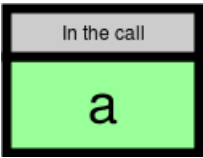


Bob is making a video conference software. Whenever a new person joins the conference, Bob displays the person's name in the interface.

However, displaying full name is tedious and takes much space. So he decided to display the shortest prefix which doesn't match with any prefix of any person who has joined earlier.

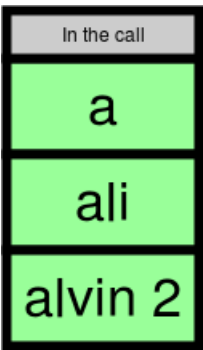
Let's suppose the first person to enter the conference is `alvin`.



Now suppose next person to join is `alice`. The shortest prefix of `alice` that doesn't match with any prefix of `alvin` is `ali`.



If the full name of a new person matches completely with the full name of any person who has joined earlier, he will display the full name and add a suffix which indicates how many times the same name has occurred in the list so far. For example, if another person name `alvin` joins, the list will look like this:



You are given the list of the persons who have joined the call in the chronological order. Your task is to figure out how the final list looks like.

Input Format

The first line contains an integer n .
The subsequent n line contains a string s_i denoting the name of the i^{th} person to join the call.

Constraints

- $1 \leq n \leq 10^5$

- $1 \leq s_i \leq 10$
- s_i will contain only lower-case english letters.

Subtask

- $1 \leq n \leq 1000$ for 60% of the maximum score

Output Format

Return a string array with n items, the i^{th} line should contain the prefix of name of the i^{th} person which doesn't match with any other person who has joined earlier.

Sample Input 0

```
3
alvin
alice
alvin
```

Sample Output 0

```
a
ali
alvin 2
```

Sample Input 1

```
6
mary
stacy
sam
samuel
sam
miguel
```

Sample Output 1

```
m
s
sa
samu
sam 2
mi
```