

## Задача А. Пересадки

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

На Новом проспекте для разгрузки было решено пустить два новых автобусных маршрута на разных участках проспекта. Все остановки на проспекте пронумерованы подряд натуральными числами. Известны конечные остановки каждого из автобусов. Определите количество остановок, на которых можно пересесть с одного автобуса на другой.

### Формат входных данных

Вводятся четыре натуральных числа — номера конечных остановок сначала первого, потом второго автобуса. Числа не превосходят  $10^9$ .

### Формат выходных данных

Ваша программа должна выводить одно число — искомое количество остановок.

### Примеры

<code>transfers.in</code>	<code>transfers.out</code>
3 6 4 2	2
3 1 5 10	0

## Задача В. Объединение отрезков

Имя входного файла: `merge.in`  
Имя выходного файла: `merge.out`  
Ограничение по времени: 1 секунда (Python3 2с)  
Ограничение по памяти: 64 мегабайта

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>merge.in</code>	<code>merge.out</code>
4	2
0 2	0 3
4 5	4 6
1 3	
5 6	

## Задача С. Дорешивание

Имя входного файла: `upsolving.in`  
Имя выходного файла: `upsolving.out`  
Ограничение по времени: 2 секунды (Python3 4с)  
Ограничение по памяти: 64 мегабайта

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 неё, то они встречаются друг с другом.

### Примеры

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

## Задача D. Минимальное покрытие

Имя входного файла: `cover.in`  
Имя выходного файла: `cover.out`  
Ограничение по времени: 1 секунда (Python3 2c)  
Ограничение по памяти: 64 мегабайта

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  $M$  ( $1 \leq M \leq 5000$ ). 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 100 000.

### 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

<code>cover.in</code>	<code>cover.out</code>
1 -1 0 -5 -3 2 5 0 0	No solution
1 -1 0 0 1 0 0	1 0 1

## Задача Е. Китайские часы

Имя входного файла: `watches.in`  
Имя выходного файла: `watches.out`  
Ограничение по времени: 1 секунда (Python3 2c)  
Ограничение по памяти: 64 мегабайта

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