2. String Anagram

An anagram of a string is another string with the same characters in the same frequency, in any order. For example *'abc', 'bca', 'acb', 'bac', 'cba', 'cab'* are all anagrams of the string *'abc'*. Given two arrays of strings, for every string in one list, determine how many anagrams of it are in the other list. Write a function that receives *dictionary* and *query*, two string arrays. It should return an array of integers where each element *i* contains the number of anagrams of *query[i]* that exist in *dictionary*.

Example

dictionary = ['hack', 'a', 'rank', 'khac', 'ackh', 'kran', 'rankhacker', 'a', 'ab', 'ba', 'stairs', 'raits']
query = ["a", "nark", "bs", "hack", "stair"]

query[0] = 'a' has *2* anagrams in *dictionary*: *'a'* and *'a'*.
query[1] = 'nark' has *2* anagrams in *dictionary*: *'rank'* and *'kran'*.
query[2] = 'bs' has *0* anagrams in *dictionary*.
query[3] = 'hack' has *3* anagrams in *dictionary*: *'hack'*, *'khac'* and *'ackh'*.
query[4] = 'stair' has *1* anagram in *dictionary*: *'raits'*. While the characters are the same in *'stairs'*, the frequency of *'s'* differs, so it is not an anagram.
The final answer is *[2, 2, 0, 3, 1]*.

Function Description

Complete the function *stringAnagram* in the editor below.

stringAnagram has the following parameters:
string dictionary[n]: an array of strings to search in
string query[q]: an array of strings to search for

Returns
int[q]: an array of integers where the i^{th} value is the answer to *query[i]*

Constraints

Language Python 3 | Environment

Autocomplete Ready

```
1  #!/bin/python3 ...
10
11 #
12 # Complete the 'stringAnagram' function below.
13 #
14 # The function is expected to return an INTEGER_ARRAY.
15 # The function accepts following parameters:
16 #  1. STRING_ARRAY dictionary
17 #  2. STRING_ARRAY query
18 #
19
20 def stringAnagram(dictionary, query):
21     # Write your code here
22
23 if __name__ == '__main__': ...
```

Test Results | Custom Input | Run Code | Run Tests | Submit