

Problem D

Palindromic Distance

Time limit: 1 second

Problem Description

The edit distance between two strings u and v is the minimum number of edit operations that turns u into v . There are three edit operations that can be applied to a string: Insert a character, delete a character, and substitute some character by a different one.

For example, we can turn `hello` into `world` with four substitutions, so the edit distance is at most 4. You can turn `wally` into `walter` with two substitutions and one insertion, so the edit distance is at most 3. Computing the edit distance between two strings is a well-known problem with many applications.

The task at hand is to compute the edit distance of a string to **one of the closest** palindromes. A palindrom is a string that is the same when read backwards, for example `madam`.

The edit distance of `hello` to the closest palindrome is only 2: We can turn `hello` into `ollo`, or `hllh`, or `elle` with two edit operations.

Write a program that can find the distance of a word to a closest palindrome.

Input Format

Each test contains multiple test cases. The first line contains the number of test cases t . The description of the test cases follows.

The only line of each test case contains a word w consisting only of lowercase letters.

Output Format

For each test case, output the edit distance of the input word w to its closest palindrome.

Technical Specification

- $1 \leq t \leq 200$
- The word w has length at least one.
- It is guaranteed that the sum of the lengths of the words w over all test cases does not exceed 3000.

Sample Input 1

```
6  
aaaaba  
hello  
palindrome  
abba  
x  
bababac
```

Sample Output 1

```
1  
2  
5  
0  
0  
1
```