Periodic Strings

CPU TIME LIMIT	MEMORY LIMIT	DIFFICULTY
1 second	1024 MB	easy (2.3)

Define a k-*periodic string* as follows:

A string s is k-*periodic* if the length of the string $|s|$ 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 can break up into substrings `abc`, `cab`, `bca`, and `abc`, and each substring (except the first) is a right-rotation of the previous substring (`abc -> cab -> bca -> abc`)

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

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The single line of input contains a string s ($1 \leq |s| \leq 100$) consisting only of lowercase letters.

Output

Output the integer k, which is the smallest k for which the input string is k-*periodic*.

Sample Input 1

```
aaaaaaaa
```

Sample Output 1

```
1
```

Sample Input 2

```
abbaabbaabba
```

Sample Output 2

```
2
```

Sample Input 3

```
abcdef
```

Sample Output 3

```
6
```

Sample Input 4

```
abccabbcaabc
```

Sample Output 4

```
3
```