Задача A. Гарри Поттер и три заклинания

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Давным-давно (наверное, ещё в первой книге) великий алхимик, создатель философского камня Николас Фламель научил Гарри Поттера трём полезным заклинаниям. Первое из них позволяет превратить a граммов песка в b граммов свинца, второе — c граммов свинца в d граммов золота, а третье — e граммов золота в f граммов песка. Когда Гарри рассказал об этих заклинаниях своим друзьям, Рон Уизли был в восторге: ведь если получится превращать песок в свинец, свинец в золото, часть золота — снова в песок и так далее, то можно будет, начиная с небольшого количества песка, получить огромное количество золота! Даже бесконечное количество золота! Гермиона Грейнджер, напротив, отнеслась к этой идее скептически. Она утверждает, что, согласно закону сохранения материи, невозможно получить бесконечное количество материи даже при помощи магии. Наоборот, количество материи при превращениях может даже уменьшаться, переходя в магическую энергию. Несмотря на то, что аргументы Гермионы выглядят убедительно, Рон не собирается ей верить. По его мнению, Гермиона придумала свой закон сохранения материи только для того, чтобы Гарри с Роном перестали заниматься ерундой, а лучше — шли учить уроки. Поэтому Рон уже набрал некоторое количество песка для экспериментов и, кажется, ссоры между друзьями не избежать...

Помогите Гарри определить, кто из его друзей прав, и всё-таки предотвратить ссору. Для этого вам придется выяснить, можно ли из некоторого конечного количества песка получить количество золота, большее любого наперёд заданного числа.

Формат входного файла

В первой строке заданы шесть целых чисел a, b, c, d, e, f ($0 \le a, b, c, d, e, f \le 1000$).

Формат выходного файла

Выведите “Ron”, если, имея некоторое конечное количество песка (и не имея вообще золота и свинца), возможно получить сколь угодно большое количество золота, то есть прав Рон. В противном случае выведите “Hermione”.

Примеры

stdin	stdout
100 200 250 150 200 250	Ron
100 50 50 200 200 100	Hermione
100 10 200 20 300 30	Hermione

Примечание

Разберём первый пример. Начнём с 500 граммов песка. Применяя 5 раз первое заклинание, превратим их в 1000 граммов свинца. Затем 4 раза применим второе заклинание и получим 600 граммов золота. Из них выделим 400 и превратим их снова в песок. Получим 500 граммов песка и 200 граммов золота. Применяя все те же операции к 500 граммам песка повторно, можно будет каждый раз получать дополнительные 200 граммов золота. Таким образом можно получить 200, 400, 600, ... граммов золота, то есть, начиная с конечного количества песка (500 граммов), можно получить количество золота, большее любого наперёд заданного числа.

Задача B. Антисортировка

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Обычно в условии задач вам говорят, что необходимо сделать. Нам это кажется скучным. В этой задаче мы поступим наоборот. Скажем, что вы дожны не сделать:

Вы не должны отсортировать данную последовательность.

Более формально, вам дана последовательность из S различных чисел. Переупорядочьте её любым способом. Единственное ограничение — полученная последовательность не должна быть отсортирована ни по возрастанию, ни по убыванию.

Формат входного файла

В первой строке вам дано число T, задающее количество тестов в файле. Перед каждым тестом есть пустая строка. Каждый тест состоит из двух строк. Первая строка содержит число N ($3 \le N \le 1000$), задающее длину последовательности. Вторая строка содержит N 32-битных знаковых чисел, разделённых пробелами. Эти числа попарно различны.

Формат выходного файла

Выведите переупорядоченные последовательности в таком же порядке и в таком же формате, как во вводе.

Пример

stdin	stdout
2	2
5 1 2 3 4 5	5 5 1 4 3 2
8 3 1 4 47 5 9 2 6	8 3 1 4 47 5 9 2 6

Задача C. Астрид и квадраты

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленькая девочка Астрид любит вырезать из бумаги различные фигуры. Особенно ей нравятся квадраты.

На столе перед Астрид лежит бумажный прямоугольник размера $m \times n$ сантиметров. Девочка хочет, чтобы на столе остались одни лишь квадраты. Пока это не так, она берёт со стола прямоугольник и одним прямолинейным разрезом отрезает от него квадрат. После этого квадрат остаётся на столе, а с остатком происходит то же самое: если он не квадратный, от него одним прямолинейным разрезом отрезается квадрат, и так далее. Наконец, после того, как очередное разрезание привело к появлению двух квадратов, они оба кладутся на стол, и разрезания заканчиваются.

Сколько квадратов окажется у Астрид на столе, когда она закончит разрезания?

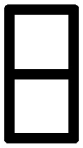
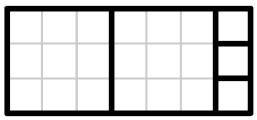
Формат входного файла

В единственной строке ввода заданы целые числа m и n — стороны исходного прямоугольника в сантиметрах ($1 \le m, n \le 1\,000\,000\,000$).

Формат выходного файла

Выведите одно число — количество квадратов, которые окажутся на столе после всех разрезаний.

Примеры

stdin	stdout	explanation
1 2	2	
7 3	5	

Пояснение

В пояснениях справа от примеров показан исходный прямоугольник. Он разделён на квадраты, которые окажутся на столе после всех разрезаний.

В первом примере от прямоугольника 1×2 отрезается квадрат 1×1. Оставшийся прямоугольник 1×1 также является квадратом, поэтому разрезания заканчиваются.

Во втором примере от прямоугольника 7×3 отрезается квадрат 3×3, и остаётся прямоугольник 4×3. От него отрезается ещё один квадрат 3×3, остаётся прямоугольник 1×3. От этого прямоугольника отрезается квадрат 1×1, остаётся прямоугольник 1×2. Наконец, при разрезании этого прямоугольника получается ещё два квадрата 1×1.

Задача D. Берт и землеройки

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленький мальчик Берт любит изучать поведение животных. В этот раз он наблюдает за поведением стайки землероек.

