Code Jam 2012 - Qualification Round

Speaking in Tongues

Problem

We have come up with the best possible language here at Google, called Googlerese. To translate text into Googlerese, we take any message and replace each English letter with another English letter. This mapping is *one-to-one* and *onto*, which means that the same input letter always gets replaced with the same output letter, and different input letters always get replaced with different output letters. A letter may be replaced by itself. Spaces are left as-is.

For example (and here is a hint!), our awesome translation algorithm includes the following three mappings: 'a' -> 'y', 'o' -> 'e', and 'z' -> 'q'. This means that "a zoo" will become "y qee".

Googlerese is based on the best possible replacement mapping, and we will never change it. It will always be the same. In every test case. We will not tell you the rest of our mapping because that would make the problem too easy, but there are a few examples below that may help.

Given some text in Googlerese, can you translate it to back to normal text?

Solving this problem

Usually, Google Code Jam problems have 1 Small input and 1 Large input. This problem has only **1 Small input**. Once you have solved the Small input, you have finished solving this problem.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow, one per line.

Each line consists of a string **G** in Googlerese, made up of one or more words containing the letters 'a' - 'z'. There will be exactly one space (' ') character between consecutive words and no spaces at the beginning or at the end of any line.

Output

For each test case, output one line containing "Case #**X**: **S**" where **X** is the case number and **S** is the string that becomes **G** in Googlerese.

Limits

Time limit: 20 seconds.
Memory limit: 1GB.
There is only one test set which has visible verdict.
$1 \leq \mathbf{T} \leq 30$.
G contains at most 100 characters.
None of the text is guaranteed to be valid English.

Sample

Sample Input	Sample Output

```
3
ejp mysljylc kd kxveddknmc re jsicpdrysi
rbcpc ypc rtcsra dkh wyfrepkym veddknkmkrkcd
de kr kd eoya kw aej tysr re ujdr lkgc jv
```

```
Case #1: our language is impossible to understand
Case #2: there are twenty six factorial possibilities
Case #3: so it is okay if you want to just give up
```