Problem A: The next

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Make a data structure that manages a set of integers S and is able to perform some operations:

- $add(i)$ - add the number i to S (if $i \in S$ already, S is kept unchanged)
- $next(i)$ - output the minimum element of the set not less than i. If there is no such element, output -1

Input

Initially S is empty. The input consists of

- one line containing n – the number of queries ($1 \leq n \leq 3 \cdot 10^5$)
- n lines containing the queries: Every query adheres to one of the formats "? i" or "+ i". Queries of the form "? i" always represent the $next(i)$ operation. Queries of the form "+ i" need further context: if such a query is given as first query or after another "+" query then it represents the $add(i)$ operation; if it stays after a "?" query with the answer y, it represents the operation $add((i + y) \mod 10^9)$. The parameter i is always in the range between 0 and 10^9.

Output

For every "?" query output the corresponding result.

Sample input and output

Input	Output
6 + 1 + 3 + 3 ? 2 + 1 ? 4	3 4

Problem B: Outlier analysis

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

After collecting integer-valued data points $a_1, ..., a_N$ corresponding to positions $1, ..., N$ on the number line, you tried to find a good linear fit for this data. You already determined the fit's parameters. However, you mistrust classical error metrics, so you want to examine the quality of the fit by hand. For that, you need a program that answers queries of the following format: Given positions ℓ and r and a value d, find the smallest deviation of any of the data points $a_\ell, ..., a_r$ from the linear fit that is an outlier in the sense that it is strictly greater than d. In other words, find

$$\min\{|a_i - f(i)| \ : \ \ell \leq i \leq r, |a_i - f(i)| > d\}.$$

Input

The input consists of

- one line containing N ($1 \leq N \leq 10^5$) – the number of data points
- one line containing integers m and b ($|m| \leq 10^3, |b| \leq 10^8$) – denoting that the linear fit is $f(t) = m \cdot t + b$
- one line containing N integers $a_1, ..., a_N$ ($|a_i| \leq 10^8$) – the data points
- one line containing Q ($1 \leq Q \leq 10^5$) – the number of queries
- Q lines each containing integers l_i, r_i, d_i ($1 \leq l_i, r_i \leq N, 0 \leq d_i \leq 10^8$) – the parameters of a query.

Output

Correctly answer each query in a separate line. If for some query there is no outlier, output -1 for that query.

Sample input and output

Input	Output
10 0 5 45 754 -123 54 1 -63 0 71 -3 -19 4 1 10 1000 6 10 5 1 5 5 5 7 4	-1 8 40 5

Problem C: Cartography

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

MapWorld is in trouble: For many decades, the small company has relied on selling the same few photo-realistic maps of the world, but throughout the steep rise of internet's popularity (and more specifically the trend to put every conceivable picture online), their sales have been declining almost as steeply. By now, the company's situation has gotten so dire that the owner sees only one way out: They need to join a recent trend and modify their online platform is such a way that people can buy customized maps, which will then be printed and sent to them.

They already have a number of high-resolution digital maps, from each of which customers should then be able to choose specific rectangular sections for printing. However, there still is a problem with that: *MapWorld*'s profit margins are relatively small and the colors they use for printing vary widely in cost, so they need to take into account exactly how much of every color is used for each of the customized maps. Of course they can't just count the respective pixels each time, instead they will need a program that quickly tells them exactly what amount printing any given rectangular section of a map would cost them (you may assume that each pixel of a map section's digital representation will be printed in the same size, regardless of the customized map's format). And there's another difficulty: As the maps they sell always need to be up-to-date, they will sometimes have to change individual pixels in their digital representations (for example due to deserts growing, forests burning down or the rising sea level turning coastal regions oceanic ones).

The program will be given one of the digital maps and should then be able to quickly process queries either asking for the printing costs of a rectangular region or telling you to change a specific pixel.

