

Задача А. Точки и отрезки

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

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 В. Пересечение прямоугольников

Имя входного файла: `rect.in`
Имя выходного файла: `rect.out`
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 прямоугольник, являющийся их пересечением.

Формат входных данных

В первой строке входного файла указано число N ($1 \leq N \leq 1500$). 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 ($-10^9 \leq x_1 \leq x_2 \leq 10^9, -10^9 \leq y_1 \leq y_2 \leq 10^9$). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

Формат выходных данных

В единственную строку выходного файла поместите описание искомого прямоугольника в том же формате, в котором заданы прямоугольники во входном файле.

Если пересечение заданных прямоугольников пусто, выведите в выходной файл единственное число -1.

Пример

<code>rect.in</code>	<code>rect.out</code>
2 0 0 2 2 1 1 3 3	1 1 2 2

Задача С. Столовая

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

В лагере детей так много, что все они не помещаются в столовую и не могут обедать одновременно. Поэтому дети стали приходить в столовую и уходить из нее как попало. Про каждого ребенка известно время его прихода в столовую и ухода из неё. Сколько других детей встретил в столовой каждый ребенок.

Формат входных данных

В первой строке записано количество детей 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 столовую и ухода из неё i -го ребенка.

Формат выходных данных

Выведите N целых чисел, i -е число должно быть равно количеству детей, которых встретил в столовой i -й ребенок.

Если в некоторый момент времени один ребенок приходит в столовую, а другой уходит из неё, то они встречаются друг с другом.

Пример

<code>dining.in</code>	<code>dining.out</code>
4	3
1 10	3
2 5	2
5 6	2
1 4	

Задача D. Кассы

Имя входного файла: `tickets.in`
Имя выходного файла: `tickets.out`
Ограничение по времени: 0.5 секунды
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

<code>tickets.in</code>	<code>tickets.out</code>
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. Том Сойер и его друзья

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

Друзья Тома Сойера по очереди красят забор разными красками. Каждый из них красит несколько идущих подряд секций забора в определенный цвет, при этом используемые цвета могут повторяться. Новая краска ложится поверх старой. Для каждой краски вычислите количество секций, которые будут покрашены этой краской после того, как все друзья закончат работу.

Формат входных данных

В первой строке входного файла содержатся два целых числа: N ($1 \leq N \leq 10^9$) и K ($1 \leq K \leq 50000$) — количество секций в заборе и количество различных красок соответственно.

Во второй строке содержится единственное число M ($0 \leq M \leq 50000$) — количество друзей Тома Сойера.

Далее следуют M строк: в i -ой строке содержится информация о работе друга, который красил забор i -ым по счету, а именно 3 целых числа c_i, l_i, r_i ($1 \leq c_i \leq K, 1 \leq l_i \leq r_i \leq N$) — номер краски, которую использовал i -й друг, номер первой и номер последней покрашенной секции соответственно.

Формат выходных данных

Выведите в единственную строку выходного файла K целых чисел: i -ое число должно быть равно количеству секций, покрашенных i -й краской.

Пример

<code>paint.in</code>	<code>paint.out</code>
5 3 4 1 3 4 2 4 5 3 2 3 1 5 5	1 1 2
5 3 3 1 1 5 2 2 4 1 3 3	3 2 0

Задача F. Объединение прямоугольников

Имя входного файла: `union.in`
Имя выходного файла: `union.out`
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 ($0 \leq N \leq 1500$). 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.

Пример

<code>union.in</code>	<code>union.out</code>
3 1 1 3 5 5 2 7 4 2 4 6 7	23

Задача G. Художник

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

Даны n отрезков на прямой. Все они разных цветов. Нужно определить порядок рисования отрезков, при котором максимальное число различных цветов будет видимо (мы рисуем отрезки последовательно, возможно перекрашивая уже нарисованное).

Формат входных данных

В первой строке написано целое число n ($1 \leq n \leq 300$). В следующих n строках будут написаны по два целых числа через пробел. В i -ой из этих строк находятся числа l_i и r_i ($-10^9 \leq l_i < r_i \leq 10^9$).

Формат выходных данных

В первой строке выведите количество цветов, которые будут видны при оптимальном порядке рисования. Во второй строке должно быть написано n чисел — i -ое число обозначает, какой отрезок надо нарисовать i -ым для достижения оптимального результата. Отрезки нумеруются, начиная с единицы, в том же порядке, в котором они заданы во входном файле.

Пример

<code>painter.in</code>	<code>painter.out</code>
4	3
1 3	4 1 2 3
2 4	
2 3	
1 4	