

Задача А. RMQ Наоборот

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

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.

Формат входных данных

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 $Q(i, j) = q$ ($1 \leq i \leq j \leq n, -2^{31} \leq q \leq 2^{31} - 1$).

Формат выходных данных

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

<code>rmq.in</code>	<code>rmq.out</code>
3 2 1 2 1 2 3 2	<code>consistent</code> 1 2 2
3 3 1 2 1 1 1 2 2 3 2	<code>inconsistent</code>

Задача В. Различные числа

Имя входного файла: `distinctnum.in`
Имя выходного файла: `distinctnum.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Сколько различных чисел на отрезке массива?

Формат входных данных

На первой строке длина массива n ($1 \leq n \leq 300\,000$). На второй строке n целых чисел от 0 до 10^9-1 . На третьей строке количество запросов q ($1 \leq q \leq 300\,000$). Следующие q строк содержат описание запросов, по одному на строке. Каждый запрос задаётся парой целых чисел l, r ($1 \leq l \leq r \leq n$).

Формат выходных данных

Выведите ответы на запросы по одному в строке.

Примеры

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

Задача С. Перестановки

Имя входного файла: permutation.in
Имя выходного файла: permutation.out
Ограничение по времени: 1.5 секунды
Ограничение по памяти: 256 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Примеры

permutation.in	permutation.out
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Задача D. Присваивание, прибавление и сумма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Есть массив из n элементов, изначально заполненный нулями. Вам нужно написать структуру данных, которая обрабатывает три вида запросов:

- присвоить всем элементам на отрезке от l до $r - 1$ значение v ,
- прибавить ко всем элементам на отрезке от l до $r - 1$ число v ,
- узнать сумму на отрезке от l до $r - 1$.

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 100000$) — размер массива и число операций. Далее следует описание операций. Описание каждой операции имеет следующий вид:

- $1\ l\ r\ v$ — присвоить всем элементам на отрезке от l до $r - 1$ значение v ($0 \leq l < r \leq n$, $0 \leq v \leq 10^5$).
- $2\ l\ r\ v$ — прибавить ко всем элементам на отрезке от l до $r - 1$ число v ($0 \leq l < r \leq n$, $0 \leq v \leq 10^5$).
- $3\ l\ r$ — узнать сумму на отрезке от l до $r - 1$ ($0 \leq l < r \leq n$).

Формат выходных данных

Для каждой операции третьего типа выведите соответствующее значение.

Примеры

стандартный ввод	стандартный вывод
5 7	8
1 0 3 3	10
2 2 4 2	4
3 1 3	
2 1 5 1	
1 0 2 2	
3 0 3	
3 3 5	

Задача Е. Марио и трубы

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

Марио собирается проходить уровень, состоящий из N последовательно расположенных труб, высота i -й трубы — a_i . Он еще не знает, где он будет располагаться изначально, и куда ему надо добраться, поэтому хочет рассмотреть несколько вариантов.

Находясь на трубе, Марио может переместиться только на соседние трубы слева и справа (если они существуют). Спускаться он может с любой высоты, также он может перемещаться между одинаковыми трубами. Подниматься Марио может только на трубу, высота которой больше высоты текущей на 1. Более формально, Марио может переместиться с трубы i на трубу j , если $|i - j| = 1$ и $a_j - a_i \leq 1$.

Однако злой динозавр Боузер хочет помешать Марио пройти уровень, для чего иногда увеличивает высоту нескольких подряд идущих труб на одно число k . Теперь Марио не может понять, удастся ли ему пройти уровень и поэтому просит вас обрабатывать два типа запросов — Боузер изменяет высоту некоторых труб, и Марио пытается пройти от одной трубы до другой.

Формат входных данных

В первой строке заданы два целых числа N и M — число труб и число запросов соответственно ($2 \leq N \leq 3 \cdot 10^5, 1 \leq M \leq 10^6$).

Следующая строка содержит N целых чисел a_i — высоты труб на уровне ($1 \leq a_i \leq 10^9$).

