

Задача А. Разрезанные таблицы

Имя входного файла: `sparse.in`
Имя выходного файла: `sparse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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

Формат входных данных

В первой строке входного файла даны 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.

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:

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

где ans_i — ответ на запрос номер i .

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

Формат выходных данных

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

Примеры

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

Замечание

Пояснение к тесту из примера: запросы и результаты.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
12345	305498	7048017	11694653	1565158	2591019	9471233	570265	13137658	1325095

#	u	v	ans
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

Задача В. Солдаты

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У генерала Гавса есть N солдат в сапогах. И генерал Гавс захотел их отсортировать по росту, но тут возникла небольшая проблема - солдаты были новобранцами и могли поменяться только в том случае, если их рост отличается не более чем на K см.

Генерал Гавс сразу понял, что может и не получится отсортировать своих солдат по росту, но подумал и решил, что ему будет достаточно получить лексикографически минимальную последовательность из своих новобранцев.

Помогите генералу Гавсу и найдите лексикографически минимальную последовательность. Переставлять своих солдат Гавс может сколько угодно.

Формат входных данных

Первая строка ввода содержит N ($1 \leq N \leq 10^5$) и K ($1 \leq K \leq 10^9$), а каждая $i + 1$ -ая строка содержит высоту i -того солдата.

Формат выходных данных

Выведите N строк, i -ая строка содержит высоту i -го солдата в решении.

Пример

стандартный ввод	стандартный вывод
5 3	6
7	7
7	7
3	2
6	3
2	

Задача С. Леха и сумма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вероятно, вы уже поняли, что Леха очень любит различные типы запросов.

Иногда, когда он идет гулять с друзьями, они идут в кино. Но каждый раз, когда Леха куда-то идет, он не может расслабиться, если его домашняя работа еще не выполнена.

Леха любезно просит вас помочь ему с домашним заданием.

В этой задаче вам дан массив p , состоящий из n элементов.

Вам нужно ответить на m запросов.

Каждый запрос имеет следующий формат: $l \ r \ x$. Ответ на запрос — $\sum_{i=l}^r (p_i \bmod x)$.

Для каждого запроса выведите ответ на отдельной строке.

Формат входных данных

В первой строке содержатся два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$).

Во второй строке содержится n целых чисел p_i ($0 \leq p_i \leq 2 \cdot 10^5$).

В следующих m строках содержится по три целых числа: l , r и x ($1 \leq l \leq r \leq n$, $1 \leq x \leq 2 \cdot 10^5$).

Формат выходных данных

Для каждого запроса выведите ответ на отдельной строке.

Примеры

стандартный ввод	стандартный вывод
10 5	11
23 32 42 50 2 33 41 5 100 3	75
2 10 3	36
1 9 23	9
3 10 12	13
2 8 4	
2 10 5	

Замечание

- $2 + 0 + 2 + 2 + 0 + 2 + 2 + 1 + 0 = 11$
- $0 + 9 + 19 + 4 + 2 + 10 + 18 + 5 + 8 = 75$
- $6 + 2 + 2 + 9 + 5 + 5 + 4 + 3 = 36$
- $0 + 2 + 2 + 2 + 1 + 1 + 1 = 9$
- $2 + 2 + 0 + 2 + 3 + 1 + 0 + 0 + 3 = 13$

Задача D. Фаброзавры-дизайнеры

Имя входного файла: `fabro.in`
Имя выходного файла: `fabro.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Фаброзавры известны своим тонким художественным вкусом и увлечением ландшафтным дизайном. Они живут около очень живописной реки и то и дело перестраивают тропинку, идущую вдоль реки: либо насыпают дополнительной земли, либо срывают то, что есть. Для того, чтобы упростить эти работы, они поделили всю тропинку на горизонтальные участки, пронумерованные от 1 до N , и их переделки устроены всегда одинаково: они выбирают часть дороги от L -ого до R -ого участка (включительно) и изменяют (увеличивают или уменьшают) высоту на всех этих участках на одну и ту же величину (если до начала переделки высоты были разными, то и после переделки они останутся разными).

Поскольку, как уже говорилось, у фаброзавров тонкий художественный вкус, каждый из них считает, что их река лучше всего выглядит с определенной высоты. Поэтому им хочется знать, есть ли поблизости от их дома место на тропинке, где высота на их взгляд оптимальна. Помогите им в этом разобраться.

Формат входных данных

Первая строка входного файла содержит два числа N и M — длину дороги и количество запросов соответственно ($1 \leq N, M \leq 10^5$). На второй строке содержатся N чисел, разделенных пробелами — начальные высоты соответствующих частей дороги; высоты не превосходят 10^4 по модулю. В следующих M строках содержатся запросы по одному на строке.

Запрос $+ L R X$ означает, что высоту частей дороги от L -ой до R -ой (включительно) нужно изменить на X . При этом $1 \leq L \leq R \leq N$, а $|X| \leq 10^4$.

Запрос $? L R X$ означает, что нужно проверить, есть ли между L -ым и R -ым участками (включая эти участки) участок, где дорога проходит точно на высоте X . Гарантируется, что $1 \leq L \leq R \leq N$, а $|X| \leq 10^9$.

