

Strings

This problem features a story of a certain Black Kitten. Perhaps you saw a funny image where a few kittens sit and think about some mundane stuff, like “I want this”, “I want that”, while the black one thinks “I want to rule the Universe”... Well, that’s true! The Black Kitten took a long sheet of paper and wrote a String on it. That String should give him power to rule the Universe!.. (Or was he just going to win ACM ICPC Finals in 2013 with it?)

The sad thing is that we’ll never know for sure because of other kittens and their stupid games. As you may know, many kittens like to shred furniture, wallpaper and other stuff like that. To make a long story short, a certain White Kitten shred the paper on which the String was written! No more ultimate power for the Black one. (What a mess.)

Luckily, the String was only cut into two parts: the First part and the Second one. The Black Kitten now performs some rituals with the two parts. That, he believes, will help him gain some of the magical power. (Or maybe he is just practicing for the Finals?)

Anyway, right now, the Black Kitten calculates the number of different substrings of the First part that occur in the Second part exactly K times. Two or more occurrences may overlap. Two substrings are considered different if they are not equal as strings. You may try and calculate the answer for him.

Perhaps some of the power will be yours, too. (At the very least, you have a chance to improve your programming and problem solving skills by doing so.)

Input

The first line of input contains an integer T that represents the number of test cases. Every following test case is three lines, The first line of test case contains the First part of the String. The second line contains the Second part.

Each of the parts is a non-empty string consisting of lowercase English letters, and the size of each is no more than 8,000 characters. The third line contains one positive integer K not exceeding the length of each of the given parts.

Output

For each test case, output on a single line “Case # T :” where T is the number of the test case, followed by a line contains one integer: the number of different substrings of the First part that occur in the Second part exactly K times.

Example

Input:

```
3
egyptnational
ecpc
1
egyptnational
ecpc
```

2
fastlast
bestmost
2

Output:

Case #1:

2

Case #2:

0

Case #3:

3