Далее идут M строк, содержащие описание запросов. Каждая строка имеет вид:

- $1 \ x \ y$ — может ли Марио пройти от трубы с номером x до трубы с номером y ($1 \leq x, y \leq N$). Гарантируется, что номера x и y не совпадают.
- $2 \ l \ r \ d$ — Боузер увеличивает высоты труб с l -й до r -й на величину d ($1 \leq l \leq r \leq N, -10^9 \leq d \leq 10^9$).

Формат выходных данных

Для каждого запроса первого типа нужно на отдельной строке вывести «YES», если Марио может прийти от одной трубы до другой и «NO» в противном случае (без кавычек).

Примеры

стандартный ввод	стандартный вывод
5 7	YES
1 2 3 4 5	NO
1 5 1	NO
2 2 4 3	YES
1 5 4	NO
1 1 3	
2 2 3 3	
1 2 4	
1 1 3	

Задача F. Окна

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

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

Формат входных данных

В первой строке входного файла записано число окон n ($1 \leq n \leq 50\,000$). Следующие n строк содержат координаты окон $x_{(1,i)}$ $y_{(1,i)}$ $x_{(2,i)}$ $y_{(2,i)}$, где $(x_{(1,i)}, y_{(1,i)})$ — координаты левого верхнего угла i -го окна, а $(x_{(2,i)}, y_{(2,i)})$ — правого нижнего (на экране компьютера y растет сверху вниз, а x — слева направо). Все координаты — целые числа, по модулю не превосходящие 10^6 .

Формат выходных данных

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенных пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

Примеры

windows.in	windows.out
2 0 0 3 3 1 1 4 4	2 1 3
1 0 0 1 1	1 0 1
4 0 0 1 1 0 1 1 2 1 0 2 1 1 1 2 2	4 1 1
5 0 0 1 1 0 1 1 2 0 0 2 2 1 0 2 1 1 1 2 2	5 1 1

Задача G. Прибавление и максимум

Имя входного файла: `addandmax.in`
Имя выходного файла: `addandmax.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Реализуйте эффективную структуру данных для хранения массива и выполнения следующих операций: увеличение всех элементов данного интервала на одно и то же число; поиск максимума на интервале.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) – количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 – элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30000$) – количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (m – найти максимум на отрезке (оба конца включительно), a – увеличить все элементы на отрезке, то есть все элементы с l -го по r -й).

Следом за m вводятся два числа – левая и правая граница отрезка.

Следом за a вводятся три числа – левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($0 \leq add \leq 100000$).

Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос m .

Примеры

<code>addandmax.in</code>	<code>addandmax.out</code>
5	4
2 4 3 1 5	104
5	104
m 1 3	
a 2 4 100	
m 1 3	
a 5 5 10	
m 1 5	

Задача Н. Объединение отрезков

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Решая задачу из контрольной по математике, Вася получил ответ в виде объединения N отрезков $[L_i, R_i]$ на числовой прямой. Однако, некоторые из этих отрезков могут пересекаться друг с другом, что не слишком нравится Васе. Ваша задача — представить Васин ответ в виде объединения минимального количества отрезков.

Формат входных данных

В первой строке указано число N ($1 \leq N \leq 50000$). В следующих N строках перечислены пары целых чисел L_i и R_i ($|L_i|, |R_i| \leq 50000$), каждая пара с новой строки, числа в парах отделены друг от друга одним или несколькими пробелами.

Формат выходных данных

В первой строке выведите число M — количество отрезков в искомом объединении. В следующих M строках выведите сами эти отрезки в том же формате, что и во входном файле. Список отрезков необходимо упорядочить по возрастанию левого конца.

Примеры

<code>stdin</code>	<code>stdout</code>
4	2
0 2	0 3
4 5	4 6
1 3	
5 6	

Задача I. Дорешивание

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как известно, после обеда в ЛКШ проходит много интересных мероприятий, но все равно каждый ЛКШОнок старается хотя бы ненадолго заглянуть в комповник, чтобы дорешать задачи, не сделанные во время практики.

