Problem A. Sum

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

You are given an array of N elements, you have to find the sum on the segments.

Input

The first line of the input contains two integers N and K ($1 \le N \le 100\,000$, $0 \le K \le 100\,000$) — the number of items in the array and the number of queries. Next K lines contain quieries:

1. A i x — set x to i-th element ($1 \le i \le n$, $0 \le x \le 10^9$);
2. Q l r — find sum of the elements on positions from l to r ($1 \le l \le r \le n$).

Initially all the items are zero.

Output

For each query Q l r return the sum on the segment on a different line.

Example

system input	system output
5 9 A 2 2 A 3 1 A 4 2 Q 1 1 Q 2 2 Q 3 3 Q 4 4 Q 5 5 Q 1 5	0 2 1 2 0 5

Problem B. Sum 2

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

Consider an array of n elements. Your task is to find sum of the elements in a range.

Input

The first line of the input contains two integers n and k ($1 \le n \le 100\,000$, $1 \le k \le 100\,000$) — the number of elements in the array and the number of queries, respectively. Each of the following k lines contains a description of the query:

- $A\,l\,r\,x$ — assign to the elements in the range from l to r value x ($1 \le l \le r \le n$, $0 \le x \le 10^9$);
- $Q\,l\,r$ — find the sum of elements of the array from l to r ($1 \le l \le r \le n$).

Initially the array is initialized with zeros.

Output

For every query in format $Q\,l\,r$ output a single number — sum of elements in the specified range.

Example

system input	system output
5 9 A 2 3 2 A 3 5 1 A 4 5 2 Q 1 3 Q 2 2 Q 3 4 Q 4 5 Q 5 5 Q 1 5	3 2 3 4 2 7

Problem C. RMQ

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

You are given an array of N elements, you have to find the maximum on the segments.

Input

The first line of the input contains two integers N and K ($1 \le N \le 100\,000$, $0 \le K \le 100\,000$) — the number of items in the array and the number of queries.

The next line contains N integers — the initial array ($|a_i| \le 10^5$).

Next K lines contain quieries:

1. add l r v — add v to elements on positions from l to r ($1 \le l, r \le n$, $|v| \le 10^5$);
2. max l r — find maximum of the elements on positions from l to r ($1 \le l \le r \le n$).

Initially all the items are zero.

Output

For each query "max l r" return the maximum on the segment on a different line.

Example

system input	system output
5 3 1 2 3 4 -5 max 1 3 add 1 2 5 max 1 3	3 7

Problem D. Signchange

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

Consider an array of n elements. Your task is to implement the data structure that allow the following queries:

- assign element a_i the value j;
- find the alternating sum of the elements of the array from l to r inclusive ($a_l - a_{l+1} + a_{l+2} - \cdots \pm a_r$).

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the length of the array. The second line contains n integer numbers $a_1, \ldots, a_n$ ($|a_i| \le 10^5$) — the elements of the array. The third line contains an integer m ($1 \le n \le 10^5$) — the number of queries:

- the query of the first kind is defined with three integers: $0\,i\,j$ ($1 \le i \le n$, $1 \le j \le 10^4$);
- the query of the second kind is defined with three integers: $1\,l\,r$ ($1 \le l \le r \le n$).

Output

For every query of the second kind output a single number — the corresponding alternating sum.

Example

system input	system 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. K-inversions

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

Consider a permutation $a_1, a_2, \ldots, a_n$ (all a_i are different integers in range from 1 to n). Let us call *k-inversion* a sequence of numbers $i_1, i_2, \ldots, i_k$ such that $1 \le i_1 < i_2 < \ldots < i_k \le n$ and $a_{i_1} > a_{i_2} > \ldots > a_{i_k}$. Your task is to evaluate the number of different k-inversions in a given permutation.

Input

The first line of the input contains two integers n and k ($1 \le n \le 20\,000$, $2 \le k \le 10$). The second line is filled with n numbers a_i.

Output

Output a single number — the number of k-inversions in a given permutation. The number must be taken modulo 10^9.

Example

system input	system output
3 2 3 1 2	2
5 3 5 4 3 2 1	10

Problem F. RMQ Inverse Problem

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

Consider an array of n elements. Let $Q(i, j)$ be the response to a query for finding the minimum among elements of the array from i to j inclusive. Your task is to restore array given some queries and their responses.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 100\,000$) — the number of the elements in the array and the number of queries, respectively. Each of the following m lines contains a description of a query: three integers i, j and q denoting that $Q(i, j) = q$ ($1 \le i \le j \le n$, $2^{-31} \le 1 \le 2^{31} - 1$).

Output

If required array does not exist, output single line containing the single word "inconsistent". In the other case the first line must contain the single word "consistent". The second line must contain n integers — elements of the required array. All integers must be in interval from 2^{-31} to $2^{31} - 1$ inclusive. If there are different solutions, output any of them.

Example

system input	system output
3 2 1 2 1 2 3 2	consistent 1 2 3
3 3 1 2 1 1 1 2 2 3 2	inconsistent

Problem G. Windows 2

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

Consider rectangular windows on screen with sides parallel to the coordinate axis. Some of them can overlap. Your task is to find area covered by this windows.

