

## Задача А. 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.

То, что это прямоугольник, докажите самостоятельно.

### 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.

### Примеры

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

## Задача В. Кассы

Имя входного файла: `tickets.in`  
Имя выходного файла: `tickets.out`  
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

<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 С. Точки и отрезки

Имя входного файла: `segments.in`  
Имя выходного файла: `segments.out`  
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  $n$  ( $1 \leq n \leq 10^5$ ) — число отрезков и  $m$  ( $1 \leq m \leq 10^5$ ) — 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

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

## Задача D. 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.

### Примеры

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

## Задача Е. Объединение прямоугольников (версия для Python)

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  $N$  ( $0 \leq N \leq 300$ ). 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>union.in</code>	<code>union.out</code>
3 1 1 3 5 5 2 7 4 2 4 6 7	23

## Задача F. Операционные системы

Имя входного файла: `os.in`  
Имя выходного файла: `os.out`  
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

<code>os.in</code>	<code>os.out</code>
10 3 1 3 4 7 3 4	1
10 4 1 3 4 5 7 8 4 6	3

## Задача G. Операционные системы (версия для Python)

Имя входного файла: `os.in`  
Имя выходного файла: `os.out`  
Ограничение по времени: 2 секунды  
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 ( $0 \leq N \leq 21000$ ). 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

<code>os.in</code>	<code>os.out</code>
10 3 1 3 4 7 3 4	1
10 4 1 3 4 5 7 8 4 6	3

## Задача Н. Тесты к задаче «Пересадки»

Имя входного файла: --  
Имя выходного файла: --  
Ограничение по времени: —  
Ограничение по памяти: —

Разработайте систему тестов для задачи «Пересадки» предыдущего дня.

Вам нужно сделать несколько тестов (от 1 до 20), удовлетворяющих условию задачи. К каждому тесту необходимо также указать правильный ответ. Тесты необходимо сохранить в файлах 001.dat, 002.dat, 003.dat и т.д., правильные ответы на них — в файлах 001.ans, 002.ans, 003.ans соответственно.

На проверку вам необходимо сдать архив в формате tar.gz. Внутри этого архива должен быть каталог с именем tests. Внутри этого каталога должно быть не более 20 тестов и ответов на них. Все тесты должны строго соответствовать формату входных и выходных данных, описанных в условии задачи.

Сданный файл получает ОК, если он имеет правильный формат и структуру, содержит от 1 до 20 корректных тестов и правильные ответы на них (правильные решения выдают такие же ответы), а все неправильные решения не проходят хотя бы один тест из числа предложенных вами.

Для создания архивов рекомендуется использовать 7-zip. Использовать встроенные средства Windows для создания архивов нельзя, так как они создают некорректные архивы.

При создании архива на OS X не забудьте исключить файл .DS\_Store из архива. Например, так:  
`tar --exclude=".DS_Store" -zcvf out.tgz tests.`