

Authors of the new, all-in-one encyclopedia have organized the titles in the order they consider most appropriate for their readers. It's not always alphabetical, because they want to observe some peculiar relationships between them. However, they still want to allow users to look up titles quickly.

They achieve this by adding a carefully calculated number of spaces before every title in the list of titles. They call this structure **a dictionary**.

A dictionary is represented by a list of words with some number of spaces before certain words. Dictionary format can be described as a set of constraints on sequences of consecutive words starting with the same letter. Any maximal sequence of consecutive words starting with the same letter should satisfy the following rules:

- The first word in the group has no spaces before it. Every subsequent word in the group has at least one leading space.
- If
 - the first word of the group is deleted and
 - one space is deleted before every remaining word and
 - the first letter is deleted from every remaining word

then resulting sequence is a dictionary.

The authors don't feel like giving you a more detailed explanation of what a dictionary is, so they have included an example (see sample input and output) that clarifies their definition.

Your task is to write a program that will convert a given list of words into a dictionary by adding some number of spaces before certain words and preserving the original order of the words.

Input

The first line of the input contains an integer indicating the number of test cases in the input. Then there is a blank line and the test cases separated by a blank line.

Each test case consists of at least one and most 100000 words. Each word consists of at least one and at most 10 lower-case letters. There will be no leading or trailing spaces. There will be no blank lines between the words.

Output

For each test case write to the output file the original words in the same order without any trailing spaces but with the appropriate number of leading spaces, so that this word list is a dictionary. There should be no blank lines between the words.

Print a blank line between test cases.

NOTE: In the Sample Output, for reading convenience, spaces are replaced with '.' characters. Your output file should contain spaces instead.

Sample Input

```
1

a
ant
antique
amaze
bargain
bridge
bride
bribe
born
bucket
tart
tan
tram
trolley
t
try
trial
zed
double
dorm
do
dormant
donate
again
agony
boost
back
born
```

Sample Output

```
a
 .ant
..antique
.amaze
bargain
.bridge
..bride
...bribe
.born
.bucket
tart
.tan
.tram
..trolley
.t
.try
..trial
zed
double
.dorm
..do
..dormant
..donate
again
.agony
boost
.back
.born
```