Input

The first line contains an integer n ($1 \le n \le 50\,000$) — the number of windows. Each of the following n lines contains coordinates of windows $x_{(1,i)}, y_{(1,i)}, x_{(2,i)}, y_{(2,i)}$, where $(x_{(1,i)}, y_{(1,i)})$ are coordinates of the left top corner, and $(x_{(2,i)}, y_{(2,i)})$ are coordinates of the right bottom corner of the i-th window (on the computer screen x coordinate grows down and y coordinate grows from left to right). All coordinates are integers and their absolute values do not exceed 10^9.

Output

The single line must contain the single integer — the area covered by windows.

Example

system input	system output
2 0 0 3 3 1 1 4 4	14

Problem H. Bus

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

Usually the residents of Flatland move in the city using buses. Unfortunately, there is a problem — buses are designed in the way that passengers feel discomfort passing each other on the way to free seats.

Spiridon works as a bus driver for a long time. So, he knows all the passengers riding on the bus day by day. Therefore, he decided to arrange seats in the way minimizing the number of passes between passengers.

It is known when every passenger enters and leaves bus. All places are arranged in a row and numbered from 1 to n starting from the closest seat from exit. When passenger sitting on the i-th place goes to the exit, he passes all the passengers who are sitting on the seats with numbers less than i.

During the ride, passengers do not move in the bus that is they occupy the same place all time.

Input

The first line contains an integer n ($1 \le n \le 100\,000$) — the number of bus passengers. Each of the following n lines contains two integers a_i, b_i ($1 \le a_i < b_i \le 2n$) — the identifiers of the bus stops on which the i-th passenger enters and leaves bus, respectively. At each bus stop at most one person enters or leaves bus.

Output

The first line must contain the single integer — the mimimal number of passes. The second line must contain n integers — for each passenger output the place on which passenger should sit.

Example

system input	system output
2 1 4 2 3	0 2 1
5 1 8 3 6 2 4 9 10 5 7	2 10 2 4 1 1

Problem I. Stacks

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

At the beginning you have n empty stacks. Then some number of operations is performed.

There are three types of operations:

1. A l r x — push x on the top of stacks from l to r;
2. G i — find the number on the top of the i-th stack;
3. R i — cancel the i-th operation of the first type: remove the corresponding number from each stack.

Perform the list of operations.

Input

The first line of the input contains n and m ($1 \le n, m \le 10^5$) — the number of stacks and the number of queries.

Each of the next m lines contains one operation in the format shown above. It is guaranteed that the number of stacks always belongs to the interval $[1, n]$, and all the numbers that are pushed belong to $[1, 10^9]$. For each cancellation request it is guaranteed that the corresponding insertion request was performed, and none of the requests is cancelled twice.

Output

For each query of the second type return one integer — the number on the top of the corresponding stack, or -1, if the corresponding stack is empty.

Example

system input	system output
3 7 A 1 3 5 A 2 3 3 G 1 G 2 R 1 G 1 G 2	5 3 -1 3

Problem J. Different colors

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

You are given n cubes of different colors. You have to perform two types of queries:

- 1 i x — change the color of i-th ($1 \le i \le n$) cube to the color x ($1 \le x \le 10^9$);
- 2 l r — return the maximal number of different colors on some segment of length k included in $[l, r]$ ($r - l + 1 \ge k$).

Input

The first line of the input contain three integers n, m and k ($1 \le n, m \le 10^5$, $1 \le k \le n$) — the number of cubes, the number of queries and the length of a segment. The next line contains n integers a_i ($1 \le a_i \le 10^6$) — the initial colors of cubes.

The next m lines contain m queries in the format described above.

Output

Output the result of each query of the second type on a separate line.

Example

system input	system output
5 3 3 1 2 1 3 3 2 1 3 1 3 3 2 1 5	2 3
6 6 3 2 3 1 1 1 9 2 1 3 2 3 5 1 3 3 2 1 5 1 2 1 2 1 6	3 1 2 3

Problem K. Sequence

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

You are given a sequence of n k-bit integers. You have to perform the queries of two types:

- 1 x y — set the number at position x to y ($1 \le x \le n$, $0 \le y \le 2^k - 1$);
- 2 l r — calculate the function $f(l, r)$ ($1 \le l \le r \le n$).

Function $f(l, r)$ is defined in the following manner:

- $f(l, l) = a_l$;
- $f(l, r) = !(f(l, r-1) \,\&\, a_r)$, where ! — the operation of bitwise negation and & — the operation of bitwise **and**.

Input

The first line contains three integers n, m and k ($1 \le n, m \le 2 \cdot 10^5$, $1 \le k \le 31$) — the length of the sequence, the number of queries and the bit-length of the integers.

The next line contains n integers a_i ($0 \le a_i \le 2^k - 1$) — the initial state of the sequence.

Output

Output the result of each query of second type on a separate line.

Example

system input	system output
4 4 5 31 0 12 4 2 1 4 2 2 3 1 3 19 2 2 4	31 31 27