Exercise 1:

We would like to implement in this part a french-english dictionary. For this purpose, we are going to use a file in which we are going to save every translation with respect to the following encoding:

French word : English word

Write a function to:

1. Give the translation of a given French word
2. Give the translation of a given English word
3. Add a given translation to the dictionary.

Exercise 2:

(From Google Code Jam Competition).

Our computers are so excited about the upcoming Google I/O that they've started storing their ones as capital letter Is and their zeroes as capital letter Os! For example, the character A, which is 65 in ASCII, would normally be stored as the byte 01000001, but our computers are storing it as OIOOOOOI.

Given a string of 8-character "bytes" consisting of Is and Os, can you translate it using ASCII? Every "byte" is guaranteed to translate to a printable character (a decimal value between 32 and 126, inclusive). Note that one of these characters (the one with decimal value 32) is a space. No translated message will begin or end with a space, but there may be internal space characters.

Input (refer to IO.in and IO.out on AREL)

The first line of the input gives the number of test cases, **T**. **T** test cases follow; each consists of two lines. The first line of each test case contains an integer representing the number **B** of "bytes" in the string to be translated. The second line of each test case contains 8 * **B** characters, all of which are either I or O.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the translated message.