

A. Simple sum

1 second, 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 .

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

Output

Output sum of all sums.

input
3 1 2 3 3 1 -1 4
output
23

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

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

B. RSQ

2 seconds, 256 megabytes

Input

The first line contains the number n — the size of the array. ($1 \leq n \leq 500\,000$) 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 $(1 \leq i \leq j \leq n)$.

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.

input
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
output
14 15 10 9 12 22 20 10

C. RMQ2

2 seconds, 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 ijx — set value x to all $a[k], i \leq k \leq j$.
- add ijx — increase $a[k], i \leq k \leq j$ by x .
- min ij — 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.

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

D. Painter

2 seconds, 256 megabytes

The Italian abstract artist F. Mandarino became interested in drawing one-dimensional black and white paintings. He is trying to find the best location and the number of black sections of the picture. To do this, he spends on the line white and black segments, and after each of these operations, he wants to know the number of black segments in the resulting picture and their total length.

Initially, the line is white. Your task is to write a program that after each of these operations outputs data in which the artist is interested.

Input

The first line of the input file contains the total number of drawn segments ($1 \leq n \leq 100\,000$). The next n lines contain the description of operations. Each operation is described by a line of the form $C\ X\ I$, where C — color of the segment (W for white segments, B for black), and the segment itself has the form $[X;X+I]$, and the coordinates of both ends — integers not exceeding in absolute value 500 000. The length is set to a positive integer.

Output

After performing each of the operations, it is necessary to output to the output file on a separate line the number of black segments in the picture and their total length, divided by one space.

input
7 W 2 3 B 2 2 B 4 2 B 3 2 B 7 2 W 3 1 W 0 10
output
0 0 1 2 1 4 1 4 2 6 3 5 0 0

Statement is not available on English language

E. Криптография

2 секунды, 256 мегабайт

Задано n матриц A_1, A_2, \dots, A_n размера 2×2 . Необходимо для нескольких запросов вычислить произведение матриц A_i, A_{i+1}, \dots, A_j . Все вычисления производятся по модулю r .

Входные данные

Первая строка входного файла содержит числа $r (1 \leq r \leq 10\,000)$, $n (1 \leq n \leq 200\,000)$ и $m (1 \leq m \leq 200\,000)$. Следующие n блоков по две строки содержащие по два числа в строке — описания матриц. Затем следуют m пар целых чисел от 1 до n , запросы на произведение на отрезке.

Выходные данные

Выведите m блоков по две строки, по два числа в каждой — произведения на отрезках. Разделяйте блоки пустой строкой. Все вычисления производятся по модулю r

входные данные
3 4 4 0 1 0 0 2 1 1 2 0 0 0 2 1 0 0 2 1 4 2 3 1 3 2 2
выходные данные
0 2 0 0 0 2 0 1 0 1 0 0 2 1 1 2

F. Sparse table

2 seconds, 256 megabytes

You are given an array consisting of n integers. Write a program that answers the queries of the following type: find the minimum between u -th and v -th element, inclusive.

Input

The first line contains three integers: n, m and a_1 ($1 \leq n \leq 10^5$; $1 \leq m \leq 10^7$; $0 \leq a_1 < 16\,714\,589$) — the number of integers in the given array, the number of queries, and the first element of the given array, respectively.

The second line contains two integers u_1 , and v_1 ($1 \leq u_1, v_1 \leq n$) — the first query.

For the sake of the input size, the array and the queries should be generated.

The array elements a_2, a_3, \dots, a_n are generated with the following formula:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

For instance, if $n = 10$, $a_1 = 12345$ the following array should be generated: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

The queries are generated in the following way:

$$u_{i+1} = ((17 \cdot u_i + 751 + r_i + 2i) \bmod n) + 1, \quad v_{i+1} = ((13 \cdot v_i + 593 + r_i + 5i) \bmod n) + 1,$$

where r_i — the answer for query i .

Be careful, u_i can be greater than v_i .

Output

Print three integers u_m, v_m and r_m (the last query, and the answer to it).

input
10 8 12345 3 9
output
5 3 1565158

Notice that you can get rid of saving u , v , and r arrays into memory.

