# **DefaultDict Tutorial**



#### **Problem Statement**

DefaultDict is a container in the collections class of Python. It is almost the same as the usual dictionary (dict) container but with one difference. The value fields' data-type is specified upon initialization. A basic snippet showing the usage follows:

```
from collections import defaultdict
d = defaultdict(list)
d['python'].append("awesome")
d['something-else'].append("not relevant")
d['python'].append("language")
for i in d.items():
    print i
```

#### This prints:

```
('python', ['awesome', 'language'])
('something-else', ['not relevant'])
```

In this challenge, you will be given 2 integers (n and m) and n words which might repeat, say they belong to a word group A. Then you'll be given m other words belonging to word group B. For each of these m words, you have to check whether the word has appeared in A or not. If it has then you have to print indices of all of its occurrences. If it has not then just print -1.

#### **Constraints**

```
\begin{split} &1 \leq n \leq 10000 \\ &1 \leq m \leq 100 \\ &1 \leq \textit{length of each word in the input} \leq 100 \end{split}
```

#### **Input Format**

The first line contains n and m.

The next n lines contain the words belonging to A.

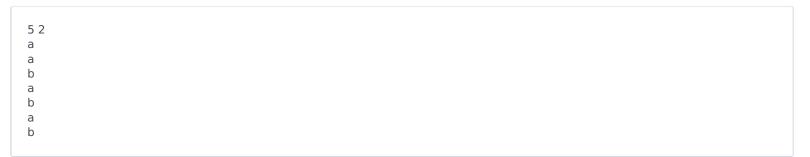
The next m lines contain the words belonging to B.

#### **Output Format**

Output m lines.

The  $i^{th}$  line should contain the 1 indexed positions of occurrences of the  $i^{th}$  word, separated by spaces, of the last m lines of the input.

## Sample Input



### **Sample Output**

## **Explanation**

'a' appeared 3 times in positions 1, 2 and 4. 'b' appeared 2 times in position 3 and 5. Hence the output. For the same word group A, had the appearances of 'c' been asked about in the word group B, you would have had to print -1 instead.