**Simple Text Queries**

Given an array of *n* strings, *sentences*, where each *sentencesi* consists of at most *w* space-separated words, we want to perform *q* queries given by an array of *q* strings named *queries* where each *queriesj* consists of at most *k* space-separated words. To answer query *queriesj*, we find all the indices in *sentences* that contain every word in *queriesj* and assemble them into an array of indices listed in ascending order. For example, if all the words in *queriesj* are only found in *sentences4*, *sentences9*, and *sentences2*, then the answer to *queriesj* is the array *[2, 4, 9]*.

Complete the *textQueries* function in the editor below. It has two parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| sentences | string array | Each *sentencesi* (where *0 ≤ i < n*) consists of at most *w* space-separated words. |
| queries | string array | Each *queriesj* (where *0 ≤ j < q*) consists of at most *k* space-separated words. |

Print *q* rows of output where each row index *j* contains the answer to *queriesj* in the form of an array of indices in ascending order; if no sentence in *sentences* contains all the words in *queriesj*, then the answer to that query is the array *[-1]* instead.

Input Format

* The first line contains an integer, *n*, denoting the number of strings in *sentences*.
* Each line *i* of the *n* subsequent lines contains at most *w* space-separated words describing *sentencesi*.
* The next line contains an integer, *q*, denoting the number of strings in *queries*.
* Each line *j* of the *q* subsequent lines contains at most *k* space-separated words describing *queriesj*.

Constraints

* *1 ≤ n ≤ 104*
* *1 ≤ q ≤ 104*
* *1 ≤ w, k ≤ 10*
* Each word has at most *11* characters.
* No word appears in more than *10* sentences.
* Each word consists of uppercase and lowercase English alphabetic letters only (i.e., the character class *[a-zA-Z]*).

Output Format

Print *q* rows of output where each row index *j* contains the answer to *queriesj* in the form of an array of indices in ascending order; if no sentence in *sentences* contains all the words in *queriesj*, then the answer to that query is the array *[-1]* instead.

Sample Case 0

Sample Input

3

jim likes mary

kate likes tom

tom does not like jim

2

jim tom

likes

Sample Output

2

0 1

Explanation

We perform the following *q = 2* queries on *sentences = ["jim likes mary", "kate likes tom", "tom does not like jim"]*:

1. Find the indices of sentences containing both the words *"jim"* and *"tom"*. The only sentence containing both words is located at index *2*, so the array *[2]* is stored in index *0* of the answer array.
2. Find the indices of sentences containing the word *"likes"*. This word appears in *sentences0* and *sentences1*, so the array *[0, 1]* is stored in index *1* of the answer array.

We then print the array *[[2], [0, 1]]* as our answer.

Sample Case 1

Sample Input

4

how it was done

are you how

it goes to

goes done are it

2

done it

it

Sample Output

0 3

0 2 3

Explanation

We perform the following *q = 2* queries on *sentences = ["how it was done", "are you how", "it goes to", "goes done are it"]*:

1. Find the indices of sentences containing both the words *"done"* and *"it"*. These words appear in *sentences0* and *sentences3*, so the array *[0, 3]* is stored in index *0* of the return array.
2. Find the indices of sentences containing the word *"it"*. This word appears in *sentences0*, *sentences2*, and *sentences3*, so the array *[0, 2, 3]* is stored in index *1* of the return array.

We then return the array *[[0, 3], [0, 2, 3]]* as our answer.

Sample Case 2

Sample Input

3

it go will away

go do art

what to will east

3

it will

go east will

will

Sample Output

0

-1

0 2

Explanation

We perform the following *q = 3* queries on *sentences = ["it go will away", "go do art", "what to will east"]*:

1. Find the indices of sentences containing both the words *"it"* and *"will"*. These words only appear in *sentences0*, so the array *[0]* is stored in index *0* of the answer array.
2. Find the indices of sentences containing the words *"go"*, *"east"*, and *"will"*. These words do not appear in any sentence, so the array *[-1]* is stored in index *1* of the answer array.
3. Find the indices of sentences containing the word *"will"*. This word appears in *sentences0* and *sentences2*, so the array *[0, 2]* is stored in index *2* of the answer array.

We then print *[[0], [-1], [0, 2]]* as our answer.