

Задача А. Раскраска в три цвета

Имя входного файла: `color.in`
Имя выходного файла: `color.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя нарисовал на бумаге n кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

Формат входного файла

Первая строка содержит два целых числа n и m — количество кружков и количество линий, которые нарисовал Петя, соответственно ($1 \leq n \leq 1000$, $0 \leq m \leq 20\,000$).

Следующая строка содержит n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — i -й из этих символов означает цвет, в который раскрашен i -й кружок ('R' — красный, 'G' — зеленый, 'B' — синий).

Следующие m строк содержат по два целых числа — пары кружков, соединенных отрезками.

Формат выходного файла

Выведите в выходной файл одну строку, состоящую из n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово `Impossible`.

Примеры

<code>color.in</code>	<code>color.out</code>
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

Задача В. Эльфы и олени

Имя входного файла: `elves.in`
Имя выходного файла: `elves.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Скоро новый год и Санта-Клаус уже начал готовить свою волшебную оленью упряжку, на которой он развозит подарки детям. Известно, что упряжку везут несколько волшебных оленей, на каждом из которых едут два эльфа.

Но волшебные олени — строптивые животные, поэтому не любые два эльфа могут ехать на любом олене. А именно, каждый олень характеризуется некоторой строптивостью a_i , а каждый эльф — темпераментом b_i . Два эльфа j и k могут ехать на i -м олене в том и только в том случае, если либо $b_j < a_i < b_k$, либо $b_k < a_i < b_j$.

Чтобы его появление было максимально зрелищным, Санта-Клаус хочет, чтобы в его упряжке было как можно больше оленей. Про каждого оленя Санта знает его строптивость, а про каждого эльфа — его темперамент.

Помогите Санте выяснить, какое максимальное количество оленей он сможет включить в упряжку, каких оленей ему следует выбрать, и какие эльфы должны на них ехать.

Формат входного файла

Первая строка входного файла содержит два целых числа m и n — количество оленей и эльфов, соответственно ($1 \leq m, n \leq 100\,000$).

Вторая строка содержит m целых чисел a_i — строптивость оленей ($0 \leq a_i \leq 10^9$). Третья строка содержит n целых чисел b_i — темперамент эльфов ($0 \leq b_i \leq 10^9$).

Формат выходного файла

На первой строке выходного файла выведите одно число k — максимальное количество оленей, которое Санта-Клаус может включить в свою упряжку. На следующих k строках выведите по три целых числа: $d_i, e_{i,1}, e_{i,2}$ — для каждого оленя в упряжке выведите его номер и номера эльфов, которые на нем поедут. Если решений несколько, выведите любое.

И эльфы, и олени пронумерованы, начиная с единицы, в том порядке, в котором они заданы во входном файле.

Пример

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

Задача С. Эксперимент

Имя входного файла: `exp.in`
Имя выходного файла: `exp.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Сегодня Игорь получил долгожданное разрешение на проведение эксперимента по изучению протекания химических реакций в магнитном поле. При этом используются две установки — генератор магнитного поля и манипулятор, соединяющий реагенты.

Эксперимент разбит на некоторое количество этапов, при этом некоторые из них могут быть выполнены только после завершения определенного набора других этапов. Правда известно, что хотя бы один способ проведения эксперимента существует. На каждом этапе Игорь должен управлять ровно одной из двух установок — либо генератором, либо манипулятором.

Игорь очень дорожит своим временем, и поэтому он хочет провести эксперимент, совершив наименьшее количество перемещений между пультами управления установками. Помогите ему узнать, в каком порядке следует выполнять этапы, чтобы этого добиться.

Формат входного файла

На первой строке входного файла записано целое число n — количество этапов эксперимента ($1 \leq n \leq 100$).

Следующие n строк содержат описание этапов. Пронумеруем этапы от 1 до n в некотором произвольном порядке. Тогда i -я из этих строк описывает i -й этап. Каждый этап описывается последовательностью целых чисел. Первое число равно нулю, если на этом этапе Игорь управляет генератором, и единице, если он управляет манипулятором. Затем записано целое число r_i — количество этапов, которые должны быть выполнены перед выполнением данного. За ним следуют номера этих этапов — r_i различных целых чисел в диапазоне от 1 до $i - 1$.

