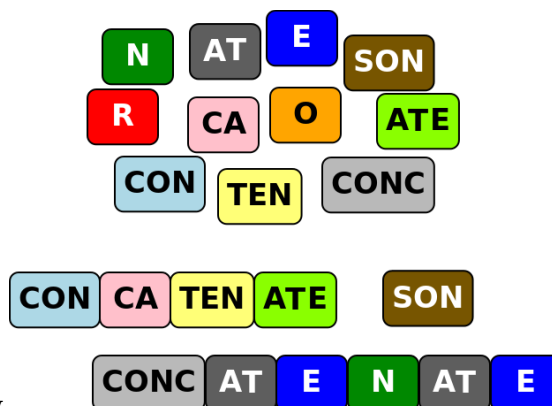


Problem H: Concatenate, Son, Concatenate!

That is all very beautiful son, but... do you know how to concatenate?

Daddy has invented a new word game for his son, Albert, who just loves letters. They have available a large pile of game pieces. Each piece has one or more letters inscribed and the main goal is to concatenate pieces in order to construct predefined sentences with one or more words. Since Albert is a little bit lazy, he wants to use as few pieces as possible. Note that the pieces cannot be broken in order to use just some of its letters. However, luckily, Albert has so many pieces that you may assume that you have an infinite quantity of each piece type.



Imagine for instance that Albert has the following 11 piece types:

CON | TEN | CONC | R | AT | N | CA | O | E | SON | ATE

If he wants to construct the sentence **CONCATENATE**, there are several different possibilities, using a different number of pieces. For example:

- Alternative 1 (4 pieces): CON | CA | TEN | ATE
- Alternative 2 (4 pieces): CONC | ATE | N | ATE
- Alternative 3 (5 pieces): CONC | AT | E | N | ATE
- Alternative 4 (6 pieces): CONC | AT | E | N | AT | E

In this case any of the first two alternatives would be preferable, using less pieces than the last two options. In fact, they correspond to the least amount of pieces that when concatenated form the sentence **CONCATENATE**.

Can you help Albert play the game?

Task

Given a set of **P** pieces, each one with one or more letters inscribed in it, and a set of **S** sentences, your task is to find the smallest number of pieces you need to use in order to form the given sentences.

Input

The first line contains an integer **P**, the number of pieces. The following **P** lines contain the description of the letters inscribed in them, one piece **P_i** per line.

The following line contains an integer **S**, the number of sentences to form. The following **S** lines contain their description, one sentence **S_i** per line.

You may assume that both in the pieces and in the sentences there are no other characters than simple uppercase letters.

Output

The output should contain exactly **S** lines, each one giving the smallest number of pieces to use in order to form the corresponding sentence. If it is impossible to form, you should print the word **no**.

Constraints

- $1 \leq P \leq 2,000$ Number of pieces
 $1 \leq S \leq 2,000$ Number of sentences to form
 $1 \leq \text{length}(\mathbf{P}_i) \leq 10$ Number of letters in each piece
 $1 \leq \text{length}(\mathbf{S}_i) \leq 200$ Number of letters in each sentence

Input Example

```
11
CON
TEN
CONC
R
AT
N
CA
O
E
SON
ATE
6
CONCATENATE
CONCATENATESONCONCATENATE
OR
WORD
TEN
RONRON
```

Output Example

```
4
9
2
no
1
6
```

This document was translated from $L^A T_E X$ by [H^EV^EA](#).