Google Code Jam '11 Round 1A Problem B - The Killer Word

Time limit: 60.0s   **Memory limit:** 1G

You are playing Hangman with your friend Sean. And while you have heard that Sean is very good at taking candy from a baby, he is not as good at this game. Can you take advantage of Sean's imperfect strategy, and make him lose as badly as possible?

```
 +--+
 |  o
 | /|\      Mystery word: _ a _ a _ a _
 | / \
 |
 +-+---+
```

Hangman is played as follows:

- There is a dictionary D of all valid words, which both you and Sean know. A word consists only of the characters `a` - `z`. In particular, there are no spaces.
- You begin by choosing any word from D, and writing it down on a blackboard with each letter replaced by a blank: `_`.
- On his turn, Sean can choose any letter and ask you if it is in the word. If it is, you reveal *all* locations of that letter. Otherwise, Sean loses a point.
- Once all letters in the word have been revealed, the round ends.
- The round never ends early, no matter how many points Sean loses.

Sean uses a very simple strategy. He makes a list L of the 26 letters in some order, and goes through the list one letter at a time. If there is at least one word in D that (a) has the letter he is thinking of, and (b) is consistent with what you have written down so far *and the result of all of Sean's previous guesses*, then Sean guesses that letter. Otherwise, he skips it. No matter what, Sean then moves on to the next letter in his list.

Given Sean's list, what word should you choose to make Sean lose as many as points as possible? If several choices are equally good, you should choose the one that appears first in D.

Example

Suppose Sean decides to guess the letters in alphabetical order (i.e., $L = $ "abcdefghijklmnopqrstuvwxyz"), and D contains the words `banana`, `caravan`, and `pajamas`. If you choose `pajamas`, the game would play out as follows:

- You begin by writing 7 blanks `_ _ _ _ _ _ _` on the blackboard. Based on the number of blanks, Sean knows immediately that the word is either `caravan` or `pajamas`.

- Sean begins by guessing [a] since it is first in L, and you reveal all locations of the letter [a] on the blackboard: `_ a _ a _ a _`.
- Sean skips [b] even though it is used in `banana`. Sean already knows that is not your word.
- He then guesses [c] because it appears in `caravan`. It does not appear in the word you actually chose though, so Sean loses a point and nothing more is revealed.
- By process of elimination, Sean now knows your word has to be `pajamas`, so he proceeds to guess [j], [m], [p], and [s] in order, without losing any more points.

So Sean loses one point if you choose `pajamas`. He would have gotten either of the other words without losing any points.

Input Specification

The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing integers N and M, representing the number of words in the dictionary and the number of lists to consider.

The next N lines contain the words in the dictionary, one per line: $D_1, D_2, \ldots, D_N$. Each word is an arbitrary sequence of characters [a] - [z].

The final M lines contain all of the lists Sean will use, one per line: $L_1, L_2, \ldots, L_M$. Each list is exactly 26 letters long, containing each letter exactly once. Sean will use these lists to guess letters as described above.

Output Specification

For each test case, output one line containing `Case #x: w₁ w₂ ... wM`, where x is the case number (starting from 1) and w_i is the word you should choose if Sean guesses the letters in order L_i. If multiple words cause Sean to lose the same number of points, you should choose the option that appears first in the dictionary.

Limits

$1 \leq T \leq 10$.

Each word in D will have between 1 and 10 characters inclusive.

No two words in D will be the same within a single test case.

Memory limit: 1 GB.

Small Dataset

$1 \leq N \leq 100$.

$1 \leq M \leq 10$.

Time limit: 30 seconds.

Large Dataset

$1 \le N \le 10\,000.$

$1 \le M \le 100.$

Time limit: 60 seconds.

Sample Input

```
2
3 2
banana
caravan
pajamas
abcdefghijklmnopqrstuvwxyz
etaoisnhrdlcumwfgypbvkjxqz
4 1
potato
tomato
garlic
pepper
zyxwvutsrqponmlkjihgfedcba
```

Sample Output

```
Case #1: pajamas caravan
Case #2: garlic
```

Note

This problem has different time limits for different batches. If you exceed the Time Limit for any batch, the judge will incorrectly display `>60.000s` regardless of the actual time taken. Refer to the **Limits** section for batch-specific time limits.