Problem C. Codejamon Cipher

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 7 points	Solve C-small
Large input 16 points	Solve C-large

Problem

The ***Codejamon*** monsters talk in enciphered messages. Here is how it works:

Each kind of monster has its own unique *vocabulary*: a list of **V** different words consisting only of lowercase English letters. When a monster speaks, it first forms a sentence of words in its vocabulary; the same word may appear multiple times in a sentence. Then, it turns the sentence into an enciphered string, as follows:

1. Randomly shuffle each word in the sentence.
2. Remove all spaces.

Understanding the monsters can bring you huge advantages, so you are building a tool to do that. As the first step, you want to be able to take an enciphered string and determine how many possible original sentences could have generated that enciphered string. For example, if a monster's vocabulary is *["this", "is", "a", "monster", "retsnom"]*, and it speaks the enciphered string *"ishtsiarestmon"*, there are four possible original sentences:

- "is this a monster"
- "is this a retsnom"
- "this is a monster"
- "this is a retsnom"

You have **S** enciphered strings from the same monster. For each one, can you figure out the number of possible original sentences?

IMPORTANT: Since the output can be a really big number, we only ask you to output the remainder of dividing the result by the prime $10^9 + 7$ (1000000007).

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of one line with two integers **V** and **S**, the size of the monster's vocabulary and the number of enciphered strings. Then, **V** lines follow; each contains a single string of lowercase English letters, representing a word in the monster's vocabulary. Finally, **S** lines follow. Each contains a string consisting only of lowercase English letters, representing an enciphered sentence. It is guaranteed that all enciphered sentences are valid; that is, each one has at least one possible original sentence.

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 a space separated list of of **S** integers: the answers (modulo $10^9 + 7$) for each enciphered sentence, in the order given in the input, as described in the problem statement.

Limits

$1 \le \mathbf{T} \le 100$.
$1 \le \mathbf{S} \le 5$.

Small dataset

1 ≤ the length of each word in the monster's vocabulary ≤ 5.
1 ≤ the length of the enciphered string ≤ 50.
$5 \le \mathbf{V} \le 10$.

Large dataset

1 ≤ the length of each word in the monster's vocabulary ≤ 20.
2000 ≤ the length of the enciphered string ≤ 4000.
$200 \le \mathbf{V} \le 400$.

Sample

Input Output

2 5 1 this is a good day sithsiaodogyad 5 3 pt ybsb xnydt qtpb kw xnydttbpqtpqb yxdtntpbsby ptptxytdnsbybpt	Case #1: 2 Case #2: 1 1 1