

Problem A. Coding

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 Mebibytes

The International Coding Procedure Committee (ICPC) deals with data manipulation tasks. One of its latest achievements is the algebraic convolution method (ACM) that allows for rapid composition of cyclic shifts in strings.

In an attempt to hire more young talented programmers, ICPC has decided to conduct a contest and present the Top Decoder award.

The participants are given a ciphertext Y and a sequence of cyclic shifts that had produced Y from string Z , the shift with parameters (i, j, k) applies to the substring $Z[i..j]$ (from i -th to j -th character, inclusive) and cyclicly rotates it to the right k times. String characters are numbered starting from one. Given the above information, your task is to guess the initial cleartext X .

Input

The first line contains the ciphertext, which is a nonempty string consisting of N lowercase English letters ($1 \leq N \leq 5 \cdot 10^4$). The second line contains the number of shifts M ($1 \leq M \leq 5 \cdot 10^4$).

The following M lines describe the sequence of cyclic shifts (in the order of their application to the cleartext). Each shift is described by three parameters i, j, k ($1 \leq i < j \leq N$, $1 \leq k \leq j - i$).

Output

Output the only line containing the cleartext.

Example

standard input	standard output
logoduck 3 1 3 1 4 5 1 1 4 1	goodluck

Problem B. DFT for Dummies

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 megabytes

The normalized discrete Fourier transform (DFT) of a sequence of complex numbers x_0, x_1, \dots, x_{n-1} is a sequence y_0, y_1, \dots, y_{n-1} which is defined according to the following formula:

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \cdot \exp\left(\frac{2\pi i j k}{n}\right).$$

Here, i stands for imaginary unit.

The Euclidean norm of a sequence of complex numbers x_0, x_1, \dots, x_{n-1} is defined according to the following formula:

$$\|x\| = \sqrt{|x_0|^2 + |x_1|^2 + \dots + |x_{n-1}|^2}.$$

You are given an array of n complex numbers. You are asked to process several queries. Each query is described by a segment $[l, r]$ and consists of three steps:

1. Replace the segment $[l, r]$ with its DFT.
2. Output the squared euclidean norm of the segment $[l, r]$.
3. Replace the segment $[l, r]$ with its DFT again.

By definition, $\exp(i \cdot \varphi) = \cos \varphi + i \cdot \sin \varphi$, where φ is any real number. The absolute value $|z|$ of a complex number $z = a + i \cdot b$ equals to $\sqrt{a^2 + b^2}$.

Input

The first line of input contains an integer number n , the length of the array ($1 \leq n \leq 10^5$). The next line contains n integer numbers a_i , the initial values in the array ($|a_i| \leq 10^6$). Note that, after the transform, the elements become complex. The following line contains an integer number q , the number of queries ($1 \leq q \leq 10^5$). Each of the next q lines describes a single query. Each description consists of two integer numbers l and r separated by a space ($1 \leq l \leq r \leq n$).

Output

For each query, output the desired sum, rounded **up** to the nearest integer number.

Examples

standard input	standard output
2	1
1 0	0
3	1
1 1	
2 2	
1 2	

Problem C. K-th Maximum

Input file: *standard output*
Output file: *standard input*
Time limit: 1 second
Memory limit: 256 megabytes

Implement a data structure that allows to add and delete elements, and also answers the k -th maximum element queries.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 100\,000$) — the number of queries to perform. Then follow n lines containing the queries. Every query is given by two numbers c_i and k_i ($|k_i| \leq 10^9$) — the type and the value of the corresponding query. The possible types are:

- +1 (or just 1): add an element with the key equal to k_i .
- 0: find and print k_i -th maximum.
- -1: delete an element with the key equal to k_i .

It is guaranteed, that during the workflow the structure will never be asked to store two elements with equal keys or to delete an element that is not present in the structure. It's guaranteed that the answer for every query of k_i -th maximum exists.

Output

For every query of the 0 type print the line containing a single integer — k_i -th maximum.

Examples

standard output	standard input
11	7
+1 5	5
1 3	3
+1 7	10
0 1	7
0 2	3
0 3	
-1 5	
+1 10	
0 1	
0 2	
0 3	

Problem D. Luxury burrow

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 megabytes

Hobbit Bilbo is very tired from his adventures and decided to build a new burrow. The hill, where he plans to buy some land, has a shape of a rectangle and it can be represented as a table consisting of N rows and M columns (rows and columns are numbered starting from 1, where 1 corresponds to the topmost row and the leftmost column). Each cell of the table corresponds to a 1×1 square piece. Hobbits like simple shapes, that's why Bilbo intends to buy a plot of land of a rectangular shape, with sides parallel to the sides of the hill. In other words he chooses x_1, x_2, y_1, y_2 ($x_1 \leq x_2, y_1 \leq y_2$) and buys all cells (x, y) , such that $(x_1 \leq x \leq x_2)$ and $(y_1 \leq y \leq y_2)$.

