

A lock you use has a code system to be opened instead of a key. The lock contains a sequence of wheels. Each wheel has the 26 letters of the English alphabet ‘a’ through ‘z’, in order. If you move a wheel up, the letter it shows changes to the next letter in the English alphabet (if it was showing the last letter ‘z’, then it changes to ‘a’). If you move the wheel down, it changes to show the previous letter in the English alphabet (if it was showing ‘a’, then it changes to ‘z’).

It is also possible to move any subsequence of contiguous wheels in the same direction with only one movement. This has the same effect of moving each of the wheels within the subsequence on that direction, but saves the effort of doing that one wheel at a time.

The lock opens when the wheels show a secret sequence of letters. Currently all wheels are showing the letter ‘a’. You want to know the minimum number of movements you need to open the lock.

## Input

The input has several test cases. Each of them is given in exactly one line containing a nonempty string of at most 1000 lowercase letters. The string represents the secret sequence of letters that opens the lock.

The last test case is followed by a line containing a single asterisk.

## Output

For each test case, output a line containing a single integer with the minimum number of movements to open the lock.

## Sample Input

```
abcxyz
abcdefghijklmnoprstuvwxyz
aaaaaaaaa
zzzzzzzzzz
zzzzbzzzz
*
```

## Sample Output

```
5
25
0
1
3
```