Plagiarism Checker

Time: 100 seconds

Memory: 10KB

Some students have submitted their computing assignment to you. You, being a suspicious teacher need to run a plagiarism checker on these assignments to see how similar they are. For this, you will use a simple string measurement algorithm called the “Levenshtein Distance (LD)”. The LD measures two input strings1 and string2 and then counts the minimum number of insertions, deletions, or substitutions that would be required to convert string1 to string2 (and vice versa).

Some examples are given below:

Distance between string “Hello” and “Helli” is 1 (1 for substituting o with i)

Distance between string “Hello world” and “Hi” is 10 (1 for substutiting e with i, and 9 for deleting llo world)

To compute the Levenshtein distance, you make use of a 2D matrix. After populating the first row
and first column, you need to compute a series of di;j values. These are calculated as:

$$d_{i,j} = \begin{cases} A_{i,j} = A_{i-\mathbf{1},j-\mathbf{1}} & \mathit{if}\ (\mathit{string}1_i == \mathit{string}2_j) \\ \min \begin{cases} (d_{i,j-\mathbf{1}}) + 1 \\ (d_{i-\mathbf{1},j}) + 1 \\ (d_{i-\mathbf{1},j-\mathbf{1}}) + 1 \end{cases} & \mathit{otherwise} \end{cases}$$

The Levenshtein distance is then value contained in the bottom-right cell, i.e., dmax(i)−1;max(j) −1.

An example for measuring the distance between string “Check” and “Cheque” is given below:

		C	h	e	q	u	e
	0	1	2	3	4	5	6
C	1	$d_{1,1}$	$d_{2,1}$	$d_{3,1}$	$d_{4,1}$	$d_{5,1}$	$d_{6,1}$
h	2	$d_{1,2}$	$d_{2,2}$	$d_{3,2}$	$d_{4,2}$	$d_{5,2}$	$d_{6,2}$
e	3	$d_{1,3}$	$d_{2,3}$	$d_{3,3}$	$d_{4,3}$	$d_{5,3}$	$d_{6,3}$
c	4	$d_{1,4}$	$d_{2,4}$	$d_{3,4}$	$d_{4,4}$	$d_{5,4}$	$d_{6,4}$
k	5	$d_{1,5}$	$d_{2,5}$	$d_{3,5}$	$d_{4,5}$	$d_{5,5}$	$d_{6,5}$

		C	h	e	q	u	e
	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
h	2	1	0	1	2	3	4
e	3	2	1	0	1	2	3
c	4	3	2	1	1	2	3
k	5	4	3	2	2	2	3

Here, Levenshtein Distance LD = 3, (2 for substituting q with c, & u with k, and 1 insertion of e).

Task

Your job is to create a plagiarism checker software which takes user assignments as text files, and computes the similarity between these assignments using the LD measure expressed as a percentage. The formula for determining the percentage is:

$$Similarity = 100 \times \frac{Levenshtein\ Distance - \max\left[length(string1), length(string2)\right]}{\max\left[length(string1), length(string2)\right]}$$

Input

The first number represents the total number of assignments submitted (ranging from 1 to 100). Sets of assignment are separated by the ~ sign. Each assignment contains four answers of upto 100 characters.

Each answer covers a single line only.

Output

The output compares each assignment with another. Each comparison is separated by a ~ character. The first line in a comparison consists of three values. The first two values represent the assignment number for whom the comparison is made. The third number is the average similarity percentage for all the questions. The other lines in a comparison display similarity indices for all questions which report more than 50% similarity.

Sample Input

```
3~
Shell is an environment facilitating user-OS interaction.
Bash is not a part of the operating system.
Find is used for finding files inside the directory tree.
Locate is used for searching from a pre-built database.
~
Shell is an intermediate user program between user-OS.
Bash is known as the Bourne Again Shell and is part of OS.
Find command searches the directory tree.
Locate searches through a pre-built database.
~
Shell us a user program meant for interactions with user.
Bash is an abbreviation for Bourne Again Shell.
Find searches a directory tree.
Locate searches from a pre-existing database.
```

Sample Output

```
1 2 38.49%
~
1 3 48.25%
Q4 53.33%
~
```

2 3 68.15%

Q2 75.75%

Q3 71.42%

Q4 87.23%