

[Back to Interview \(/interview?candidate\\_interview\\_id=1273\)](/interview?candidate_interview_id=1273)

Time Left: Unlimited

## Problem

Given a list of sets of words, determine the frequency with which other words occur together with all of the words in another given set of words. The list of sets of words is provided as a CSV file. Using the data in this file, write a function that takes a set of words as parameter, and returns a dictionary of the frequency of OTHER titles that occur in a line that contains ALL WORDS in the given input set.

For example, if this were the input file:

```
AmericanDad,FamilyGuy,Cosmos
AmericanDad,FamilyGuy,Blacklist
FamilyGuy,Cosmos,Blacklist,Glee
AmericanDad,Cosmos,Blacklist,Glee
####
FamilyGuy,AmericanDad
Blacklist,Cosmos
```

Then each line after the "####" line consists of a query input set to be evaluated against all the data that preceded the "####" line. We will now give the correct results (with explanation) for this example input file:

Query 1:

```
FamilyGuy,AmericanDad
```

Answer 1:

```
{"Blacklist": 1, "Cosmos": 1}
```

Note that only lines 1 and 2 of the input data contain BOTH inputs, so only those lines are considered for the output.

Query 2:

```
Blacklist,Cosmos
```

Answer 2:

```
{"AmericanDad": 1, "FamilyGuy": 1, "Glee": 2}
```

Note that only lines 3 and 4 of the input data contain BOTH inputs, so only those lines are considered for the

output.

## Sample Input:

Download Sample Input (/download\_sample\_input?candidate\_problem\_id=1794)

```
AmericanDad,FamilyGuy,Cosmos
AmericanDad,FamilyGuy,Blacklist
FamilyGuy,Cosmos,Blacklist,Glee
AmericanDad,Cosmos,Blacklist,Glee
####
FamilyGuy,AmericanDad
Blacklist,Cosmos
```

## Sample Output:

Download Sample Output (/download\_sample\_output?candidate\_problem\_id=1794)

```
{"Blacklist": 1, "Cosmos": 1}
{"AmericanDad": 1, "FamilyGuy": 1, "Glee": 2}
```

## Input Format:

Read your input from a file. The file consists of two separate sections, separated by line consisting of "####". The first section consists of a large list of sets of words to be used as training data. The second section consists of individual sets of words to be interpreted as queries against the data from the first section.

- The file consists of ASCII-encoded text.
- Each line of the file contains a set of comma-separated words.
- Aside from the separating commas and the "####" line, there is no other punctuation in the file.
- None of the words contain whitespace.
- Words are case-sensitive.
- No word appears more than once in same line.

## Output Format:

For each test case, use the following function to output your result dictionary:

```
from json import dumps

def output(dictionary):
    print(dumps(dictionary, sort_keys=True))
```

## Bounds:

These are assumptions you can make about the input. You do not need to test for them.

**N** (number of input lines) will be very large.

**M** (number of words per input line)  $\leq 15$

Test

Submit

---

# Test your code on the sample input

## Step 1. Run your program on the sample input

Download Sample Input (/download\_sample\_input?candidate\_problem\_id=1794)

## Step 2. Upload Your Output

Choose File

No file chosen

Test

Summary

Correct

Incorrect

---

## Test Results Summary

Run your program on the sample input. Submit the output in order to test.