Problem 1: Rebranding

(Taken from: http://codeforces.com/contest/591/problem/B)

The name of one small but proud corporation consists of n lowercase English letters. The corporation has decided to try rebranding by changing their name.

For this purpose the corporation has consecutively hired m designers. Once a company hires the i-th designer, he immediately contributes to the creation of a new corporation name as follows: he takes the newest version of the name and replaces all the letters x_i by y_i, and all the letters y_i by x_i. This results in the new version. It is possible that some of these letters do not occur in the string. It may also happen that x_i coincides with y_i. The version of the name received after the work of the last designer becomes the new name of the corporation.

Manager Arkady has recently been hired by this company, but is already soaked in the spirit of teamwork and is very worried about the success of the rebranding. Satisfy Arkady's curiosity and tell him the final version of the name.

Input
The first line of the input contains two integers n and m ($1 \leq n, m \leq 200000$), the length of the initial name and the number of designers hired, respectively. The second line consists of n lowercase English letters and represents the original name of the corporation.

Lastly, m lines contains the descriptions of the designers' actions. The i-th of them contains two space-separated lowercase English letters, x_i and y_i.

Output
Print the new name of the corporation.

Examples

- **Input**
  6 1
  police
  p m

  Output
  molice

- **Input**
  11 6
  abacabadaba
  a b
  b c
  a d
  e g
  f a
  b b

  Output
  cdcbcdcfcdc

Problem 2: Light Switch

(Taken from: ICPC SoCal Regionals 2013)

A lighting company has announced a new product - a blinking string of lights taht can be ordered with a variable number of bulbs (N) on it. This string of lights is unique because instead of blinking on and off in unison, there is a special pattern that governs how bulbs get turned on and off. Every second, one or more bulbs toggle from on to off or off to on depending on their last states and their positions from the beginning of the string - a bulb is toggled if its position is a multiple of time t. Assume that at time $t = 0$ all bulbs are off. At time $t = 1$ all bulbs would toggle on (1, 2, 3 ,4 etc.) At time $t = 2$ only even numbered bulbs (2, 4, 6, 8, etc.) change state, at time $t = 3$ every third bulb (3, 6, 9, 12, etc.) changes states, and so on. This continues until time $N + 1$ at which point all bulbs reset to off (as they were at $t = 0$). The light all toggle on again at time $N + 2$ and the process repeats.

The company's Quality Control department is having a hard time verifying that the bulbs are turning on and off at the appropriate time. Your team has been asked to write a verification program that, given the number of bulbs on the strand N, a particular time t, and a bulb number b, will determine if that bulb should be on or off.

You are given that $3 \leq N < 2^{54}$, $1 \leq t < 2^{54}$, $1 \leq b < 2^{54}$, $b \leq N$.

Input to your program will be a series of lines, each containing values for N, t, and b in that order, separated from each other by one or more spaces. Input is completed at the end of file. No input line will exceed 80 columns.

For each input line, your program is to print a line containing the bulb number followed by a colon, a single space, and the string "On" if the bulb is on, or "Off" if the bulb is off. No leading or trailing whitespace is to appear on an output line.

Sample Input 55 10 24
55 68 24
20 70 5

Sample Output 24: Off
24: On
5: Off

Problem 3: Anagram Pyramids

(Taken from: ICPC SoCal Regionals 2014)

Back in the 20th century anagram puzzles were often found in the back pages of many newspapers. Although they never reached the level of popularity of Sudoku puzzles, they were still a reliable means for killing a few minutes. One type of such puzzles was the anagram pyramid, which consisted of a sequence of words stacked on top of each other. The word at the bottom had N letters, the second-to-last word had $N-1$ letters, the third-to-last word had $N-2$ letters, and so on. Each word was formed by rmeoving one letter from the previous word and shuffling the remaining letters. Here is one example of an anagram pyramid:

- PIN
- SNIP
- PAINS
- PIANOS

Mr. Zino Ponzi, a retired financier, was reminiscing of the good old times one day when he got the idea of creating a few anagram puzzles to share with his friends at the Ziggurat Retirement Home. Although he loved constructing anagram pyramids by hand, often he would get stuck unable to think of suitable word in the middle of the pyramid. Finally he decided to hire a student programming team to write a program to make things easier. He didn't want to kill all the fun of doing it by hand, so he only wanted the program to tell him whether an anagram pyramid was possible for a pair of top and bottom words.

Input to your program will consist of a dictionary of N words ($N < 1,000,000$), followed by M pairs ($M < 100$) of top and bottom words in the following format:

N
word1
...
word$N$
M
top1 bottom1
...
top$M$ bottom$M$

All input will begin in the first column. Words in the top and bottom word pairs will be separated from each other by single spaces. The top word will be shorter than the bottom word. The top and bottom words will appear in the dictionary. No word will be longer than thirty letters. Words will contain only upper-case letters A through Z.

The output is to consists of M lines, one for each pair of top and bottom words, with each line containing only the word "yes" or "no" depending on whether an anagram pyramid was possible using the given input dictionary. No leading or trailing whitespace is to appear on an output line.

Sample Input

```
8
SPAIN
PIANOS
PEN
SNIP
PAINS
SNIPER
PIN
PINE
2
PIN PIANOS
PEN SNIPER
```

Sample Output

```
yes
no
```