Given a string of lowercase letters in the range ascii[a-z], determine a character that can be removed to make the string a [palindrome](). There may be more than one solution, but any will do. For example, if your string is "*bcbc*", you can either remove '*b*' at index 0 or '*c*' at index 3. If the word is already a palindrome or there is no solution, return *-1*. Otherwise, return the index of a character to remove.

Function Description

Complete the *palindromeIndex* function in the editor below. It must return the index of the character to remove or -1.

palindromeIndex has the following parameter(s):

- *s*: a string to analyze

Input Format

The first line contains an integer q, the number of queries.
Each of the next q lines contains a query string s_i.

Constraints

- $1 \leq q \leq 20$
- $1 \leq |s| \leq 10^5 + 5$
- All characters are in the range ascii[a-z].

Output Format

Print an integer denoting the *zero-indexed* position of the character to remove to make s a palindrome. If s is already a palindrome or no such character exists, print -1.

Sample Input

```
3
aaab
baa
aaa
```

Sample Output

```
3
0
-1
```

Explanation

Query 1: "aaab"
Removing '*b*' at index 3 results in a palindrome, so we print 3 on a new line.

Query 2: "baa"
Removing '*b*' at index 0 results in a palindrome, so we print 0 on a new line.

Query 3: "aaa"
This string is already a palindrome, so we print -1. Removing any one of the characters would result in a palindrome, but this test comes first.

Note: The custom checker logic for this challenge is available [here]().