Формат выходных данных

На каждый запрос второго типа нужно вывести в выходной файл на отдельной строке одно слово «YES» (без кавычек), если нужный участок существует, и «NO» в противном случае.

Примеры

<code>fabro.in</code>	<code>fabro.out</code>
10 5	NO
0 1 1 3 3 3 2 0 0 1	YES
? 3 5 2	YES
+ 1 4 1	
? 3 5 2	
+ 7 10 2	
? 9 10 3	

Задача Е. Перестановка и запросы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана перестановка p из n элементов. Перестановка из n элементов — это массив длины n , в котором каждое целое число от 1 до n встречается ровно по одному разу. Например, $[1, 2, 3]$ и $[4, 3, 5, 1, 2]$ — это перестановки, но $[1, 2, 4]$ и $[4, 3, 2, 1, 2]$ — это не перестановки. Вам нужно выполнить q запросов.

Есть два типа запросов:

- $1\ x\ y$ — поменять местами p_x и p_y .
- $2\ i\ k$ — вывести число, которым станет i , если присвоить $i = p_i\ k$ раз.

Формат входных данных

В первой строке находятся два целых числа n и q ($1 \leq n, q \leq 10^5$).

Во второй строке находятся n целых чисел p_1, p_2, \dots, p_n .

В каждой из следующих q строк находится по три целых числа. Первое число t ($1 \leq t \leq 2$) — тип запроса. Если $t = 1$, то следующие два целых числа — это x и y ($1 \leq x, y \leq n; x \neq y$) — запрос первого типа. Если $t = 2$, то следующие два целых числа — это i и k ($1 \leq i, k \leq n$) — запрос второго типа.

Гарантируется, что есть хотя бы один запрос второго типа.

Формат выходных данных

Для каждого запроса второго типа выведите одно целое число в новой строке — ответ на этот запрос.

Примеры

стандартный ввод	стандартный вывод
5 4	4
5 3 4 2 1	1
2 3 1	2
2 1 2	
1 1 3	
2 1 2	
5 9	3
2 3 5 1 4	5
2 3 5	4
2 5 5	2
2 5 1	3
2 5 3	3
2 5 4	3
1 5 4	1
2 5 3	
2 2 5	
2 5 1	

Замечание

В первом примере $p = \{5, 3, 4, 2, 1\}$.

Первый запрос — вывести p_3 . Ответ — 4.

Второй запрос — вывести p_{p_1} . Ответ — 1.

Третий запрос — поменять местами p_1 и p_3 . Теперь $p = \{4, 3, 5, 2, 1\}$.

Четвёртый запрос — вывести p_{p_1} . Ответ — 2.

Задача F. Безумие и Отвага

Имя входного файла: `heroes.in`
Имя выходного файла: `heroes.out`
Ограничение по времени: 10 секунд
Ограничение по памяти: 512 мегабайт

Многие из нас с детства мечтали создавать компьютерные игры, а для некоторых это даже стало причиной, по которой они начали изучать информатику и программирование. Мишина мечта сбылась, и теперь он работает в известной и уважаемой корпорации «Метель», выпустившей в своё время такие шедевры, как «Искусство войны» и «Звёздное ремесло».

Недавно Миша присоединился к проекту новой ролевой игры «Безумие и отвага». Её ключевой особенностью является возможность на каждом из уровней заново выбирать персонажа для его прохождения.

Перед стартом очередного уровня игроку доступны N героев. Каждый герой характеризуется силой атаки a_i и запасом здоровья b_i . Уровень представляет собой длинную пещеру, содержащую M монстров. Каждый монстр также имеет свою силу атаки c_i и запас здоровья d_i . Зайдя в пещеру, герой сначала сражается с первым монстром, затем, если остаётся жив, сражается со вторым и так далее, пока не погибнет или не дойдёт до конца. Количество жизней героя не восстанавливается между боями, то есть каждую следующую драку он начинает с меньшим запасом здоровья, чем предыдущую.

Бой между монстром и героем состоит в одновременном обмене ударами. Каждый из них, нанося удар, уменьшает запас здоровья противника на величину, равную силе своей атаки. Как только запас здоровья кого-либо из сражающихся становится неположительным, он умирает, и бой прекращается. Обратите внимание, что при такой схеме боя возможна ситуация, когда оба противника погибнут одновременно.

Компания планирует распространять игру бесплатно, получая доход за счёт продажи разнообразных бонусов, реализовать один из которых и поручено Мише. Данный бонус позволяет игроку узнать, сколько монстров убьёт каждый из героев, если игрок выберет именно его для прохождения данного уровня. Так как монстров и героев может быть очень много, Миша столкнулся со сложностями при вычислении необходимых значений и обратился за помощью к вам.

Формат входных данных

В первой строке ввода записаны два целых числа N и M — количество доступных игроку героев и количество монстров в пещере соответственно ($1 \leq N, M \leq 200\,000$).

Следующие N строк описывают героев. Каждая из них содержит два целых числа a_i и b_i , задающих силу атаки и запас здоровья i -го героя ($1 \leq a_i, b_i \leq 10^9$).