After all adventures he took part in, Bilbo is very rich and is not worried about the total price. However, he thinks a lot about his reputation and thus the price of the cheapest cell he will buy must be as high as possible. He also wants his new burrow to be enough to fit all of his furniture, that means the square of a rectangular plot he will buy has to have an area greater than or equal to K . If there are multiple possible plots, Bilbo chooses the one with the largest area.

Bilbo asks you to find the best plot of land for him.

Input

The first line of input file contains three positive integers N, M and K , divided by exactly one space — the number of rows, the number of columns and the minimal possible square of land plot Bilbo can live on, respectively. Each of the following N lines contains M numbers — the prices of the corresponding cells. Hence, the number j in the line $i + 1$, corresponds to the cell (i, j) . K does not exceed the total number of cells in the table, all prices belong to range from 1 to 10^9 , inclusive.

Output

As the answer, output two numbers separated by space. First goes the price of the cheapest cell in the resulting rectangular plot. Second is the total square of this plot.

Examples

standard input	standard output
3 3 3 1 1 1 1 2 2 1 2 2	2 4
1 10 5 4 3 2 5 10 7 6 5 1 100	5 5
3 5 2 5 7 5 5 5 8 5 5 7 5 8 5 8 8 8	8 3

Problem E. Period

Input file: *standard input*
Output file: *standard output*
Time limit: 10 seconds
Memory limit: 512 megabytes

The period of a string s is a string t of minimum possible length, such that s can be represented as a concatenation of some copies of t . For example, the period of the string **abaaba** is the string **aba**, and the period of the string **aaaaa** is the string **a**, while the period of the string **abaabaa** is the string **abaabaa** itself.

You are given the string s . Perform three types of queries:

- Insert the symbol c in the string s ;
- Delete some symbol from the string s ;
- Print the period of the current string s .

Input

The first line of the input contains a single integer n ($1 \leq n \leq 500\,000$) — the length of the string s .

The second line of the input contains the string s consisting of n small English letters.

The third line of the input contains a single integer q ($1 \leq q \leq 500\,000$) — the number of queries to be processed.

Then follow q lines with queries descriptions.

- The line $\langle + i c \rangle$ ($0 \leq i \leq |s|$) correspond to the query «insert symbol c to the i -th position». For example, if $i = |s|$, then the symbol is added to the end of the string, otherwise it is inserted before the current i -th symbol (positions are numbered starting from zero). All symbols are guaranteed to be small English letters.
- The line $\langle - i \rangle$ ($0 \leq i \leq |s| - 1$) corresponds to the query «delete i -th symbol of the string» (again, symbol are numbered starting from 0).
- The line $\langle ? \rangle$ corresponds to the query «find the current period of the string».

There will be no more than 100 000 queries of the third type in each test.

It's guaranteed that s is a non-empty string at any moment of time.

Output

For every $\langle ? \rangle$ query print the single integer on a separate line — the period of the string s .

Examples

standard input	standard output
6	3
abaaba	5
7	2
?	5
- 2	
?	
- 4	
?	
+ 0 a	
?	

Problem F. Urns and Balls

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 megabytes

Consider n different urns and n different balls. Initially, there is one ball in each urn.

There is a special device designed to move the balls. Using this device is simple. First, you choose some range of consecutive urns. The device then lifts all the balls from these urns. After that, you specify the destination which is another range of urns having the same length. The device then moves the lifted balls and places them in the destination urns. Each urn can contain any number of balls.

Given a sequence of movements for this device, find where will each of the balls be placed after all these movements.

Input

First line contains two integers n and m , the number of urns and the number of movements ($1 \leq n \leq 100\,000$, $1 \leq m \leq 50\,000$). Each of the next m lines contain three integers $count_i$, $from_i$ and to_i which mean that the device simultaneously moves all balls from urn $from_i$ to urn to_i , all balls from $from_i + 1$ to urn $to_i + 1$, ..., all balls from urn $from_i + count_i - 1$ to urn $to_i + count_i - 1$ ($1 \leq count_i, from_i, to_i \leq n$, $\max(from_i, to_i) + count_i \leq n + 1$).

Output

Output exactly n numbers from 1 to n : the final positions of all balls. The first number is the final position of the ball which was initially in urn 1, the second number is the final position of the ball from urn 2, and so on.

Examples

standard input	standard output
2 3 1 1 2 1 2 1 1 2 1	1 1
10 3 1 9 2 3 7 3 8 3 1	1 2 1 2 3 4 1 2 2 8