Find A Sub-Word

We define a *word character* to be any of the following:

- An English alphabetic letter (i.e., `a-z` and `A-Z`).
- A decimal digit (i.e., `0-9`).
- An underscore (i.e., `_`, which corresponds to ASCII value 95).

We define a *word* to be a contiguous sequence of one or more word characters that is preceded and succeeded by one or more occurrences of non-word-characters or line terminators. For example, in the string `I l0ve-cheese_?`, the words are `I`, `l0ve`, and `cheese_`.

We define a *sub-word* as follows:

- A sequence of word characters (i.e., English alphabetic letters, digits, and/or underscores) that occur in the same exact order (i.e., as a contiguous sequence) inside another word.
- It is preceded and succeeded by word characters *only*.

Given n sentences consisting of one or more words separated by non-word characters, process q queries where each query consists of a single string, s. To process each query, count the number of occurrences of s as a *sub-word* in all n sentences, then print the number of occurrences on a new line.

Input Format

The first line contains an integer, n, denoting the number of sentences.
Each of the n subsequent lines contains a sentence consisting of words separated by non-word characters.
The next line contains an integer, q, denoting the number of queries.
Each line i of the q subsequent lines contains a string, s_i, to check.

Constraints

- $1 \leq n \leq 100$
- $1 \leq q \leq 10$

Output Format

For each query string, s_i, print the total number of times it occurs as a sub-word *within all words in all* n *sentences*.

Sample Input

```
1
existing pessimist optimist this is
1
is
```

Sample Output

```
3
```

Explanation

We must count the number of times $s =$ `is` occurs as a sub-word in our $n = 1$ input sentence(s):

- s occurs 1 time as a sub-word of `ex`**`is`**`ting`.

- s occurs 1 time as a sub-word of pessim**is**t.
- s occurs 1 time as a sub-word of optim**is**t.
- While s is a substring of the word th**is**, it's followed by a blank space; because a blank space is non-alphabetic, non-numeric, and not an underscore, we do not count it as a sub-word occurrence.
- While s is a substring of the word **is** in the sentence, we do not count it as a match because it is preceded and succeeded by non-word characters (i.e., blank spaces) in the sentence. This means it doesn't count as a sub-word occurrence.

Next, we sum the occurrences of s as a sub-word of all our words as $1 + 1 + 1 + 0 + 0 = 3$. Thus, we print 3 on a new line.