Left Shifting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

We say a string is beautiful, if its first character is the same as its last character.

Given a string $S = s_0 s_1 \cdots s_{n-1}$ of length n, let $f(S, d)$ be the string obtained by shifting S to the left d times. That is $f(S, d) = s_{(d+0) \bmod n} s_{(d+1) \bmod n} \cdots s_{(d+n-1) \bmod n}$. Find the smallest non-negative integer d such that $f(S, d)$ is beautiful.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first and only line contains a string $s_0 s_1 \cdots s_{n-1}$ ($1 \le n \le 5 \times 10^5$) consisting only of lower-cased English letters.

It's guaranteed that the sum of n of all test cases will not exceed 5×10^5.

Output

For each test case, output one line containing one integer, indicating the smallest non-negative integer d such that $f(S, d)$ is beautiful. If it's impossible to find such d, output `-1` instead.

Example

standard input	standard output
4 helloccpc abcdcba x abc	3 0 0 -1

Note

For the first sample test case, $f(S, 3) =$ `loccpchel`. As its first and last characters are both `l`, it is a beautiful string. Although $f(S, 6) =$ `cpchelloc` is also beautiful, we need to answer the smallest non-negative d. So the answer is 3.