String Compression

Time Limit: 1 s
Memory Limit: 256 MB

Does anyone else feel like their being watched?

Central Intelligence Agency (CIA)
*They're

Just now · Like · Reply

Description

The CIA is planning to invent a new mechanism to compress data. The data consist of several string that contains only non-capital alphabet characters. You as one of the CIA programmer are asked to write the implementation by composing several functions and using only recursion (no looping). The mechanism will convert each word to a new compressed string by the following steps.

First, if a character in the string occurs only once, leave it as it is. But if it's more than once, it will be translated to the form `[xc]`, where `x` is the number of occurrence of that character, and `c` is the character itself. For example if you have string `aaaaaa`, the compressed string would be `[6a]`. Finally, print all the characters in alphabetical order. By this mechanism, it is possible for different strings to have the same value after the compression, only if they have the same length and contains the same amount of characters (see *Example* and *Explanation* for more details)

Input

- The first line contains one integer n, the number of string to be compressed
- The next n lines will contain the string str to be compressed. It is guaranteed that the string will contain non-capital alphabet characters only and is between 1-64 characters long

Output

For each line, output the compressed string as in the *Description*

Constraints

$1 \le \boldsymbol{n} \le 10$

$1 \le strlen(\boldsymbol{str}) \le 64$

$'a' \le \boldsymbol{str}[i] \le 'z'$

Example

#1

Input

```
2
aabbcc
bcbaac
```

Output

```
[2a][2b][2c]
[2a][2b][2c]
```

#2

Input

```
2
teknik
informatika
```

Output

```
ei[2k]nt
[2a]f[2i]kmnort
```

Explanation

In *Example #1*, there are two strings `aabbcc` and `bcbaac`. Each of them has the same amount of characters. 'a' = 2, 'b' = 2, and 'c' = 2, so the compressed string will be the same, `[2a][2b][2c]`.

In *Example #2*, there are two strings, `teknik` and `informatika`. The string `teknik` has `k` repeated twice. So it will be converted to `[2k]`. The final string would be `ei[2k]nt` when printed in alphabetical order. Next, `informatika` have `a` and `i` repeated twice, so `[2a]` and `[2i]`, and the final string would be `[2a]f[2i]kmnort`