Difficulty: Easy Points: 20

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.

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

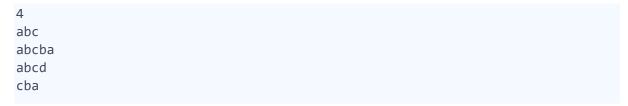
length of string

All characters are lower case English letters.

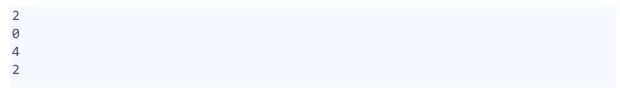
Output Format

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

Sample Input



Sample Output



Explanation

Difficulty: Easy Points: 20

- 1. For the first test case, $abc \rightarrow abb \rightarrow aba$.
- 2. For the second test case, *abcba* is already a palindromic string.
- 3. For the third test case, $abcd \rightarrow abcc \rightarrow abcb \rightarrow abca = abca \rightarrow abba$.
- 4. For the fourth test case, *cba* -> *bba* -> *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++)</pre>
       {
         cin >> str;
         cout<<palindrome_operation(str)<<endl;</pre>
    return 0;
  }
```