Формат выходного файла

На первой строке выходного файла выведите минимальное количество перемещений, которые придется совершить Игорю. На второй строке выведите перестановку чисел от 1 до n — последовательность, в которой следует выполнять этапы. Если решений несколько, выведите любое.

Примеры

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

Задача D. Шоссе

Имя входного файла:	highways.in
Имя выходного файла:	highways.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Во Флатландии n городов, расположенных в различных точках плоскости. Известно, что никакие три города не лежат на одной прямой.

Правительство решило построить в стране сеть сверхскоростных шоссе. Сеть шоссе должна быть такой, чтобы из любого города можно было проехать в любой другой по построенным шоссе. А в целях экономии средств было решено, что путь, соединяющий любые два города, должен быть единственным. Каждое шоссе представляет собой отрезок, соединяющий некоторую пару городов.

Завод, выполняющий этот госзаказ, подготовил проект сети шоссе. Проект представляет собой описание $n - 1$ шоссе. Каждое шоссе задается городами, которые оно соединяет. В целях секретности вместо названий городов в проекте были использованы коды — числа от 1 до n .

Однако когда дело дошло до реализации проекта, выяснилось, что документ, в котором было указано соответствие номеров городам, утерян. Поскольку проект приурочен к пятистолетию культурной столицы Флатландии, переделывать проект полностью оказалось невозможно. Поэтому было решено установить некоторое новое соответствие номеров городам.

При попытке это сделать разработчики проекта столкнулись со следующей проблемой. В соответствии с техническими нормами строительства, недопустимо, чтобы шоссе пересекались вне городов. Поэтому не любое сопоставление номеров городам допустимо. После пары бессонных ночей главный инженер завода решил поручить спасение проекта вам.

Ваша задача — таким образом сопоставить числам от 1 до n города, чтобы после реализации проекта шоссе не пересекались вне городов, которые они соединяют.

Формат входного файла

В первой строке входного файла содержится целое число n — количество городов во Флатландии ($2 \leq n \leq 1500$).

Далее следует n описаний городов. Описание каждого города состоит из двух строк. Первая строка содержит название города — строку, состоящую из символов с ASCII-кодами от 33 до 127. Названия различных городов не совпадают. Длина названия города не превышает 60 символов. Вторая строка описания города содержит два целых числа x и y — координаты города. Координаты не превышают 10^4 по абсолютной величине.

Далее следуют $n - 1$ строк, которые описывают проект строительства сети шоссе в его текущем состоянии. Каждая строка содержит по два целых числа — номера городов, соединенных шоссе в проекте. Никакое шоссе в проекте не соединяет город сам с собой, никакие два города не соединены более чем одним шоссе.

Формат выходного файла

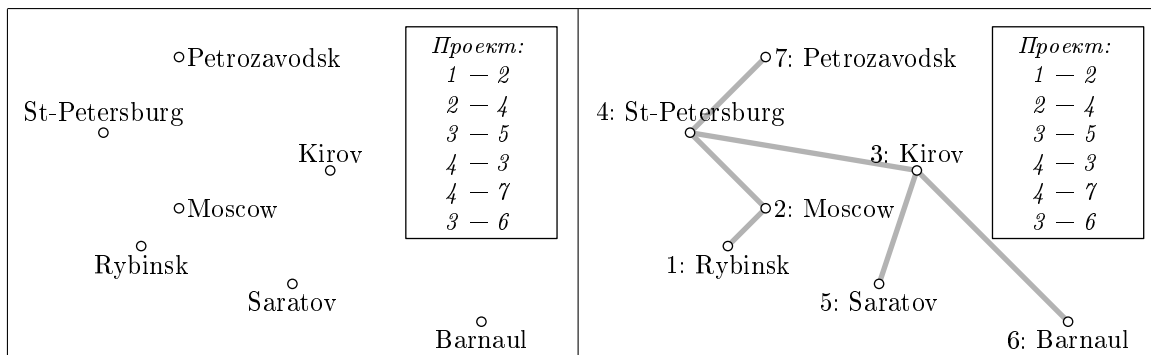
Выведите в выходной файл n строк, i -я из этих строк должна содержать название города, который следует сопоставить числу i в проекте. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл «No solution».

Пример

highways.in	highways.out
7	Rybinsk
Moscow	Moscow
2 2	Kirov
St-Petersburg	St-Petersburg
0 4	Saratov
Kirov	Barnaul
6 3	Petrozavodsk
Saratov	
5 0	
Rybinsk	
1 1	
Petrozavodsk	
2 6	
Barnaul	
10 -1	
1 2	
2 4	
3 5	
4 3	
4 7	
3 6	

Иллюстрация



Задача Е. Автомобильные номера

Имя входного файла: `numbers.in`
Имя выходного файла: `numbers.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

При расследовании дорожно-транспортных происшествий часто возникают проблемы с розыском автомобилей, водители которых покинули место происшествия.

Получение свидетельских показаний — непростая работа. Ситуация осложняется тем, что очень часто свидетели могут только приблизительно вспомнить номер автомобиля. При этом с большой вероятностью опрашиваемый может перепутать порядок цифр или букв в номере.

По полученному от свидетеля происшествия номеру, подсчитайте, сколько различных номеров может получиться из него перестановкой букв и/или цифр, а также выведите все такие номера.

Напомним, что автомобильные номера в России состоят из трех букв и трех цифр, упорядоченных следующим образом: буква, три цифры, затем две буквы. Фрагмент номера, который идентифицирует регион, в котором зарегистрирован автомобиль, мы будем игнорировать.

В номере могут использоваться следующие буквы: «А», «В», «С», «Е», «Н», «К», «М», «О», «Р», «Т», «Х», «У» (эти буквы имеют схожие по написанию аналоги как в русском, так и в латинском алфавите). В этой задаче во входном файле будут использоваться буквы латинского алфавита.

Формат входного файла

Входной файл содержит одну строку, которая представляет собой корректный автомобильный номер.

Формат выходного файла

В первой строке выходного файла выведите число k — количество номеров, которые могут получиться из заданного перестановкой букв и/или цифр.

В последующих k строках выведите все такие номера в произвольном порядке.

Примеры

<code>numbers.in</code>	<code>numbers.out</code>
X772KX	9 X277XK X277KX X727XK X727KX X772XK X772KX K277XX K727XX K772XX

Задача F. Планета Плюк

Имя входного файла: `pluk.in`
Имя выходного файла: `pluk.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На планете Плюк, поверхность которой мы будем считать абсолютно плоской, был разработан новый принцип перемещения единственного имеющегося там транспортного средства — пепелаца. А именно, на расстоянии одного километра друг от друга в точках $(0, 0)$ и $(1, 0)$ были построены две станции управления пепелцами A и B . С помощью них можно мгновенно переместить любой пепелац, повернув его на 90 градусов по или против часовой стрелки относительно точки A или B . Расстояние от пепелаца до соответствующей станции при этом не меняется. Следующее перемещение можно делать как относительно той же станции, так и относительно другой.

Например, если повернуть пепелац, находящийся в точке $(3, 1)$ на 90 градусов против часовой стрелки относительно станции A , то он переместится в точку $(-1, 3)$, если его затем повернуть на 90 градусов по часовой стрелке относительно станции B , то он переместится в точку $(4, 2)$, если затем повернуть его вокруг станции B по часовой стрелке еще раз, он переместится в точку $(3, -3)$.

Один житель планеты недавно решил отправиться на своем пепелце в гости к другу. Житель проживает около точки с координатами (x_1, y_1) , а его друг — около точки с координатами (x_2, y_2) . Помогите жителю с помощью станций управления пепелцем оказаться как можно ближе к месту, где проживает его друг, чтобы потом меньше было идти по пустыне.

Поскольку перемещения мгновенные и абсолютно бесплатные, то минимизировать количество перемещений не надо.

Формат входного файла

Входной файл содержит четыре целых числа — x_1 , y_1 , x_2 и y_2 , они не превышают 10^4 по абсолютной величине.

