcaabc
```

The above string breaks up into substrings `abc`, `cab`, `bca`, and `abc`, and each substring (except the first) is a rotation of the previous substring (`abc` → `cab` → `bca` → `abc`).

Given a string, determine the smallest k for which the string is k-periodic.

Input

The input file contains several test cases, each of them will be a single line containing a string s, $(1 \leq |s| \leq 100)$, consisting only of lower-case letters.

Output

For each test case, print a single line containing an integer denoting the smallest value of k for which the input string is *k-periodic*.

Sample Input

```
aaaaaaaa
abbaabbaabba
abcdef
abccabbcaabc
```

Sample Output

```
1
2
6
3
```