В этом году погода стоит особо жаркая, поэтому в комповнике очень душно и важно следить за тем, чтобы в комповнике не находилось одновременно очень много школьников. Поэтому завуч записал время прихода и ухода из комповника каждого ЛКШОнка.

Теперь завуч хочет узнать, сколько ЛКШАт встретил в комповнике каждый ЛКШОнок.

Формат входных данных

В первой строке записано количество ЛКШАт N ($1 \leq N \leq 10^5$). В i -й из следующих N строк через пробел записаны целые числа S_i и T_i ($0 \leq S_i \leq T_i \leq 10^9$) — время прихода в комповник и ухода из него i -го ЛКШОнка.

Формат выходных данных

Программа должна вывести N целых чисел, i -е число должно быть равно количеству ЛКШАт, которых встретил в комповнике i -й ЛКШОнок.

Если в некоторый момент времени один ЛКШОнок приходит в комповник, а другой уходит из него, то они встречаются друг с другом.

Примеры

<code>stdin</code>	<code>stdout</code>
4	3
1 10	3
2 5	2
5 6	2
1 4	

Замечание

Ограничение времени работы для языка Python — 4 секунды.

Задача J. Кассы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 64 мегабайта

На одном из московских вокзалов билеты продают N касс. Каждая касса работает без перерыва определенный промежуток времени по фиксированному расписанию (одному и тому же каждый день). Требуется определить, на протяжении какого времени в течение суток работают все кассы одновременно.

Формат входных данных

Сначала вводится одно целое число N ($0 < N \leq 10000$).

В каждой из следующих N строк через пробел расположены 6 целых чисел, первые три из которых обозначают время открытия кассы в часах, минутах и секундах (часы — целое число от 0 до 23, минуты и секунды — целые числа от 0 до 59), оставшиеся три — время закрытия в том же формате. Числа разделены пробелами.

Время открытия означает, что в соответствующую ему секунду касса уже работает, а время закрытия — что в соответствующую секунду касса уже не работает. Например, касса, открытая с 10 ч 30 мин 30 с до 10 ч 35 мин 30 с, ежедневно работает 300 секунд.

Если время открытия совпадает с временем закрытия, то касса работает круглосуточно. Если первое время больше второго, то касса начинает работу до полуночи, а заканчивает — на следующий день.

Формат выходных данных

Требуется вывести одно число — суммарное время за сутки (в секундах), на протяжении которого работают все N касс.

Примеры

stdin	stdout
3 1 0 0 23 0 0 12 0 0 12 0 0 22 0 0 2 0 0	7200
2 9 30 0 14 0 0 14 15 0 21 0 0	0
2 14 0 0 18 0 0 10 0 0 14 0 1	1

Задача К. Точки и отрезки [геометрия HARD]

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано n отрезков на числовой прямой и m точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b , если выполняется двойное неравенство $\min(a, b) \leq x \leq \max(a, b)$.

Формат входных данных

Первая строка содержит два целых числа n ($1 \leq n \leq 50000$) — число отрезков и m ($1 \leq m \leq 50000$) — число точек. В следующих n строках по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходных данных

В выходной файл выведите m чисел — для каждой точки количество отрезков, в которых она содержится.

Примеры

stdin	stdout
3 2	2
0 5	0
-3 2	
7 10	
1 6	
1 3	0
10 -10	0
-100 100 0	1

Задача L. Операционные системы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Васин жесткий диск состоит из M секторов. Вася последовательно устанавливал на него различные операционные системы следующим методом: он создавал новый раздел диска из последовательных секторов, начиная с сектора номер a_i и до сектора b_i включительно, и устанавливал на него очередную систему. При этом если очередной раздел хотя бы по одному сектору пересекается с каким-то ранее созданным разделом, то ранее созданный раздел «затирается», и операционная система, которая на него была установлена, больше не может быть загружена.

