

1A. RSQ

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Входные данные

В первой строке находится число n — размер массива. ($1 \leq n \leq 500\,000$) Во второй строке находится n чисел a_i — элементы массива. Далее содержится описание операций, их количество не превышает $1\,000\,000$. В каждой строке находится одна из следующих операций:

- `set i x` — установить $a[i]$ в x .
- `sum i j` — вывести значение суммы элементов в массиве на отрезке с i по j , гарантируется, что $(1 \leq i \leq j \leq n)$.

Все числа во входном файле и результаты выполнения всех операций не превышают по модулю 10^{18} .

Выходные данные

Выведите последовательно результат выполнения всех операций `sum`. Следуйте формату выходного файла из примера.

Пример

входные данные	Скопировать
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	
выходные данные	Скопировать
14 15 10 9 12 22 20 10	

1B. RMQ

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Входные данные

В первой строке находится число n — размер массива. ($1 \leq n \leq 500000$) Во второй строке находится n чисел a_i — элементы массива. Далее содержится описание операций, их количество не превышает 1000000. В каждой строке находится одна из следующих операций:

- `set i x` — установить $a[i]$ в x .
- `min i j` — вывести значение минимального элемента в массиве на отрезке с i по j , гарантируется, что $(1 \leq i \leq j \leq n)$.

В массив помещаются только целые числа, не превышающие по модулю 10^9 .

Выходные данные

Выведите последовательно результат выполнения всех операций `min`. Следуйте формату выходного файла из примера.

Пример

входные данные

[Скопировать](#)

```
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
```

выходные данные

[Скопировать](#)

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

1С. Художник

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Итальянский художник-абстракционист Ф. Мандарино увлекся рисованием одномерных черно-белых картин. Он пытается найти оптимальное местоположение и количество черных участков картины. Для этого он проводит на прямой белые и черные отрезки, и после каждой из таких операций хочет знать количество черных отрезков на получившейся картине и их суммарную длину.

Изначально прямая — белая. Ваша задача — написать программу, которая после каждой из таких операций выводит в выходной файл интересующие художника данные.

Входные данные

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leq n \leq 100\,000$). В последующих n строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где c — цвет отрезка (w для белых отрезков, b для черных), а сам отрезок имеет вид $[x; x + l)$, причем координаты обоих концов — целые числа, не превосходящие по модулю **500 000**. Длина задается положительным целым числом.

Выходные данные

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество черных отрезков на картине и их суммарную длину, разделенные одним пробелом.

Пример

входные данные	Скопировать
<pre>7 W 2 3 B 2 2 B 4 2 B 3 2 B 7 2 W 3 1 W 0 10</pre>	
выходные данные	Скопировать
<pre>0 0 1 2 1 4 1 4 2 6 3 5 0 0</pre>	

1D. RMQ2

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Входные данные

В первой строке находится число n — размер массива. ($1 \leq n \leq 10^5$) Во второй строке находится n чисел a_i — элементы массива. Далее содержится описание операций, их количество не превышает $2 \cdot 10^5$. В каждой строке находится одна из следующих операций:

- `set i j x` — установить все $a[k]$, $i \leq k \leq j$ в x .
- `add i j x` — увеличить все $a[k]$, $i \leq k \leq j$ на x .
- `min i j` — вывести значение минимального элемента в массиве на отрезке с i по j , гарантируется, что ($1 \leq i \leq j \leq n$).

Все числа во входном файле и результаты выполнения всех операций не превышают по модулю 10^{18} .

Выходные данные

Выведите последовательно результат выполнения всех операций `min`. Следуйте формату выходного файла из примера.

Пример

входные данные

Скопировать

```
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
```

выходные данные

Скопировать

```
2
1
1
2
5
5
8
8
```

1E. RMQ наоборот

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: rmq.in

вывод: rmq.out

Рассмотрим массив $a[1..n]$. Пусть $Q(i, j)$ — ответ на запрос о нахождении минимума среди чисел $a[i], \dots, a[j]$. Вам даны несколько запросов и ответы на них. Восстановите исходный массив.

Входные данные

Первая строка входного файла содержит число n — размер массива, и m — число запросов ($1 \leq n, m \leq 100\,000$).
Следующие m строк содержат по три целых числа i, j и q , означающих, что $Q(i, j) = q$ ($1 \leq i \leq j \leq n$,
- $2^{31} \leq q \leq 2^{31} - 1$).

