

The Love-Letter Mystery

James found a love letter his friend Harry has written for his girlfriend. James is a prankster, so he decides to meddle with the letter. He changes all the words in the letter into [palindromes](#).

To do this, he follows two rules:

1. He can reduce the value of a letter, e.g. he can change *d* to *c*, but he cannot change *c* to *d*.
2. In order to form a palindrome, if he has to repeatedly reduce the value of a letter, he can do it until the letter becomes *a*. Once a letter has been changed to *a*, it can no longer be changed.

Each reduction in the value of any letter is counted as a single operation. Find the minimum number of operations required to convert a given string into a palindrome.

Input Format

The first line contains an integer T , i.e., the number of test cases.

The next T lines will contain a string each. The strings do not contain any spaces.

Constraints

$$1 \leq T \leq 10$$

$$1 \leq \text{length of string} \leq 10^4$$

All characters are lower case English letters.

Output Format

A single line containing the number of minimum operations corresponding to each test case.

Sample Input

```
4
abc
abcba
abcd
cba
```

Sample Output

```
2
0
4
2
```

Explanation

1. For the first test case, **abc** -> **abb** -> **aba**.
2. For the second test case, **abcba** is already a palindromic string.
3. For the third test case, **abcd** -> **abcc** -> **abcb** -> **abca** = **abca** -> **abba**.
4. For the fourth test case, **cba** -> **bba** -> **aba**.