**Sub Sequence String**

**Input:** Standard Input, **Output:** Standard Output

**Time Limit:** 1 second(s)

**Memory Limit:** 256 megabytes

**Problem Statement:**

Given three non empty strings **a, b** and **c**. Let **b** have concatenation with **c** to it's **beginning** and **ending** position. If the **order** of the letters are exact, then you have to calculate **How Many strings of c + b + c are occurred in the string a as a subsequence**.

You have to consider an occurrence of a subsequence, if the **letters** of string **b** are **contiguous** in string **a** but letters of **c** may not be contiguous. Assume that, string **c** contains **all same** characters.

For Example: let, a = "opoabcoaboco" , b = "abco" , c = "o" and concatenation of c + b + c = "oabcoo".Here, b = "abco" is only contiguous in position 4 (1 base index).

Now, the occurrence of subsequences in positions (1 based index):

1. 1,4,5,6,7,10

2. 1,4,5,6,7,12

3. 3,4,5,6,7,10

4. 3,4,5,6,7,12

so , total number of subsequence is 4.

**Input:**

There are **T <= 100** test cases. In each case, three non empty strings **a , b , c** consists lower case letters are given where **|a|, |b| <= 200000** and **|c|<= 200**.It is guaranteed that **all characters of c are same.**

**Output:**

In each case, output one integer - the total number of subsequences. As the result is pretty big, so output **modulo 1e9 + 7**.

**Sample Input/Output:**

|  |  |
| --- | --- |
| **Sample Input** | **Sample Output** |
| 3  opoabcoaboco  abco  o  aabcdaabacadaa  bcd  aa  aabc  c  a | 4  15  0 |