

## 892 Finding words

A common problem when processing incoming text is to isolate the words in the text. This is made more difficult by the punctuation; words have commas, “quote marks”, (even brackets) next to them, or hy-phens in the middle of the word. This punctuation doesn't count as letters when the words have to be looked up in a dictionary by the program.

For this problem, you must separate out “clean” words from text, that is, words with no attached or embedded non-letters. A “word” is any continuous string of non-whitespace characters, with whitespace characters on each side of it. For this problem, a “whitespace” character is a space character or an end-of-line character, or the start or end of the file (so that, for example, if the input file consists of ‘Anne Bob’, where there is a space character between the A and B but no other, then there are two words, ‘Anne’ and ‘Bob’).

### Input

Input will consist of lines with no more than 60 characters in each line. Every line will be terminated by a character which isn't whitespace (which will be followed immediately by an end-of-line character). The input will be terminated by a line consisting of a single ‘#’.

### Output

Output must be the lines of the incoming text, with the non-letters stripped away from each word. A non-letter is any character which is not a letter (a - z and A - Z) and not a whitespace character. Your program must not change the letters and space characters (although space characters at the ends of lines will be ignored by the judges). When a non-letter occurs in the middle of a word (ie there is no whitespace character next to it), it must be simply removed (see what happens to the word ‘doesn't’ in the example). A word which consists entirely of non-letters will therefore be removed entirely.

There is a special rule for a hyphen (‘-’) when it is the very last character in a line:

- the word part before the hyphen and the first word part on the next line form a single word;
- this complete word must be written on a line by itself;
- you can assume that there will always be a space before the word part on the first line, and a space after the word part on the second line.

### Sample Input

```
A common problem when processing incoming text is to isolate
the words in the text. This is made more difficult by the
punctuation; words have commas, "quote marks",
(even brackets)      next to them, or hy-
phens in the middle of the word. This punctuation doesn't
count as letters when the words have to be looked up in a
# dictionary by the 12345 "***&! program.
#
```

## Sample Output

A common problem when processing incoming text is to isolate the words in the text. This is made more difficult by the punctuation words have commas quote marks even brackets next to them or hyphens in the middle of the word. This punctuation doesn't count as letters when the words have to be looked up in a dictionary by the program.