Напишите программу, которая по информации о том, какие разделы на диске создавал Вася, определит, сколько в итоге работающих операционных систем установлено и в настоящий момент работает на Васином компьютере.

Формат входных данных

Сначала вводятся натуральное число M — количество секторов на жестком диске ($1 \leq M \leq 10^9$) и целое число N — количество разделов, которое последовательно создавал Вася ($0 \leq N \leq 100000$). Далее идут N пар чисел a_i и b_i , задающих номера начального и конечного секторов раздела ($1 \leq a_i \leq b_i \leq M$).

Формат выходных данных

Выведите одно число — количество работающих операционных систем на Васином компьютере.

Примеры

стандартный ввод	стандартный вывод
10 3 1 3 4 7 3 4	1
10 4 1 3 4 5 7 8 4 6	3

Задача М. Китайские часы

Имя входного файла: `watches.in`
Имя выходного файла: `watches.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Когда программист Вова был в Китае, он обнаружил, что российские часы “Заря” продаются там в десять раз дешевле, чем в России. Вова решил проверить выгодное дело и закупил огромную партию с тем, чтобы привезти часы на родину и продать их за полцены (то есть впятеро дороже, чем купил). Но дома обнаружилось, что многие часы идут вразнобой, к тому же, от малейшего толчка часы останавливаются (либо наоборот, начинают идти). Очевидно, это были не настоящие часы фирмы “Заря”, а их точные копии. Чтобы быстро продать всю партию часов, Вова хочет поставить их все на одинаковое время (тогда будет неважно, правильное это время или нет — можно будет сказать, что это местное время завода-изготовителя) и перед открыванием крышки просто встряхнуть чемодан, чтобы часы одновременно пошли.

Для установки времени на часах есть заводная головка, вращение которой позволяет быстро поворачивать стрелки, при этом часовая стрелка крутится в 60 раз медленнее минутной, а минутная — в 60 раз медленнее секундной. Один оборот заводной головки поворачивает секундную стрелку на полный оборот, и хотя на такой поворот уходит всего одна секунда, чтобы изменить время на 6 часов, потребуется 6 минут. Вращать заводную головку и стрелки можно только по часовой стрелке, чтобы не повредить хрупкий механизм часов. Помогите Вова минимизировать усилия по предпродажной подготовке часов, выбрав время, которое будет установлено на всех часах.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leq n \leq 50000$) — количество часов в партии. Далее следует n строк, $(i+1)$ -я строка ввода содержит время на i -х часах в формате $h : mm : ss$, где целое число h ($1 \leq h \leq 12$) означает час, а двухразрядные целые числа mm и ss ($00 \leq mm, ss \leq 59$) — минуты и секунды соответственно.

Формат выходных данных

Выведите время, которое нужно установить на всех часах, в формате, указанном выше.

Примеры

<code>watches.in</code>	<code>watches.out</code>
3 11:30:00 12:10:01 6:10:18	12:10:01

Задача N. Объединение прямоугольников

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На плоскости задано N прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти площадь их объединения.

Формат входных данных

В первой строке входного файла указано число N ($1 \leq N \leq 3500$). В следующих N строках заданы по 4 целых числа x_1, y_1, x_2, y_2 — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ($0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

Формат выходных данных

В выходной файл выведите единственное число — ответ на задачу.

Примеры

<code>stdin</code>	<code>stdout</code>
3 1 1 3 5 5 2 7 4 2 4 6 7	23

Задача О. Объединение прямоугольников

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На плоскости задано N прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти площадь их объединения.

Формат входных данных

В первой строке входного файла указано число N ($1 \leq N \leq 100'000$). В следующих N строках заданы по 4 целых числа x_1, y_1, x_2, y_2 — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ($0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

Формат выходных данных

В выходной файл выведите единственное число — ответ на задачу.

Примеры

<code>stdin</code>	<code>stdout</code>
3 1 1 3 5 5 2 7 4 2 4 6 7	23