Берт поймал n землероек и выпустил их стайкой на лужайку перед своим домом. Как известно, землеройки предпочитают индивидуальный образ жизни, поэтому они стремятся скрыться от Берта и друг от друга как можно скорее. Каждую секунду последовательно происходят два события:

1. В начале секунды каждая стайка, в которой больше одной землеройки, разделяется ровно на две стайки. В каждой получившейся стайке должна быть хотя бы одна землеройка.
2. В конце секунды одна землеройка из каждой стайки прячется, зарывшись в траву.

Изначально все землеройки находятся в одной стайке. От того, как землеройки делятся на стайки в начале каждой секунды, зависит, сколько секунд пройдёт, прежде чем все они спрячутся. Какое минимальное и максимальное количество секунд может пройти от начала эксперимента, прежде чем все землеройки спрячутся?

Формат входного файла

В единственной строке ввода задано целое число n — количество землероек в начале эксперимента ($1 \le n \le 1\,000\,000\,000$).

Формат выходного файла

Выведите два числа, разделив их пробелом — минимальное и максимальное количество секунд, которое может пройти, прежде чем все землеройки спрячутся.

Примеры

stdin	stdout	explanation
2	1 1	$2 = 1 + 1 \xrightarrow{t=1}$ $2 = 1 + 1 \xrightarrow{t=1}$
5	2 3	$5 = 3 + 2 \xrightarrow{t=1} 2 + 1 = 1 + 1 + 1 \xrightarrow{t=2}$ $5 = 4 + 1 \xrightarrow{t=1} 3 = 2 + 1 \xrightarrow{t=2} 1 = 1 \xrightarrow{t=3}$

Пояснение

В пояснениях справа от примеров показаны варианты разделения землероек на стайки. В первой строке показан один из возможных вариантов, позволяющих землеройкам спрятаться за минимальное количество секунд, а во второй — за максимальное. Выражения вида $a_1 + a_2 + \ldots = b_1 + b_2 + \ldots$ означают, что в результате разделения стаек из $a_1, a_2, \ldots$ землероек образовались стайки из $b_1, b_2, \ldots$ землероек. Стрелка $\xrightarrow{\texttt{t}=x}$ означает конец x-й секунды. В этот момент одна землеройка из каждой стайки прячется, зарывшись в траву.

В первом примере стайка из двух землероек в начале первой секунды разделится на две стайки по одной землеройке, а в конце первой секунды обе землеройки спрячутся.

Во втором примере стайка из пяти землероек может в начале первой секунды разделиться на 3 и 2 землеройки, а может на 4 и 1 землеройку.

В первом случае в конце первой секунды на лужайке останется две стайки: из 2 землероек и из 1 землеройки. Первая из них в начале второй секунды разделится, и в конце второй секунды все три оставшиеся землеройки спрячутся.

Во втором случае в конце первой секунды осталась одна стайка из трёх землероек. В начале второй секунды она разделится на 2 и 1 землеройку. В конце второй секунды спрячутся все землеройки, кроме одной. Эта последняя землеройка спрячется в конце третьей секунды.

Задача E. Марсианская архитектура

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Кролик Крис нашел следы древней цивилизации Марса. В небольшой телескоп отважному астроному удалось увидеть архитектурный шедевр — "Дорогу к Солнцу". Это сооружение состоит из одинаковых по размеру кубических камней. Фундамент разбивает всю "дорогу" на ячейки, в которые плотно ложатся кубокамни. Таким образом, каждой ячейке фундамента можно приписать координату. Для того, чтобы стать вождём, марсианин должен быть проложить путь к Солнцу, то есть построить из этих кубокамней на заданном фундаменте лестницу. Лестницу можно описать количеством камней в начальной координате и координатами начала и конца лестницы. Каждая следующая ячейка в направлении возрастания координаты должна содержать на один кубокамень больше, чем предыдущая. Причём, если в ячейке уже были до этого камни — в текущем строительстве они не считаются, лестницу просто строили поверх них. Другими словами, пусть строится лестница с координатой начала l, координатой конца r и количеством камней в начальной координате x. Это означает, что в ячейке l **добавится** x камней, в $l+1$ добавится $x+1$ камней, ..., в ячейке r добавится $x+r-l$ камней.

Крису удалось отыскать древний манускрипт, содержащий описания всех лестниц. Теперь он хочет сравнить эти данные, чтобы быть уверенным в том, что нашёл именно "Дорогу к Солнцу". Для этого он выбрал некоторые ячейки дороги и посчитал суммарное количество кубокамней, которые накопились за всю марсианскую историю, а потом попросил вас с помощью манускрипта вычислить, чему должна быть равна эта сумма, для проверки.

Формат входного файла

Первая строка содержит три целых числа, разделённых пробелами: n, m и k ($1 \le n, m \le 10^5$, $1 \le k \le \min(100, n)$) — количество ячеек, количество "Дорог к Солнцу" и количество запросов соответственно. Каждая из последующих m строк содержит по три целых числа, разделённых пробелами: a_i, b_i и c_i ($1 \le a_i \le b_i \le n$, $1 \le c_i \le 1000$) — описание лестницы, содержащее координаты её начала и конца, а также высоту начальной ячейки. Далее следует строка, содержащая k различных целых чисел b_i, разделённых пробелами. Все эти числа в пределах от 1 до n — ячейки, количество камней в которых интересует Криса.

Формат выходного файла

Требуется вывести в единственной строке одно число — суммарное количество камней во всех интересующих Криса ячейках.

Примеры

stdin	stdout
5 2 1 1 5 1 2 4 1 3	5
3 2 1 1 3 1 1 3 1 2	4
3 2 1 1 3 1 1 3 1 3	6

Задача F. Камилла и язык программирования WR

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленькая девочка Камилла любит программировать. Она придумала и реализовала свой собственный язык программирования.

Этот язык называется "WR", и он работает с *wr-строками* — строками, состоящими только из букв "w" и "r". Программа на этом языке — это wr-строка, результат работы программы также является wr-строкой.

В языке WR всего две команды. Команда записи (write) начинается с маленькой буквы "w" и состоит из двух символов: первый из них — сама буква "w", а второй — буква, которую нужно вывести. Команда повтора (repeat) состоит из одного символа — маленькой буквы "r" — и означает, что нужно один раз повторить выполнение предыдущей команды.

