



## DefaultDict Tutorial ★

46/115 challenges solved

Rank: 69890 | Points: 505 !



Your DefaultDict Tutorial submission got 20.00 points.

Share

Tweet

[Try the next challenge](#) | [Try a Random Challenge](#)

Problem

Submissions

Leaderboard

Editorial 𐀀

The defaultdict tool is a container in the collections class of Python. It's similar to the usual dictionary (dict) container, but the only difference is that a defaultdict will have a default value if that key has not been set yet. If you didn't use a defaultdict you'd have to check to see if that key exists, and if it doesn't, set it to what you want.

For example:

```
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***. There are ***n*** words, which might repeat, in word group ***A***. There are ***m*** words belonging to word group ***B***. For each ***m*** words, check whether the word has appeared in group ***A*** or not. Print the indices of each occurrence of ***m*** in group ***A***. If it does not appear, print **-1**.

Constraints

 $1 \leq n \leq 10000$  $1 \leq m \leq 100$  $1 \leq \text{length of each word in the input} \leq 100$ 

Input Format

The first line contains integers, ***n*** and ***m*** separated by a space.The next ***n*** lines contains the words belonging to group ***A***.The next ***m*** lines contains the words belonging to group ***B***.

Output Format

Output ***m*** lines.The ***i*<sup>th</sup>** line should contain the **1**-indexed positions of the occurrences of the ***i*<sup>th</sup>** word separated by spaces.

Sample Input

```
5 2
a
a
b
a
b
a
```

b

**Sample Output**

```
1 2 4
3 5
```

**Explanation**

'a' appeared **3** times in positions **1**, **2** and **4**.

'b' appeared **2** times in positions **3** and **5**.

In the sample problem, if 'c' also appeared in word group **B**, you would print **-1**.

[Change Theme](#)

Python 3



```
1 from collections import defaultdict
2
3 def check_words_positions(groups):
4     b = groups['B']
5
6     for w in b:
7         if w in groups['A']:
8             for i, p in enumerate(groups['A']):
9                 if w == p:
10                    print(i + 1, end=' ')
11            else:
12                print(-1, end=' ')
13            print()
14
15 if __name__ == '__main__':
16     n, m = list(map(int, input().split()))
17
18     groups = defaultdict(list)
19
20     for _ in range(n):
21         groups['A'].append(input())
22
23     for _ in range(m):
24         groups['B'].append(input())
```

Line: 6 Col: 16

☒ Upload Code as File ☐ Test against custom input[Run Code](#)[Submit Code](#)

You have earned 20.00 points!

46/115 challenges solved.

40%

**Congratulations**

You solved this challenge. Would you like to challenge your friends?

[Next Challenge](#)**Earn a certificate in Python**

Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

[Get Certified](#)

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Input (stdin)

1	5 2
2	a
3	a
4	b
5	a
6	b
7	a
8	b

Download