

Задача А. 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

stdin	stdout
4	2
0 2	0 3
4 5	4 6
1 3	
5 6	

Задача В. 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 0 5 -3 2 7 10 1 6	2 0
1 3 10 -10 -100 100 0	0 0 1

Задача С. 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

stdin	stdout
4	3
1 10	3
2 5	2
5 6	2
1 4	

Замечание

Ограничение времени работы для языка Python — 4 секунды.

Задача D. 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 Е. Посчитай покрытие

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

Задано n отрезков на координатной прямой; концы отрезков имеют целочисленные координаты. Отрезки могут вырождаться в точки. Отрезки могут пересекаться друг с другом, вкладываться и даже совпадать.

Ваша задача: для каждого $k \in [1, n]$ посчитать количество таких точек с целочисленными координатами, что количество отрезков, покрывающих эти точки, было не меньше k . Отрезок с границами l_i и r_i покрывает точку x тогда и только тогда, когда $l_i \leq x \leq r_i$.

Формат входных данных

Первая строка входных данных содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество отрезков.

Следующие n строк содержат отрезки. Строка с индексом i состоит из пары целых чисел l_i, r_i ($0 \leq l_i \leq r_i \leq 10^{18}$) — концы i -го отрезка.

Формат выходных данных

Выведите n целых чисел через пробел $cnt_1, cnt_2, \dots, cnt_n$, где cnt_i равно количеству таких точек, что количество отрезков, покрывающих эти точки, не меньше i .

Примеры

стандартный ввод	стандартный вывод
3 0 3 1 3 3 8	9 3 1
3 1 3 2 4 5 7	7 2 0

Задача F. Минимальное покрытие

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

На прямой задано некоторое множество отрезков с целочисленными координатами концов $[L_i, R_i]$. Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок $[0, M]$, (M — натуральное число), содержащее наименьшее число отрезков.

Формат входных данных

В первой строке указана константа M ($1 \leq M \leq 50000$). В каждой последующей строке записана пара чисел L_i и R_i ($|L_i|, |R_i| \leq 50000$), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 1 000 000.

Формат выходных данных

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка $[0, M]$. Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно.

Если покрытие отрезка $[0, M]$ исходным множеством отрезков $[L_i, R_i]$ невозможно, то следует вывести единственную фразу “No solution”.

Примеры

cover.in	cover.out
1 -1 0 -5 -3 2 5 0 0	No solution
1 -1 0 0 1 0 0	1 0 1

Замечание

Ограничение времени работы для языка Python — 3 секунды.

Задача G. Лягушачья диета

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

На пруду устроены соревнования для лягушек по ловле мошек, которые проводятся в один заход и по необычным правилам. Мошки прилетают к лягушке не все сразу, а по очереди: каждая i -я мошка ($1 \leq i \leq n$) появляется над прудом в свой момент времени s_i . Когда прилетает очередная мошка, лягушка должна сразу решить, будет она её ловить или нет. Если лягушка выбирает эту мошку, то у неё уходит t_i минут на то, чтобы её поймать и проглотить, причём в течение этого времени она не может переключиться на ловлю другой мошки. Если же лягушка отказывается от этой мошки, то в дальнейшем вернуться к ней уже нельзя. В тот момент, когда заканчивается время, потраченное на ловлю текущей мошки, лягушка может начать ловить другую мошку, появившуюся ровно в этот же момент, если такая есть, или ждать появления следующей мошки. При этом за успешно пойманную i -ю мошку лягушка получает c_i очков сытости.

Лягушка Артур, представляющая на соревнованиях один из региональных прудов, понимает, что здесь важную роль играет не только умение ловить мошек, но и правильный стратегический расчёт того, каких мошек надо ловить, а каких пропустить. Ему, как и всем участникам, до начала заплыва известно, в какой момент времени появится каждая мошка, сколько времени займёт её ловля и сколько очков сытости за неё можно получить. Артур является очень ловкой лягушкой и поэтому гарантированно успеет поймать и проглотить любую мошку, которую выберет для ловли на соревновании.

