

---

### Gaming fan's nadir

---

Two strings are **anagrams** of one another if, ignoring capitalisation, punctuation and whitespace they contain the same characters. For instance `Finding Anagrams` and `Gaming fan's nadir` are anagrams. Utilities such as `I rearrangement servant` can help find anagrams which arise most commonly in cryptic crossword puzzles but also in other sorts of wordplay. The intent of this étude is to ask you to reproduce some of the functionality of such utilities.

Specifically, given a dictionary and a string you're going to be asked to find the "best" anagram for the string from words in the dictionary. The definition of "better" (and hence "best") in this context is as follows.

- Anagrams using longer words earlier are better than ones using shorter words. So any anagram that begins with a five-letter word is better than any anagram that begins with a four-letter word.
- If two anagrams start with a word of the same length, then the one where that word is earlier in alphabetical order is better. If they start with the same word, you compare first the lengths of the second word (longer is better), or again by alphabetical order etc.

---

### Task

Write a program that finds the best anagrams for each of a number of strings from a given dictionary. For the purpose of the following description, a "word" is just a string whose characters come from the range `a-z` (i.e., lower case letters). Input from `stdin` will be of the following form:

- a series of lines consisting of words,
- an empty line (or one consisting only of whitespace),
- another series of words.

The first group of words in the input are the words for which we seek anagrams. The second group (after the empty line) are the source dictionary which might consist of up to 100,000 words.

Output should be to `stdout`. For each word from the first group (and in the same order as they were input) a single line consisting of that word, a colon, a space, and then the best anagram for that word from the dictionary. If there are no anagrams for the word from the dictionary then the word, colon and space should still be printed.

For instance if the input is:

```
apple  
appleapple  
frog
```

```
app  
el  
leap  
pel
```

Then the output should be:

```
apple: app el  
appleapple: leap app pel  
frog:
```

---

## Standards

For an achieved standard, the program must work correctly on valid input representing words of length 12 or less and dictionaries of at most 1000 words.

Merit criteria include the ability to handle much larger dictionaries and words efficiently, gracefully dealing with poorly-formatted input (e.g., not all lower case), and clearly written and well-structured code.

Excellence criteria include some significant extension to the functionality of the program, or an investigation of general properties of the problem.

---

## Objectives

1.2, 1.4, 2.2, 2.3, 2.7, 2.9, 3.3, 3.4, 3.5, 3.6, 4.3.

(Pair)