Password Protection

filename: `protect`
Difficulty Level: `Medium-Hard`
Time Limit: `8 seconds`

You have recently made an account on a website. You used a password manager to produce a random password. One of the constraints was that your password does not contain your first or last name. You assumed that containment here referred to having a substring (contiguous sequence of characters) that were identical to the sequence of characters that comprised either your first or last name.

You were unlucky enough that your first password did not satisfy this criterion. You want to know how often such an unfortunate turn of events could happen. Write a program to help in this endeavor by simply counting the number of passwords that contain either a given first name or last name as a substring. For example, if your first name was “ali” (quotes for clarity), then the password “california” contains your first name as a substring but the password “applied” does not contain your first name as a substring, even though “ali” is a subsequence of this password.

Note that this is an “inclusive or” which means either or both, i.e., a password is not valid if it contains the first name or the last name or both.

The Problem:

Given a first name, last name, and password length, determine the number of randomly generated passwords that will contain your first or last name as a substring (contiguous subsequence of characters). You can assume that passwords generated will contain only lowercase letters.

The Input:

The first input line contains an integer, n ($1 \le n \le 10^7$), representing the length of the password that is randomly generated. The second input line contains a string of x lowercase letters ($1 \le x \le 128$), representing the first name to check. The third input line contains a string of y lowercase letters ($1 \le y \le 128$), representing the last name to check.

The Output:

Print an integer representing the number of invalid passwords, i.e., they contain the first or last name. Since the answer can be quite large, your program should report the number as the smallest non-negative remainder when divided by 1,000,000,007.

(Sample Input/Output on the next page)

Sample Input	**Sample Output**
6 ali orooji	70304
5 arup guha	104