# Språk från rymden

After years of study, people at Adlibris have discovered an alien language transmitted from a faraway planet. The alien language is unique in that every word consists of exactly **L** lowercase letters. Also, there are exactly **D** words in this language.

Once the dictionary of all the words in the alien language was built, the next breakthrough was to discover that the aliens have been transmitting messages to Earth for the past decade. Unfortunately, the signals with these messages are weakened due to the distance between our two planets and some of the words may be misinterpreted. In order to help them decipher these messages, the scientists have asked you to devise an algorithm that will determine the number of possible interpretations for a given pattern.

A pattern consists of exactly **L** tokens. Each token is either a *single lowercase letter* (the scientists are very sure that this is the letter) or a *group of unique possible lowercase letters* surrounded by parenthesis ( and ). For example: (ab)d(dc) means the first letter is either a or b, the second letter is definitely d and the last letter is either d or c. Therefore, the pattern (ab)d(dc) can stand for either one of these 4 possibilities: add, adc, bdd, bdc.

## User Input

The first line of input is the configuration of the language and it contains 3 integers, **L**, **D** and **N** separated by a space. After that, **D** lines will follow, each containing one word of length **L**. These are the words that are known to exist in the alien language. **N** alien messages then follow, each on its own line and each consisting of a pattern as described above. You may assume that all known words provided are unique.

## Output

For each alien message, the output should be

“Message #**X**: **K”**

where **X** is the number of the message starting from 1, and **K** indicates how many words in the alien language match the pattern.

Limits

1 ≤ **L** ≤ 10   
1 ≤ **D** ≤ 25   
1 ≤ **N** ≤ 10

# Example

## If the user inputs:

3 5 4  
abc  
bca  
dac  
dbc  
cba  
(ab)(bc)(ca)  
abc  
(abc)(abc)(abc)  
(zyx)bc

## The resulting output should be:

Message #1: 2  
Message #2: 1  
Message #3: 3  
Message #4: 0