30^{th} Annual
Rowan University
Programming Contest

hosted by the
Computer Science Department

Saturday, 23 April 2016

Contest Problem

1 Introduction

Automatic spelling correction makes using a computer easier, by comparing words entered against a list of known words, and fixing common errors. These can include omitting a letter, such as typing 'leter' instead of 'letter'; inserting a letter, such as typing 'occassion' instead of 'occasion'; transposing letters, such as typing 'begni' instead of 'begin'; substituting a letter, such as typing 'cavana' instead of 'cabana'; and incorrect capitalization, such as typing 'london' instead of 'London', or 'Mckinley' instead of 'McKinley'.

Write a program to read in a list of words which are spelled correctly, and then a list of words that are to be checked, and report which of the corrections above (omitted letter, inserted letter, transposed letters, wrong letter, incorrect capitalization) can be used to find a word on the list of correct words.

2 Detailed Descriptions

2.1 Correct Words

This is *always* the primary consideration. A word is to be considered correct if:

1. It is on the list of correct words.
2. It is on the list of correct words, except that its first letter is uppercase and the word on the list starts with lowercase. For example, if 'testing' is on the list of correct words, and you get 'Testing', that is considered correct because it might be at the beginning of a sentence.

   This **does not** work the other way: if the correct word list includes 'Trenton', you are to regard 'trenton' as misspelled unless it is also listed in all-lowercase.
3. It is on the list of correct words, but it is in all-uppercase letters. If 'not' is on the correct word list, and your program gets 'NOT', that is considered correct because the writer might be SHOUTING.

2.2 Special-case Words

If a word is not on the word list, it does not need to be checked for errors if:

1. It contains a non-alphabetic character, as in '4Score&7YearsAgo', 'python3', or 'Terminator2', because it might be a password, or a programming language, or a movie sequel. In such a case you report it as having non-alphabetic characters and do not apply spell-checking corrections.
2. The word is entirely upper-case, as in 'NATO' or 'CPU'. Such words should be reported as possible acronyms, and not examined for spell-checking corrections.

2.3 Omitted Letters

If a word is one letter shorter than a word on the official list, and is the same as that word if one letter is added, then it should be reported as a possible omitted letter.

There are three cases:

1. The letter is omitted from the beginning. If the correct word list includes 'correct', and you read the word 'orrect', then you can identify this by checking every letter starting at the end of each word, and ensuring a match for each as you count to the beginning of the shorter word.
2. The letter is omitted from the end. If the correct word list includes 'banana', and you read the word 'banan', then you can identify this by checking every letter starting at the beginning of each word, and ensure a match for each as you count to the end of the shorter word.

3. The letter is omitted from the middle. You can identify this by doing each of the above steps and stopping when the match fails. If the failure points are separated by one, then that's where the missing letter goes. For example, if the correct word list includes 'example', and you read the word 'examle', counting from the beginning you would get to 'l' and stop, with the last match at 'm'. Counting from the end you would get to 'p' and stop, with the last match at 'l'. In 'examle', the 'm' and 'l' are one letter apart, so you could fix that by putting in the 'p'.

2.4 Inserted Letters

If a word is one letter longer than a word on the official list, and is the same as that word if one letter is removed, then it should be reported as a possible inserted letter.

Testing for this is similar to testing for omitted letters.

2.5 Transposed Letters

If the test word matches a word on the correct word list except for two letters which are adjacent and out of order, it should be reported as a possible transposition error.

The words must be the same length, and must match exactly except in two positions. The positions which are different must be adjacent, and the test word must have the letters that the other word has in those same positions.

You should consider 'latpop' a transposition from 'laptop', but you should **not** consider 'paltop' a transposition, because the swapped letters are not adjacent.

2.6 Wrong Letter Substituted

In the event a test word matches a word on the word list exactly except for one and only one character, that should be considered a 'wrong letter' mistake. The words must be the same length: 'begim' is a wrong letter mistake for 'begin', but 'pable' is not a wrong letter mistake for 'babble'.

Note: a one-letter difference that is only about case ('DeForest' is correct and you're testing 'Deforest', or 'clock' is correct and you're testing 'cLock') does not count as substitution, but as capitals. (See the next extry.)

2.7 Incorrect Capitals

If a word appears on the correct word list with a capital letter, and a test word matches except for case, that is to be reported as an incorrect capitalization error. If 'MacAlpin' is on the list, 'macalpin' and 'Macalpin' should both be listed as incorrect capitalization errors. (This is **not** a substitution error.) If 'NATO' is on the list, 'nato' should be listed as an incorrect capitalization error. Similarly, if 'Glassboro' is on the list, 'GLassboro' should be reported as an incorrect capitalization error. All-uppercase is okay; 'GLASSBORO' would be correct.

2.8 Words On The Correct Word List Are Always Correct

If the correct word list includes both 'Reading' (the city in PA) and 'reading' (what you are doing right now with this problem description), you should not report 'reading' as an incorrect capitalization error. The word 'reading' was on the list in that form, and is to be considered correctly spelled. This takes priority over every other test.

