Killjee and k-th letter | Problem Code: **KILLKTH**

Killjee is trying to unlock a treasure. The key to the treasure is encrypted using a string **S** and **Q** queries. In each query, you need to find the **K**-th letter of a hidden string which is formed from the string **S**.

To form the hidden string, you should sort all substrings of **S** in lexicographical order and concatenate them. For example, if **S** = "abc", the hidden string would be "aababcbbcc". (See the sample explanation for details.)

In each query, the value of **K** is encoded in the following way:

- You're given two integers **P** and **M**.
- Let's define **G** as the sum of ASCII values of answers to all previous queries (therefore, **G** = 0 for the first query).
- The value of **K** for the current query is **(P · G) % M + 1**, where **%** denotes the modulo operator.

Input

- The first line of the input contains a single string **S**.
- The second line contains a single integer **Q**.
- **Q** lines follow. Each of these lines contains two space-separated integers **P** and **M**.

Output

For each query, print a single line containing one character — the **K**-th letter of the hidden string.

Constraints

- $1 \leq |S| \leq 2 \cdot 10^5$
- $1 \leq Q \leq 2 \cdot 10^5$
- $1 \leq K,M \leq$ length of hidden string
- $1 \leq P \leq 10^9$
- **S** will consist only of lowercase English letters

Subtasks

Subtask #1 (5 points): $1 \leq |S| \leq 50$

Subtask #2 (15 points):

- $1 \leq |S| \leq 2000$

- 1 ≤ **Q** ≤ 25000

Subtask #3 (20 points): 1 ≤ **Q** ≤ 10

Subtask #4 (60 points): original constraints

Example

```
Input:

abc
3
1 1
2 3
5 6

Output:

a
b
a
```

Explanation

The substrings of **S** are "a", "b", "c", "ab", "abc", "bc". The lexicographical order of these strings is "a", "ab", "abc", "b", "bc", "c", so the hidden string is "a"+"ab"+"abc"+"b"+"bc"+"c" = "aababcbbcc".

For query 1, **G** = 0, so **K = (P · G) % M + 1 = (1 · 0) % 1 + 1 = 1**. The 1-st character of the hidden string is **'a'**. We add the ASCII value of 'a' (97) to **G**.

For query 2, **G** = 97, so **K = (2 · 97) % 3 + 1 = 3**. The 3-rd character of the hidden string is **'b'**. We add the ASCII value of 'b' (98) to **G**.

For query 3, **G** = 195, so **K = (5 · 195) % 6 + 1 = 4**. The 4-th character of the hidden string is **'a'**. We add the ASCII value of 'a' (97) to **G**.

Author: 6★ killjee (/users/killjee)

Date Added: 8-12-2017

Time Limit: 1 secs

Source Limit: 50000 Bytes