

Submissions

Centauri Prime

7pt	Not attempted 28/30 users correct (93%)
8pt	Not attempted 13/27 users correct (48%)

Music Collection

8pt	Not attempted 12/12 users correct (100%)
12pt	Not attempted 12/12 users correct (100%)

Extreme Escalator Pogo

5pt	Not attempted 2/3 users correct (67%)
10pt	Not attempted 1/2 users correct (50%)

Top Scores

PeterLiang	45
trollmonkey1	30
prettypinkponies	30
jahooma	30
sjhakorea	30
jboning	20
Peterliang	20
WillDevanny	15
tjwilson	15
DFlat2	15

Problem B. Music Collection

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 1  
8 points

Solve B-small-1

Small input 2  
12 points

Solve B-small-2

Problem

Audio Phil has a huge music collection, and he is very particular about the songs he listens to. Each song has a name that is a string of characters. His music player has a search feature that lets Phil type a substring into the search box, and the player then lists all songs whose names contain the substring. If there is exactly one song that matches the search, then Phil can hit the Enter key to play that song.

Phil hates using the mouse, and he doesn't like typing too much, so he insists on always typing the shortest possible substring that will match exactly the one song that he wants to play at this moment. Could you help him find his optimal search query?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each one starts with a line containing a single number **N**. The next **N** lines each contain one song name -- these are all of the songs in Phil's collection.

Each song name will consist of only letters, spaces and the hyphen character (-). All songs in Phil's collection will be unique and at most 100 characters in length. Song names are case insensitive, so "dZihan" is the same is "Dzihan". The search algorithm is also case insensitive.

Output

For each test case, output one line containing "Case #**x**:", where **x** is the case number (starting from 1). After that, print **N** lines, one for each song in Phil's collection, in the order that the songs were given in the input. For each song, print the shortest string of characters that will uniquely find that song. If there are several correct answers, print the lexicographically smallest one. Put double quotes around each string. If there is no correct answer, print ":(" without the double quotes.

Note that upper case letters come lexicographically before lower case letters, hyphen comes before all letters, and space comes before hyphen.

Limits

1 ≤ **T** ≤ 100.

Small dataset

1 ≤ **N** ≤ 10.

Large dataset

1 ≤ **N** ≤ 100.

Sample

Input	Output
5	
6	Case #1:
A Perfect Circle - Gravity	"V"
Aimee Mann - You Do	" D"
Aqualung - Cinderella	"Q"
Arcade Fire - Haiti	" F"
Art of Noise - Pleure	"S"
ATB - Marrakech	"B"
2	Case #2:
Hybrid - Altitude	"A"
Kings of Convenience - The Build-up	"C"
3	Case #3:
aaaaaaaabb	"AAAAAAA"
aaaaaaaabbb	"BBB"
ababababab	"BA"
3	Case #4:
butter	:(
fly	:(

butterfly  
1  
Unknown Artist - Track One

"RF"  
Case #5:  
" "

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform