

## Задача А. Поручения

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

Учитель Сплинтер всегда держит своих учеников в тонусе. Он дал им  $n$  поручений: им нужно помочь Кейси Джонсу в уличной драке, спасти Землю от нападок Шреддера и сорвать коварные планы Кренга, а в довершение еще сделать кучу дел по дому. Причем черепашкам необходимо выполнить все эти задания.

Очевидно, каждое поручение — не из приятных и доставляет черепашкам какое-то количество боли и страданий. Когда черепашки выполняют очередное задание, боль, которую оно приносит, может добавиться к усталости черепашек. Однако, это происходит только в том случае, если любое из заданий, выполненных ими раньше, приносило им меньше боли, чем последнее выполненное. Страдание добавляется к усталости по таким же правилам.

Теперь черепахи хотят выполнять задания в таком порядке, чтобы после выполнения всех заданий усталость была минимальна.

### Формат входного файла

В первой строке входного файла задано одно целое число  $n$  — количество заданий, которые получили черепашки. ( $1 \leq n \leq 10^5$ ). Далее следуют  $n$  строк, где для каждого  $i$ -го задания задано два целых числа — количество боли  $a_i$  и страдания  $b_i$  ( $1 \leq a_i, b_i \leq 10^9$ ). Гарантируется, что все  $a_i$  различны и все  $b_i$  различны.

### Формат выходного файла

В первую строку выходного файла выведите минимальную усталость черепашек после выполнения всех заданий. Во второй строке выходного файла выведите перестановку чисел от 1 до  $n$  — порядок, в котором следует выполнять задания. Если существует несколько оптимальных ответов, выведите любой.

### Примеры

<code>missions.in</code>	<code>missions.out</code>
3	8
3 2	2 1 3
2 3	
1 1	

### Пояснение

Решения, работающие в случае, когда  $n \leq 10$ , будут оцениваться в 30 баллов.

## Задача В. Точки

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

Недавно черепашки получили письмо от учителя Сплинтера. В нем было сказано, что они должны прийти в центр Нью-Йорка ровно в полночь (а до нее оставалось всего несколько часов). Леонардо заподозрил что-то неладное и предположил, что это ловушка, а учитель в беде и его необходимо срочно спасать. Донателло заметил набор точек, расположенных на одной прямой, который был нарисован в углу листка, и тут же вспомнил шифр, который они недавно придумали вместе с учителем Сплинтером.

Сам шифр заключался в следующем. Введем на прямой с точками такую систему координат, что крайние точки имеют координаты ноль и один. Координаты всех остальных точек будут рациональными числами, лежащими в интервале от нуля до единицы. Известно, что изначально на прямой были нарисованы только две крайние точки.

Также известно, что учитель ставил очередную точку только строго посередине между какими-то двумя уже поставленными. Например, третьей точкой он мог поставить точку с координатой  $\frac{1}{2}$ , четвертой — точку с координатой  $\frac{1}{4}$ , а пятой — точку с координатой  $\frac{5}{8} = \frac{1+1}{2}$ . Таким образом учитель поставил  $n - 2$  точки.

Донателло хочет проверить, является ли набор точек, найденных на листе, описанным шифром.

### Формат входного файла

В первой строке входного файла записано целое число  $n$  ( $2 \leq n \leq 1500$ ) — количество точек на рисунке в письме. В следующих  $n$  строках записано по два целых числа  $x_i$  и  $y_i$  ( $0 \leq x_i \leq y_i \leq 10^9, y_i \neq 0$ ), означающие, что  $i$ -я точка имеет координату  $\frac{x_i}{y_i}$ .

Гарантируется, что в множестве есть точки с координатами ноль и один и что все точки различны. Также гарантируется, что все дроби, данные в условии, несократимые.

Необходимо определить, можно ли было поставить эти точки описанным образом.

### Формат выходного файла

В выходной файл в первой строке выведите YES, если точки могли быть поставлены описанным образом, и NO в противном случае.

Если множество получить можно, в следующих  $n - 2$  строках выведите по три числа — номер точки, которую нужно добавить на очередном шаге, а также номера двух точек, которые уже были добавлены, таких, что новая точка лежит ровно посередине между ними. Считайте, что точки с координатами 0 и 1 уже добавлены.

Если же множество нельзя получить подобным образом, то во второй строке выведите номер любой точки, которую нельзя получить. Все точки нумеруются с 1.

## Примеры

points.in	points.out
4 0 1 1 4 1 2 1 1	YES 3 1 4 2 1 3
4 0 1 1 4 3 4 1 1	NO 2
5 5 8 1 4 1 2 1 1 0 1	YES 3 5 4 2 3 5 1 2 4

## Пояснение

Решения, работающие, когда количество точек не превосходит 500, будут оцениваться в 40 баллов.

## Задача С. Канализация

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

Как известно, черепашки-ниндзя вместе со своим учителем Сплинтером всю свою жизнь проводят в канализации. Ведь только там можно так быстро передвигаться по городу, скользя на панцире!

Из канализации в город можно выбраться только с помощью  $n$  канализационных люков, расположенных в разных частях города. Некоторые люки соединены друг с другом трубами, по которым черепашки могут передвигаться в обе стороны. Между каждой парой люков существует ровно один путь по трубам. Это, в частности, значит, что в канализации ровно  $n - 1$  труба. Люки пронумерованы начиная с 1.

Когда черепашки-ниндзя были маленькими, они постоянно забывали маршруты от одного люка до другого, и спрашивали учителя Сплинтера, куда же им скользить, чтобы попасть в какой-то люк. Сплинтер сообщал черепашкам номер первого люка на пути из люка номера  $l$  к люку номер  $r$ .

Вскоре учителю Сплинтеру надоело, что черепашки постоянно отвлекают его от медитации, и они вместе с Донателло написали программу, которая по двум номерам люков  $l$  и  $r$  сообщает номер первого люка на пути из  $l$  в  $r$ . А вы сможете написать такую программу?

### Формат входного файла

В первой строке входного файла заданы два целых числа  $n$  и  $m$  — количество люков в канализации и количество запросов соответственно ( $1 \leq n, m \leq 10^5$ ).

В следующих  $n - 1$  строке заданы описания труб — два числа  $a$  и  $b$ , которые задают номера люков, которые соединяет эта труба ( $1 \leq a, b \leq n$ ). По трубе можно скользить в обоих направлениях. Гарантируется, что в канализации можно добраться от каждого люка до каждого.

В следующих  $m$  строках заданы запросы, по одному запросу в строке — два числа  $l$  и  $r$ , которые задают стартовый и конечный люк на пути соответственно ( $1 \leq l, r \leq n$ ). Гарантируется, что  $l$  и  $r$  — различны.

### Формат выходного файла

В выходной файл выведите ответы на запросы по одному в строке, в том порядке, в котором они следуют во входном файле. Ответом на запрос является номер первого люка на пути с  $l$  по  $r$ .

### Примеры

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

### Пояснение

Решения, работающие, когда количество люков не превосходит  $10^3$ , а количество запросов не превосходит  $10^4$ , будут оцениваться в 20 баллов.

Решения, работающие, когда количество люков не превосходит  $10^3$ , будут оцениваться в 40 баллов.

## Задача D. Пицца-марафон

Имя входного файла: `pizza.in`  
Имя выходного файла: `pizza.out`  
Ограничение по времени: 5 секунды  
Ограничение по памяти: 512 мегабайт

Устав от тяжелых трудовых будней подземных супергероев, черепашки-ниндзя решили взять отпуск. А как лучше всего можно провести свободное время, если вы зеленого цвета и носите на спине панцирь? Конечно устроив недельный пицца-марафон!

У черепашек есть меню их любимой пиццерии. В нем записано название каждой пиццы и ее стоимость. При этом, что интересно, размер пиццы совпадает с длиной ее названия. Поскольку некоторые пиццы черепашки уже пробовали, то заказы они делают следующим образом: выбирают пиццу, которая им понравилась и заказывают все, похожие на нее, размером не меньше некоторого числа  $k$ . Считается, что пицца  $i$  похожа на пиццу  $j$ , если ее название  $s_i$  является префиксом названия  $j$ -й пиццы  $s_j$ . Так как сами черепашки не могут свободно передвигаться по городу, они попросили Эйприл сообщать им об изменениях в работе пиццерии, в частности, об изменении цен на пиццы.

В течение всей недели Донателло, которому было поручено следить за количеством потраченных на пиццу денег, делал в своем блокноте записи о заказах пиццы и изменениях цен. Когда же в конце недели он решил посчитать стоимость всех заказов, он обнаружил, что их оказалось невероятно много, поэтому он попросил вас помочь с этой задачей.

### Формат входного файла

В первой строке входного файла записано два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10^6, 1 \leq m \leq 10^6$ ) — число пицц в меню и число заметок в блокноте. В следующих  $n$  строк содержится описание меню пиццерии: в  $i$ -й строке через пробел записаны название  $i$ -й пиццы  $s_i$  и ее начальная стоимость  $c_i$  ( $0 < c_i \leq 10^9$ ). Названия пицц состоят только их малых латинских букв. Гарантируется, что суммарная длина всех строк не превышает  $10^6$ . Гарантируется, что все названия пицц различны.

В следующих  $m$  строках находятся записи из блокнота Донателло в следующем формате:

- "`? i k`" — черепашки заказали пиццы, похожие на  $i$ -ю и размером не меньше  $k$  ( $1 \leq i \leq n, 1 \leq k \leq 10^6$ ).
- "`! i j`" — Эйприл принесла сообщение, что стоимость  $i$ -й пиццы увеличилась на  $j$  ( $1 \leq i \leq n, |j| \leq 10^9$ ).

### Формат выходного файла

Для заказа каждой пиццы в отдельной строке выведите её стоимость.

### Примеры

<code>pizza.in</code>	<code>pizza.out</code>
<code>2 4</code>	<code>30</code>
<code>margarita 20</code>	<code>20</code>
<code>marga 10</code>	<code>10</code>
<code>? 1 5</code>	<code>0</code>
<code>? 1 6</code>	
<code>? 2 5</code>	
<code>? 2 6</code>	
<code>2 2</code>	<code>70</code>
<code>supreme 20</code>	
<code>supremesuper 30</code>	
<code>! 2 20</code>	
<code>? 2 2</code>	

## Пояснение

Решения, работающие в случае, когда  $m \leq 100$  и сумма длин всех строк не превышает  $10^5$ , будут оцениваться в 30 баллов.

Решения, работающие в случае, когда  $m \leq 10000$  и максимальная длина строк не превышает  $10^4$ , будут оцениваться в 60 баллов.