


Alien Numbers

Problem ID: aliennumbers**CPU Time limit:** 1 second**Memory limit:** 1024 MB**Difficulty:** 2.0**Source:** Google Code Jam Practice Problems**License:** 

The decimal numeral system is composed of ten digits, which we represent as "0123456789" (the digits in a system are written from lowest to highest). Imagine you have discovered an alien numeral system composed of some number of digits, which may or may not be the same as those used in decimal. For example, if the alien numeral system were represented as "oF8", then the numbers one through ten would be (F, 8, Fo, FF, F8, 8o, 8F, 88, Foo, FoF). We would like to be able to work with numbers in arbitrary alien systems. More generally, we want to be able to convert an arbitrary number that's written in one alien system into a second alien system.

Input

The first line of input gives the number of cases, $T \leq 100$. T test cases follow. Each case is a line formatted as

```
alien_number source_language target_language
```

Each language will be represented by a list of its digits, ordered from lowest to highest value. No digit will be repeated in any representation, all digits in the alien number will be present in the source language, and the first digit of the alien number will not be the lowest valued digit of the source language (in other words, the alien numbers have no leading zeroes). Each digit will either be a number 0–9, an uppercase or lowercase letter, or one of the following symbols !"#%&'()*+,-./:;<>?@[\\]^_`{|}~

You may assume that the number in decimal is positive and at most 1000000000. All languages will have at most 94 digits.

Output

For each test case, output one line containing "Case # x : " followed by the alien number translated from the source language to the target language.

Sample Input 1

```
4
9 0123456789 oF8
Foo oF8 0123456789
13 0123456789abcdef 01
CODE O!CDE? A?JM!.
```

Sample Output 1

```
Case #1: Foo
Case #2: 9
Case #3: 10011
Case #4: JAM!
```