При выполнении программа разбивается на команды слева направо, а затем команды выполняются последовательно. Если что-то из этого невозможно, вся программа считается некорректной. Это происходит в двух случаях:

1. Очередная команда начинается с буквы "w", но второго символа в этой команде нет, так как эта буква "w" — последняя буква программы.
2. Команда повтора — первая в программе, поэтому она не может повторить предыдущую команду.

Результат работы некорректной программы — пустая строка.

Камилле нравится, что на её языке можно написать программу, которая выведет другую программу, которую можно тоже выполнить и получить третью программу, и так далее. Девочка хочет последовательно выписать все программы, которые получились при таком процессе. Сначала она выбрала некоторую исходную wr-строку и объявила её *исполняемой*. После этого Камилла выполняет следующий алгоритм:

1. Выписывает исполняемую строку.
2. В результате работы исполняемой строки как программы на языке WR получает новую строку.
3. Если полученная новая строка пуста, заканчивает работу, а иначе объявляет эту новую строку исполняемой и вновь переходит к шагу 1.

По заданной исходной wr-строке выясните, какие строки и в каком порядке выписала Камилла.

Формат входного файла

В единственной строке ввода задана wr-строка — последовательность букв "w" и "r". Она содержит от 1 до 100 букв и не содержит пробелов.

Формат выходного файла

Выведите все строки, которые выписала Камилла, в порядке их выписывания.

Примеры

stdin	stdout	explanation
wwrrwrr	wwrrwrr wwwrr wrr rr	write w, repeat, repeat, write r, repeat write w, write r, repeat write r, repeat repeat <ERROR>
wwrwr	wwrwr wwr ww w	write w, repeat, write r write w, repeat write w write <ERROR>

Задача G. Ворчливые коровы

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

У фермера Джона N коров. Каждый вечер они выстраиваются в очередь, чтобы их подоили. У каждой коровы есть показатель ворчливости — целое число. Показатели ворчливости всех коров попарно различны.

Ворчливые коровы могут повредить оборудование фермы, поэтому фермер Джон хочет переупорядочить их так, чтобы доить коров в порядке возрастания их ворчливости. Фермер может выбрать двух любых коров (не обязательно стоящих рядом) с ворчливостями X и Y и поменять их местами за время $X + Y$. Эту операцию можно повторять последовательно сколько угодно раз.

Помогите фермеру Джону вычислить минимальное время, которое потребуется для того, чтобы переставить коров в очереди таким образом, чтобы доить их в порядке возрастания ворчливости.

Формат входного файла

В первой строке ввода задано целое число N — количество коров ($1 \le N \le 10\,000$). Следующие N строк содержат по одному числу каждая. Это ворчливости коров в порядке очереди на дойку. Гарантируется, что ворчливости — попарно различные целые числа от 1 до 100 000.

Формат выходного файла

Выведите одно число — минимальное время, за которое можно переупорядочить коров в порядке возрастания ворчливости.

Пример

stdin	stdout
3 2 3 1	7

Пояснение

В примере три коровы. Изначально первой стоит корова с ворчливостью 2, второй — с ворчливостью 3, а третьей — с ворчливостью 1.

Фермер Джон может сначала поменять коров с ворчливостями 1 и 3 за время $1 + 3 = 4$ и из порядка 2 3 1 получить порядок 2 1 3. Затем фермер может поменять коров с ворчливостями 1 и 2 за время $1 + 2 = 3$ и получить требуемый порядок 1 2 3.

Задача H. Давид и осёл

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленький мальчик Давид любит играть в настольные игры. Научившись играть в шахматы и шашки, он решил придумать свою фигуру, которая ходит по доске по определённым правилам.

Давиду очень нравится, как ходит шахматный конь, поэтому он захотел придумать что-нибудь похожее. Перебрав в уме известных ему животных, мальчик решил посвятить свою фигуру ослу.

Осёл, придуманный Давидом — это фигура, которая умеет перемещаться по прямоугольной клетчатой доске. Он ходит не так быстро, как шахматный конь. Кроме того, осёл упрям и не ходит в некоторые стороны. Следующая схема показывает все возможные ходы осла.

Центральная клетка схемы — начальное положение, а стрелки показывают, куда осёл может переместиться за один ход: на одну клетку вверх, влево, по диагонали влево-вниз, вправо и по диагонали вправо-вниз. Если осёл находится у края доски, можно делать только те из перечисленных ходов, которые не приводят к выходу за границы доски.

Давид гордится выносливостью своего осла: он может обойти любую прямоугольную доску размера $w \times h$ клеток, побывав в каждой её клетке ровно один раз.

Для заданных размеров доски — ширины w и высоты h — выясните, как это сделать. Разрешается начать маршрут из любой клетки доски. Возвращаться в исходную клетку не нужно.

Формат входного файла

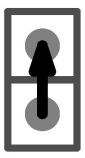
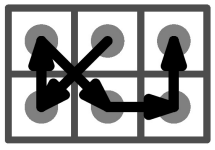
В единственной строке ввода заданы целые числа w и h — ширина и высота доски ($1 \le w, h \le 30$).

Формат выходного файла

Выведите $w \cdot h$ строк. В каждой из них выведите два числа, разделив их пробелом — номер столбца x и номер строки y посещённой ослом клетки. Столбцы доски нумеруются слева направо от 1 до w, а строки — сверху вниз от 1 до h. Клетки должны быть перечислены в порядке их посещения. Из каждой выведенной клетки, кроме последней, должно быть возможно за один ход переместить осла в следующую выведенную клетку согласно правилам. Каждая клетка доски должна встречаться в выводе ровно один раз.

Если возможных обходов доски несколько, выведите любой из них.

Примеры

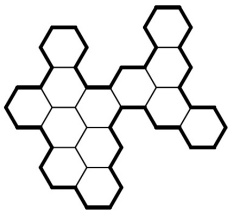
stdin	stdout	explanation
1 2	1 2 1 1	
3 2	2 1 1 2 1 1 2 2 3 2 3 1	

Задача I. Эмма и соты

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленькая девочка Эмма любит наблюдать за пчёлами. Недавно она нашла заброшенный улей, в котором сохранился кусочек рамки с пчелиными сотами.

Пчелиные соты, найденные Эммой, при взгляде сверху выглядят как одинаковые правильные шестиугольники со стороной 3 миллиметра, соединённые по сторонам. Пример того, как могут выглядеть соты, показан на рисунке.



