Problem A. Simple sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

You have to answer requests "sum of numbers on the segment".

Array doesn't change. There're many requests. You should answer on each in $\mathcal{O}(1)$ time.

Input

First line contains four integers: n, x, y and a_0 — length of the array and numbers which generates array a: $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$.

Next line contains four integers: m, z, t and b_0 — number of requests and numbers which generates array b: $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$.

Array c is generating in the following way: $c_i = b_i \bmod n$.

Request number i is to find sum on segment from $\min(c_{2i}, c_{2i+1})$ to $\max(c_{2i}, c_{2i+1})$ in the array a.

$1 \le n \le 10^7$, $0 \le m \le 10^7$. All other number are from 0 to 2^{16}. t can also be equal to -1.

Output

Output sum of all sums.

Example

standard input	standard output

`3 1 2 3`
`3 1 -1 4`
	`23`

Note

$a = \{3, 5, 7\}, b = \{4, 3, 2, 1, 0, 2^{30} - 1\}, c = \{1, 0, 2, 1, 0, 0\}$,

Requests $= \{[0, 1], [1, 2], [0, 0]\}$, sums $= \{8, 12, 3\}$.

Problem B. RMQ

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Input

First line contains integer n — length of array ($1 \leq n \leq 500000$). Second line contains n integers a_i — array elements. After that requests description is given. Number of requests doesn't exceed 1 000 000. Every line contains on of the following requests:

- `set` i x — set $a[i]$ value x ($|x| \leq 10^9$).
- `min` i j — output minimum in the array from position i to j, it's guaranteed that ($1 \leq i \leq j \leq n$).

Output

Output answer on every request of the second type.

Example

standard input	standard output
5 1 2 3 4 5 min 2 5 min 1 5 min 1 4 min 2 4 set 1 10 set 2 3 set 5 2 min 2 5 min 1 5 min 1 4 min 2 4	2 1 1 2 2 2 3 3

Problem C. RSQ

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

Input

The first line contains the number n — the size of the array. ($1 \le n \le 500\,000$) The second line contains n numbers a_i — elements of the array. The following is the description of operations, their number doesn't exceed 1 000 000. Each line contains one of the following operation:

- `set` i x — set value x to $a[i]$.
- `sum` i j — print the sum of the elements in the array in the range from i to j, it's guaranteed, that ($1 \le i \le j \le n$).

All numbers in the input file and the results of all operations do not exceed 10^{18} in absolute value.

Output

Print successively all the operations `sum`. Follow the output file format from the example.

Example

standard input	standard output
5 1 2 3 4 5 sum 2 5 sum 1 5 sum 1 4 sum 2 4 set 1 10 set 2 3 set 5 2 sum 2 5 sum 1 5 sum 1 4 sum 2 4	14 15 10 9 12 22 20 10

Problem D. Sign alternation

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

Implement data structure of n elements $a_1, a_2 \dots a_n$, with the following operations:

- assign the element a_i value j;
- find alternating sign sum in the range from l to r inclusive ($a_l - a_{l+1} + a_{l+2} - \dots \pm a_r$).

Input

The first line of the input file contains a natural number n ($1 \le n \le 10^5$) — the length of the array. The second line contains the initial values of the elements (non-negative integers not exceeding 10^4).

The third line contains a positive integer m ($1 \le m \le 10^5$) — the number of operations. The following m lines contain operations:

- the operation of the first type is given by three numbers `0 i j` ($1 \le i \le n$, $1 \le j \le 10^4$).
- an operation of the second type is given by three numbers `1 l r` ($1 \le l \le r \le n$).

Output

For each operation of the second type, print on a separate line the corresponding sign alternating sum.

Example

standard input	standard output

3
1 2 3
5
1 1 2
1 1 3
1 2 3
0 2 1
1 1 3
	-1
	2
	-1
	3

Problem E. RMQ2

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

Input

The first line contains the number n — the size of the array. ($1 \leq n \leq 10^5$) The second line contains n numbers a_i — elements of the array. The following is a description of the operations, their number does not exceed $2 \cdot 10^5$. Each line contains one of the following operations:

- `set` i j x — set value x to all $a[k]$, $i \leq k \leq j$.
- `add` i j x — increase $a[k]$, $i \leq k \leq j$ by x.
- `min` i j — print the value of the minimum element in the array between i and j, it is guaranteed that ($1 \leq i \leq j \leq n$).

All numbers in the input file and the results of all operations do not exceed 10^{18} in absolute value.

Output

Print successively all the operations `sum`. Follow the output file format from the example.

Example

standard input	standard output
5	2
1 2 3 4 5	1
min 2 5	1
min 1 5	2
min 1 4	5
min 2 4	5
set 1 3 10	8
add 2 4 4	8
min 2 5	
min 1 5	
min 1 4	
min 2 4	

Problem F. Backward RMQ

Input file:	`rmq.in`
Output file:	`rmq.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Consider array $a[1..n]$. Let $Q(i, j)$ be a minimum among numbers $a[i], \ldots, a[j]$. You are given multiple triples i, j and $Q(i, j)$. Restore the array.

Input

First line contains two integers n and m — length of the array and number of requests ($1 \le n, m \le 100\,000$). Next m lines contains three integers i, j and q, meaning that $Q(i, j) = q$ ($1 \le i \le j \le n$, $-2^{31} \le q \le 2^{31} - 1$).

Output

If there's no suitable array exist, output "`inconsistent`".

Otherwise, first line should contain "`consistent`". Second line should contain array. Array elements should be integers from -2^{31} to $2^{31} - 1$ inclusively. If more than one solution exist, output any.

Examples

rmq.in	rmq.out

`3 2`
`1 2 1`
`2 3 2`
	`consistent`
	`1 2 2`

`3 3`
`1 2 1`
`1 1 2`
`2 3 2`
	`inconsistent`

Problem G. Distinct num

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

How many different numbers are there in an array segment?

Input

On the first line, there's the length of the array is n $(1 \le n \le 300\,000)$. On the second line, there are n integers from 0 to 10^9-1. On the third line, there's the number of queries q $(1 \le q \le 300\,000)$. The following q lines contain a description of the queries, one per line. Each query is given by a pair of integers l, r $(1 \le l \le r \le n)$.

Output

Print the answers to the queries one per line.

Example

`standard input`	`standard output`
`5`	`3`
`1 1 2 1 3`	`2`
`3`	`3`
`1 5`	
`2 4`	
`3 5`	