

Задача А. Разрезанные таблицы

Имя входного файла: `sparse.in`
Имя выходного файла: `sparse.out`
Ограничение по времени: 1 секунда
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. Интеллектуальный отпуск

Имя входного файла: e.in
Имя выходного файла: e.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Туристическая отрасль в этом сезоне столкнулась с серьёзными сложностями. Добросовестные туроператоры ищут новые рекламные ходы для продажи своих туров. Как известно, наиболее благоприятная для отдыха погода меняется плавно, причём не только от одного дня к другому, но и в течение суток.

Для большинства туристических направлений есть многолетние посекундные результаты измерений различных климатических параметров, например, температуры или влажности. У каждого человека своё понимание того, насколько различными могут быть подобные значения во время отпуска, но всех интересуют непрерывные туры как можно большей продолжительности.

Пусть мы зафиксировали туристическое направление и некоторый климатический параметр. Будем называть изменчивостью тура разницу между максимальным и минимальным значением выбранного параметра за всё время поездки. Для каждого туриста известно максимальное приемлемое значение изменчивости k_i .

Даны результаты измерений некоторого климатического параметра на одном из курортов и значения k_i для нескольких туристов. Требуется для каждого из них определить максимальный диапазон, подходящий для отпуска.

Формат входных данных

В первой строке входного файла находится целое число N ($1 \leq N \leq 600\,000$) — количество сделанных измерений. Во второй строке — N целых чисел, по модулю не превосходящих 10^9 — данные посекундных измерений.

В третьей строке входного файла находится число M ($1 \leq M \leq 100$) — количество туристов, для которых необходимо найти оптимальный диапазон. В четвёртой строке — M целых чисел k_1, k_2, \dots, k_M ($0 \leq k_i \leq 10^9$) — максимальная возможная разница между выбранным климатическим параметром в непрерывном диапазоне дней для каждого из туристов.

Формат выходных данных

В выходной файл для каждого из M запросов в отдельной строке выведите два числа: номер первого измерения диапазона и номер последнего измерения, входящего в диапазон. Нумерация измерений ведётся с единицы. Если для некоторого туриста существует несколько подходящих диапазонов максимальной длины, выведите границы любого из них.

Примеры

e.in	e.out
7	3 5
10 1 10 12 11 1 11	4 5
2	
2 1	
9	3 4
1 5 2 3 6 4 7 8 9	1 9
6	7 9
1 10 2 4 5 0	2 6
	1 6
	1 1

Задача С. Сумма на отрезке

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

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка входного файла содержит два целых числа N и K — количество чисел в массиве и количество запросов ($1 \leq N \leq 100\,000$, $0 \leq K \leq 100\,000$). Следующие K строк содержат следующие запросы:

- A i x — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$);
- Q l r — найти сумму чисел в массиве на позициях от l до r ($1 \leq l \leq r \leq n$).

Изначально в массиве живут нули.

Формат выходных данных

На каждый запрос вида Q l r нужно вывести единственное число — сумму на отрезке.

Примеры

<code>sum.in</code>	<code>sum.out</code>
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Замечание

TL для Python 4 секунды

Задача D. Река

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

Во Флатландии протекает богатая рыбой река Большой Флат. Много лет назад река была поделена между n рыболовными предприятиями, каждое из которых получило непрерывный отрезок реки. При этом i -е предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длиной a_i .

С тех пор с рыболовными предприятиями во Флатландии k раз происходили различные события. Каждое из событий было одного из двух типов: банкротство некоторого предприятия или разделение некоторого предприятия на два. При некоторых событиях отрезок реки, принадлежащий предприятию, с которым это событие происходит, делится на две части. Каждый такой отрезок имеет длину большую или равную 2. Деление происходит по следующему правилу. Если отрезок имеет четную длину, то он делится на две равные части. Иначе он делится на две части, длины которых различаются ровно на единицу, при этом часть, которая ближе к истоку реки, имеет меньшую длину.

При банкротстве предприятия происходит следующее. Отрезок реки, принадлежавший обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседей двое, то отрезок реки делится на две части описанным выше способом, после чего каждый из соседей присоединяет к своему отрезку ближайшую к нему часть. При разделении предприятия отрезок реки, принадлежавший разделяемому предприятию, всегда делится на две части описанным выше способом. Разделившееся предприятие ликвидируется, и образуются два новых предприятия. Таким образом, после каждого события каждое предприятие владеет некоторым отрезком реки.

Министерство финансов Флатландии предлагает ввести налог на рыболовные предприятия, пропорциональный квадрату длины отрезка реки, принадлежащего соответствующему предприятию. Чтобы проанализировать, как будет работать этот налог, министр хочет по имеющимся данным узнать, как изменялась величина, равная сумме квадратов длин отрезков реки, принадлежащих предприятиям, после каждого произошедшего события.

Требуется написать программу, которая по заданному начальному разделению реки между предприятиями и списку событий, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальный момент времени и после каждого события.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и p — исходное количество предприятий ($2 \leq n \leq 100000$) и номер подзадачи ($0 \leq p \leq 4$) (считайте его просто так).

Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n — длины исходных отрезков реки.

Третья строка входного файла содержит целое число k — количество событий, происходивших с предприятиями ($1 \leq k \leq 100000$).

Последующие k строк содержат описания событий, i -я строка содержит два целых числа: e_i и v_i — тип события и номер предприятия, с которым оно произошло. Значение $e_i = 1$ означает, что предприятие, которое после всех предыдущих событий является v_i -м по порядку, если считать с единицы от истока реки, обанкротилось, а значение $e_i = 2$ означает, что это предприятие разделилось на два.

Гарантируется, что значение v_i не превышает текущее количество предприятий. Гарантируется, что если отрезок предприятия при банкротстве или разделении требуется поделить на две части, то он имеет длину большую или равную 2. Гарантируется, что если на реке осталось единственное предприятие, оно не банкротится.

Формат выходных данных

Выходной файл должен содержать $(k + 1)$ целых чисел, по одному в строке. Первая строка должна содержать исходную сумму квадратов длин отрезков реки, а каждая из последующих k строк — сумму квадратов длин отрезков реки после очередного события.

Примеры

river.in	river.out
4 0	75
3 5 5 4	105
5	73
1 1	101
2 1	83
1 3	113
2 2	
1 3	

Задача Е. 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

holes.in	holes.out
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	

Задача F. 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 ландшафтными 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 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:

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