Input

There are four different types of pixels:

symbol	**type**	**color**
w	water	blue
f	forest	green
i	ice	white
d	desert	yellow

The input consists of

- one line containing N and M $(1 \leq N, M \leq 1000)$ – the numbers of rows and columns of pixels in the digital map
- one line containing integers b, g, w and y $(1 \leq b, g, w, y \leq 100)$ – the costs of printing a pixel colored blue, green, white or yellow, respectively
- N lines each containing M characters (without whitespace between them), representing the digital map
- one line containing Q $(1 \leq Q \leq 10^4)$ – the number of queries
- Q lines containing the queries, each of them in one of the following formats:
  - "R r_1 r_2 c_1 c_2" $(0 \leq r_1 \leq r_2 < N, 0 \leq c_1 \leq c_2 < M)$, denoting a request for the cost of printing the map consisting of all pixels with coordinates (r, c) for which $r_1 \leq r \leq r_2$ and $c_1 \leq c \leq c_2$.
  - "C r c t" $(0 \leq r < N, 0 \leq c < M, t \in \{\text{w, f, i, d}\})$, denoting that the pixel with coordinates (r, c) changed to type t.

Output

Correctly answer each request-query in a separate line.

Sample input and output

Input	Output
5 10 40 20 10 20 wwwwwwwddw wwwiiwwwdw wwiiiwwwww wwwwwwfffw wwwwwfffww 4 R 0 4 0 9 R 0 1 0 2 C 1 2 i R 0 1 0 2	1670 240 210

Problem D: Go to the front

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Captain Joe likes commanding his crew. His favorite command is ”Go to the front”. He lines up his crew and roars a sequence of commands. Every command is like ”Crewmen from l_i to r_i - go to the front!”.

Initially the crewmen are numerated from 1 to n, from left to right. By the command ”Crewmen from l_i to r_i - go to the front!” the men standing at positions l_i to r_i inclusively are ordered to go to the front of the line, while keeping their initial numbers and internal order.

For instance, assume that at some point the crewmen stand in the order 2, 3, 6, 1, 5, 4. Then the command “Crewmen from 2 to 4 - go to the front!” would change the order to 3, 6, 1, 2, 5, 4.

Find out the final order of Captain Joe's crew after the given sequence of commands.

Input

The input consists of

- one line containing integers n and m $(2 \leq n \leq 10^5, 1 \leq m \leq 10^5)$ – the amounts of crewmen and commands of the captain
- m lines giving the commands, with the i-th line containing integers ℓ_i and r_i $(1 \leq \ell_i \leq r_i \leq n)$, denoting the i-th command is ”Crewmen from ℓ_i to r_i - go to the front!”.

Output

Output n integers – the final order of the crewmen after obeying the commands.

Sample input and output

Input	Output
6 3 2 4 3 5 2 2	1 4 5 2 3 6

Problem E: A treap

Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Given N pairs of numbers (a_i, b_i). You are to interpret each pair as a vertex and try to to build a treap out of them, where the a_i are the keys of a binary search tree and the b_i those of a heap.

Input

The input consists of

- one line containing N $(1 \leq N \leq 5 \cdot 10^4)$ – the number of pairs
- N lines providing the pairs, with the i-th line containing a_i and b_i $(|a_i|, |b_i| \leq 3 \cdot 10^4)$, denoting that vertex i is the pair (a_i, b_i). It is guaranteed that $a_i \neq a_j$ and $b_i \neq b_j$ for all $i \neq j$.

Output

If it's possible to build a treap as specified above, output `YES`, otherwise output `NO`. In the first case you must also give a description of the treap: To do so, output N more lines, with the i-th line describing the position of vertex i in the treap using three other vertices: its parent, its left son and its right son. When describing a root or leaf, use zeros to indicate the absence of parents/children as in the example below.

Sample input and output

Input	Output
7 5 4 2 2 3 9 0 5 1 3 6 6 4 11	YES 2 3 6 0 5 1 1 0 7 5 0 0 2 4 0 1 0 0 3 0 0