

Задача А. 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 -го до R -го 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 -й до R -й (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 -м и R -м 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

fabro.in	fabro.out
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 В. Варенье

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

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

Малыш 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.

Малышу хочется определить для каждой банки i наименьший номер операции, после которой в ней станет хотя бы b_i грамм варенья. Помогите Малышу: 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 С. Инверсии на отрезке

Имя входного файла: `riq.in`
Имя выходного файла: `riq.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Дана перестановка из n элементов.

Ответьте на m запросов про число инверсий для подотрезка перестановки от l до r .

Инверсией называется пара индексов i, j такая, что $i < j$ и $a_i > a_j$, где a_i - это i -ый элемент перестановки.

Формат входных данных

В первой строке задано число n ($1 \leq n \leq 10^5$).

Во второй строке задана перестановка из n элементов (элементы перестановки — попарно различные целые числа от 1 до n).

В третьей строке задано число m ($1 \leq m \leq 10^5$).

В последующих m строках содержится по два числа, l и r , границы запроса ($1 \leq l, r \leq n$).

Формат выходных данных

Выведите m строк, ответы на данные запросы.

Примеры

riq.in	riq.out
5	2
4 5 2 3 1	2
3	8
1 3	
3 5	
1 5	
6	1
5 2 4 3 1 6	4
4	8
4 6	8
2 5	
1 5	
1 6	

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

power.in	power.out
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. Сычи

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

Мурка, не ходи, там сыч
На подушке вышит,
Мурка серый, не мурлычь,
Дедушка услышит.
Няня, не горит свеча,
И скребутся мыши.
Я боюсь того сыча,
Для чего он вышит?

Анна Ахматова

Есть n сычей. Известны пары друзей. Три сыча образуют компанию, если все три попарно дружат. Требуется найти количество компаний сычей.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество сычей и дружеских связей соответственно ($1 \leq n, m \leq 3 \cdot 10^5$). Каждая из следующих m строк содержит по два целых числа от 1 до n — номера сычей, которые дружат. Гарантируется, что никакая пара не указана два раза и сыч не дружит сам с собой.

Формат выходных данных

В единственной строке выходного файла выведите количество компаний сычей.

Примеры

<code>owls.in</code>	<code>owls.out</code>
6 6 1 2 2 3 3 1 4 2 3 4 5 1	2

Задача F. Машинное обучение

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

На курсе машинного обучения вам выдали первое домашнее задание — вам предстоит проанализировать некоторый массив из n чисел.

В частности, вы интересуетесь так называемой *равномерностью* массива. Предположим, что в массиве число b_1 встречается k_1 раз, b_2 — k_2 раз, и т.д. Тогда *равномерностью* массива называется такое минимальное целое число $c \geq 1$, что $c \neq k_i$ для любого i .

В рамках вашего исследования вы хотите последовательно проделать q операций.

- Операция $t_i = 1, l_i, r_i$ задаёт запрос исследования. Необходимо вывести равномерность массива, состоящего из элементов на позициях от l_i до r_i включительно.
- Операция $t_i = 2, p_i, x_i$ задаёт запрос уточнения данных. Начиная с этого момента времени p_i -му элементу массива присваивается значения x_i .

Формат входных данных

Первая строка содержит n и q ($1 \leq n, q \leq 100\,000$) — размер массива и число запросов соответственно.

Во второй строке записаны ровно n чисел — $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Каждая из оставшихся q строк задаёт очередной запрос.

Запрос первого типа задаётся тремя числами $t_i = 1, l_i, r_i$, где $1 \leq l_i \leq r_i \leq n$ — границы соответствующего отрезка.

Запрос второго типа задаётся тремя числами $t_i = 2, p_i, x_i$, где $1 \leq p_i \leq n$ — позиция в которой нужно заменить число, а $1 \leq x_i \leq 10^9$ — его новое значение

Формат выходных данных

Для каждого запроса первого типа выведите одно число — равномерность соответствующего отрезка массива.

Примеры

стандартный ввод	стандартный вывод
10 4	2
1 2 3 1 1 2 2 2 9 9	3
1 1 1	2
1 2 8	
2 7 1	
1 2 8	

Задача G. Нестабильность сети

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

Вася — системный администратор в большой компании под названием Глюкософт. В сети компании n компьютеров, и какие-то пары компьютеров напрямую соединены сетевыми кабелями, всего таких соединений ровно m , при чем никакой кабель не соединяет компьютер с самим собой, и между любой парой компьютеров не больше одного кабеля.

