DNA Compression

Alex works at the biotech division of the Association of Crazy Mathematicians (ACM). She helps her organization catch criminals by doing DNA fingerprinting. DNA contains four different bases - adenine (abbreviated A), cytosine (C), guanine (G), and thymine (T). A sequence of DNA can therefore be represented as a sequence of A's, C's, G's, and T's.

However, a strand of DNA could contain millions of bases! Alex, being a clever computer scientist, developed a simple algorithm to compress any DNA sequence. She encodes long runs of the same base as a number followed by a letter. For example, She can write "AAAGGTCCCCCCCCCCCCTA" as "3A2GT12C1TA" because AAA-GG-T-CCCCCCCCCCCC-T-A has three A's, followed by two G's, followed by a T (note that the 1 is optional in the encoding), followed by twelve C's, a T, and an A. As you can see, the shorthand representation can be much shorter that the original.

Help Alex test her implementation of this compressing algorithm, and she'll have her friends at Amazon give you a special prize.

Input

The first line will contain a single integer N <= 1000 denoting the number of test cases. Each of the following N lines will be of the form "X Y" where X is a DNA sequence written out in the long form, and Y is a (valid) compressed DNA sequence. You must check if X and Y represent the same DNA.

Output

For each test case, print out a line containing the single word "Yes" if the two DNA sequences are identical or the single word "No" otherwise.

Example

Input

```
3
AAAGGTCCCCCCCCCCCCTA 3A2GT12C1TA
AGGG 1A2G
A 1A
```

Output

```
Yes
No
Yes
```

Golf Fine

Land development company Developers-R-Us has been in constant battle with environmentalists for decades. In recent years, the company has been responsible for destroying the habitat of the Michigan monkey flower, and has faced large fines as a consequence.

The lawyers for this land development company have thought up a new idea - a sort of loophole in the system. The company has purchased a large plot of land, but will not develop all of it, thus incurring fines only for those areas containing monkey flowers that are adjacent (horizontally, vertically, or diagonally) to developed land.

The environmental engineers of Developers-R-Us have provided you, the software engineer, with a series of grid maps representing the area where a new golf course will be built. They would like to determine the area which will be covered by the proposed golf course, as well as the fines the company will have to pay for building it. You will be provided with a 10 x 10 grid, representing the 100 square acres being used to build the golf course. Each acre of monkey flowers along the path costs $50 000.

Input

Input will consist of ten lines, each containing ten characters, where:

- . - represents land not being developed
- s - represents the start of the golf course; there will be exactly one such acre in the whole grid map
- d - represents an acre of developed land; note that there may be developed land which is not connected (horizontally, vertically, or diagonally) to the golf course, but this is not your problem
- m - represents an acre of land containing Michigan monkey flowers

Output

Provide, on two separate lines, the number of acres being developed for the golf course, and the fine for building the course next to areas containing Michigan monkey flowers. Pay close attention to the output format for the fine: there must be a dollar sign at the beginning of the line, and a space should be used as thousands separator.

Example

Input

```
..........
.sd.......
..d.......
..dm......
..d.......
..d...m...
..ddddm...
..........
...ddd....
...mmm....
```

Output

```
10
$150 000
```

Vocabulary Check

Your English teacher gave you a homework to prepare you for the SAT. She tells you a list of vocabulary words, and you must write text that contains all of the words at least once. What the teacher didn't anticipate is how long it would take to grade this assignment. Therefore she is now calling on your programming skills to make her life easier.

You must take blocks of text and discard the ones which obviously don't satisfy the assignment's requirements. Those that contain all the words in the list will be flagged for further review by your teacher. Your program should be able to handle differences in capitalization, both at the beginning and within words. Because spelling mistakes do happen, your program should also tolerate one letter substitution error in each word. And of course, you should ignore all non-alphabetic characters because they aren't parts of words so you might as well treat them as whitespace.

Are you up to the task? Help your teacher and you might get more than just extra credit. You'll get Lockheed Martin's prize!

Input

The first line of input will contain a single positive integer W <= 100. The next line is the vocabulary list, W words (each at most 50 characters in length) separated by a comma and a space. Then, there will be a line containing a single positive integer T <= 100, the number of text blocks to check. Each of the next T lines will be a text block. It may be composed of multiple sentences, but will not exceed 1000 characters in length.

Output

For each text block, check if it contains all the words in the vocabulary list, allowing for punctuation, capitalization differences, and at most one letter substitution per word. If the block passes, print out "FLAGGED" on a single line, or "NOT FLAGGED" otherwise.