Далее следуют M строк, описывающих находящиеся в пещере монстров. Каждое описание состоит из двух целых чисел c_i и d_i , обозначающих параметры i -го монстра ($1 \leq c_i, d_i \leq 200\,000$). Порядок расположения монстров в пещере совпадает с порядком их описания, то есть первым необходимо убить монстра, описанного в строке $N + 2$, а последним — в строке $N + M + 1$.

Формат выходных данных

Выведите N чисел по одному в строке. i -я строка должна содержать ответ для i -го героя.

Примеры

<code>heroes.in</code>	<code>heroes.out</code>
5 3	0
1 2	1
2 2	2
10 10	3
100 10	3
1 100	
2 2	
7 2	
3 20	

Замечание

Бой между первым героем и первым монстром в пещере продлится один ход, после которого герой погибнет, а монстр останется в живых.

Параметры второго героя совпадают с параметрами первого монстра, поэтому они убьют друг друга на первом же ходу боя. Ответ для данного героя равен одному.

Если игрок выберет для прохождения уровня третьего героя, то после боя с первым монстром его запас здоровья будет равен восьми, а после боя со вторым — единице. Для убийства третьего монстра ему необходимо сделать два удара, но он умрёт после первой же его атаки.

У четвёртого героя столько же жизней, сколько и у третьего, но сила атаки гораздо больше, поэтому он пройдёт уровень полностью, хотя и погибнет в последней драке.

Пятый герой обладает минимально возможной силой атаки, но при этом у него большой запас здоровья, поэтому он сможет пройти весь уровень и остаться в живых. После первого боя его запас здоровья будет равен 96, после второго — 82, а в конце игры останется только 22.

Задача G. Жесть

Имя входного файла: `sqrtrev.in`
Имя выходного файла: `sqrtrev.out`
Ограничение по времени: 10 секунд
Ограничение по памяти: 256 мегабайт

Дан массив из N чисел. Нужно уметь обрабатывать 3 типа запросов:

- `get(L, R, x)` — сказать, сколько элементов отрезка массива $[L..R]$ не меньше x .
- `set(L, R, x)` — присвоить всем элементам массива на отрезке $[L..R]$ значение x .
- `reverse(L, R)` — перевернуть отрезок массива $[L..R]$.

Формат входных данных

Число N ($1 \leq N \leq 10^5$) и массив из N чисел. Далее число запросов M ($1 \leq M \leq 10^5$) и M запросов. Формат описания запросов предлагается понять из примера. Для всех отрезков верно $1 \leq L \leq R \leq N$. Исходные числа в массиве и числа x в запросах — целые от 0 до 10^9 .

Формат выходных данных

Для каждого запроса типа `get` нужно вывести ответ.

Примеры

<code>sqrtrev.in</code>	<code>sqrtrev.out</code>
5	3
1 2 3 4 5	1
6	3
get 1 5 3	1
set 2 4 2	
get 1 5 3	
reverse 1 2	
get 2 5 2	
get 1 1 2	

Задача Н. Лунки

Имя входного файла: `holes.in`
Имя выходного файла: `holes.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маленький Петя очень любит играть. Больше всего он любит играть в игру «Лунки». Это игра для одного игрока со следующими правилами:

Есть N лунок, расположенных в ряд, пронумерованных слева направо числами от 1 до N . У каждой лунки изначально установлена своя сила выброса (у лунки с номером i она равна a_i). Если вбросить шарик в лунку i , то он тут же вылетит из нее и попадет в лунку $i + a_i$, после чего он опять вылетит и т.д.. Если же лунки с таким номером нету, то он просто вылетит за край ряда. На каждом из M ходов игрок выбирает одно из двух действий:

- Установить силу выброса лунки a равной b .
- Вбросить шарик в лунку a и посчитать количество прыжков шарика, прежде чем он вылетит за край ряда, а так же записать номер лунки, после выпрыгивания из которой шарик вылетел за край.

У Пети есть некоторые проблемы с математикой, поэтому, как Вы уже догадались, именно Вам предстоит произвести все подсчеты.

Формат входных данных

Первая строка содержит два числа N и M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$) — количество лунок в ряду и количество ходов. Следующая строка содержит N целых положительных чисел, не превышающих N — начальные силы выброса лунок. Следующие M строк задают ходы, сделанные Петей. Каждая строка может быть двух типов:

- 0 a b
- 1 a

Тут, первый тип означает что требуется установить силу выброса лунки a равной b , а второй означает что требуется вбросить мячик в лунку с номером a . Числа a и b — целые положительные и не превышают N .

Формат выходных данных

Для каждого хода второго типа (задающего вбрасывание шарика) в порядке следования во входном файле выведите два числа через пробел в отдельной строке — номер последней лунки перед вылетом шарика за край и количество прыжков.

Примеры

<code>holes.in</code>	<code>holes.out</code>
8 5	8 7
1 1 1 1 1 2 8 2	8 5
1 1	7 3
0 1 3	
1 1	
0 3 4	
1 2	