A Basic Spell Checker

A Basic Spell Checker

This challenge will introduce you to the basics of Spell Checking. Right from what you type in a search box, or the red squiggles you see as you enter text via your browser, or the articles you write using an online or offline word processor; spell checking is an important tool on the PC and on the Internet.

Most frequent spelling mistakes

People are prone to making spelling mistakes as they type in a hurry. It has been observed that the most common spelling mistakes occur for the following reasons:

Deletion, Replacement, Transposition, Insertion **Assume only the letters a-z are involved.**

1. One character in the string gets deleted incorrectly. Example: The user enters *Ordnary* instead of *Ordinary* (i.e. leaves out the i)
2. One character in the string is incorrectly replaced by another one. Example: The user enters *Accedent* instead of *Accident*.
3. While typing hurriedly, the user ends up swapping one pair of consecutive characters. Example: The user enters *Noramlly* instead of *Normally*.
4. The user ends up inserting one extra character somewhere in the string. Example: The user enters *Heello* instead of *Hello*. The extra character will only be a letter from [a-z] for the purpose of solving this problem.

So, generally, the correct string is just *one step* of *one edit distance* away from what the user erroneously types in.

Please take note, that in each of the four popular cases above, the mistake occurs only at one particular character (or, pair of characters in case 3).

If a spell checker is able to detect these simple but common mistakes, it will be able to handle sixty to seventy percent of all spelling mistakes which people make while typing text on their computers.

What you need to do

You will be provided with a Corpus of text which you can read in as a file in your program. Assume it is placed in the same folder as your program. Read in this text, and build up a dictionary of words and the frequencies with which those words occur.Words are string of letters, and they might contain hyphens and/or apostrophes. The end of the corpus file is marked by "END-OF-CORPUS"

In case you would like to try out the corpus locally, download this file, and use it from the same directory as your program code: corpus

Then, via the standard input, you will be provided with a set of (possibly) mistyped words.

Your program should recommend the likeliest known word from the dictionary you built up, for each of those mistyped words. If the given word exists in your dictionary, output it as it is.

Guidelines

1. Consider the four popular mistakes described earlier in the description. Think of the candidate words which might have led to the given mis-typed versions.
2. Among the candidate words, restrict your choice to the words which do exist in your dictionary. Among these, pick up the word which occurred most frequently, as the best possible suggestion which you can find. If there are multiple such words which occurred most frequently, then output the

one which occurs first in lexicographical order.

3. If your program cannot come up with any suggestion, output the original (possibly mis-typed) word itself.
4. Output should be in lower case.

For the purpose of building a dictionary and language model, details about the corpus have already been provided.

Input Format

The first line of the input contains N. N lines follow, each line having 1 possibly misspelt word.

Output Format

For each word, output the correct spelling of the word. If the word is not misspelt, print it as is.

Sample Input

```
5
contan
seroius
pureli
dose
note
```

Sample Output

```
contain
serious
purely
dose
note
```

Explanation

5 words are given all of them are misspelt. Each word in the output is the correct spelling of the corresponding misspelt word of the input.

Note

- For this problem, as described in the statement, only stick to generating suggestions which are just one "edit distance" or "mistake" away from the original word, otherwise the answer we expect will be different from yours.
- Corpus can be downloaded here

Scoring

Scoring is proportional to the answers you compute correctly.

Score for each test case = (100 * correctAnswers/TotalNumberOfTests)

Total Score = Average of Scores for all test cases which are run on your submission.

** A Note Regarding the Corpus **

We are aware of the fact that the corpus might have more spelling errors than desired, but we are also constrained by the fact that there are only a few sources which we can use without violating terms and conditions of providers.