Example

Input

```
3
car, cat, dog
3
The dog, the fat are not allowed to drive the car.
The Dog ran over the cat and dog.
The Cat ran over the cat and dog.
```

Output

```
FLAGGED
NOT FLAGGED
FLAGGED
```

Rides

Disney is building a new amusement park on an archipelago! The individual islands are only large enough to fit a single ride each, however, so they'll have to build bridges to connect the different rides together. You have been hired to verify that every ride is reachable from every other ride, by going through some sequence of bridges from one ride to the other. A Disney-sponsored reward is reserved for the first to solve this problem.

Input

The first line of input will contain a single integer T <= 10, the number of test cases. An individual test case will start with a line containing two integers R and B, both no greater than 100, separated by a space. R is the number of rides in the amusement park, B is the number of bridges Disney is planning to build. B lines follow, describing one of the bridges. A bridge is identified by the names of the two rides it connects. There will be at most one bridge connecting two rides. Ride names are unique and at most 32 characters long. They may contain uppercase letters and underscores.

Output

For each test case, print out "Yes" if all the rides are connected, and "No" otherwise.

Example

Input

```
2
3 2
DONALDS_ADVENTURE MICKEY_ROLLER_COASTER
FAMILY_FUNLAND MICKEY_ROLLER_COASTER
2 0
```

Output

```
Yes
No
```

Lock Code

Alberta went to the beach thinking it would be just another lazy vacation, until she saw what looked like a treasure chest floating to shore, and her world changed forever. Examining the locking mechanism, she immediately realized she came across part of the Unusually Fascinating Pirate Treasure (UFPT), a series of puzzles released by the Association of Crazy Mathematicians (ACM) to reward those who are skilled in the mathematical arts and punish those who are not.

This particular puzzle had six dials of the digits 0 through 9 and the inscription: "I'll give you two positive integers no greater than a billion. Tell me the final digits of raising the first to the power of the second... or else!" Sadness overcame Alberta as she read this. With numbers as big as a billion, the multiplications could take years, but she couldn't count on having more than a few seconds to give the correct answer.

If only she had paid attention in number theory! This is the wildcard prize problem: if the prizes for any sponsored problems are not claimed, they will be given to the teams who solve this problem.

Input

Input will consist of multiple lines, each containing two integers A and B ($1 <= A, B <= 1000000000$), separated by a space. A line containing two zeros signals the end of input and should not be processed.

Output

For each line of input, print out the correct dial settings, i.e. the final digits of A^B.

Example

Input

```
421 1988
628 496
0 0
```

Output

```
362161
493696
```

Shift Fix

Oh noes! Your Shift key broke so you can't capitalize properly anymore! You're too cheap to buy a new keyboard. You'll have to either type everything lowercase, getting marked off on all your homework, or toggle Caps and type everything UPPERCASE, losing any respect you might still have left on online forums. I don't know which is worse!

Then it hits you... programming can save the day yet again. For starters, make everything lowercase. Then, capitalize the first word of every sentence. Feed in a list of exceptions (names, acronyms, etc.), replacing any instances with the proper casing, and you're set.

See? You don't really need a Shift key anyway. All you need is a program to process everything you type. H4X!!!1!!!!!one!!!!!!2+e^(i*pi)!!!

Input

The first line of input will contain a single integer N <= 100. Each of the next N lines contains one word of at most 50 characters each, an exception to the casing rules. Next, there is a single integer L <= 100, followed by L lines containing the text you are supposed to fix.

Output

Output the text given in the input, with casing fixed as described in the problem. In this problem, sentences may end with a period (.), an exclamation mark (!), or a question mark (?).

Example

Input

```
6
Rick
Astley
Jason
McAninley
Miorel
Palii
4
if you're enjoying this contest, please thank miorel palii.
if you're not, the person to blame is jason mcaninley.
do you know who wrote this problem?
(no, not jason. good guess though.)
```

Output

```
If you're enjoying this contest, please thank Miorel Palii.
If you're not, the person to blame is Jason McAninley.
Do you know who wrote this problem?
(No, not Jason. Good guess though.)
```

Data Distortion

Willie's hard drive is getting full, and he's convinced it's because of all his Word documents from the writing assignments he's had since middle school.

There's a lot of them, but Willie is still attached to them for sentimental reasons, so he doesn't want to delete them outright.

As he's looking at the documents, he realizes that lots of words are taking up more space than necessary by repeating letters multiple times in a single word.

Willie figures that he can save a lot of space simply by transforming his documents so that, in each word, a given letter only appears one time.

To make sure that he can still figure out what the words are, he wants to make sure that the letters appear in the same order in the transformed word as they did in the original word.

Your task is to write a program to help Willie transform his documents.

Collapse strings such that each letter only appears once, in the order in which they appear in the original string.

e.g.

banana -> ban

abacus -> abcu

programming -> progamin

Input

Each line will contain one test case. Only lowercase letters will appear. The end of input is signaled by EOF.

Output

For each test case, output on a new line the transformed word.

Example

Input
```
banana
abacus
programming
```

Output
```
ban
abcu
progamin
```