Digit Shifts

You will be given a **non-negative integer X**. You need to perform **Q** queries on that integer. Each query will consist of a single decimal digit **D**. After every query, you need to move all occurrences of digit **D** in the integer to the end, while keeping the relative position of every other digit intact.

For example, suppose **X = 123123**, and suppose **Q = 3**.

1. For the first query, **D = 1**, then after the digits are shifted **X = 232311**.
2. For the second query, **D = 2**, then after the digits are shifted **X = 331122**.
3. For the third query, **D = 3**, then after the digits are shifted **X = 112233**.

After every query, you need to output the value of the integer **X**. Since it can be really large, output it modulo **1000000007 (10^9 + 7)**. Please note that if at any point after a query, **X** contains leading zeros, then the leading zeros should be **discarded**. Therefore, if **X = 2022** and if **D = 2**, then after the query, **X** will become **222**.

Input

The first line will contain a single integer **T (1 ≤ T ≤ 20)**. Each starts with a single line, which will contain the integer **X**. Then, in the next line there is a single integer **Q (1 ≤ Q ≤ 10^5)**, denoting the number of queries. Each of the next **Q** lines denotes a query, containing a decimal digit **D (0 ≤ D ≤ 9)**. You can safely assume that **X** won't contain leading zeros initially and that **X** will never have more than **10^5** digits.

Output

For each test case, first print the case number in a single line like "**Case V:**". Then for each query, output the value of **X** modulo **1000000007**. Refer to the sample I/O for more clarity.

Sample Input

```
1
123123
3
1
2
3
```

Sample Output

```
Case 1:
232311
```

331122
112233