There is a collection of input strings and a collection of query strings. For each query string, determine how many times it occurs in the list of input strings. Return an array of the results.

Example

$strings = ['ab', 'ab', 'abc']$
$queries = ['ab', 'abc', 'bc']$

There are 2 instances of 'ab', 1 of 'abc' and 0 of 'bc'. For each query, add an element to the return array, $results = [2, 1, 0]$.

Function Description

Complete the function matchingStrings in the editor below. The function must return an array of integers representing the frequency of occurrence of each query string in strings.

matchingStrings has the following parameters:

- string strings[n] - an array of strings to search
- string queries[q] - an array of query strings

Returns

- int[q]: an array of results for each query

Input Format

The first line contains and integer n, the size of $strings[]$.
Each of the next n lines contains a string $strings[i]$.
The next line contains q, the size of $queries[]$.
Each of the next q lines contains a string $queries[i]$.

Constraints

$1 \leq n \leq 1000$
$1 \leq q \leq 1000$
$1 \leq |strings[i]|, |queries[i]| \leq 20$.

Sample Input 1

Copy Download

```
4
aba
baba
aba
xzxb
3
aba
xzxb
ab
```

Sample Output 1

```
2
1
0
```

Sample Input 2

Copy Download

```
3
def
de
fgh
3
de
lmn
fgh
```

Change Theme Language Python 3

```
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'matchingStrings' function below.
#
# The function is expected to return an INTEGER_ARRAY.
# The function accepts following parameters:
#  1. STRING_ARRAY strings
#  2. STRING_ARRAY queries
#

'''
O(n) Time | O(1) Space: where n is the length of the input strings
'''
def matchingStrings(strings, queries):
    result = [0 for query in queries]

    (string_frequency, idx1) = ({}, 0)

    while idx1 < len(strings):
        string_frequency[strings[idx1]] = string_frequency.get(strings[idx1], 0) + 1
        idx1 += 1

    for idx, query in enumerate(queries):
        result[idx] = string_frequency.get(query, 0)

    return result

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    strings_count = int(input().strip())

    strings = []

    for _ in range(strings_count):
        strings_item = input()
        strings.append(strings_item)

    queries_count = int(input().strip())

    queries = []

    for _ in range(queries_count):
        queries_item = input()
        queries.append(queries_item)

    res = matchingStrings(strings, queries)

```

Line: 60 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

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