Формат выходного файла

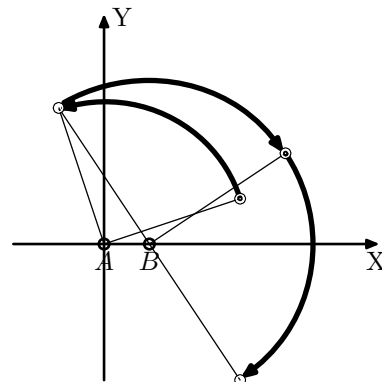
Выведите в выходной файл последовательность перемещений с использованием станций управления, которая перемещает пепелац из точки (x_1, y_1) как можно ближе к точке (x_2, y_2) .

Поворот по часовой стрелке относительно станции A обозначается как «+A», поворот против часовой стрелки относительно станции A обозначается как «-A», соответствующие повороты относительно станции B обозначаются как «+B» и «-B». Выводите по одному перемещению на строке.

Выведенная последовательность не обязана быть минимальной по количеству перемещений, но должна содержать не более 10^6 действий.

Примеры

<code>pluk.in</code>	<code>pluk.out</code>
3 1	-A
3 -3	+B
	+B
0 0	-B
3 0	-B



Задача G. Полигон

Имя входного файла: `polygon.in`
Имя выходного файла: `polygon.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Одной из первоочередных задач, стоящих перед министерством обороны Флатландии, является модернизация вооружения. В связи с этим было решено построить новый испытательный полигон.

По форме полигон представляет собой выпуклый многоугольник. Для демонстрации военных испытаний на полигоне различным чиновникам, неподалеку от него решено было построить наблюдательный центр. В результате длительных исследований было установлено, что основной характеристикой местоположения наблюдательного центра является *степень* этого центра относительно полигона.

Степень точки A относительно многоугольника вычисляется по следующему правилу. Рассмотрим все лучи с вершиной в точке A , имеющие общие точки с многоугольником. Для каждого такого луча найдем минимальное и максимальное расстояние вдоль него от точки A до некоторой точки многоугольника: d_{min} и d_{max} . Степенью точки относительно данного многоугольника назовем минимум величины $d_{min} \times d_{max}$ по всем таким лучам.

Военные не справляются с задачей вычисления степени наблюдательного центра относительно полигона и решили подключить к этой задаче вас. Помогите им!

Формат входного файла

Будем считать, что наблюдательный центр находится в точке $(0,0)$. Входной файл содержит описание полигона.

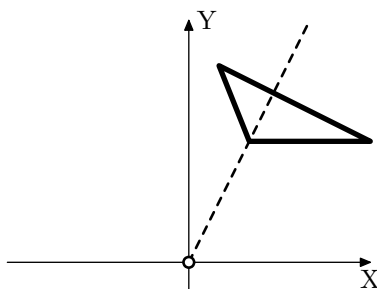
Первая строка входного файла содержит число n — количество вершин полигона ($3 \leq n \leq 100$). Следующие n содержат по два вещественных числа — координаты вершин полигона в порядке обхода их против часовой стрелки. Координаты не превышают 1000 по абсолютной величине. Гарантируется, что наблюдательный центр находится вне полигона, полигон представляет собой выпуклый невырожденный многоугольник, никакие три его последовательных вершины не лежат на одной прямой. Никакая сторона многоугольника не лежит на луче с центром в начале координат.

Формат выходного файла

Выведите в выходной файл одно число — степень наблюдательного центра относительно полигона. Ответ должен отличаться от правильного не более чем на 10^{-4} .

Пример

<code>polygon.in</code>	<code>polygon.out</code>
3 1.0 2.0 3.0 2.0 0.5 3.25	7.000000000



Задача Н. Красно-черные деревья

Имя входного файла: `rbtrees.in`
Имя выходного файла: `rbtrees.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Широкое распространение в стандартных библиотеках многих языков программирования получила реализация сбалансированных деревьев на основе так называемых *красно-черных деревьев*. В данной задаче вам предлагается посчитать количество красно-черных деревьев заданной формы.

