

Organizing CS contests didn't prove very lucrative for Mirko, so he has opened an ice cream and pastry shop. The business was flourishing until, one day, the European Union health inspection decided to pay him a visit.

A new directive specifies **M** banned ingredients which cannot appear in food even in trace amounts. Each ingredient has a serial number consisting of digits 0 through 9. The declaration on each food package lists all the serial numbers of ingredients contained in the respective food item.

Mirko must check whether any of his products has a banned ingredient serial number listed on its declaration. However, Mirko, being inept and reckless as always, decided to concatenate all the serial numbers into one long number with length **N** believing that it will make his job easier. He has borrowed a robot from his friend Slavko. The robot is programmed to check whether a serial number **A** contains another serial number **B** as a substring. Let us denote the length of **B** by **L**. The robot carries out search as follows:

- First, it compares the segment of **A** from position 1 to position **L**, digit by digit, with the digits in **B**. Comparison is stopped when a differing digit is found or when the segments are determined to be equal. If the segments are equal, the search is stopped and the match reported.
- If the segments are not equal, the procedure above is repeated with the segment from 2 to **L**+1. If those segments aren't equal either, the search continues with segments 3 to **L**+2, 4 to **L**+3 etc.
- If the robot doesn't have a sufficient number of digits to obtain a full segment of length **L** (for example, starting at character 5 in a serial number with length 8, a segment with length 6 is needed), it will pad the number with '#' signs. For example, the segment of "563232" from position 4 to position 10 is "232#####".
- If the robot reaches the end of the serial number (having tried out all **N** segments) without having found **B**, the absence of match is reported.

The robot takes one second for each comparison between two digits, and Slavko charges Mirko one dollar per second for using the robot.

Help Mirko determine how much money he'll have to pay Slavko for pattern matching!

INPUT

The first line of input contains the positive integer **N** ($1 \leq \mathbf{N} \leq 100\,000$), the length of the long serial number.

The second line of input contains **N** digits from 0 to 9, the long serial number.

The third line of input contains the positive integer **M** ($1 \leq \mathbf{M} \leq 50\,000$), the number of banned ingredients.

Each of the following **M** rows contains a single banned serial number,

A banned serial number will not exceed 100 000 digits in length.

The total length of all banned serial numbers will not exceed 3 000 000 digits.

OUTPUT

Output **M** integers, one per line. Line *i* must contain the dollar amount that Mirko needs to pay Slavko for the search for ingredient serial number *i*.

SCORING

In test data worth at least 20% of total points, the following constraints hold:

- $1 \leq N \leq 1000$
- $1 \leq M \leq 500$
- length of a single banned ingredient serial number doesn't exceed 1000

SAMPLE TESTS

input	input	input
7	10	3
1090901	5821052680	001
4	4	1
87650	210526	11
0901	2105	
109	582	
090	105268	
output	output	output
7	8	4
10	6	
3	3	
4	9	

Clarification of the first example:

First serial number: the robot finds differing first digits for every segment – a total of 7 comparisons.

Second serial number: tries first position, finding difference immediately (1 comparison). Tries second position, finding difference on the fourth digit (4 comparisons). Tries third position, finding difference immediately (1 comparison). Tries fourth position, finding a match (4 comparisons). Total: 10 comparisons.

Third serial number: finds match immediately (3 comparisons).

Fourth serial number: finds match at second position ($1 + 3 = 4$ comparisons).

Clarification of the third example:

The robot compares the serial number '11' in order with segments '00' (1 comparison), '01' (1 comparison) and '1#' (2 comparisons). Total: 4 comparisons.