The same applies to all other kinds of corrections: If 'lamp' and 'lump' are both on the correct word list, reporting 'lamp' as a misspelled version of 'lump' would make no sense.

Note: There is no guarantee that either word list is in any particular order.

3 Input

3.1 Input Specification

For text input, your program should accept input in the following format:

1. An integer, $\mathcal{D}$, where $1 \leq \mathcal{D} \leq 100$, which is the number of datasets in this file.
2. $\mathcal{D}$ data sets, each of which is in this format:
   (a) One line with one integer, $\mathcal{C}$, where $1 \leq \mathcal{C} \leq 1000$, the number of words on the correct word list.
   (b) $\mathcal{C}$ lines, each with exactly one word, which are the correct words for this data set.
   (c) One line with one integer, $\mathcal{T}$, where $1 \leq \mathcal{T} \leq 1000$, which is the number of words to be tested.
   (d) $\mathcal{T}$ lines, each with exactly one word, each to be tested against the correct word list.

3.2 Sample Input #1

Data in file	Item #	Meaning in plain English
1	1	*this file has 1 data set*
7	2.a	*Data Set 1 has 7 correct words*
car cave cap cane cape cage care	2.b	*the 7 correct words*
10	2.c	*there are 10 words to be tested*
cop core bore cra crae age Cage cae can RAGE	2.d	*the 10 test words*

(This input is on the website as **sample1.txt**.)

You may choose to have your program read the input from the keyboard, or ask the user for a filename and then read the file. Users of GUI-based programming environments may prefer to use text boxes into which the values can be entered, and buttons to begin their calculation. Any reasonable variation in the spirit of the problem is acceptable.

You need not do error-checking on the input. Each line will have exactly the number of items described with no stray characters or extra spaces. There will be no blank lines.

All sample and test data sets are available at <http://elvis.rowan.edu/rupc/2016>

4 Output

4.1 Output Specification

For each data set configuration, your program must generate output as follows:

1. The text '`Analyzing D data sets`', where D is the number of data sets in the input.
2. For each data set, print the following:
   (a) '`Data Set D`', where D is the number of the data set being reported on.
   (b) '`Correct Words: C`', where C is the number of words on the correct word list.
   (c) '`Test Words: T`', where T is the number of words to be tested.
   (d) '`Test Word N: testword`', where N in the number of the test word, and *testword* is the word itself.
   (e) For each test word, one of:
      i. The text '`word is correct`', if the word is on the correct word list,
      ii. The text '`word has non-alphabetic characters`', if the word has any non-alphabetic characters.
      iii. The text '`word is possible acronym`', if the word is in all-uppercase letters.
      iv. For each kind of error described above, if a word on the list corresponds to that error, print it and the words that may apply. If no words apply, do not print that kind of error. Possibilities are: '`Omissions: A B C ...`', '`Insertions: D E F ...`', '`Transposed: G H I ...`', '`Substituted: J K L ...`', and '`Capitals: M N O ...`'.
      In this sample, A, B, . . . O are the possible corrections for the test word by corresponding to the listed correction. (Some may have more than three possible corrections, some may have none.) Only print those lines that have possible corrections.
      v. '`(no suggestions)`', if nothing was printed by any of the previous options.

4.2 Sample Output 1

```
Analyzing 1 data set(s)
Data Set 1
   Correct Words: 7
   Test Words: 10
   Test Word 1: cop
      Substituted: cap
   Test Word 2: core
      Substituted: care
   Test Word 3: bore
      (no suggestions)
   Test Word 4: cra
      Transposed: car
   Test Word 5: crae
      Transposed: care
   Test Word 6: age
      Omissions: cage
   Test Word 7: Cage
      word is correct
   Test Word 8: cae
      Omissions: cave cane cape cage care
      Substituted: car cap
   Test Word 9: can
      Omissions: cane
      Substituted: car cap
   Test Word 10: RAGE
      word is possible acronym
```

(This output corresponds to Sample Input 1 from page 4.)

Your output does **not** have to duplicate the sample output exactly with respect to spacing or use of upper/lower case. Your output should be neat, but need not match character-for-character.

5 Test Data

Run your program on this input and print the results. **You must submit printed output to earn full points.** Your program will also be run on data known only to the judges.

5.1 Test Input #1

```
2
10
Abe
able
ace
adze
age
aid
ale
ape
axe
azure
10
aze
Able
abe
axure
pea
xae
aex
xea
aga
azrue
10
stale
steal
least
slate
tales
post
spot
stop
tops
opts
12
stael
stela
psot
POST
TALES
p0st
st3al
sla#e
OPST
pots
OPTS
tpos
```

5.2 Test Input #2

```
2
5
test
tests
Luke
MacAlpin
NATO
22
TEST
Test
T&ST
TST
tests
tet
teste
texst
est
tes
atest
text
tset
luke
LUKE
Luke
asdf5f
nato
Nato
Macalpin
macalpin
NATO
5
alerting
altering
integral
relating
triangle
7
aletring
TRIANGLE
altering
integra
relting
riangle
aletring
```