

## 7090 Anagram Pyramids

Back in the 20th century, anagram puzzles were often found in the back pages of newspapers. Although they never reached the level of popularity of Sudoku puzzles, they were still a reliable means for killing a few minutes. One variant was the *anagram pyramid*, which consisted of a sequence of words stacked on top of each other. The word at the base of the pyramid had  $N$  letters, the second word had  $N - 1$  letters, the third word had  $N - 2$  letters, and so on. Each word (other than the word at the base) was formed by removing one letter from the word below and shuffling the remaining letters. Here is one example of an anagram pyramid:

```
PIN
SNIP
PAINS
PIANOS
```

Mr. Zino Ponzi, a retired financier, was reminiscing about the good old times one day when he got the idea of creating a few anagram puzzles to share with his friends at the Zigurat Retirement Home. Although he loved constructing anagram pyramids by hand, often he would get stuck, unable to think of a suitable word in the middle of the pyramid. Finally, he decided to hire a computer science student to write a program to make things easier. He didn't want to kill all the fun of doing it by hand, so he only wanted the program to tell him whether an anagram pyramid was possible for a given pair of top and bottom words and a dictionary.

### Input

There may be multiple cases to consider. Process until an end-of-file marker is detected. Words in this problem will be strings of from one to thirty letters. Treat upper and lower case versions of the same letter as equal. The input for each case will consist of a dictionary of  $N$  words ( $N < 1,000,000$ ) to be used to form pyramids, followed by  $M$  ( $M < 100$ ) pairs of top/bottom words from the dictionary with the top word shorter than the bottom in the following format (Strict pairing of words per line in second part is NOT to be implied by this sample):

```
N
word1
...
wordN
M
top1 bottom1
...
topM bottomM
```

### Output

The answers for each case should consist of  $M$  lines, one for each pair of top/bottom words, saying either 'yes' or 'no' depending on whether an anagram pyramid was possible given the input dictionary. Follow this format exactly: 'Case', one space, the case number, a colon and no trailing space on the first line for a case, and the answers on separate lines, again with no trailing spaces.

**Sample Input**

```
8
PEN
PIN
SNIP
PINE
PAINS
SPAIN
PIANOS
SNIPER
2
PIN PIANOS
PEN SNIPER
```

**Sample Output**

```
Case 1:
yes
no
```