LEVENSHTEIN DISTANCE

CHALLENGE DESCRIPTION:

Two words are friends if they have a Levenshtein distance of 1 (For details see Levenshtein distance). That is, you can add, remove, or substitute exactly one letter in word X to create word Y. A word’s social network consists of all of its friends, plus all of their friends, and all of their friends’ friends, and so on. Write a program to tell us how big the social network for the given word is, using our word list.

INPUT SAMPLE:

The first argument will be a path to a filename, containing words, and the word list to search in. The first N lines of the file will contain test cases, they will be terminated by string 'END OF INPUT'. After that there will be a list of words one per line. E.g

```
recursiveness
elastic
macrographies
END OF INPUT
aa
aahed
aahs
aalii
...
...
zymoses
zymosimeters
```

OUTPUT SAMPLE:

For each test case print out how big the social network for the word is. In sample the social network for the word 'elastic' is 3 and for the word 'recursiveness' is 1. E.g.

```
1
3
1
```

Constraints:
Number of test cases N in range(15, 30)
The word list always will be the same and it's length will be around 10000 words