

6980 Afghan Jerseys

Afghanistan is playing its maiden ICC Cricket World Cup in 2015. After all the war, their cricket board wants to put on a good show to get the people's mind off the strife, and guide it towards the more important kind of conflict — on the pitch. It has therefore decided to give all the players in the country a brand new jersey, with the player's name printed to inspire them to kick butt.

Afghani names being long, they do not fit properly on the jerseys. So the board has decided to compress the names into patterns that match parts of the players' names. To save on the fabric printing cost, each pattern also has to match as many names as possible. The problem has been given to you to solve with your programming skills.

The pattern is given as a string P . Each character of the pattern P is either a lower-case character between 'a' to 'z' or '*'. A '*' matches any number of characters (including 0). For example, the string 'abc' matches patterns '*', 'a*', and '*b*', but not 'b*'.

As the number of players in the country is huge, it's a tedious task to match them individually. Hence, they have collected the most frequently occurring chunks (only lowercase alphabetical letters) from players' names as a string T .

Given a string T , consisting only of lower-case characters 'a' to 'z', they want to know how many distinct subsequences of T match a given pattern P modulo 1000000007.

A subsequence of sequence is a sequence of characters that can be derived from the given sequence by deleting some elements without changing the order of the remaining elements. For example, 'ace' is a valid subsequence of 'abcde', while 'aec' is not.

Input

The first line contains the number of test cases, Q . Q test cases follow. Each test case contains the text T on the first line, followed by the pattern P on the next line.

Output

Output the answer for each test case on a new line. All answers must be output modulo 1000000007.

Constraints:

- $1 \leq Q \leq 50$
- $1 \leq |T| \leq 1000$
- $1 \leq |P| \leq 50$

Explanation:

For the first example, all subsequences (including the empty string) match the pattern '*'.

For the second example, note that the string 'a' occurs as a subsequence of the text twice, but it should only be counted once.

Sample Input

```
4
abc
*
aba
```

*
ababa
ab
abcbaba
a*b

Sample Output

8
7
1
14