3711   Java vs C++

Apologists of Java and C++ can argue for hours proving each other that their programming language is the best one. Java people will tell that their programs are clearer and less prone to errors, while C++ people will laugh at their inability to instantiate an array of generics or tell them that their programs are slow and have long source code.

Another issue that Java and C++ people could never agree on is identifier naming. In Java a multiword identifier is constructed in the following manner: the first word is written starting from the small letter, and the following ones are written starting from the capital letter, no separators are used. All other letters are small. Examples of a Java identifier are `javaIdentifier`, `longAndMnemonicIdentifier`, `name`, `nEERC`.

Unlike them, C++ people use only small letters in their identifiers.

To separate words they use underscore character '`_`'. Examples of C++ identifiers are `c_identifier`, `long_and_mnemonic_identifier`, `name` (you see that when there is just one word Java and C++ people agree), `n_e_e_r_c`.

You are writing a translator that is intended to translate C++ programs to Java and vice versa. Of course, identifiers in the translated program must be formatted due to its language rules — otherwise people will never like your translator.

The first thing you would like to write is an identifier translation routine. Given an identifier, it would detect whether it is Java identifier or C++ identifier and translate it to another dialect. If it is neither, then your routine should report an error. Translation must preserve the order of words and must only change the case of letters and/or add/remove underscores.

Input

The input file contains several test cases, each of them consists of one line that contains an identifier. It consists of letters of the English alphabet and underscores. Its length does not exceed 100.

Output

For each test case, and on a line by itself:

If the input identifier is Java identifier, output its C++ version.

If it is C++ identifier, output its Java version.

If it is none, output '`Error!`' instead.

Sample Input

```
long_and_mnemonic_identifier
anotherExample
i
bad_Style
```

Sample Output

```
longAndMnemonicIdentifier
another_example
i
Error!
```