# Palindrome Index

Given a string of lowercase letters, determine the index of the character whose removal will make the string a palindrome. If the string is already a palindrome, then print **$-1$**. There will always be a valid solution.

### Input Format

The first line contains $T$ (the number of test cases).
The $T$ subsequent lines of test cases each contain a single string to be checked.

### Constraints

$1 \le T \le 20$
$1 \le$ *length of string* $\le 100005$
All characters are Latin lower case indexed.

### Output Format

Print the *zero-indexed* position (integer) of a character whose deletion will result in a palindrome; if there is no such character (i.e.: the string is already a palindrome), print **-1**. Any correct answer will be accepted; e.g.: for a string such as `bcbc`, we can either remove *b* at index $0$ or *c* at index $3$—both answers are acceptable.

### Sample Input

```
3
aaab
baa
aaa
```

### Sample Output

```
3
0
-1
```

### Explanation

*Test Case 1($aaab$):* Removing *b* at index $3$ results in a palindrome, so we print **3**.

*Test Case 2($baa$):* Removing *b* at index $0$ results in a palindrome, so we print **0**.

*Test Case 3($aaa$):* This string is already a palindrome, so we print **-1**; however, **0**, **1**, and **2** are also all acceptable answers, as the string will still be a palindrome if any one of the characters at those indices are removed.

### Custom Checker logic

https://gist.github.com/shashank21j/58df3865a95bf960092c