# E. Cipher

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sherlock Holmes found a mysterious correspondence of two VIPs and made up his mind to read it. But there is a problem! The correspondence turned out to be encrypted. The detective tried really hard to decipher the correspondence, but he couldn't understand anything.

At last, after some thought, he thought of something. Let's say there is a word $s$, consisting of $|s|$ lowercase Latin letters. Then for one operation you can choose a certain position $p$ $(1 \le p < |s|)$ and perform one of the following actions:

- either replace letter $s_p$ with the one that alphabetically **follows** it and replace letter $s_{p+1}$ with the one that alphabetically **precedes** it;
- or replace letter $s_p$ with the one that alphabetically **precedes** it and replace letter $s_{p+1}$ with the one that alphabetically **follows** it.

Let us note that letter "z" doesn't have a defined following letter and letter "a" doesn't have a defined preceding letter. That's why the corresponding changes are not acceptable. If the operation requires performing at least one unacceptable change, then such operation cannot be performed.

Two words coincide in their meaning iff one of them can be transformed into the other one as a result of zero or more operations.

Sherlock Holmes needs to learn to quickly determine the following for each word: how many words can exist that coincide in their meaning with the given word, but differs from the given word in at least one character? Count this number for him modulo $1000000007$ $(10^9 + 7)$.

## Input

The input data contains several tests. The first line contains the only integer $t$ $(1 \le t \le 10^4)$ — the number of tests.

Next $t$ lines contain the words, one per line. Each word consists of lowercase Latin letters and has length from $1$ to $100$, inclusive. Lengths of words can differ.

## Output

For each word you should print the number of different **other** words that coincide with it in their meaning — not from the words listed in the input data, but from all possible words. As the sought number can be very large, print its value modulo $1000000007$ $(10^9 + 7)$.

### Examples

**input**

```
1
ab
```

**output**

```
1
```

**input**

```
1
aaaaaaaaaaa
```

**output**

```
0
```

| input |
| --- |
| 2<br>ya<br>klmbfxzb |

| output |
| --- |
| 24<br>320092793 |

## Note

Some explanations about the operation:

- Note that for each letter, we can clearly define the letter that follows it. Letter "b" alphabetically follows letter "a", letter "c" follows letter "b", ..., "z" follows letter "y".
- Preceding letters are defined in the similar manner: letter "y" precedes letter "z", ..., "a" precedes letter "b".
- Note that the operation never changes a word's length.

In the first sample you can obtain the only other word "ba". In the second sample you cannot obtain any other word, so the correct answer is $0$.

Consider the third sample. One operation can transform word "klmbfxzb" into word "klmcexzb": we should choose $p = 4$, and replace the fourth letter with the following one ("b" $\rightarrow$ "c"), and the fifth one — with the preceding one ("f" $\rightarrow$ "e"). Also, we can obtain many other words from this one. An operation can transform word "ya" only into one other word "xb".

Word "ya" coincides in its meaning with words "xb", "wc", "vd", ..., "ay" (overall there are $24$ other words). The word "klmbfxzb has many more variants — there are $3320092814$ other words that coincide with in the meaning. So the answer for the first word equals $24$ and for the second one equals $320092793$ — the number $3320092814$ modulo $10^9 + 7$