



## PROBLEM 2 – ABBREVIATIONS

In this part of the world it is quite common to abbreviate words – “Uni” for “University”, and “Bas” for “Basil” are obvious examples.

In this problem, you will be given sets of words. In each set you must decide on the shortest abbreviation that will uniquely match each word within the set. In this context, an abbreviation is one or more adjacent letters from the word, starting with the first letter. So valid abbreviations for “Basil” would be “B”, “Ba”, “Bas”, “Basi” and “Basil”. The last is not strictly an abbreviation, but it is included to allow us to distinguish words such as “wrong” and “wrongly”.

### INPUT FORMAT

Input for this problem consists of a sequence of one or more scenarios. Each scenario describes a set of words.

- Each set of words will start with a single integer,  $N$ ,  $1 \leq N \leq 100$ , on a line of its own, which is the number of words in the set.
- This will be followed by  $N$  lines, each consisting of one word. A word consists of a non-empty sequence of up to 100 (inclusive) lowercase letters. Words within a set will be unique, but not necessarily sorted.

The input will be terminated by a line consisting of a zero (0). This line should not be processed.

### SAMPLE INPUT:

```
4
cat
dog
mouse
horse
3
ant
antelope
anteater
0
```



## OUTPUT FORMAT

For each input scenario, you must output the scenario number, starting with 1, on a line on its own, followed by one line for each word in the set, in the same order as they appear in the input. The word must be displayed, followed by a space, followed by the shortest abbreviation (as defined above, in lower case) that will uniquely match the word within the set.

## SAMPLE OUTPUT:

```
1
cat c
dog d
mouse m
horse h
2
ant ant
antelope antel
anteater antea
```