

4. Filtering Messages

Users of a chat application have been having some issues. Developers have discovered a virus that may be included in the messages and created a filter pattern to remove the virus. A *good message string* does not contain the virus pattern. Given two message strings of equal lengths, *messageA* and *messageB*, and a virus pattern string, *virusC*, determine the number of good message strings that can be created that are:

1. the same length as strings *messageA* and *messageB*, and
2. alphabetically greater than or equal to *messageA* and
3. alphabetically smaller than or equal to *messageB*.

Since the number can be large, return the value modulo (10^9+7) .

Note:

Formally, the string *p* of length *n* is alphabetically smaller than or equal to string *q* of the same length, if $p_1 = q_1, p_2 = q_2, \dots, p_{k-1} = q_{k-1}, p_k < q_k$ for some k ($1 \leq k \leq n$) or $p_1 = q_1, p_2 = q_2, \dots, p_{n-1} = q_{n-1}, p_n = q_n$.

Similarly, the string *p* of length *n* is alphabetically greater than or equal to string *q* of the same length, if $p_1 = q_1, p_2 = q_2, \dots, p_{k-1} = q_{k-1}, p_k > q_k$ for some k ($1 \leq k \leq n$) or $p_1 = q_1, p_2 = q_2, \dots, p_{n-1} = q_{n-1}, p_n = q_n$.

For example, the two strings *messageA* = *ab* and *messageB* = *da*. The virus pattern has only one character, *virusC* = *c*. Create all strings of length 2, the lengths of *a* and *b*, that do not have the virus pattern within them. There are 24 strings beginning with the letter *a* that are greater than or equal to *messageA*: *ab, ad, ae, af, ..., az* and another 25 beginning with the letter *b*: *ba, bb, bd, be, bf, ...*. No strings can begin with the letter *c*, and those beginning with the letter *d* are limited to *da*, which equals *messageB*. In total, there are $24 + 25 + 1 = 50$ valid message strings, and $50\%(10^9+7) = 50$.

Function Description

Complete the function *filteringMessages* in the editor below. The function must return the number of good message strings, modulo (10^9+7) .

filteringMessages has the following parameter(s):

messageA : a string
messageB : a string
virusC: a string

Constraints

- $1 \leq |messageA|, |messageB| \leq 1000$
- $1 \leq |virusC| \leq 100$
- All the strings will contain lowercase English letters only, *ascii[a-z]*.

► Input Format For Custom Testing

▼ Sample Case 0

Sample Input For Custom Testing

b
f
c

Sample Output

4

Explanation

There are 4 strings of length 1 that match the criteria: *b, d, e, f*.

► Sample Case 1

Python 3

Autocomplete Loading...

🌙 ⌚ ? ...

```
1 > #!/bin/python3 ...
10
11 #
12 # Complete the 'filteringMessages' function below.
13 #
14 # The function is expected to return an INTEGER.
15 # The function accepts following parameters:
16 # 1. STRING messageA
17 # 2. STRING messageB
18 # 3. STRING virusC
19 #
20
21 def filteringMessages(messageA, messageB, virusC):
22     # Write your code here
23
24 > if __name__ == '__main__': ...
```

Line: 10 Col: 1

Test Results

Custom Input

Run

▲

Submit Code