The queries, and the answers to them:

i	u_i	v_i	r_i
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

This problem probably is not solvable by using standard Python 2 and Python 3 interpreters. Use PyPy compilers instead.

Statement is not available on English language

G. Окна

2 секунды, 256 мегабайт

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

Входные данные

В первой строке входного файла записано число окон n ($1 \leq n \leq 50000$). Следующие n строк содержат координаты окон $X(1, i)$ $Y(1, i)$ $X(2, i)$ $Y(2, i)$, где $(X(1, i), Y(1, i))$ — координаты левого верхнего угла i -го окна, а $(X(2, i), Y(2, i))$ — правого нижнего (на экране компьютера Y растёт сверху вниз, а X — слева направо). Все координаты — целые числа, по модулю не превосходящие $2 \cdot 10^5$.

Выходные данные

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенные пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т.е. покрывающими свои граничные точки.

входные данные
2 0 0 3 3 1 1 4 4
выходные данные
2 1 3

входные данные
1 0 0 1 1
выходные данные
1 0 1

H. Backward RMQ

2 seconds, 256 megabytes

Consider array $a[1..n]$. Let $Q(i, j)$ be a minimum among numbers $a[i], \dots, a[j]$. 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 ($1 \leq n, m \leq 100\,000$). Next m lines contains three integers i, j and q , meaning that $Q(i, j) = q$ ($1 \leq i \leq j \leq n, -2^{31} \leq q \leq 2^{31} - 1$).

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.

input
3 2 1 2 1 2 3 2

output

consistent
1 2 2

input

3 3
1 2 1
1 1 2
2 3 2

output

inconsistent

Statement is not available on English language

I. Горы

2 секунды, 256 мегабайт

В парке развлечений «Ай-ой-ай» открылся новейший аттракцион: польские горки. Трек состоит из n рельс, присоединенных одна к концу другой. Начало первой рельсы находится на высоте 0. Оператор Петя может конфигурировать аттракцион, изменяя по своему желанию подъем нескольких последовательных рельс. При этом подъем всех остальных рельс не изменяется. При каждом изменении конфигурации рельс положение следующих за изменяемыми подбирается таким образом, чтобы весь трек оставался связным.

Каждый запуск вагонетки осуществляется с энергией, достаточной для достижения высоты h . Это значит, что вагонетка будет двигаться до тех пор, пока высота не превысит h , либо пока не закончится трек.

По записям о всех изменениях конфигурации рельс и временах запусков вагонетки для каждого запуска определите, сколько рельс вагонетка проедет до остановки.

Трек можно представить как последовательность n подъемов d_i по одному на рельс. Изначально рельсы горизонтальны, то есть $d_i = 0$ для всех i .

Каждое изменение конфигурации определяется числами a , b и D : все рельсы с a -й по b -ю включительно после этого действия имеют подъем, равный D .

Каждый запуск вагонетки определяется единственным целым числом h — максимальной высотой, на которую способна подняться вагонетка.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^9$) — число рельс. Следующие строки содержат запросы трех видов:

- I $a\ b\ D$ — изменение конфигурации. Рельсы с a -й по b -ю включительно после выполнения запроса имеют подъем, равный D .
- Q h — запуск вагонетки. Требуется найти число рельс, которое проедет вагонетка, которая способна подняться на высоту h .
- E — конец ввода. Этот запрос встретится ровно один раз в конце файла.

В любой момент времени высота любой точки трека лежит в промежутке от 0 до 10^9 . Во вводе не более 100 000 строк.

Выходные данные

Для каждого запроса Q выведите единственное целое число — количество рельс, которое проедет вагонетка.

входные данные

4
Q 1
I 1 4 2
Q 3
Q 1
I 2 2 -1
Q 3
E

выходные данные

4
1
0
3

Statement is not available on English language

J. Великая Китайская Стена

2 секунды, 256 мегабайт

В этой задаче мы проследим альтернативную историю Великой Китайской Стены.

Великая Китайская Стена состоит из n метровых участков, пронумерованных по порядку целыми числами от 1 до n . Каждый участок характеризуется своей высотой в метрах — целым неотрицательным числом. До начала нашей истории Стена ещё не построена, поэтому высота каждого участка равна нулю.

Происходят события двух видов.

