

4051 I18n

Internationalization and localization are long words that are usually abbreviated as i18n and l10n. The numbers in between i-n and l-n refer to the number of letters that were omitted. It is a very powerful abbreviation method that can be naturally used for any words.

A word may be abbreviated only when it has previously occurred in a given text. A word is abbreviated by omitting all the letters in the word except for the first and last letter and replacing omitted letters with a number of omitted letters.

Your task is to write a program that expands such abbreviations in a given text whenever possible. Expansion is possible if it is *valid* and *unambiguous*.

Expansion is valid if expanded word has previously occurred in the text and its abbreviated form corresponds to the abbreviation that is being expanded. Case is ignored for the purposes of validness. For example, expansion from i18n to internationalization is valid in this problem statement (even as internationalization was previously mentioned only with capital letter I). But expansion of p14n to parameterization is not valid since the word parameterization has never occurred before its abbreviation, and expansion of a11n to abbreviation is not valid, since it is not a correct abbreviation for the word abbreviation (correct one is a10n).

Expansion is unambiguous if there is exactly one valid expansion for it. For example, expansion from l10n to localization is unambiguous in this problem statement, but expansion from p5m cannot be made unambiguously, since both problem and program are abbreviated to p5m.

Input

Input consists on several datasets separated by a blank line. Each dataset contains at most 1000 lines with at most 80 characters in each line. Each line contains one or more words separated by spaces and special symbols: '-', ',', '.', '"', '(', ')', ':', ';', '!', '?'. There are no trailing spaces, but other separators are allowed at the end of line.

Words may be either *full* or *abbreviated*. Full word consists of one to 32 Latin letters and may be written in one of three *capitalization* forms: all lowercase, First Capital Letter, or ALL CAPITAL LETTERS. Abbreviated word consists of a Latin letter, followed by a number from 2 to 30 (no leading zero), followed by a Latin letter. Abbreviated words also have three corresponding capitalization forms: all l7e, F3t C5l L4r, or ALL C5L L5S.

Output

For each dataset, write to the output file original text with original separators while expanding abbreviated words into full words whenever possible (see problem statement). Capitalization of the expanded full word shall correspond to the capitalization of the abbreviation that is being expanded.

Invalid or ambiguous abbreviations shall be left in the text as is (abbreviated). Note, that lines in the output file may be longer than 80 characters.

Write a blank line between datasets.

Sample Input

```
The first line of sample input.  
The s4d l2e of s4e i3t.  
Lone, lone, l4e...  
S4e input last l4e!
```

Sample Output

The first line of sample input.

The s4d line of sample input.

Lone, lone, l4e...

Sample input last l4e!