

## 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

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

## Задача В. 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 С. 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 D. Точки и отрезки

Имя входного файла: `segments.in`  
Имя выходного файла: `segments.out`  
Ограничение по времени: 1 секунда  
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  $n$  ( $1 \leq n \leq 10^5$ ) — число отрезков и  $m$  ( $1 \leq m \leq 10^5$ ) — число точек. В следующих  $n$  строках записаны по два целых числа  $a_i$  и  $b_i$  — координаты концов соответствующего отрезка. В последней строке записаны  $m$  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 выведите количество отрезков, в которых она содержится.

### Примеры

<code>segments.in</code>	<code>segments.out</code>
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1

## Задача Е. Реклама 2

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

В супермаркете решили время от времени транслировать рекламу новых товаров. Для того, чтобы составить оптимальное расписание трансляции рекламы, руководство супермаркета провело следующее исследование: в течение дня для каждого покупателя, посетившего супермаркет, было зафиксировано время, когда он пришел в супермаркет, и когда он из него ушел.

Менеджер по рекламе предположил, что такое расписание прихода-ухода покупателей сохранится и в последующие дни. Он хочет составить расписание трансляции рекламных роликов, чтобы каждый покупатель услышал не меньше двух рекламных объявлений. В тоже время он выдвинул условие, чтобы два рекламных объявления не транслировались одновременно и, поскольку продавцам все время приходится выслушивать эту рекламу, общее число рекламных объявлений за день было минимальным.

Напишите программу, которая составит такое расписание трансляции рекламных роликов. Рекламные объявления можно начинать транслировать только в целые моменты времени. Считается, что каждое рекламное объявление заканчивается до наступления следующего целого момента времени. Если рекламное объявление транслируется в тот момент времени, когда покупатель входит в супермаркет или уходит из него, покупатель это объявление услышать успеваает.

### Формат входных данных

Во входном файле записано сначала число  $N$  — количество покупателей, посетивших супермаркет за день ( $1 \leq N \leq 3000$ ). Затем идет  $N$  пар натуральных чисел  $A_i, B_i$ , задающих соответственно время прихода и время ухода покупателей из супермаркета ( $1 \leq A_i < B_i \leq 10^6$ ).

### Формат выходных данных

В выходной файл выведите сначала количество рекламных объявлений, которое будет протранслировано за день. Затем выведите в возрастающем порядке моменты времени, в которые нужно транслировать рекламные объявления. Если решений несколько, выведите любое из них.

### Примеры

стандартный ввод	стандартный вывод
5	5
1 10	9 10 12 23 24
10 12	
1 10	
1 10	
23 24	

## Задача F. Межрегиональная олимпиада

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

На межрегиональной олимпиаде по программированию роботов соревнования проводятся в один тур и в необычном формате. Задачи участникам раздаются последовательно, а не все в самом начале тура, и каждая  $i$ -я задача ( $1 \leq i \leq n$ ) становится доступной участникам в свой момент времени  $s_i$ . При поступлении очередной задачи каждый участник должен сразу определить, будет он ее решать или нет. В случае, если он выбирает для решения эту задачу, то у него есть  $t_i$  минут на то, чтобы сдать ее решение на проверку, причем в течение этого времени он не может переключиться на решение другой задачи. Если же участник отказывается от решения этой задачи, то в будущем он не может к ней вернуться. В тот момент, когда закончилось время, отведенное на задачу, которую решает участник, он может начать решать другую задачу, ставшую доступной в этот же момент, если такая задача есть, или ждать появления другой задачи. При этом за правильное решение  $i$ -й задачи участник получает  $c_i$  баллов.

Артур, представляющий на межрегиональной олимпиаде один из региональных центров искусственного интеллекта, понимает, что важную роль на такой олимпиаде играет не только умение решать задачи, но и правильный стратегический расчет того, какие задачи надо решать, а какие пропустить. Ему, как и всем участникам, до начала тура известно, в какой момент времени каждая задача станет доступной, сколько времени будет отведено на ее решение и сколько баллов можно получить за ее решение. Артур является талантливым школьником и поэтому сможет успешно решить за отведенное время и сдать на проверку любую задачу, которую он выберет для решения на олимпиаде.

Требуется написать программу, которая определяет, какое максимальное количество баллов Артур сможет получить при оптимальном выборе задач, которые он будет решать, а также количество и перечень таких задач.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество задач на олимпиаде.

Последующие  $n$  строк содержат описания задач, по три числа на каждой строке:  $s_i$  — момент появления  $i$ -й задачи в минутах,  $t_i$  — время, отведенное на ее решение в минутах, и  $c_i$  — сколько баллов получит участник за решение этой задачи ( $1 \leq s_i, t_i, c_i \leq 10^9$ ).

### Формат выходных данных

Первая строка выходного файла должна содержать одно число — максимальное количество баллов, которое сможет получить Артур на олимпиаде. **Восстанавливать ответ не требуется.**

### Примеры

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

## Задача G. 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 Н. Объединение прямоугольников

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

От вас требуется решить задачу объединения прямоугольников

### Формат входных данных

В первой строке входных данных дано число  $n \leq 50000$  – количество прямоугольников. В следующих  $n$  строках задаются прямоугольники в формате  $x_1, y_1, x_2, y_2$  ( $0 \leq x_1, y_1, x_2, y_2 \leq 50000$ ) – координаты левого нижнего и правого верхнего углов прямоугольника

### Формат выходных данных

В единственной строке выходной данных выведите одно число – площадь объединения всех данных прямоугольников.

### Примеры

стандартный ввод	стандартный вывод
2 0 0 2 2 1 3 2 4	5
3 0 0 2 4 4 1 6 3 1 3 5 6	23