

James found a love letter that his friend Harry has written to 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 only reduce the value of a letter by **1**, i.e. he can change *d* to *c*, but he cannot change *c* to *d* or *d* to *b*.
2. The letter *a* may not be reduced any further.

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.

For example, given the string *s* = **cde**, the following two operations are performed: *cde* → *cdd* → *cdc*.

Function Description

Complete the *theLoveLetterMystery* function in the editor below. It should return the integer representing the minimum number of operations needed to make the string a palindrome.

theLoveLetterMystery has the following parameter(s):

- *s*: a string

Input Format

The first line contains an integer *q*, the number of queries.

The next *q* lines will each contain a string *s*.

Constraints

$$1 \leq q \leq 10$$

$$1 \leq |s| \leq 10^4$$

All strings are composed of lower case English letters, *ascii[a-z], with no spaces.

Output Format

A single line containing the minimum number of 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* → *abba*.
4. For the fourth test case, *cba* → *bba* → *aba*.