Требуется написать программу, которая определяет, какое максимальное количество очков сытости Артур сможет получить при оптимальном выборе мошек, которых он будет ловить, а также количество и перечень таких мошек.

Формат входных данных

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 100000$) — количество мошек на соревновании.

Последующие n строк содержат описания мошек, по три числа на каждой строке: s_i — момент появления i -й мошки в минутах, t_i — время, необходимое на её ловлю в минутах, и c_i — сколько очков сытости получит лягушка за её поимку ($1 \leq s_i, t_i, c_i \leq 10^9$).

В данной задаче все c_i одинаковы.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — максимальное количество очков сытости, которое сможет получить Артур на соревновании. Вторая строка должна содержать одно целое число m — количество мошек, которых надо поймать при оптимальном выборе. Третья строка должна содержать m разделённых пробелом целых чисел — номера этих мошек в порядке их ловли. Мошки пронумерованы, начиная с единицы, в порядке их описания во входном файле. Если оптимальных ответов несколько, необходимо вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
5	12
1 2 6	2
2 3 6	1 4
1 2 6	
3 1 6	
3 2 6	

Задача Н. 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

stdin	stdout
3 1 1 3 5 5 2 7 4 2 4 6 7	23

Задача I. Радиочастоты

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

В городе Радиоград работает ровно n радиостанций. Городской регулятор частот выделил каждой станции собственный диапазон вещания: i -я станция вещает на всех частотах с a_i по b_i килогерц включительно, причём $a_i < b_i$.

Регулятор строго следит за порядком в эфире. Он считает, что станция j перекрывает станцию i , если диапазон станции j строго содержит внутри себя диапазон станции i , то есть $a_j < a_i$ и $b_i < b_j$. Станцию, которую перекрывает хотя бы одна другая станция, регулятор помечает как заглушённую: её всё равно не слышно за более мощным соседом.

Чтобы навести порядок в эфире, регулятору нужно узнать, сколько станций оказались заглушёнными. Помогите ему посчитать это число.

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 10^5$) — количество радиостанций.

В следующих n строках записаны описания диапазонов вещания, по одному на строке. В $(i+1)$ -й строке находятся два целых числа a_i и b_i ($1 \leq a_i < b_i \leq 10^9$) — нижняя и верхняя границы диапазона i -й станции в килогерцах.

Гарантируется, что все $2n$ значений границ различны, то есть $a_i \neq a_j$, $a_i \neq b_j$, $b_i \neq a_j$ и $b_i \neq b_j$ для всех $i \neq j$. Станции заданы в произвольном порядке.

Формат выходных данных

Выведите единственное целое число — количество заглушённых станций, то есть станций, диапазон каждой из которых строго содержится внутри диапазона хотя бы одной другой станции.

Примеры

стандартный ввод	стандартный вывод
5 1 10 2 9 3 8 4 7 5 6	4
5 1 100 2 50 51 99 52 98 10 60	4
1 1 1000000000	0

Замечание

В первом примере диапазон пятой станции содержится в диапазоне четвёртой, четвёртой — в третьей, третьей — во второй, а второй — в первой. Заглушены четыре станции.

Во втором примере все станции, кроме первой, содержатся в диапазоне первой станции $[1, 100]$, поэтому заглушёнными оказываются четыре станции.

В третьем примере станция всего одна, перекрывать её некому, поэтому ответ равен нулю.

Задача J. Прыжки по кувшинкам

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

На большом болоте живут лягушки, которые прыгают по кувшинкам, растущим вдоль одного длинного узкого канала. Кувшинки расположены в ряд, и лягушки перемещаются с одной кувшинки на другую строго вдоль этого канала. Канал узкий, поэтому в один момент времени по нему может двигаться только одна лягушка — совсем как однопутная дорога.

Каждый день на болоте длится h часов, каждый час состоит из m минут, причём m обязательно чётное. Известно, что сейчас n старых лягушек совершают свой ежедневный прыжок по разу в день: i -я лягушка прыгает в h_i часов и m_i минут.

