

Задача А. 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 В. 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 С. Варенье

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

Мальш и Карлсон решили пойти на прогулку. Они знают, что прогулка будет совсем скучной, если перед ней не опустошить несколько банок варенья.

Мальш достал из кладовки N банок варенья и выставил их в ряд. В банке номер i содержится ровно a_i грамм варенья. Карлсон немного подумал и решил, что в некоторых банках недостаточно варенья, и что в банке номер i должно быть хотя бы b_i грамм варенья.

Выходить из этой ситуации Карлсон хочет в M этапов. На каждом этапе он выбирает числа l , r , x и y , а затем выполняет следующие операции: в банку номер l он добавляет x грамм варенья, в банку номер $l + 1 - x + y$ грамм варенья, в банку номер $l + 2 - x + 2 \cdot y$, и так далее. В банку номер r наш герой добавит $x + y \cdot (r - l)$ грамм варенья.

Мальшу хочется определить для каждой банки i наименьший номер операции, после которой в ней станет хотя бы b_i грамм варенья. Помогите Мальшу: найдите соответствующее число для каждой банки.

Формат входных данных

В первой строке входного файла задано одно число N ($1 \leq n \leq 10^5$) — количество банок. Во второй строке заданы N чисел a_i ($0 \leq a_i \leq 2 \cdot 10^9$) — изначальное количество варенья в банке номер i . В третьей строке заданы N чисел b_i ($0 \leq b_i \leq 2 \cdot 10^9$) — минимальное количество варенья, которое должно быть в банке номер i .

В четвертой строке задано M ($0 \leq M \leq 10^5$) — число этапов добавления варенья в банки, которые выполнит Карлсон. В следующих M строках описаны сами этапы в хронологическом порядке. Каждый этап задан четырьмя числами l , r , x и y ($1 \leq l \leq r \leq N$, $0 \leq x, y \leq 3 \cdot 10^5$).

Формат выходных данных

Выведите N чисел в одной строке, разделенные пробелом. Число номер i должно быть равно нулю, если в банке номер i изначально было достаточно варенья, номеру этапа, после которого в ней станет хотя бы b_i варенья, или -1 , если даже после выполнения всех этапов, в этой банке будет недостаточно варенья. Этапы нумеруются с единицы.

Примеры

jam.in	jam.out
5	1 2 0 3 -1
5 4 4 2 1	
7 7 4 7 7	
3	
1 2 2 0	
2 5 1 1	
3 4 2 2	

Задача D. Мощные юнги

Имя входного файла: `power.in`
Имя выходного файла: `power.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Имеется список из n юнг, для каждого из которых известен его рост a_1, a_2, \dots, a_n . Рассмотрим некоторый его подсписок a_l, a_{l+1}, \dots, a_r , где $1 \leq l \leq r \leq n$, и для каждого натурального числа s обозначим через K_s число юнг с ростом s в этом подсписке. Назовем *мощностью* подсписка сумму произведений $K_s \cdot K_s \cdot s$ по всем различным натуральным s . Так как количество различных чисел в массиве конечно, сумма содержит лишь конечное число ненулевых слагаемых.

Необходимо вычислить мощности каждого из t заданных подсписков.

Формат входных данных

Первая строка содержит два целых числа n и t ($1 \leq n, t \leq 200000$) — длина списка и количество запросов соответственно.

Вторая строка содержит n натуральных чисел a_i ($1 \leq a_i \leq 10^6$) — рост юнг.

Следующие t строк содержат по два натуральных числа l и r ($1 \leq l \leq r \leq n$) — индексы левого и правого концов соответствующего подсписка.

Формат выходных данных

Выведите t строк, где i -ая строка содержит единственное натуральное число — мощность подсписка i -го запроса.

Примеры

<code>power.in</code>	<code>power.out</code>
3 2	3
1 2 1	6
1 2	
1 3	
8 3	20
1 1 2 2 1 3 1 1	20
2 7	20
1 6	
2 7	

Задача Е. Озера и лягушки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В Берендеевском лесу живет много лягушек. Кроме того, этот лес славится тем, что в нем находится n небольших озер, в котором и живут квакушки. Эти озера соединены между собой m ручейками. Чюваки, живущие неподалеку, уже обошли их все и точно знают, что ручеек не может течь из одного озера в него же и что между любыми двумя озерами течет не более одного ручейка.



В одну прекрасную среду вы пришли к этим озерам и заметили, что в каждом озере лягушки имеют собственный цвет. Понаблюдав еще немного, вы узнали, что иногда лягушки в каком-то из озер меняют цвет.

