Monica is the Head-chef at Javu's restaurant. Javu has **N** kitchen workers, numbered from 1 to N. Each kitchen worker has expertise in exactly one of the following fields:

1. VEG - denoted as **V**
2. NONVEG - denoted as **N**
3. GRILL - denoted as **G**
4. TOAST - denoted as **T**
5. SANDWICH - denoted as **S**
6. BURGER - denoted as **B**
7. PIZZA - denoted as **P**

You'll be given the string **S1** of length **N**, where **ith** character defines the field of expertise of **ith** worker, **1 ≤ i ≤ N**.  
Now, a customer's order contains 3 fields:

| 1st field | 2nd field | 3rd field |
| --- | --- | --- |
| **VEG or NONVEG (V or N)** | **GRILL or TOAST (G or T)** | **SANDWICH or BURGER or PIZZA (S or B or P)** |

You'll be given a string **S2** of length **3**, where **jth** character defines the **jth** field of the order, **1 ≤ j ≤ 3**.  
The first character will either be **V** or **N**. Second character will either be **G** or **T**. Third character will either be **S** or **B** or **P**.

According to the customer order, Monica has to form a team of workers who can prepare the order. Javu has received **M**orders at the moment. You have to find in how many ways can Monica prepare a team of **3** workers for each of the orders.

**Note:** One worker can only work in one team. See the sample test case for clarification.

**Input:**  
First line of input contains **N** - number of workers.  
Second line contains a string **S1** of length **N**, where **ith** character defines the field of expertise of **ith** worker, **1 ≤ i ≤ N**.  
Third lines contains **M** - number of orders.  
Each of the next M lines contain an order query **S2**.

**Output:**   
For each order query, print the **number of ways** Monica can form a team of **3** workers who can prepare the order.   
Since the answer can be very large, print it modulo **1000000007** ( **109 + 7** )

**Constraints:**  
**1 ≤ N ≤ 1000000**   
String **S1** will be of length **N** and will only contain upper case character **[V, N, G, T, S, B, P]**   
**1 ≤ M ≤ 300000**   
String **S2** will be of length **3** and will only contain upper case character **[V, N, G, T, S, B, P]**

**Sample Input**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/broadridge-codebee/problems/head-chef-monica-1/dec264d8-e-sample-input-dec2611.txt?Signature=g4BaBglnKqHXYn%2BFrYSuxLMgfiI%3D&Expires=1430938682&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ)

11

VBVNSSTSBGG

5

VGS

NTS

NGS

VGS

VGB

**Sample Output**

[(Plaintext Link)](https://he-s3.s3.amazonaws.com/media/hackathon/broadridge-codebee/problems/head-chef-monica-1/ded9cc9a-e-sample-output-ded9c91.txt?Signature=2hgLBDvE6jdiBHXb0KG%2FEtX85B8%3D&Expires=1430938683&AWSAccessKeyId=AKIAJLE6MUHDYS3HN6YQ)

12

2

0

1

0

Explanation

Initially we have following number of employees of each field:  
{'B': 2, 'G': 2, 'N': 1, 'P': 0, 'S': 3, 'T': 1, 'V': 2}

For order VGS, Monica can make a team in 2*2*3 = 12 ways.  
After 1st order we decrement 1 employee each from V, G and ; so available number of employees are:  
{'B': 2, 'G': 1, 'N': 1, 'P': 0, 'S': 2, 'T': 1, 'V': 1}  
For order NTS, there are 1*1*2 = 2 ways.  
After 2nd order, available employees are:  
{'B': 2, 'G': 1, 'N': 0, 'P': 0, 'S': 1, 'T': 0, 'V': 1}

For order NGS, since number of employees for 'N' is 0, we cannot prepare a team even though the other employees for 'G' and 'S'. So the answer is zero. Since the team is not formed we don't decrement the number for 'G' and 'S'.  
After 3rd order, available employees are again:  
{'B': 2, 'G': 1, 'N': 0, 'P': 0, 'S': 1, 'T': 0, 'V': 1}  
For order VGS, there is 1*1*1 = 1 way.  
Similarly we can find for order VGB as well.