TEX is a typesetting language developed by Donald Knuth. It takes source text together with a few typesetting instructions and produces, one hopes, a beautiful document. Beautiful documents use “ and ” to delimit quotations, rather than the mundane " which is what is provided by most keyboards. Keyboards typically do not have an oriented double-quote, but they do have a left-single-quote ` and a right-single-quote '. Check your keyboard now to locate the left-single-quote key ` (sometimes called the “backquote key”) and the right-single-quote key ' (sometimes called the “apostrophe” or just “quote”). Be careful not to confuse the left-single-quote ` with the “backslash” key \. TEX lets the user type two left-single-quotes `` to create a left-double-quote “ and two right-single-quotes '' to create a right-double-quote ”. Most typists, however, are accustomed to delimiting their quotations with the un-oriented double-quote ".

familiar

typesetting

spacing

delimit

establish

mundane

boring

If the source contained

```
"To be or not to be," quoth the bard, "that is the question."
```

then the typeset document produced by TEX would not contain the desired form:

“To be or not to be,” quoth the bard, “that is the question.”

In order to produce the desired form, the source file must contain the sequence:

```
``To be or not to be,'' quoth the bard, ``that is the question.''
```

You are to write a program which converts text containing double-quote (") characters into text that is identical except that double-quotes have been replaced by the two-character sequences required by TEX for delimiting quotations with oriented double-quotes. The double-quote (") characters should be replaced appropriately by either `` if the " opens a quotation and by '' if the " closes a quotation. Notice that the question of nested quotations does not arise: The first " must be replaced by ``, the next by '', the next by ``, the next by '', the next by ``, the next by '', and so on.

Input

Input will consist of several lines of text containing an even number of double-quote (") characters. Input is ended with an end-of-file character.

Output

The text must be output exactly as it was input except that:

- the first " in each pair is replaced by two ` characters: `` and
- the second " in each pair is replaced by two ' characters: ''.

Sample Input

```
"To be or not to be," quoth the Bard, "that
is the question".
The programming contestant replied: "I must disagree.
To `C' or not to `C', that is The Question!"
```

Sample Output

```
``To be or not to be,'' quoth the Bard, ``that
is the question''.
The programming contestant replied: ``I must disagree.
To `C' or not to `C', that is The Question!''
```