На каждом компьютере в Глюкософте установлена специальная программа, поддерживаемая разработчиками Глюкософта, под названием СетеБаг. Новые версии этой программы выходят почти каждый день, но система обновлений в компании совершенно непостижима. Так, обновление версии программы происходит не синхронно для всех компьютеров. Это происходит следующим образом — выбирается один компьютер, и на него устанавливается какая-то версия СетеБага. Поскольку система обновлений настроена очень странным образом, вполне возможно, что на компьютер установят не последнюю версию СетеБага, а какую-то другую, например, ту что уже установлена на нем, или даже какую-то более раннюю.

После нескольких месяцев работы, Вася обнаружил, что наиболее частая причина нестабильной работы сети в несовместимости версий СетеБага двух компьютеров в Глюкософте. Если два компьютера, напрямую соединенных кабелем, имеют установленный Сетебаг разных версий, то соединение между этими компьютерами *нестабильно*: попытка передачи данных между ними может вызвать отказ системы. Чем больше нестабильных соединений между компьютерами, тем больше вероятность отказа системы. С другой стороны, обмен данными между компьютерами с одинаковой версией СетеБага обычно не приводит ни к каким ошибкам.

Задача Васи — предотвратить и исправить ошибки в сети Глюкософта. Сделать он это может только если он находится на рабочем месте целый день. Вася хочет быть на работе, когда вероятность отказа системы достаточно высока. Он хочет распланировать свое расписание на следующий год, но оценка вероятностей отказа системы — довольно сложная задача.

Помогите Васе найти число нестабильных соединений между компьютерами после каждого обновления версий СетеБага.

Формат входных данных

В первой строке даны два целых числа n, m — количество компьютеров и соединений между ними ($1 \leq n, m \leq 10^5$).

Во второй строке даны n целых чисел $v_1, v_2, \dots, v_n$ — версии СетеБага, изначально установленные на компьютеры Глюкософта.

В следующих m строках даны пары целых чисел a_i, b_i — номера компьютеров, соединенных i -м кабелем ($1 \leq a_i, b_i \leq n, a_i \neq b_i$). Гарантируется, что никакие два компьютера не соединены больше, чем одним кабелем.

В следующей строке дано целое число q — количество запланированных обновлений версий СетеБага ($1 \leq q \leq 10^5$).

В следующих q строках даны пары целых чисел c_i, v_i — номер компьютера, на котором обновляется версия СетеБага, и новая версия СетеБага, которая установится на компьютер ($1 \leq c_i \leq n, 1 \leq v_i' \leq 10^5$). Все обновления даны в хронологическом порядке, и никакие два обновления не происходят одновременно.

Формат выходных данных

Выведите для каждого запроса изменения одно целое число — количество нестабильных соединений сразу после очередного обновления версии СетеБага.

Примеры

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

Задача Н. 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 Луна 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

str-qry.in	str-qry.out
4 2	2
a	2
b	
ab	
bab	
1 3 a	
1 4 ab	

Задача I. Японский кроссворд

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Дан ряд из n квадратов, пронумерованных от 1 до n , и последовательность a длины n , состоящая из целых положительных чисел. Изначально первый квадрат покрашен в черный цвет, а остальные — в белый. На первом квадрате расположена фишка.

Вы можете повторить следующую операцию любое количество раз, возможно, нулевое:

- Если фишка находится в квадрате с номером i , выберите целое положительное целое число x и переместите фишку в квадрат $i + a_i \cdot x$. Обратите внимание, вы не можете сделать ход, если $i + a_i \cdot x > n$.
- Покрасьте квадрат $i + a_i \cdot x$ в черный цвет.

Найдите количество различных строк, которые могут получиться после выполнения операций, по модулю 998244353.

Формат входных данных

В первой строке вводится одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина строки и последовательности.

Во второй строке вводится n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 2 \cdot 10^5$) — последовательность a .

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 1 1	8
1 200000	1

Замечание

В первом примере есть 8 возможных рядов, где в черный цвет покрашены квадраты:

- 1
- 1, 2
- 1, 2, 4
- 1, 2, 4, 5
- 1, 3
- 1, 4
- 1, 4, 5
- 1, 5

Задача J. Охота за общим префиксом

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Строка x называется *префиксом* строки y , если x может быть получен удалением последних букв y ноль или более раз. Например, строка “abac” имеет следующие префиксы: “abac”, “aba”, “ab”, “a”, и пустую строку.