Напомним, что двоичным деревом называется набор *вершин*, организованных в виде *дерева*. Каждая вершина имеет не более двух *детей*, один из которых называется *левым*, а другой — *правым*. Как левый, так и правый ребенок, а также оба могут отсутствовать.

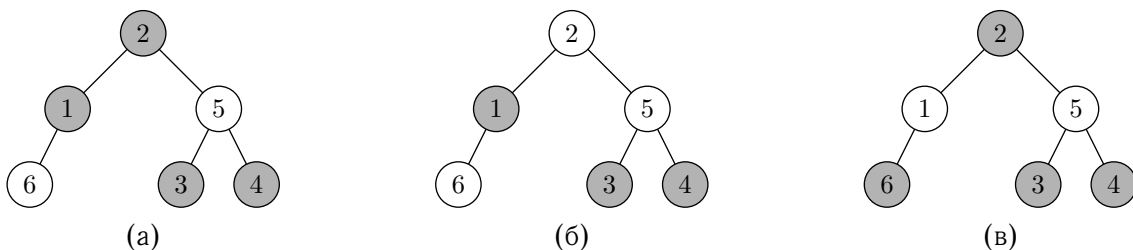
Если вершина Y — ребенок вершины X , то говорят, что вершина X является *родителем* вершины Y . У каждой вершины дерева, кроме одной, есть ровно один родитель. Единственная вершина, не имеющая родителя, называется *корнем* дерева.

Соединим каждую вершину кроме корня с ее родителем. Заметим, что для каждой вершины существует ровно один путь, ведущий в нее от корня.

Двоичное дерево называется *красно-черным*, если каждая его вершина раскрашена в красный либо в черный цвет, причем выполняются следующие условия:

1. если вершина красная, то ее родитель — черный;
2. количество черных вершин на пути от корня до любой вершины, у которой отсутствует хотя бы один ребенок, одно и то же.

Примеры двоичного дерева, вершины которого раскрашены в два цвета, приведены на следующем рисунке.



Если считать закрашенные вершины черными, а незакрашенные — красными, то дерево на рисунке (a) является красно-черным деревом, а деревья на рисунках (б) и (в) — нет. Для дерева на рисунке (б) нарушается первое свойство — у красной вершины 5 родитель 2 также красный, а в дереве на рисунке (в) нарушается второе свойство — на пути от корня до вершины 1 одна черная вершина, а, например, на пути от корня до вершины 3 — две.

Для заданного двоичного дерева подсчитайте число способов раскрасить его вершины в черный и красный цвет так, чтобы оно стало красно-черным деревом.

Формат входного файла

Первая строка входного файла содержит число n — количество вершин в дереве ($1 \leq n \leq 1000$).

Пусть вершины дерева пронумерованы числами от 1 до n . Следующие n строк содержат по два числа — для каждой вершины заданы номера ее левого и правого ребенка. Если один из детей отсутствует, то вместо его номера записан ноль. Гарантируется, что входные данные корректны, то есть набор чисел во входном файле действительно задает двоичное дерево.

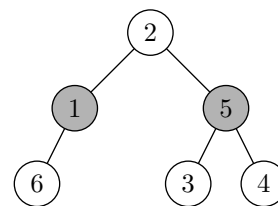
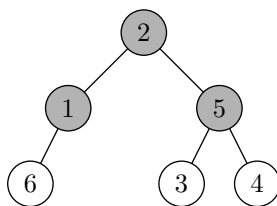
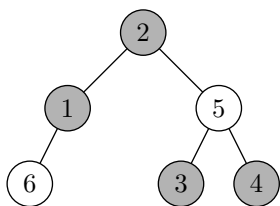
Формат выходного файла

Выведите в выходной файл одно число — количество способов раскрасить вершины заданного во входном файле двоичного дерева в красный и черный цвета так, чтобы оно стало красно-черным деревом.

Примеры

rbtrees.in	rbtrees.out
6 6 0 1 5 0 0 0 0 3 4 0 0	3
4 2 0 3 0 4 0 0 0	0

Все допустимые способы раскрасить вершины дерева из первого примера приведены на следующем рисунке.