Эмма решила записать, как выглядят найденные соты, чтобы потом нарисовать их. Она заметила, что граница сот — одна непрерывная ломаная линия, не пересекающая и не касающаяся сама себя, и решила, что придумать описание границы будет проще всего. Подумав, девочка решила записывать границу так: мысленно встать на неё, выбрать направление обхода и обойти всю границу, побывав в каждой её точке ровно один раз.

Рассмотрим подробнее этот мысленный эксперимент. Сначала Эмма встаёт в какой-то угол шестиугольника на границе сот. В этом углу встречаются два отрезка границы. Эмма поворачивается в сторону того из отрезков, для которого получается, что справа от неё соты, а слева — пустота. Затем девочка начинает двигаться по границе. Как можно увидеть на рисунке выше, граница состоит из прямых участков длиной 3 миллиметра, после каждого из которых линия границы поворачивает на 60 градусов либо направо, либо налево. Эмма выписывает последовательность поворотов, обозначая поворот направо заглавной буквой “`R`”, а поворот налево — заглавной буквой “`L`”, пока не окажется в исходной точке границы смотрящей в исходном направлении.

По полученной Эммой записи границы выясните, из скольких правильных шестиугольников со стороной 3 миллиметра состоят найденные соты.

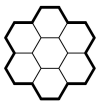
Формат входного файла

В единственной строке ввода записана последовательность поворотов при обходе границы. Эта строка состоит из букв “`L`” и “`R`”, имеет длину от 6 до 100 букв и не содержит пробелов. Гарантируется, что данная последовательность поворотов корректно задаёт границу некоторых сот в соответствии с условием, и эта граница не пересекает и не касается сама себя.

Формат выходного файла

Выведите одно число — количество правильных шестиугольников со стороной 3 миллиметра, из которых состоят соты, имеющие заданную границу.

Примеры

stdin	stdout	explanation
LRRRLRRRLRRR	3	
RLRRLRRLRRLRRLRRLR	7	

Пояснение

В пояснениях справа от примеров показаны соты, имеющие заданную границу.

В первом примере на границе 12 поворотов, и эта граница ограничивает соты, состоящие из трёх правильных шестиугольников со стороной 3 миллиметра.

Во втором примере на границе 18 поворотов, и эта граница ограничивает соты, состоящие из семи правильных шестиугольников со стороной 3 миллиметра.

Задача J. Побег

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Принцесса собирается сбежать из пещеры дракона. Это нужно как следует спланировать.

Принцесса бегает со скоростью v_p миль в час, дракон летает со скоростью v_d миль в час. Дракон обнаружит побег через t часов и немедленно бросится в погоню. Затея кажется безнадёжной, но принцесса заметила, что дракон очень жаден и не очень умён. Чтобы задержать его, принцесса решает прихватить с собой несколько драгоценностей из его сокровищницы. Когда дракон догоняет принцессу, она может бросить на землю драгоценность; тогда дракон остановится, подберёт упавшее, вернётся в пещеру и потратит f часов в самой пещере на возвращение вещицы на место и наведение порядка в сокровищнице, после чего вновь отправится в погоню, начиная всё с самого начала.

Предполагая, что принцесса будет бежать по прямой без остановок, выясните, сколько драгоценностей ей нужно взять с собой, чтобы успеть добраться до королевского замка, расположенного на расстоянии c миль от пещеры дракона? Если дракон догоняет принцессу в тот момент, когда она добегает до замка, считается, что она успела скрыться в замке раньше (дополнительная драгоценность не нужна).

Формат входного файла

Входные данные содержат целые числа v_p, v_d, t, f и c ($1 \le v_p, v_d \le 100$, $1 \le t, f \le 10$, $1 \le c \le 1000$), каждое число записано в отдельной строке.

Формат выходного файла

Выведите минимальное количество драгоценностей, необходимое для того, чтобы побег удался.

Примеры

stdin	stdout
1 2 1 1 10	2
1 2 1 1 8	1

Пояснение

В первом примере через час после побега принцесса будет находиться на расстоянии 1 от пещеры, а дракон обнаружит побег. Через 2 часа после побега дракон догонит принцессу на расстоянии 2 от пещеры, и ей нужно будет расстаться с первой драгоценностью. Возвращение в пещеру и возня в сокровищнице займёт у дракона ещё два часа; за это время принцесса убежит на расстояние 4 от пещеры. Во второй раз дракон догонит принцессу на расстоянии 8 от пещеры, и ей понадобится вторая драгоценность, после чего она спокойно добежит до замка.

Второй пример аналогичен первому, но второй раз дракон догонит принцессу ровно в тот момент, когда она добежит до замка, и вторая драгоценность ей не понадобится.

Задача K. Петя и Java

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленький Петя недавно начал посещать кружок по программированию. Естественно, перед ним появилась задача выбрать язык, на котором он будет программировать. После долгих размышлений он понял, что Java — лучший выбор. Главным аргументом в пользу выбора Java было то, что в ней есть целочисленный тип данных, позволяющий работать с очень большими числами: BigInteger.

Однако, после посещения занятий кружка Петя понял, что не все задачи требуют использования типа BigInteger. Как оказалось, в некоторых задачах намного удобнее использовать маленькие типы данных. Поэтому возникает вопрос: "Какой целочисленный тип использовать, если нужно хранить натуральное число n?"

Петя знает лишь пять целочисленных типов:

1. Тип **byte** занимает 1 байт и позволяет хранить числа от -128 до 127.
2. Тип **short** занимает 2 байта и позволяет хранить числа от $-32\,768$ до $32\,767$.
3. Тип **int** занимает 4 байта и позволяет хранить числа от $-2\,147\,483\,648$ до $2\,147\,483\,647$.
4. Тип **long** занимает 8 байт и позволяет хранить числа от $-9\,223\,372\,036\,854\,775\,808$ до $9\,223\,372\,036\,854\,775\,807$.
5. Тип **BigInteger** позволяет хранить любое целое число, но при этом не является примитивным типом, и операции с ним выполняются гораздо медленнее, чем с перечисленными выше типами.

Для всех указанных выше типов значения границ включаются в диапазон значений.

Из этого списка Петя хочет выбрать самый маленький тип, в котором можно хранить натуральное число n. Так как BigInteger работает гораздо медленнее остальных типов, Петя рассматривает его в последнюю очередь. Помогите ему.

Формат входного файла

В первой строке записано целое положительное число n. Оно состоит не более чем из 100 цифр и не содержит ведущих нулей. Число n не может являться пустой строкой.

Формат выходного файла

Выведите первый тип из списка "`byte`, `short`, `int`, `long`, `BigInteger`", в котором можно хранить натуральное число n в соответствии с данными, приведёнными выше.

Примеры

stdin	stdout
127	byte
130	short
12345678910111213141516171819202122232 4	BigInteger

Задача L. Цепная дробь

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Цепной дробью называется выражение вида

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots \cfrac{1}{a_m}}}},$$

где все a_k целые, $a_k > 0$ при $k > 0$ и $a_m > 1$, если $m > 0$. Эти, на первый взгляд странные, правила позволяют каждому вещественному числу x единственным образом сопоставить цепную дробь так, чтобы выполнялось равенство

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots \cfrac{1}{a_m}}}}.$$

Например, цепная дробь для числа $7/18$ выглядит как

$$0 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{1 + \frac{1}{3}}}}:$$

$$\begin{aligned} 7/18 &= 0 + 7/18 \\ &= 0 + 1/(18/7) \\ &= 0 + 1/(2 + 4/7) \\ &= 0 + 1/(2 + 1/(7/4)) \\ &= 0 + 1/(2 + 1/(1 + 3/4)) \\ &= 0 + 1/(2 + 1/(1 + 1/(4/3))) \\ &= 0 + 1/(2 + 1/(1 + 1/(1 + 1/3))). \end{aligned}$$

Цепную дробь можно записывать как $[\mathtt{a_0}; \mathtt{a_1}, \mathtt{a_2}, \ldots, \mathtt{a_m}]$; в этой записи $7/18 = [\mathtt{0}; \mathtt{2}, \mathtt{1}, \mathtt{1}, \mathtt{3}]$.

Не все числа имеют конечную цепную дробь; так, для числа $\sqrt{2} \approx 1,41421356\ldots$ цепная дробь имеет вид $[\mathtt{1}; \mathtt{2}, \mathtt{2}, \mathtt{2}, \mathtt{2}, \mathtt{2}, \mathtt{2}, \ldots]$ — воспользовавшись равенством $\sqrt{2} - 1 = 1/(\sqrt{2} + 1)$, получаем:

$$\begin{aligned} \sqrt{2} &= 1 + (\sqrt{2} - 1) \\ &= 1 + 1/(\sqrt{2} + 1) \\ &= 1 + 1/(2 + (\sqrt{2} - 1)) \\ &= 1 + 1/(2 + 1/(\sqrt{2} + 1)) \\ &= \ldots \end{aligned}$$

Цепная дробь такого вида называется *периодической*; в данном случае предпериод — это число $a_0 = 1$, а период — повторяющееся бесконечное количество раз число 2. Для удобства записи период цепной дроби можно записывать в круглых скобках; для $\sqrt{2}$ сокращённая запись будет выглядеть как $[\mathtt{1}; (\mathtt{2})]$. Можно доказать, что для всякого целого $N \geq 0$ цепная дробь числа $\sqrt{N}$ является конечной, если $\sqrt{N}$ — целое число, и периодической в противном случае.

По данному числу N выведите цепную дробь числа $\sqrt{N}$ в сокращённой записи.

Формат входного файла

В первой строке входного файла записано целое число N ($1 \leq N \leq 1000$).

Формат выходного файла

В первую и единственную строку выходного файла выведите сокращённую запись цепной дроби числа $\sqrt{N}$. Необходимо как можно более точно соблюдать такой же формат вывода, как в примерах. Гарантируется, что правильный ответ на каждый тест содержит не более 10 000 символов.

Примеры

stdin	stdout
1	[1]
2	[1;(2)]
3	[1;(1,2)]
4	[2]
5	[2;(4)]
6	[2;(2,4)]
7	[2;(1,1,1,4)]
76	[8;(1,2,1,1,5,4,5,1,1,2,1,16)]

Задача M. Шестерёнки

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Несколько шестерёнок соприкасается с зубчатой осью, одним зубцом каждая. Когда ось поворачивается на один зубец по часовой стрелке, все шестерёнки поворачиваются на один зубец против часовой стрелки. На шестерёнках может быть разное количество зубцов, но на каждой есть один, покрашенный зелёной краской.

Ось поворачивается с постоянной скоростью — на один зубец в секунду по часовой стрелке. Известно, что, если продолжать крутить ось бесконечно долго, в какой-то момент все зелёные зубцы одновременно соприкоснутся с осью. Также для каждой шестерёнки известно, сколько секунд требуется, чтобы из начального положения получить такое, в котором зелёный зубец этой шестерёнки впервые соприкоснётся с осью.

Выясните, в какой момент времени все зелёные зубцы впервые соприкоснутся с осью одновременно.

Формат входного файла

В первой строке входного файла записано число N ($2 \le N \le 4$) — количество шестерёнок. Вторая строка содержит N чисел A_1, A_2, …, A_N ($5 \le A_k \le 50$), записанных через пробел — количество зубцов на шестерёнках. В третьей же строке записано N чисел B_1, B_2, …, B_N ($0 \le B_k < A_k$), также через пробел — это моменты первого соприкосновения зелёного зубца каждой шестерёнки с осью. Другими словами, зелёный зубец k-й шестерёнки соприкасается с осью в моменты B_k, $B_k + A_k$, $B_k + 2A_k$, $B_k + 3A_k$, ….

Формат выходного файла

В выходной файл выведите одно число — номер секунды, в которую все зелёные зубцы впервые соприкоснутся с осью одновременно.

Примеры

stdin	stdout
2 5 6 0 0	0
3 5 6 7 1 2 3	206
4 50 50 50 40 25 25 25 25	25

Задача N. Задача на НОК

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Несколько дней назад я узнал, что существует такая штука, как наименьшее общее кратное (НОК). Теперь я часто играю с этим понятием — хочу сделать большое число с помощью НОК.

