Problem: 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](https://en.wikipedia.org/wiki/Palindrome).

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 , i.e., the number of test cases.   
The next  lines will contain a string each. The strings do not contain any spaces.

**Constraints**   
   
 *length of string*    
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, ab**c** -> ab**b** -> aba.
2. For the second test case, *abcba* is already a palindromic string.
3. For the third test case, *abc****d****-> abc****c****-> abc****b****-> abc****a****= ab****c****a -> ab****b****a*.
4. For the fourth test case, ***c****ba ->****b****ba -> aba*.

Solution

int palindrome\_operation(string str)

{

int counter=0;

int length = str.length();

for(int i=0; i<length/2; i++)

{

counter += abs( str[i] - str[length-i-1] );

}

return counter;

}

int main()

{

int cases;

string str;

cin>>cases;

for(long i=0; i<cases; i++)

{

cin >> str;

cout<<palindrome\_operation(str)<<endl;

}

return 0;

}

`’Anshul AgGarwal