

Problem A. Naïve RMQ

Input filename: `stupid.rmqs.in`
Output filename: `stupid.rmqs.out`
Time limit: 2 seconds
Memory limit: 256 Mb

You are given an array a of n elements, and m queries of the form (l, r) . For each query, find the minimal element of a with index from l to r inclusive.

Input file format

The first line contains the length $1 \leq n \leq 1\,000$ of the array a , The second line contains n elements of the array, $-10^9 \leq a_i \leq 10^9$. The next line contains the number $1 \leq m \leq 1\,000$ of queries. The last m lines contain queries in the form $1 \leq l, r \leq n$.

Output file format

For every query print the answer on a new line.

Sample tests

stupid.rmqs.in	stupid.rmqs.out
5 1 -2 3 4 1 3 1 3 3 4 1 5	-2 3 -2
2 -100000 -1111111 2 1 1 1 2	-100000 -1111111

Problem B. Range Variation Query

Input filename: `rvq.in`
Output filename: `rvq.out`
Time limit: 2 seconds
Memory limit: 256 Mb

Initially there is a sequence a_n given by the recurrence $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

You need to process multiple queries of the following types:

- find the variation (difference between max and min) on the segment $a_i, a_{i+1}, \dots, a_{j-1}, a_j$;
- assign the value j to the element a_i .

Input file format

The first line contains the number $k \leq 100\,000$ of queries. The following k lines contain queries, one per line. The i -th query is described by two numbers x_i, y_i .

If $x_i > 0$ then you need to find the variation of $a_{x_i} \dots a_{y_i}$, and $1 \leq x_i \leq y_i \leq 100\,000$.

If $x_i < 0$ then you need to assign the value y_i to the element $a_{|x_i|}$, and $-100\,000 \leq x_i \leq -1$ and $|y_i| \leq 100\,000$.

Output file format

For every query of the first type, print one line containing the variation on the corresponding segment.

Sample tests

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

Problem C. Sum

Input filename: `sum.in`
Output filename: `sum.out`
Time limit: 2 seconds
Memory limit: 256 Mb

Given an array of N elements, you will need to answer range sum queries.

Input file format

The first line of the input contains two integers $1 \leq N \leq 100\,000$ and $0 \leq K \leq 100\,000$ denoting the size of the array and number of queries. The following K lines contain queries:

- **A** $l\ r\ x$ — assign the value x to elements with indices from l to r inclusive ($1 \leq l \leq r \leq n$, $0 \leq x \leq 10^9$);
- **Q** $l\ r$ — find the sum of elements with indices from l to r inclusive ($1 \leq l \leq r \leq n$).

Output file format

For every query of the type **Q** $l\ r$ print the sum on the corresponding segment.

Sample tests

sum.in	sum.out
5 9	3
A 2 3 2	2
A 3 5 1	3
A 4 5 2	4
Q 1 3	2
Q 2 2	7
Q 3 4	
Q 4 5	
Q 5 5	
Q 1 5	

Problem D. Sum 2

Input filename: `sum2.in`
Output filename: `sum2.out`
Time limit: 2 seconds
Memory limit: 256 Mb

Input file format

Given an array of n elements, you will need to answer m range sum queries.

Output file format

The first line contains the length $1 \leq n \leq 100\,000$ of the array a , The second line contains n elements of the array, $-10^9 \leq a_i \leq 10^9$. The next line contains the number $1 \leq m \leq 100\,000$ of queries. The last m lines contain queries in the form $1 \leq l, r \leq n$.

Sample tests

sum2.in	sum2.out
5 1 2 3 4 5 3 1 5 1 1 4 5	15 1 9
6 -1 1 -1 1 -1 1 2 1 6 3 3	0 -1

Problem E. Sparse Tables

Input filename: `sparse.in`
Output filename: `sparse.out`
Time limit: 2 seconds
Memory limit: 256 Mb

Given an array of n integers, you will need to answer multiple range minimum query (between indices u and v inclusive).

Input file format

First line of the input file contains three integers $1 \leq n \leq 10^5$, $1 \leq m \leq 10^7$ and $0 \leq a_1 \leq 16\,714\,589$ — the size of the array, the number of queries and the first element of the array. The second line contains two integers $1 \leq u_1, v_1 \leq n$ — the first query.

Elements of the array are defined via the following recurrence:

$$a_{i+1} = (23 \cdot a_i + 21\,563) \bmod 16\,714\,589.$$

For example, for $n = 10$ and $a_1 = 12\,345$, you will get the following array: $a = (12\,345, 305\,498, 7\,048\,017, 11\,694\,653, 1\,565\,158, 2\,591\,019, 9\,471\,233, 570\,265, 13\,137\,658, 1\,325\,095)$.

The queries are generated in the following way:

$$\begin{aligned} u_{i+1} &= ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1, \\ v_{i+1} &= ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1, \end{aligned}$$

where ans_i is the answer to i -th query.

Output file format

Output three integers: u_m , v_m , and ans_m (the last query and its answer).

Sample tests

<code>sparse.in</code>	<code>sparse.out</code>
10 8 12345 3 9	5 3 1565158

Problem F. RMQ

Input filename: `rmq.in`
Output filename: `rmq.out`
Time limit: 2 seconds
Memory limit: 64 Mb

The *Giggle* company is about to open a new office in Sudislavl', and you are invited to the interview. You need to create a data structure that represents an array of integers. Initially the array is empty. You need to support two types of queries:

- `? i j` — find minimum between i -th and j -th elements, inclusive;
- `+ i x` — insert a value x after the i -th element of the array. If $i = 0$ then element is appended to the front of the array.

Input file format

The first line contains the single integer $1 \leq n \leq 200\,000$ denoting the number of queries. The following n lines describe the operations. All insertion operations are guaranteed to be valid. All elements of the array do not exceed 10^9 in absolute value.

Output file format

For every `? i j` query print its answer on a separate line.

Sample tests

rmq.in	rmq.out
8	4
+ 0 5	3
+ 1 3	1
+ 1 4	
? 1 2	
+ 0 2	
? 2 4	
+ 4 1	
? 3 5	

The table below shows how the state of the array changes under these operations:

Operation	Array state after
<i>initially</i>	<i>empty</i>
+ 0 5	5
+ 1 3	5, 3
+ 1 4	5, 4, 3
+ 0 2	2, 5, 4, 3
+ 4 1	2, 5, 4, 3, 1