# Problem 3 - File Fix-it

## Problem

On Unix computers, data is stored in *directories*. There is one *root directory*, and this might have several directories contained inside of it, each with different names. These directories might have even more directories contained inside of them, and so on.

A directory is uniquely identified by its name and its parent directory (the directory it is directly contained in). This is usually encoded in a *path*, which consists of several parts each preceded by a forward slash ('/'). The final part is the name of the directory, and everything else gives the path of its parent directory. For example, consider the path:

```
/home/gcj/finals
```

This refers to the directory with name "finals" in the directory described by "/home/gcj", which in turn refers to the directory with name "gcj" in the directory described by the path "/home". In this path, there is only one part, which means it refers to the directory with the name "home" in the root directory.

To create a directory, you can use the *mkdir* command. You specify a path, and then*mkdir* will create the directory described by that path, but *only if* the parent directory already exists. For example, if you wanted to create the "/home/gcj/finals" and "/home/gcj/quals" directories from scratch, you would need four commands:

```
mkdir /home

mkdir /home/gcj

mkdir /home/gcj/finals

mkdir /home/gcj/quals
```

Given the full set of directories already existing on your computer, and a set of new directories you want to create if they do not already exist, how many *mkdir* commands do you need to use?

## Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each case begins with a line containing two integers **N** and **M**, separated by a space.

The next **N** lines each give the path of one directory that already exists on your computer. This list will include every directory already on your computer other than the root directory. (The root directory is on every computer, so there is no need to list it explicitly.)

The next **M** lines each give the path of one directory that you want to create.

Each of the paths in the input is formatted as in the problem statement above. Specifically, a path consists of one or more lower-case alpha-numeric strings (i.e., strings containing only the symbols

'a'-'z' and '0'-'9'), each preceded by a single forward slash. These alpha-numeric strings are never empty.

**Output**

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the number of *mkdir* you need.

**Limits**

1 ≤ **T** ≤ 100.
No path will have more than 100 characters in it.
No path will appear twice in the list of directories already on your computer, or in the list of directories you wish to create. A path may appear once in both lists however. (See example case #2 below).
If a directory is listed as being on your computer, then its parent directory will also be listed, unless the parent is the root directory.
The input file will be no longer than 100,000 bytes in total.

Small dataset

0 ≤ **N** ≤ 10.
1 ≤ **M** ≤ 10.

Large dataset

0 ≤ **N** ≤ 100.
1 ≤ **M** ≤ 100.

**Sample**

| Input | Output |
|---|---|
| 3<br>0 2<br>/home/gcj/finals<br>/home/gcj/quals<br>2 1<br>/chicken<br>/chicken/egg<br>/chicken<br>1 3<br>/a<br>/a/b<br>/a/c<br>/b/b | Case #1: 4<br>Case #2: 0<br>Case #3: 4 |