Для двух строк x и y , обозначим $\text{LCP}(x, y)$ как длину наибольшего общего префикса строк x и y . Например, $\text{LCP}(\text{abacab}, \text{abacbbba}) = 4$ потому что наибольший общий префикс этих двух строк “abac”. Заметим, что $\text{LCP}(x, y)$ определен для любых строк x и y потому что пустая строка всегда является общим префиксом.

Вам даны n строк $s_1, \dots, s_n$ и m строк $t_1, \dots, t_m$ содержащие маленькие английские буквы. Потом вам дано q запросов. В каждом запросе вам дана последовательность $a_1, \dots, a_k$. Пусть u будет конкатенацией $t_{a_1}, \dots, t_{a_k}$. Ваша задача посчитать $\sum_{i=1}^n \text{LCP}(u, s_i)$.

Формат входных данных

Первая строка содержит одно число n ($1 \leq n \leq 2 \cdot 10^5$). Каждая из следующих n строк содержит не пустую строку s_i состоящую из маленьких английских букв.

Следующая строка содержит одно число m ($1 \leq m \leq 2 \cdot 10^5$). Каждая из следующих m строк содержит не пустую строку t_j состоящую из маленьких английских букв.

Следующая строка содержит одно число q , число запросов ($1 \leq q \leq 2 \cdot 10^5$). Затем следует q запросов. Каждый запрос описывается одной строкой как “ $k \ a_1 \ \dots \ a_k$ ” ($1 \leq k \leq 2 \cdot 10^5$; $1 \leq a_i \leq m$).

Сумма длин s_i не превосходит $2 \cdot 10^5$. Сумма длин t_i не превосходит $2 \cdot 10^5$. Сумма k По всем запросам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Выведите q строк. В i -м должно быть одно число: ответ на i -й запрос.

Пример

стандартный ввод	стандартный вывод
5	4
abcde	9
aaa	3
a	1
ab	0
bcd	
5	
a	
bc	
de	
aaaa	
b	
5	
1 1	
3 1 2 3	
2 2 3	
5 5 4 3 2 1	
3 3 3 3	

Задача К. Расширение Всадников

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Четыре Всадника продолжают свои тщательно спланированные представления по всему миру. Они то исчезают, то снова появляются на сцене, оставляя за собой множество загадок.

Рассмотрим динамическое множество команд иллюзионистов. Каждая команда с номером i характеризуется двумя параметрами x_i и k_i (обратите внимание, что оба параметра у двух разных команд могут совпадать).

Начиная с дня $t = 1$, команда живёт по бесконечному циклу:

- в течение первых k_i дней команда активно выступает с интенсивностью x_i ;
- затем следующие k_i дней команда полностью исчезает, готовя новые фокусы и не внося никакого вклада в общую «интенсивность магии».

Затем цикл повторяется: k_i дней выступлений, k_i дней тишины, и так далее.

Необходимо обрабатывать запросы от разных наблюдателей (Интерпол, полиция, фанаты и таинственные заказчики), которые хотят знать, какая суммарная «интенсивность магии» наблюдается в тот или иной день.

Поддерживаются три типа запросов:

1. $+ x k$ — на сцену выходит новая команда иллюзионистов с параметрами x и k ;
2. $- x k$ — команда с параметрами x и k прекращает участвовать в игре;
3. $? t$ — требуется узнать суммарную интенсивность всех команд в день с номером t .

Помогите наблюдателям узнавать ответы на все запросы третьего типа.

Формат входных данных

Первая строка содержит одно целое число q — количество запросов ($1 \leq q \leq 4 \cdot 10^5$).

Каждая из следующих q строк содержит один запрос одного из трёх описанных выше типов.

В запросах 1 и 2 типа заданы целые числа x и k , они описывают добавление и удаление команды с параметрами x и k соответственно ($1 \leq x \leq 10^3$; $1 \leq k \leq 2 \cdot 10^5$).

В запросе третьего типа задано целое число t — номер дня ($1 \leq t \leq 2 \cdot 10^5$).

Гарантируется, что для каждого запроса второго типа соответствующая команда ранее была добавлена запросом первого типа и к этому моменту ещё не была удалена.

Формат выходных данных

Для каждого запроса третьего типа выведите в отдельной строке одно целое число — суммарную интенсивность магии в интересующий день.

Примеры

стандартный ввод	стандартный вывод
5 + 1 1 ? 1 ? 2 ? 3 ? 4	1 0 1 0
12 + 3 2 + 5 3 ? 1 ? 2 ? 3 + 4 2 ? 4 - 5 3 ? 5 ? 6 + 2 4 ? 7	8 8 5 0 7 7 0