Выходные данные

Если искомого массива не существует, выведите строку «inconsistent».

В противном случае в первую строку выходного файла выведите «consistent». Во вторую строку выходного файла выведите элементы массива. Элементами массива должны быть целые числа в интервале от -2^{31} до $2^{31} - 1$ включительно. Если решений несколько, выведите любое.

Примеры

входные данные	Скопировать
3 2 1 2 1 2 3 2	
выходные данные	Скопировать
consistent 1 2 2	

входные данные	Скопировать
3 3 1 2 1 1 1 2 2 3 2	
выходные данные	Скопировать
inconsistent	

1F. Горы

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

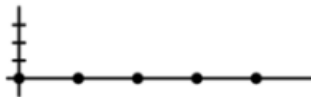
вывод: стандартный вывод

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.

Пример

входные данные	Скопировать
<pre>4 Q 1 I 1 4 2 Q 3 Q 1 I 2 2 -1 Q 3 E</pre>	
выходные данные	Скопировать
<pre>4 1 0 3</pre>	

2A. Разреженные таблицы

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Входные данные

В первой строке заданы три натуральных числа n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 10^7$) и a_1 ($0 \leq a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leq u_1, v_1 \leq n$) — первый запрос.

Для того, чтобы размер ввода был небольшой, массив и запросы генерируются.

Элементы a_2, a_3, \dots, a_n задаются следующей формулой:

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

Например, при $n = 10, a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

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

где r_i — ответ на запрос номер i .

Обратите внимание, что u_i может быть больше, чем v_i .

Выходные данные

В выходной файл выведите u_m, v_m и r_m (последний запрос и ответ на него).

Примеры

входные данные	Скопировать
10 8 12345 3 9	
выходные данные	Скопировать
5 3 1565158	

K -й максимум

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k -й максимум.

Входные данные

Первая строка входного файла содержит натуральное число n — количество команд ($n \leq 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \leq 10^9$). Поддерживаемые команды:

- 1: Добавить элемент с ключом k_i .
- 0: Найти и вывести k_i -й максимум.
- -1: Удалить элемент с ключом k_i .

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i -го максимума, он существует.

Выходные данные

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i -й максимум.

Пример

входные данные

[Скопировать](#)

```
11
1 5
1 3
1 7
0 1
0 2
0 3
-1 5
1 10
0 1
0 2
0 3
```

выходные данные

[Скопировать](#)

```
7
5
3
10
7
3
```

4В. Переместить в начало

ограничение по времени на тест: 6 секунд
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Вам дан массив $a_1 = 1, a_2 = 2, \dots, a_n = n$ и последовательность операций: переместить элементы с l_i по r_i в начало массива. Например, для массива 2, 3, 6, 1, 5, 4, после операции (2, 4) новый порядок будет 3, 6, 1, 2, 5, 4. А после применения операции (3, 4) порядок элементов в массиве будет 1, 2, 3, 6, 5, 4.

Выведите порядок элементов в массиве после выполнения всех операций.

Входные данные

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 100\,000, 1 \leq m \leq 100\,000$) — число элементов в массиве и число операций. Следующие m строк содержат операции в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Выходные данные

Выведите n целых чисел — порядок элементов в массиве после применения всех операций.

Пример

входные данные	Скопировать
6 3 2 4 3 5 2 2	
выходные данные	Скопировать
1 4 5 2 3 6	

4D. Развороты

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Вам дан массив $a_1 = 1, a_2 = 2, \dots, a_n = n$ и последовательность операций: переставить элементы с l_i по r_i в обратном порядке. Например, для массива 1, 2, 3, 4, 5, после операции (2, 4) новый порядок будет 1, 4, 3, 2, 5. А после применения операции (3, 5) порядок элементов в массиве будет 1, 4, 5, 2, 3.

Выведите порядок элементов в массиве после выполнения всех операций.

Входные данные

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — число элементов в массиве и число операций. Следующие m строк содержат операции в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Выходные данные

Выведите n целых чисел — порядок элементов в массиве после применения всех операций.

Пример

входные данные	Скопировать
5 3 2 4 3 5 2 2	
выходные данные	Скопировать
1 4 5 2 3	