You want to encode a string changing its letters. For each letter in the original string, where a letter is in the range `a-z`, you should replace it according to some encoding strategy. But that is not all! Your encoding strategy can also change and depends on the current position of the original string that you are analyzing.

In the beginning, there is an initial encoding strategy, specifying how you should replace the letters of the original string. At each step you can also have some additional rules, specifying a new enconding strategy for some letter. If there is not a new encoding strategy for a letter you should keep the old one. For example, we can estabilish that every `a` in the original string should be replaced by `b`, then you should estabilish that every `a` that appears since position 4 of the original string should be replaced by `c`.

Input

There are several test cases. Each test case starts with a string S, you can assume that the length of S is at most 100. Then 26 lines will follow, specifying the encoding strategy for each letter, where the first line has the letter that should replace an `a` in the string S, the second line has the letter that should replace a `b` in the original string and so on.

After this, there is a line with an integer $0 \leq R \leq 1000$ indicating the number of additional rules. Each one of the next R lines will have a rule, where each rule has the form $P\ X\ Y$, indicating that since position $P \geq 0$ of the original string you should replace the letter X by Y. The first letter of a string is at position 0. You can assume that the initial position P of a rule in a test case is never less than the initial position of a previous rule in the same test case. If there are two rules for the same letter starting in the same position you should consider the rule that appears later in the input. There is a blank line after each test case and the input is finished by end of file (EOF).

Output

For each test case you must print the message: '`Case #`N`: The encoding string is `E`.`', where N is the number of the test case, starting from 1, and E is the string that you get after apply the set of encoding rules over the original string. After each test case you should print a blank line.

Sample Input

```
ufrn
t
o
w
k
q
z
f
n
y
i
c
m
s
j
n
r
g
l
d
s
u
s
g
y
e
u
10
0 q t
0 j f
1 v d
1 f d
1 r o
2 e p
2 v e
3 y p
3 t m
3 u k
```

Sample Output

```
Case #1: The encoding string is udoj.
```