Вы решили посидеть еще немного с лягушками и поиграть в игру. А именно каждую минуту вы либо видите, что в озере с номером v лягушки поменяли свой цвет на цвет c , либо вы хотите посчитать для озера с номером v количество различных цветов лягушек в соседних озерах (соседними озерами вы называете те озера, которые соединены ручейком).

В конце игры у вас остался листочек с изначальным цветом лягушек в n озерах, а также q действий которые происходили в тот день.

Вам хочется еще поиграть в эту увлекательную игру, поэтому для простоты вы хотите написать программу, которая будет считать количество различных цветов лягушек в соседних озерах за вас.

Формат входных данных

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

Во второй строке через пробел даны n целых чисел $1 \leq c_1, \dots, c_n \leq n$ – изначальные цвета лягушек в озерах.

В следующих m строках заданы по два целых числа u и v – номера озер, которые соединены очередным ручейком ($1 \leq u, v \leq n, u \neq v$).

В следующих q строках заданы запросы. Каждый запрос имеет один из двух типов:

- $v c$ — поменять цвет лягушек в озере с номером v на цвет c ($1 \leq v, c \leq n$);
- v — вычислить количество различных цветов лягушек среди соседей озера под номером v ($1 \leq v \leq n$).

Формат выходных данных

Для каждого запроса второго типа выведите в отдельной строке ответ на него.

Пример

стандартный ввод	стандартный вывод
4 3 3	3
1 2 3 4	2
1 2	
1 3	
1 4	
2 1	
1 4 3	
2 1	

Задача F. Substring Query

Имя входного файла: `str-qry.in`
Имя выходного файла: `str-qry.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Пупы есть n строк S_1, S_2, \dots, S_n . Однажды его друг Лупа пришёл в бухгалтерию и попросил его ответить на q вопросов: сколько строк среди $S_{l_i}, S_{l_i+1}, \dots, S_{r_i}$ содержат P_i как подстроку?

Но в бухгалтерии все перепутали, и ответить на запросы за Пупу придется Вам.

Формат входных данных

На первой строке 2 целых числа n, q ($1 \leq n, q \leq 200\,000$). Каждая из следующих n строк содержит ровно одну строку S_i ($|S_1| + |S_2| + \dots + |S_n| \leq 200\,000$). Последние q строк содержат запросы. Каждый запрос задаётся двумя целыми числами l_i, r_i и строкой P_i . ($1 \leq l_i \leq r_i \leq n, |P_1| + |P_2| + \dots + |P_n| \leq 200\,000$) Все строки состоят из букв «a» и «b».

Формат выходных данных

На каждый вопрос выведите одно число — ответ на вопрос.

Примеры

<code>str-qry.in</code>	<code>str-qry.out</code>
4 2	2
a	2
b	
ab	
bab	
1 3 a	
1 4 ab	

Задача G. Бар у Мо

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

В баре у Мо дела идут как всегда прекрасно. Чтобы дела шли еще лучше, Мо озадачился вопросом разнообразия сортов пива, которые разливают в его баре. Для составления стратегического плана по улучшению списка сортов пива Мо должен получить ответы на ряд вопросов.



В данный момент у барной стойки сидят n посетителей и у каждого посетителя есть кружка с пивом одного из сортов. Мо пронумеровал посетителей слева направо числами от 1 до n . После этого он пронумеровал сорта пива числами от 1 до m . Далее Мо вспомнил число k , которое было рекомендовано в гороскопе на эту неделю. Наконец, Мо составил список из q вопросов, ответы на которые помогут его бизнесу стать еще успешней. В каждом вопросе спрашивается следующее: каково количество сортов пива, которые можно встретить ровно k раз в кружках посетителей с номерами от l до r включительно.

Формат входных данных

В первой строке даны четыре целых числа n, m, q и k ($1 \leq n, q \leq 2 \cdot 10^5, 1 \leq m \leq 10^6, 0 \leq k \leq 3$).

Во второй строке даны n целых чисел $1 \leq a_1, \dots, a_n \leq m$ – номера сортов пива у посетителей.

В следующих q строках заданы по два целых числа l и r , которые задают запрос для посетителей с номерами от l до r ($1 \leq l \leq r \leq n$).

Формат выходных данных

Для каждого запроса выведите в отдельной строке количество сортов пива, которые на отрезке от l до r встречаются ровно k раз.

Пример

стандартный ввод	стандартный вывод
5 5 2 2	2
1 2 2 1 5	1
1 5	
2 4	

Задача Н. 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	