

## 6420 Similarity

You are developing a search engine to dethrone AltaVista (ok, so maybe you travelled 15 years back in time first). The secret sauce to your search engine is the way in which you allow the users to misspell their search keywords, but still be able to find the results they were looking for.

The distance from  $word\_A$  to  $word\_B$  is the minimum cost of the changes that must be applied to transform  $word\_A$  into  $word\_B$  or to transform  $word\_B$  into  $word\_A$ . You allow the following changes, with their associated cost:

1. A character may be inserted into a word at a cost of 2 units, or
2. A character may be deleted from a word at a cost of 2 units, or
3. A character may be replaced with another character. If the replacement differs only in case (e.g., 'A' replaced with 'a' or vice versa), then the cost is 1 unit. For all other replacements, the cost is 2 units.

For example, consider distance from the name **pieter** to the name **peter**. We can transform **peter** into **pieter** by inserting the character 'i' after the 'p', which amounts to only one change.

If instead we measure the distance between **pieter** and **pierre**, we find that three changes are required: either we replace the last three characters of **pierre** with 'ter', or we delete the first 'r', replace the next 'r' with a 't', and insert another 'r' at the end. From these distances we can deduce that **pieter** is more similar to **peter** than it is to **pierre**.

Let  $d(keyword, candidate)$  denote the minimum cost of the of changes required to transform  $keyword$  into  $candidate$ , or vice versa.

The user has entered the term  $keyword$  in your search engine, and you have identified (by mysterious means, if necessary) a possible candidate word from your dictionary, referred to as  $candidate$ . Your task is to compute  $d(keyword, candidate)$ .

### Input

Your input consists of an arbitrary number of records (no more than 20), with each record containing two words separated by white space (one or more space and/or newline characters). Records are separated by a single newline character. Each word consists of  $n$  characters from the set  $\{[a..z] \cup [A..Z] \cup [0..9]\}$ , with  $1 \leq n \leq 10000$ . The two words in each record correspond to the keyword and candidate, respectively.

The end of input is indicated by a line containing only the value '-1'.

### Output

For each input record, output

$k$

where  $k$  denotes the minimum cost of the changes required to transform  $keyword$  into  $candidate$ .

### Sample Input

```
TATTACTA
TATTAAATA
polyvinyl Polynomial
-1
```

**Sample Output**

4  
11