К началу брачного сезона решено выпустить на болото молодых лягушек, которые тоже будут прыгать вдоль канала. Их расписание нужно согласовать со старыми лягушками, потому что канал узкий.

Из-за высокой активности молодых лягушек решено, чтобы ровно 2 молодые лягушки прыгали каждый час. Более того, по законам болота промежутки времени между прыжками молодых лягушек должны быть равны $\frac{m}{2}$ минутам: предыдущая молодая лягушка прыгает ровно за $\frac{m}{2}$ минут до данной, а следующая — ровно через $\frac{m}{2}$ минут после. Кроме того, молодую лягушку надо посадить на стартовую кувшинку за k минут до прыжка. Пока молодая лягушка сидит на стартовой кувшинке, с неё не могут прыгать старые лягушки. При этом разрешается сажать молодую лягушку на кувшинку в ту же самую минуту, когда с неё прыгает предыдущая старая лягушка. А также прыжки происходят настолько быстро, что разрешено запустить молодую лягушку и следующую за ней старую лягушку в одну и ту же минуту.

Поскольку молодые лягушки прыгают каждый день с интервалом $\frac{m}{2}$ минут, первая молодая лягушка прыгнет в 0 часов и t минут, причём $t < \frac{m}{2}$. Обратите внимание, что если $t < k$, то стартовая кувшинка будет занята ещё и последние $k - t$ минут предыдущего дня.

Разумеется, не всегда удаётся составить корректное расписание прыжков молодых лягушек вместе со всеми старыми. Тогда придётся отменить прыжки некоторых старых лягушек, чтобы стало возможным запустить молодых с соблюдением всех законов болота. При этом, чтобы не разгневать старых лягушек слишком сильно, нужно отменить как можно меньше их прыжков. Составление такого расписания оказалось трудной задачей, поэтому это было поручено вам. Найдите такое время t для запуска первой молодой лягушки, при котором число отменяемых прыжков старых лягушек минимально.

Формат входных данных

Первая строка содержит четыре целых числа n , h , m , k ($1 \leq n \leq 100\,000$, $1 \leq h \leq 10^9$, $2 \leq m \leq 10^9$, $1 \leq k \leq \frac{m}{2}$) — число старых лягушек, количество часов и минут на болоте, а также время, которое молодая лягушка сидит на стартовой кувшинке. Гарантируется, что число минут m чётное.

В следующих n строках вводится по два целых числа h_i и m_i ($0 \leq h_i < h$, $0 \leq m_i < m$) — время прыжка i -й старой лягушки, часы и минуты соответственно. Гарантируется, что все старые лягушки прыгают в разное время.

Формат выходных данных

Выведите два числа: минимальное количество отменённых прыжков старых лягушек и t — время запуска первой за день молодой лягушки в минутах.

Во второй строке через пробел выведите номера старых лягушек, чьи прыжки отменяются.

Примеры

стандартный ввод	стандартный вывод
2 24 60 15 16 0 17 15	0 0
2 24 60 16 16 0 17 15	1 0 2

Замечание

В первом примере первую молодую лягушку надо запустить в 0:00. Тогда старая лягушка в 16:00 прыгнет сразу после молодой, а молодую лягушку в 17:30 надо будет посадить на стартовую кувшинку сразу после прыжка старой лягушки в 17:15.

Во втором примере посадить молодую лягушку на кувшинку надо за 16 минут до её прыжка. Сделать это без отмены прыжка какой-либо старой лягушки не получится: если запускать молодую лягушку в $t \in [1, 15]$, то к 16:00 она уже должна сидеть на стартовой кувшинке, а с этой кувшинки в это время прыгает первая старая лягушка. Если $t \in \{0, [16, 29]\}$, то вторая старая лягушка в 17:15 не сможет прыгнуть с кувшинки, потому что на ней уже будет сидеть следующая молодая лягушка.