Но я не хочу использовать слишком много чисел, поэтому я выберу три целых положительных числа (не обязательно различных), каждое из которых не превышает n. Помогите мне найти максимально возможное наименьшее общее кратное таких трёх целых чисел.

Формат входного файла

В первой строке записано целое число n ($1 \le n \le 10^6$) — максимальное число, которое можно выбрать.

Формат выходного файла

Выведите единственное целое число — максимально возможное наименьшее общее кратное трёх не обязательно различных целых чисел, каждое из которых не превосходит n.

Примеры

stdin	stdout
9	504
7	210

Пояснение

Наименьшее общее кратное нескольких положительных целых чисел — это наименьшее положительное целое число, кратное им всем.

Результат может получиться достаточно большим. Возможно, 32-битного целого числа не будет достаточно для его хранения. Поэтому рекомендуется использовать 64-битные целые числа.

В последнем примере мы можем выбрать числа 7, 6, 5, их НОК равен $7 \cdot 6 \cdot 5 = 210$. Это — максимальный НОК, который мы можем получить.

Задача O. Сон Студента

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Как утверждает статистика, студенты спят не более трёх часов в сутки. Но даже в мире грёз, мерно посапывая, они не всегда могут избавиться от чувства нависшей угрозы.

Спит Бедный Студент, и снится ему, как сидит он на экзамене по математическому анализу у самого страшного преподавателя современности, трёхкратного героя Советского Союза, лауреата нобелевской премии по отчислению студентов, многоуважаемого профессора Петра Палыча.

Ни на один вопрос не смог ответить Бедный Студент. Значит, вместо большого просторного офиса придётся ему идти работать на ториевые шахты. Но подождите! Пётр Палыч решил дать Студенту последний шанс! Да, такое может быть только во сне.

И профессор начал: “Встретились венерянская девочка и марсианский мальчик на Земле и захотели прогуляться, держась за руки. Но вот беда: у девочки на левой руке a_l пальцев, а на правой — a_r пальцев. У мальчика, соответственно, b_l и b_r. Им будет удобно держаться за руки только тогда, когда никакая пара пальцев девочки не будет соприкасаться между собой, то есть между любыми двумя пальцами девочки есть палец мальчика. И вместе с этим, никакая тройка пальцев мальчика не должна соприкасаться между собой. Определите, смогут ли они взяться за руки так, чтобы обоим было удобно?”

Мальчику и девочке всё равно, кто пойдёт слева, а кто справа. Разница лишь в том, что, если мальчик пойдёт слева от девочки, то он возьмёт своей правой рукой её левую руку, а если он пойдёт справа — то наоборот.

Формат входного файла

В первой строке даны два целых положительных числа, не превышающие 100: количество пальцев на левой и правой руках венерянской девочки соответственно. Во второй строке даны два натуральных числа, не превышающие 100: количество пальцев на левой и правой руках марсианского мальчика соответственно.

Формат выходного файла

Выведите “YES” или “NO”: ответ на вопрос Петра Палыча.

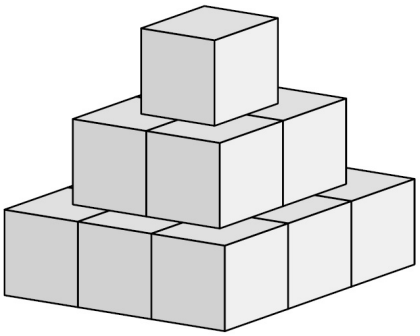
Примеры

stdin	stdout
5 1 10 5	YES
4 5 3 3	YES
1 2 11 6	NO

Задача P. Постройка пирамиды

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Великий Фараон Эхнатон, Сын Солнца, повелел построить новую пирамиду. Для постройки были доставлены n каменных блоков, имеющих форму куба пяти локтей в длину, пяти в ширину и пяти в высоту (локоть — древнеегипетская мера длины, равная примерно 466 миллиметрам). Пирамида строится так: сначала выкладывается основание $k \times k$ блоков, на него кладётся следующий ярус размера $(k-1) \times (k-1)$ блоков, и так далее до последнего k-го яруса, состоящего всего лишь из одного каменного блока. Пирамида высоты в три каменных блока, состоящая из $9 + 4 + 1 = 14$ блоков, показана на рисунке.



Главный землемер фараона хочет узнать, какова максимальная высота пирамиды, которую можно построить из n блоков — не зная заранее высоты, он не может начать строительство. К сожалению, эта задача оказалась ему не по силам, и он обратился за помощью к вам — своему помощнику.

Формат входного файла

В первой строке входного файла записано одно целое число n — количество каменных блоков ($1 \le n \le 32\,000$).

Формат выходного файла

В первой строке выходного файла выведите одно целое число — максимальную возможную высоту пирамиды в блоках.

Примеры

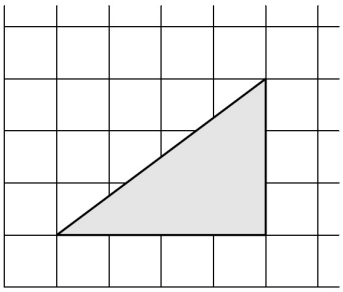
stdin	stdout
9	2
14	3

Задача Q. Треугольная комната

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

После укладки основания пирамиды (это оказался квадрат из 45×45 блоков) пришла пора вырубить в ней комнату — будущую усыпальницу для одного из родственников фараона. Эхнатон решил, что усыпальница в этот раз, вопреки всем канонам, будет треугольной. Стороны её должны иметь длины, кратные размерам блоков, из которых построена пирамида. Одна сторона должна быть в длину как a блоков, стоящих в ряд друг за другом, другая — как b блоков, третья — как c блоков. Числа a, b и c — это три знаменательных даты в жизни родственника фараона.

Главный землемер, руководящий строительством, хочет, кроме того, чтобы каждый из трёх углов комнаты находился на стыке четырёх блоков, из которых составлена пирамида. Иными словами, если нарисовать основание пирамиды на плоскости и ввести координаты так, чтобы один угол пирамиды оказался в начале координат, а противоположный — в точке $(45, 45)$, то вершины треугольника, соответствующего комнате, должны оказаться в точках с целыми координатами. Кроме того, углы комнаты должны быть строго внутри основания пирамиды. Пример такой комнаты для $a = 3$, $b = 5$ и $c = 4$ показан на рисунке.