Задача I. Изображение таблицы

Имя входного файла: `table.in`
Имя выходного файла: `table.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

При разработке программ для просмотра веб-страниц одной из наиболее сложных задач является корректное отображение таблиц. Компания «Kozilla», в которой вы работаете, планирует разработать новую версию браузера «Waterrat» для работы в терминальном режиме, и просит вас написать фрагмент ядра отображения веб-страниц, ответственный за формирование структуры таблиц.

Фрагмент, который вы должны написать, получает на вход информацию о количестве строк таблицы и ячейках этих строк и должен сгенерировать структуру таблицы и передать ее модулю, который занимается отображением таблицы на экране.

Таблица состоит из строк, каждая строка состоит из одной или нескольких ячеек, j -я ячейка i -й строки имеет ширину $a_{i,j}$.

По заданным параметрам таблицы постройте символическое изображение ее структуры.

Формат входного файла

Первая строка входного файла содержит n — количество строк в таблице ($1 \leq n \leq 100$). Следующие n строк входного файла содержат описание строк таблицы.

Описание каждой строки включает число m_i — количество ячеек этой строки, и m_i целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,m_i}$ — ширину каждой из ячеек строки ($1 \leq m_i \leq 10, 1 \leq a_{i,j} \leq 20$).

Формат выходного файла

Выведите в выходной файл символическое изображение структуры таблицы.

Изображение i -й строки таблицы должно начинаться изображением горизонтальной линии, составленным из символов «+» (плюс) и «-» (минус). Затем должна следовать строка, содержащая пробелы и символы «|» (вертикальная черта). Первым символом строки должна быть вертикальная черта, затем $a_{i,1}$ пробелов, затем вертикальная черта, затем $a_{i,2}$ пробелов, и так далее, всего m_i блоков пробелов. После последнего блока должна следовать вертикальная черта. После последней строки таблицы также должно следовать изображение горизонтальной линии.

В изображении горизонтальной линии используйте символ «+», если сверху или снизу от этой позиции находится вертикальная черта, и «-» в противном случае. Горизонтальная линия должна иметь минимальную возможную длину, чтобы над каждым символом вертикальной черты следующей строки и под каждым символом вертикальной черты предыдущей строки были символы «+».

Примеры

<code>table.in</code>	<code>table.out</code>
4	+---+-----+--+
3 3 5 1	
1 2	+---+-----+--+
1 2	
2 5 1	+---+
	+---+---+--+
	+-----+--+

Задача J. Неправильная считалка

Имя входного файла: `wrong.in`
Имя выходного файла: `wrong.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ребята во дворе решили поиграть в прятки. Чтобы выбрать ведущего, который будет искать, они решили воспользоваться считалкой. Считалка состоит из k слов, и используется следующим образом.

Все n ребят становятся в круг и один из них, начиная с себя, по очереди указывает на ребят в порядке, в котором они стоят по кругу, называя слова считалки. Тот, на кого указывает считающий, называя последнее слово считалки, выбывает из круга. После этого считалка повторяется сначала, а счет начинается со следующего за выбывшим. Так продолжается до тех пор, пока в круге не останется один человек. Он то и будет ведущим.

Но на этот раз ребята так увлеклись идеей предстоящей игры, что забывали выходить из круга, после того как считающий указывал на них, называя последнее слово считалки. В результате считающий снова указывал на них при следующих повторениях считалки.

Ребята заметили это только тогда, когда после очередного повторения считалки считающий снова указал на последнем слове на участника, который уже должен был покинуть круг. Теперь их заинтересовал вопрос — а на скольких ребят в этот момент считающий все еще не указал, что они должны покинуть круг.

Помогите им ответить на этот вопрос.

Формат входного файла

Входной файл содержит два целых числа — n и k ($1 \leq n \leq 1000$, $1 \leq k \leq 10^9$).

Формат выходного файла

Выведите в выходной файл одно число — количество ребят, на которых ведущий, еще не указал, что они должны покинуть круг, когда ведущий повторно укажет на кого-либо на последнем слове считалки.

Пример

<code>wrong.in</code>	<code>wrong.out</code>
6 14	3
6 13	0