Your Palindrome Index submission got 25.00 points.
Try the Next Challenge | Try a Random Challenge

Palindrome Index


by amititkgp

Problem | Submissions | Leaderboard | Discussions | Editorial | Topics

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
$1 \leq T \leq 20$
$1 \leq$ *length of string* ≤ 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

Related Topics

Palindrome

Solved score: 25.00pts

Submissions: 19574

Max Score: 25

Difficulty: Easy

More

Current Buffer (saved locally, editable)

Python 2

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
def is_palindrome(s):
    for i in xrange(len(s) // 2):
        if (s[i] != s[len(s)-i-1]):
            return False
    return True

def palindrome_index(s):
    for idx in xrange((len(s)+1)//2):
        if s[idx] != s[len(s)-idx-1]:
            if is_palindrome(s[:idx]+s[idx+1:]):
                return idx
            else:
                return len(s)-idx-1
    return -1

test = int(raw_input().strip())

for _ in xrange(test):
    print palindrome_index(raw_input().strip())
```

Line: 20 Col: 48

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14

Next Challenge