Помогите главному землемеру и в этот раз — выясните, каким образом вырубить комнату с данными длинами сторон так, чтобы выполнить его требования, или укажите, что это невозможно.

Формат входного файла

В первой строке входного файла записаны через пробел три целых числа a, b и c — длины стен треугольной комнаты ($1 \le a, b, c \le 43$). Числа a, b и c таковы, что из трёх отрезков с такими длинами можно составить треугольник, площадь которого строго положительна.

Формат выходного файла

Если комнату с данными длинами стен указанным образом поместить в пирамиду невозможно, в первой строке выходного файла выведите "**NO**". В противном случае в первой строке выведите "**YES**", а в следующих трёх строках — координаты углов комнаты, по одному углу на строке. Порядок углов не имеет значения. Все координаты должны быть целыми числами в промежутке от 1 до 44, включительно.

Примеры

stdin	stdout
3 5 4	YES 1 1 5 1 5 4
2 3 4	NO

Задача R. SMS

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Маленький морж Клычок, как и все современные моржи, любит общаться с помощью SMS. Однажды он столкнулся с проблемой: при отправке больших текстов они разделяются на части по n символов (размер одного SMS сообщения), и разрываются целые предложения или даже слова!

Клычку это не понравилось, и перед ним встала задача разбить текст на сообщения самостоятельно таким образом, чтобы ни одно предложение не было разбито на части при отправке, а количество SMS сообщений для отправки было бы минимальным. Если два подряд идущих предложения находятся в разных сообщениях, то пробел между ними можно не учитывать (Клычок этот пробел не набирает).

Текст маленького моржа имеет следующий вид:

```
TEXT ::= SENTENCE | SENTENCE SPACE TEXT
SENTENCE ::= WORD SPACE SENTENCE | WORD END
END ::= {'.', '?', '!'}
WORD ::= LETTER | LETTER WORD
LETTER ::= {'a'..'z', 'A'..'Z'}
SPACE ::= ' '
```

Под SPACE следует подразумевать символ пробела.

Сколько же сообщений отправил Клычок?

Формат входного файла

В первой строке задаётся целое число n — размер одного сообщения ($2 \le n \le 255$). В следующей строке находится сам текст. Длина текста не превышает 10^4 символов. Гарантируется, что текст удовлетворяет описанному выше формату. В частности, из этого следует, что текст не пуст.

Формат выходного файла

В первой и единственной строке выведите количество SMS сообщений, которое потребуется Клычку. Если разбить текст невозможно, выведите "Impossible" без кавычек.

Примеры

stdin	stdout
25 Hello. I am a little walrus.	2
2 How are you?	Impossible
19 Hello! Do you like fish? Why?	3

Пояснение

Рассмотрим третий пример. Текст будет разбит на три сообщения: "Hello!", "Do you like fish?" и "Why?".

Задача S. Неквадраты

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Задано целое положительное число n. Выясните, может ли оно быть представлено в виде произведения k целых чисел, ни одно из которых *не* является квадратом целого числа.

Формат входного файла

Первая строка ввода содержит целое число t ($1 \le t \le 10$) — количество тестовых случаев. Каждая из последующих t строк содержит один тестовый случай, состоящий из двух целых чисел n ($1 \le n \le 1\,000\,000\,000$) и k ($2 \le k \le 50$).

Формат выходного файла

Для каждого тестового случая выведите в отдельной строке слово "YES" в том случае, если существует такой набор из k целых чисел a_i, что $n = a_1 \cdot a_2 \cdots a_k$ и ни одно из a_i не является квадратом целого числа, и слово "NO" в противном случае.

Пример

stdin	stdout
2 1 3 7 2	NO YES

Задача T. Перестановка цифр

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Нет ничего прекраснее, чем целое число.

Вам задано целое число n. Запишите его в десятичной записи без ведущих нулей. Пусть M — количество цифр в получившейся записи. Далее следует ровно k раз выполнить один шаг следующего вида:

Выберите i и j — две различные позиции в записи числа так, что $1 \le i < j \le M$ (цифры нумеруются с первой). Поменяйте местами цифры на этих двух позициях. Если после шага в числе появляются ведущие нули (то есть цифра на позиции j равна нулю, а $i = 1$), такой шаг делать нельзя.

Выведите наибольшее возможное число, которое может получиться после выполнения описанной процедуры. Если ровно k шагов сделать невозможно, выведите -1.

Формат входного файла

В первой строке ввода задано два целых числа n и k ($1 \le n \le 1\,000\,000$, $1 \le k \le 10$).

Формат выходного файла

Выведите одно целое число — ответ на задачу.

Примеры

stdin	stdout
16375 1	76315
432 1	423
90 4	-1
5 2	-1

Пояснение

В первом примере оптимально будет поменять местами цифры 1 и 7.

Во втором примере результат получается даже меньше, чем исходное число.

В третьем примере мы не можем сделать ни одного шага, поскольку в результате любого шага в числе появятся ведущие нули.

В четвёртом примере мы не можем выбрать две различных позиции для шага.

Задача U. Пристрастный учитель

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

По итогам учебного года учитель решил поощрить своих учеников — раздать им немного ирисок. Он попросил n учеников встать в ряд. Поскольку учитель очень пристрастен, он руководствуется следующим правилом при раздаче ирисок.

Он смотрит на первых двух учеников и даёт больше ирисок тому из них, у которого выше оценки. Если у этих двух учеников одинаковые оценки, то они получают одинаковое количество ирисок. Процесс раздачи проходит аналогичным образом для любой пары рядом стоящих учеников, начиная с первого и заканчивая последним.

Известно, что каждый ученик получит как минимум одну ириску. Вам следует определить, сколько ирисок учитель может дать каждому ученику, таким образом, чтобы общее количество розданных ирисок было наименьшим.

Формат входного файла

В первой строке ввода записано количество учеников n ($2 \le n \le 1000$). Во второй строке содержится $(n-1)$ знаков, каждый из которых — "L", "R" или "=". Для пары рядом стоящих учеников "L" означает, что у ученика слева оценки выше, "R" означает, что выше оценки у ученика справа, а "=" означает, что их оценки одинаковые. Первый знак соответствует отношению между оценками первого и второго учеников, второй знак — между оценками второго и третьего, и так далее. Первый ученик в ряду считается стоящим слева от всех, последний — справа.

