code jam
hello, world!

Practice Mode

Contest scoreboard | Sign in

Qualification Round 2008

A. Saving the Universe

B. Train Timetable

C. Fly Swatter

Contest Analysis

Questions asked 7

– Submissions

Saving the Universe

5pt	Not attempted **6760/10473 users** correct (65%)
20pt	Not attempted **6258/7836 users** correct (80%)

Train Timetable

5pt	Not attempted **5076/6516 users** correct (78%)
20pt	Not attempted **4408/5491 users** correct (80%)

Fly Swatter

5pt	Not attempted **1007/1536 users** correct (66%)
20pt	Not attempted **652/1274 users** correct (51%)

– Top Scores

rem	75
ymatsux	75
Reid	75
Jacek	75
krijgertje	75
inazz	75
gawry	75
t3hg0suazn	75
RomanLipovsky	75
jasonw	75

Problem A. Saving the Universe

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 5 points	Solve A-small Judge's response for last submission: Correct.
Large input 20 points	Solve A-large Judge's response for last submission: Correct.

Problem

The urban legend goes that if you go to the Google homepage and search for "Google", the universe will implode. We have a secret to share... It is true! Please don't try it, or tell anyone. All right, maybe not. We are just kidding.

The same is not true for a universe far far away. In that universe, if you search on any search engine for that search engine's name, the universe does implode!

To combat this, people came up with an interesting solution. All queries are pooled together. They are passed to a central system that decides which query goes to which search engine. The central system sends a series of queries to one search engine, and can switch to another at any time. Queries must be processed in the order they're received. The central system must never send a query to a search engine whose name matches the query. In order to reduce costs, the number of switches should be minimized.

Your task is to tell us how many times the central system will have to switch between search engines, assuming that we program it optimally.

Input

The first line of the input file contains the number of cases, **N**. **N** test cases follow.

Each case starts with the number **S** -- the number of search engines. The next **S** lines each contain the name of a search engine. Each search engine name is no more than one hundred characters long and contains only uppercase letters, lowercase letters, spaces, and numbers. There will not be two search engines with the same name.

The following line contains a number **Q** -- the number of incoming queries. The next **Q** lines will each contain a query. Each query will be the name of a search engine in the case.

Output

For each input case, you should output:

```
Case #X: Y
```

where **X** is the number of the test case and **Y** is the number of search engine switches. Do not count the initial choice of a search engine as a switch.

Limits

$0 < \mathbf{N} \le 20$

Small dataset

$2 \le \mathbf{S} \le 10$

$0 \le \mathbf{Q} \le 100$

Large dataset

$2 \le \mathbf{S} \le 100$

$0 \le \mathbf{Q} \le 1000$

Sample

```
Input          Output
2              Case #1: 1
5              Case #2: 0
Yeehaw
NSM
Dont Ask
B9
Googol
10
Yeehaw
Yeehaw
Googol
B9
Googol
NSM
B9
NSM
Dont Ask
Googol
5
Yeehaw
NSM
Dont Ask
B9
Googol
7
Googol
Dont Ask
NSM
NSM
Yeehaw
Yeehaw
Googol
```

In the first case, one possible solution is to start by using Dont Ask, and switch to NSM after query number 8.
For the second case, you can use B9, and not need to make any switches.