Если отменить прыжок второй старой лягушки, то можно выбрать $t = 0$, тогда молодые лягушки будут прыгать в 0 и 30 минут каждый час, а конфликта с первой старой лягушкой не будет. Также можно отменить только прыжок первой старой лягушки и выбрать, например, $t = 13$.

Задача К. Перемежающиеся строки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Назовём строку из цифр $d_1d_2 \dots d_k$ перемежающейся, если цифры на чётных и нечётных позициях образуют два непересекающихся множества.

Например, 121212 и 01234567 — перемежающиеся, а 11 и 5435 — нет.

Ваша программа должна по строке из цифр найти самую длинную перемежающуюся подстроку. В случае, если решений несколько, выведите самое левое из них.

Формат входных данных

Входной файл содержит строку из цифр длиной от 1 до 30 000 символов.

Формат выходных данных

Выходной файл должен содержать два целых числа P и L — позиция и длина перемежающейся подстроки.

Примеры

<code>input.txt</code>	<code>output.txt</code>
7777	1 1
01234567802345	2 13
9	1 1

Задача L. Проклятие Черной жемчужины

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

Всем известно, что корабль «Черная жемчужина» на самом деле существует. Долгое время им командовал всем известный капитан Джек Воробей. И этот корабль, вместе со своим капитаном прошел огонь, воду и медные трубы. Так же учеными установлено, что «Черная жемчужина» является самым быстроходным кораблем в мире. Она даже быстрее, чем «Разящий» и «Летучий голландец», про который говорят, что он ходит быстрее ветра.

После ограбления сокровищницы на Исла де Муэрте на команду «Чёрной Жемчужины» легло проклятие, которое повлияло и на сам корабль: паруса корабля порвались, а само судно стал окружать жутковатый туман. Снять проклятие было достаточно сложно, и поэтому никто не стал этого делать.

В трюме корабля, на самой гнилой доске самой дальней стены, написана загадка, которая является ключом к снятию проклятия. Доска очень старая, и из-за этого некоторые буквы на ней стерлись. Согласно легендам, в загадке не было пробелов, то есть она выглядела как одно слово. Поскольку загадку восстановить уже нельзя, снять проклятие не представляется возможным. Однако, его можно попробовать смягчить.

Если верить Мудрецу, для смягчения проклятия нужно выбросить за борт большой мешок с золотом. За каждую подстроку слова-загадки, в которой, при каком-нибудь заполнении пропусков буквами, все буквы могли оказаться одинаковыми, в мешок необходимо положить одну монету.

Вам дана загадка, определите, сколько монет нужно выбросить за борт для смягчения проклятия.

Формат входных данных

Во входном файле дана строка, длина которой не превышает 10^6 . Строка состоит из строчных латинских букв и знаков вопроса, обозначающих стертую букву (пропуск).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

<code>theblackpearl.in</code>	<code>theblackpearl.out</code>
<code>ab?c</code>	6
<code>aa???b?c</code>	19

Замечание

Описанным в условии требованиям отвечают ровно шесть подстрок слова-загадки из первого примера:

- четыре подстроки из одного символа («a», «b», «c», и «?» при заполнении пропуска произвольным символом)
- подстрока «b?» при заполнении пропуска символом «b»
- подстрока «?c» при заполнении пропуска символом «c»

Задача М. Тест на тупость

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

У Саши есть n палочек с длинами $a_1, \dots, a_n$. Ваша задача — определить количество различных тупоугольных треугольников, которые можно составить из них. Каждый треугольник должен быть составлен ровно из трёх палочек набора.

Формат входных данных

Входной файл содержит n строк, каждая из которых содержит очередное число a_i . Все числа натуральные, все не превышают 7000.

Формат выходных данных

Выведите строку «Sasha can compose k triangles», где k — количество тупоугольных треугольников.

Примеры

<code>obtuse.in</code>	<code>obtuse.out</code>
3	Sasha can compose 2 triangles
4	
5	
5	
6	