

Question 1

Correct

Marked out of 10.00

A string is said to be palindrome, if it reads the same from both the ends. Given a string S , you are allowed to perform cyclic shifts. More formally, you can pick any one character from any end (head or tail) and you can append that character at the other end. For example, if the string is abc , then if we do a shift using the character at head position then the string becomes bca . Similarly, if we do the shift using the character at the tail then the input string becomes cab . Your task is to find out the minimum number of shifts needed to make the given string, a palindrome.

In case, we can't convert the string to palindrome then print -1.

Input Format

First line starts with T i.e. number of test cases, and then T lines will follow each containing a string S .

Output Format

Print the minimum number of cyclic shifts for each string if it can be made a palindrome, else -1.

Constraints

$1 \leq T \leq 100$

$1 \leq |S| \leq 300$, S will contains only lower case alphabets $a-z$.

Sample Input and Output

Input

4

abbb

aaabb

aabb

abc

Output

-1

1

1

-1

Explanation:

For Test Case 2 (aaabb):

Shift the character at the tail to the head and the result will be baaab, which is a palindrome. This is an operation which requires minimum number of shifts to make the given string a palindrome.

For Test Case 3 (aabb):

One way to convert the given string to palindrome is, shift the character at the head to the tail, and the result will be abba, which is a palindrome. Another way is to shift the character at the tail to the head, and the result will be baab, which is also a palindrome. Both require only one shift.

For example:

Input	Result
4	-1
abbb	1
aaabb	1
aabb	-1
abc	

Answer: (penalty regime: 0 %)

```

1 for i in range(int(input())):
2     c=input()
3     rot=c
4     p=0
5     for j in range(len(c)):
6         rot=rot[1:]+rot[:1]
```

```
7 ▼         if rot==rot[::-1]:
8             p=1
9             break
10 ▼     if p==1:
11         print("1")
12 ▼     else:
13         print("-1")
14
15
```

	Input	Expected	Got	
✓	4	-1	-1	✓
	abbb	1	1	
	aaabb	1	1	
	aabb	-1	-1	
	abc			

Passed all tests! ✓