Формат выходного файла

Выведите n целых — сколько ирисок получит каждый ученик в ряду, начиная с первого и заканчивая последним.

Примеры

stdin	stdout
5 LRLR	2 1 2 1 2
5 =RRR	1 1 2 3 4

Задача V. Поезда

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Вася живет посередине Программистской ветки метро. У него есть две девушки: Даша и Маша, живущие на разных концах ветки, каждая не ведающая о существовании другой.

Когда у Васи появляется свободное время, он едет к одной из своих девушек. Он спускается в метро в некоторый момент времени, ждёт первого поезда и едет на нём до конца ветки к соответствующей девушке. Однако поезда ходят с разной частотой: в Дашином направлении станцию проезжает один поезд в a минут, а в Машином — один поезд в b минут. В случае, если поезда подошли одновременно, Вася едет в направлении с меньшей частотой хождения поездов, то есть к той девушке, в чью сторону поезда ходят реже (см. пояснение к третьему примеру).

Известно, что поезда начинают своё хождение одновременно до появления Васи. То есть расписание поездов таково, что существует момент времени, когда два поезда приезжают одновременно.

Помогите Васе посчитать, к какой девушке он будет попадать чаще.

Формат входного файла

В первой строке записано два целых числа a и b ($a \neq b$, $1 \leq a, b \leq 10^6$).

Формат выходного файла

Выведите "`Dasha`", если Вася будет чаще попадать к Даше, "`Masha`", если к Маше, и "`Equal`", если одинаково часто к обеим девушкам.

Примеры

stdin	stdout
3 7	Dasha
5 3	Masha
2 3	Equal

Пояснение

Разберём третий пример.

Пусть поезда начали своё хождение в нулевой момент времени. Понятно, что моменты прибытия поездов будут периодичны с периодом 6. Поэтому достаточно показать, что, спускаясь в метро в момент времени внутри полуинтервала $(0, 6]$, Вася будет попадать к обеим девушкам одинаково часто.

Если Вася спустился в момент от 0 до 2, он уезжает к Даше на поезде, прибывающем ко второй минуте.

Если Вася спустился в момент от 2 до 3, он уезжает к Маше на поезде, прибывающем к третьей минуте.

Если Вася спустился в момент от 3 до 4, он уезжает к Даше на поезде, прибывающем к четвёртой минуте.

Наконец, если Вася спустился в момент от 4 до 6, то он дожидается прибытия обоих поездов к шестой минуте и уезжает к Маше, потому что в её направлении поезда ходят реже.

Суммарно на Дашу и Машу приходится поровну — по три минуты, значит, к обеим девушкам он будет попадать одинаково часто.

Задача W. Ловушки

Вход:	stdin
Выход:	stdout
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Эхнатон, Сын Солнца, хочет защитить новую пирамиду от воров и мародёров. Для этого в пирамиде, кроме усыпальницы, будет вырублено ещё несколько комнат поменьше. Все комнаты, включая усыпальницу, будут соединены системой коридоров таким образом, чтобы из любой комнаты в любую другую существовал ровно один путь, не проходящий ни по какому коридору дважды. Во многих комнатах будут установлены смертоносные ловушки.

У Эхнатона есть верный слуга — шаман родом из далёких южных стран, владеющий магией вуду. Фараон повелел шаману наложить заклятие на пирамиду: как только непрошеный посетитель оказывается в тупике — комнате, в которую ведёт только один коридор — волны отчаяния и безысходности должны захлестнуть его разум и подтолкнуть несчастного к ближайшей ловушке. Заклятие, однако, не должно действовать в священной комнате — усыпальнице, — даже если из неё ведёт всего один коридор.

Чтобы наложить заклятие, шаману необходимо знать точное количество тупиков в пирамиде. Чтобы узнать это число, он позвал вас — помощника главного землемера. К счастью, у вас сохранился план пирамиды. Ответьте шаману, сколько тупиков насчитывается на этом плане.

Формат входного файла

В первой строке входного файла записаны через пробел два целых числа n и m — количество комнат и количество коридоров, соответственно ($2 \le n \le 20$, $m > 0$). Следующие m строк содержат описания коридоров; в i-й из этих строк записаны два числа u_i и v_i через пробел, означающие, что i-й коридор соединяет комнаты с номерами u_i и v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$). Комнаты нумеруются с единицы; комната с номером 1 — это усыпальница. Любые две комнаты соединены не более чем одним коридором.

Формат выходного файла

В первой строке выходного файла выведите одно целое число — количество тупиков. Помните, что усыпальница тупиком не считается.

Примеры

stdin	stdout
3 2 1 2 1 3	2
3 2 1 2 2 3	1

Задача X. Магия вуду

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	0.5 с
Ограничение по памяти:	256 Мб

Чтобы наложить на пирамиду заклятие, шаман Эхнатона должен использовать один или несколько магических кристаллов. Каждый кристалл характеризуется целым числом — своей силой. От силы используемых кристаллов зависит мощность заклятия: при использовании k кристаллов, имеющих силы f_1, f_2, ..., f_k, заклятие будет иметь силу $(f_1 \cdot f_2 \cdot \ldots \cdot f_k) \bmod m$, где m — магическое число вуду, равное $2^{30} = 1\,073\,741\,824$. Выражение "$a \bmod b$" означает остаток от деления a на b.

Шаман уже убедился в ваших способностях к вычислениям, и дал новое задание. Он снабдил вас информацией об имеющихся у него магических кристаллах. Теперь шаман просит сказать ему, какой максимальной мощности может достичь заклятие при правильном выборе из имеющихся кристаллов одного или нескольких, которые надо использовать.

Формат входного файла

В первой строке входного файла записано целое число n — количество магических кристаллов, имеющихся в распоряжении шамана ($1 \le n \le 12$). Во второй строке записаны n целых чисел p_1, p_2, ..., p_n через пробел — силы магических кристаллов ($1 \le p_i \le 10\,000$).

Формат выходного файла

В первой строке выходного файла выведите одно число — максимальную возможную силу заклятия.

Примеры

stdin	stdout
2 1 3	3
3 10000 11 9999	99990000