

RATE THIS CHALLENGE



We're going to make our own Contacts application! The application must perform two types of operations:

- add *name*, where *name* is a string denoting a contact name. This must store *name* as a new contact in the application.
- find *partial*, where *partial* is a string denoting a partial name to search the application for. It must count the number of contacts starting with *partial* and print the count on a new line.

Given *n* sequential add and find operations, perform each operation in order.

Example

Operations are requested as follows:

```
add ed
add eddie
add edward
find ed
add edwina
find edw
find a
```

Three *add* operations include the names 'ed', 'eddie', and 'edward'. Next, *find ed* matches all **3** names. Note that it matches and counts the entire name 'ed'. Next, add 'edwina' and then find 'edw'. **2** names match: 'edward' and 'edwina'. In the final operation, there are **0** names that start with 'a'. Return the array **[3, 2, 0]**.

Function Description

Complete the contacts function below.

contacts has the following parameters:

- string queries[n]: the operations to perform

Returns

- int[]: the results of each find operation

Input Format

The first line contains a single integer, *n*, the number of operations to perform (the size of *queries*[]).

Each of the following *n* lines contains a string, *queries*[*i*].

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq \text{length of } name \leq 21$
- $1 \leq \text{length of } partial \leq 21$
- name* and *partial* contain lowercase English letters only.
- The input does not have any duplicate *name* for the *add* operation.

Sample Input

STDIN	Function
-----	-----
4	queries[] size n = 4

```
queries = ['add hack', 'add hackerrank', 'find hac', 'find hak']
add hack
add hackerrank
find hac
find hak
```

Sample Output

```
2
0
```

Explanation

1. Add a contact named hack.
2. Add a contact named hackerrank.
3. Find the number of contact names beginning with hac. Both name start with hac, add **2** to the return array.
4. Find the number of contact names beginning with hak. neither name starts with hak, add **0** to the return array.

[Change Theme](#)

Language

C#



```
1  using System.CodeDom.Compiler;
2  using System.Collections.Generic;
3  using System.Collections;
4  using System.ComponentModel;
5  using System.Diagnostics.CodeAnalysis;
6  using System.Globalization;
7  using System.IO;
8  using System.Linq;
9  using System.Reflection;
10 using System.Runtime.Serialization;
11 using System.Text.RegularExpressions;
12 using System.Text;
13 using System;
14
15 class Result
16 {
17
18     /*
19      * Complete the 'contacts' function below.
20      *
21      * The function is expected to return an INTEGER_ARRAY.
22      * The function accepts 2D_STRING_ARRAY queries as parameter.
23      */
24
25     public static List<int> contacts(List<List<string>> queries)
26     {
27
```

Line: 55 Col: 1

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

Run Code

Submit Code