

E. Strange Sorting

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

How many specific orders do you know? Ascending order, descending order, order of ascending length, order of ascending polar angle... Let's have a look at another specific order: d -sorting. This sorting is applied to the strings of length at least d , where d is some positive integer. The characters of the string are sorted in following manner: first come all the 0-th characters of the initial string, then the 1-st ones, then the 2-nd ones and so on, in the end go all the $(d - 1)$ -th characters of the initial string. By the i -th characters we mean all the character whose positions are exactly i modulo d . If two characters stand on the positions with the same remainder of integer division by d , their relative order after the sorting shouldn't be changed. The string is zero-indexed. For example, for string 'qwerty':

Its 1-sorting is the string 'qwerty' (all characters stand on 0 positions),

Its 2-sorting is the string 'qetwry' (characters 'q', 'e' and 't' stand on 0 positions and characters 'w', 'r' and 'y' are on 1 positions),

Its 3-sorting is the string 'qrwtey' (characters 'q' and 'r' stand on 0 positions, characters 'w' and 't' stand on 1 positions and characters 'e' and 'y' stand on 2 positions),

Its 4-sorting is the string 'qtwyer',

Its 5-sorting is the string 'qywert'.

You are given string S of length n and m shuffling operations of this string. Each shuffling operation accepts two integer arguments k and d and transforms string S as follows. For each i from 0 to $n - k$ in the increasing order we apply the operation of d -sorting to the substring $S[i..i + k - 1]$. Here $S[a..b]$ represents a substring that consists of characters on positions from a to b inclusive.

After each shuffling operation you need to print string S .

Input

The first line of the input contains a non-empty string S of length n , consisting of lowercase and uppercase English letters and digits from 0 to 9.

The second line of the input contains integer m – the number of shuffling operations ($1 \leq m \cdot n \leq 10^6$).

Following m lines contain the descriptions of the operations consisting of two integers k and d ($1 \leq d \leq k \leq n$).

Output

After each operation print the current state of string S .

Examples

input
qwerty 3 4 2 6 3 5 2
output
qertwy qtewry qetyrw

Note

Here is detailed explanation of the sample. The first modification is executed with arguments $k = 4$, $d = 2$. That means that you need to apply 2-sorting for each substring of length 4 one by one moving from the left to the right. The string will transform in the following manner:

qwerty \rightarrow **qewr**ty \rightarrow q**erwt**y \rightarrow qert**wy**

Thus, string S equals 'qertwy' at the end of first query.

The second modification is executed with arguments $k = 6$, $d = 3$. As a result of this operation the whole string S is replaced by its 3-sorting:

qertwy \rightarrow **qtewry**

The third modification is executed with arguments $k = 5$, $d = 2$.

qtewry \rightarrow **qertwy** \rightarrow qet**yrw**