- 1. *Укрепление Стены* (запись: «defend $a\ b\ c$ »). Император вызывает к себе вассалов из приграничных провинций и велит им сделать так, чтобы промежуток Стены, охватывающий участки от a до b включительно, имел высоту не менее c метров. Это значит, что все участки меньшей высоты на этом промежутке нужно достроить до высоты c , а остальные оставить нетронутыми. Приказ императора выполняется немедленно, то есть до наступления следующего события.
- 2. *Нападение варваров* (запись: «attack $d\ e$ »). Варвары подходят к Стене снаружи и занимают позиции напротив промежутка Стены, охватывающего участки от d до e включительно. После этого они находят такой участок на этом промежутке, у которого высота как можно меньше, и пытаются через него проникнуть на территорию Китая. Нападение также происходит немедленно, до наступления следующего события.

Для восстановления достоверной альтернативно-исторической картины не хватает одного: для каждого нападения варваров указать минимальную высоту Стены на соответствующем промежутке, а также какой-нибудь участок из этого промежутка с такой высотой. По заданной последовательности событий найдите эти числа.

Входные данные

В первой строке заданы через пробел два целых числа n и m — длина Стены в метрах и количество событий соответственно ($1 \leq n \leq 10^6, 0 \leq m \leq 10^5$). В следующих m строках описаны события в порядке их следования. Если событие описывает укрепление Стены, оно задано в форме «defend $a\ b\ c$ » ($1 \leq a \leq b \leq n, 1 \leq c \leq 10^7$). Если же событие описывает нападение варваров, оно задано в форме «attack $d\ e$ » ($1 \leq d \leq e \leq n$).

Выходные данные

В ответ на каждое нападение варваров выведите строку, содержащую два числа, разделённые пробелом. Первое из этих чисел — минимальная высота Стены на соответствующем промежутке. Второе — номер любого метрового участка Стены на этом промежутке, имеющего такую высоту.

входные данные
5 4 defend 1 3 10 attack 1 4 attack 2 3 attack 1 2
выходные данные
0 4 10 2 10 1

Statement is not available on English language

К. Парковка

2 секунды, 256 мегабайт

На кольцевой парковке есть n мест пронумерованных от 1 до n . Есть два вида событий прибытие машину на парковку и отъезд машины с парковки. Если машина приезжает на парковку, а её место занято, то она едет далее по кругу и встаёт на первое свободное место.

Входные данные

В первой строке входного файла находится два числа n и m — размер парковки и количество запросов ($1 \leq n, m \leq 100000$). В следующих m строках находятся события. Каждая из этих строк имеет следующий вид:

- enter X — приехала машина, которая хочет встать на место X . Для каждой такой команды выведите какое место займёт эта машина.
- exit X — уехала машина занимавшая место X . Гарантируется, что на этом месте была машина.

Выходные данные

Выведите последовательно результаты выполнения всех операций enter.

входные данные
3 5 enter 1 enter 1 exit 1 enter 2 enter 2

выходные данные

1
2
3
1

Statement is not available on English language

L. Звезды

2 секунды, 256 мегабайт

Вася любит наблюдать за звездами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером $n \times n \times n$. Этот куб поделен на маленькие кубики размером $1 \times 1 \times 1$. Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звезд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

Входные данные

Первая строка входного файла содержит натуральное число $1 \leq n \leq 128$. Координаты кубиков — целые числа от 0 до $n-1$. Далее следуют записи о происходивших событиях по одной в строке. В начале строки записано число m . Если m равно:

- 1, то за ним следуют 4 числа — x, y, z ($0 \leq x, y, z < n$) и k ($-20000 \leq k \leq 20000$) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;
- 2, то за ним следуют 6 чисел — $x_1, y_1, z_1, x_2, y_2, z_2$ ($0 \leq x_1 \leq x_2 < n, 0 \leq y_1 \leq y_2 < n, 0 \leq z_1 \leq z_2 < n$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$;
- 3, то это означает, что Васе надоело наблюдать за звездами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней.

Количество записей во входном файле не больше 100 002.

Выходные данные

Для каждого Петиного вопроса выведите искомое количество звезд.

входные данные

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

выходные данные

0
1
4
2