

А. Студенческий автобус

Ограничение по времени: 1 секунда
Ограничение по памяти: 256МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

Из поселка Северный в университет регулярно ездит студенческий автобус. Водитель автобуса заметил одну особенность. Любой студент выбирает ближайший к водителю ряд, в котором есть хотя бы одно пустое место. В этом ряду студент выбирает самое левое (на схеме — верхнее) место из пустых. В каждом ряду, кроме последнего, есть пропуск для прохода по салону. Известно, что сейчас в автобусе N студентов. Распечатайте схему расположения студентов в автобусе. Формат схемы смотрите в примере.

Входные данные

В единственной строке указано целое число N ($0 \leq N \leq 49$).

Выходные данные

Выведите ровно семь строк символов — схему расположения студентов в автобусе. Пустое место обозначайте символом «#». Занятое место обозначайте символом «*». Остальные символы — декоративные (их тоже необходимо вывести). Строго придерживайтесь формата из примера.

Пример

Входные данные	Результат
9	<pre>+-----+ #.#.#.#.#.#.#.#.#.#.*.*.* D) #.#.#.#.#.#.#.#.#.#.#.*.*.* . #..... #...#.#.#.#.#.#.#.#.#.#.*.*... . #...#.#.#.#.#.#.#.#.#.#.#.#.*... .) +-----+</pre>
5	<pre>+-----+ #.#.#.#.#.#.#.#.#.#.#.#.*.* D) #.#.#.#.#.#.#.#.#.#.#.#.*.* . #..... #...#.#.#.#.#.#.#.#.#.#.#.#.*... . #...#.#.#.#.#.#.#.#.#.#.#.#.#... .) +-----+</pre>

Не забывайте, что для удобства вы можете скачать электронную версию условия с примерами.

В. Железная дорога 1

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной файл: стандартный поток ввода

Выходной файл: стандартный поток вывода

Железные дороги — один из наиболее популярных видов транспорта для дальних поездок в Китае. Сеть железных дорог в Китае организована следующим образом. Имеется S (Station) железнодорожных станций. Станции пронумерованы числами от 1 до S . Некоторые из них соединены железными дорогами. Между станциями проложены маршруты. Каждый из R (Route) маршрутов задаётся последовательностью из C (Count) номеров станций, через которые он проходит. Никакая станция не встречается дважды в одном маршруте. Поезда по каждому маршруту ходят в обоих направлениях. Можно садиться на поезд и выходить на любой станции.

Новый генеральный директор China Railways по имени Че Фа принял решение о разработке новой клиент-серверной системы, которая будет обслуживать продажу билетов. Разработка новой ИС была поручена вашей компании «FIFO Technologies». Вся система довольно сложна, но на первой итерации от вас требуется разработать лишь небольшую её часть. А именно, система должна уметь отвечать на самый важный для потенциальных пассажиров вопрос — возможно ли вообще доехать от станции A до станции B , используя любое количество пересадок. Следует считать, что от станции всегда можно доехать до самой себя.

Ваша программа должна ответить на последовательность из Q (Query) таких вопросов.

Входные данные

В первой строке теста указаны числа S и R ($1 \leq S \leq 10^4$, $0 \leq R \leq 10^4$) — общее количество станций и маршрутов соответственно. Далее следует R строк, описывающих маршруты. Каждая строка начинается с числа C_i ($2 \leq C_i \leq 100$) — количества станций в i -ом маршруте. Затем следует C_i чисел — номеров станций, через которые проходит маршрут (станции нумеруются с 1 до S).

Затем, в $(R + 2)$ -ой строке теста, указано число Q ($1 \leq Q \leq 10^4$) — количество запросов. Далее следует Q строк, описывающих запросы. Каждая строка содержит числа A и B — номера станций, между которыми требуется определить возможность проезда.

Выходные данные

Для каждого из Q запросов выведите в отдельной строке «Yes» или «No» — ответ на вопрос, можно ли проехать от станции A до станции B , используя любое количество пересадок.

Пример

Входные данные	Результат
10 3	Yes
3 1 2 3	Yes
4 2 4 5 6	No
3 7 8 9	Yes
5	No
1 3	
1 6	
3 8	
7 9	
10 5	

C. Yet Another Timetable App

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной файл: стандартный поток ввода

Выходной файл: стандартный поток вывода

Вася Пупкин — студент-первокурсник университета X-OSPU (eXtreme Open Source Programming University). Вася — довольно перспективный студент, а ещё он очень любит свой смартфон на RobotOS и мечтает научиться под него программировать. И вот, недавно в X-OSPU закончился первый семестр, и начались каникулы. Вася не хочет тратить время зря и, как, наверное, почти каждый уважающий себя студент-программист, решил написать своё приложение для просмотра расписания занятий на смартфоне (ну а вы посмотрите, сколько их в магазине Bubble Play).

Одной из важных функций Васиного приложения будет хранение номеров аудиторий для каждого занятия. При этом Вася хочет, чтобы приложение могло сообщать, на каком этаже находится эта аудитория. Номера всех аудиторий в X-OSPU состоят из ровно трёх десятичных цифр и одной маленькой английской буквы, написанной сразу после цифр без пробела. Буква обозначает корпус, первая цифра — номер этажа в корпусе (этажи нумеруются начиная 1), а вторая и третья цифры — номер аудитории на этаже. Например, аудитория 401f (которую студенты X-OSPU называют «Комната ужасов») находится на четвёртом этаже корпуса F (Факультета конечных Автоматов и Вычислений Тернарными операторами).

Помогите Васе реализовать эту функцию.

Входные данные

В единственной строке входных данных указан номер аудитории в описанном выше формате (три десятичные цифры, за которыми неразрывно следует маленькая английская буква).

Выходные данные

Выведите номер этажа, на котором находится аудитория с указанным номером.

Пример

Входные данные	Результат
401f	4
123x	1
908z	9

D. Студенты и сессия

Ограничение по времени: 1 секунда
Ограничение по памяти: 256МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

Как известно, после сдачи сессии студенты любят отдохнуть. Один из них решил поиграть в свою любимую компьютерную игру. Это игра, в которой необходимо строить роботов и собирать из них отряды и армии. Всего в игре имеются N типов роботов. Каждый тип роботов описывается тремя числами: количество роботов, которые уже построены; количество роботов данного типа, которое должно входить в каждый отряд, и цена за постройку одного робота в монетах. Всего у студента M монет. Чтоб победить в этой игре, необходимо узнать, какое максимальное количество отрядов можно создать. Новый отряд можно создать только в том случае, если роботы всех типов имеются или могут быть построены в достаточном для одного отряда количестве.

Входные данные

В первой строке через пробел даны 2 числа N и M ($1 \leq N \leq 10^5$, $0 \leq M \leq 10^{12}$) — количество типов роботов и начальное количество монет. В следующих N строках записано по три числа $start_i$, $count_i$ и $price_i$ ($0 \leq start_i, count_i \leq 100$, $1 \leq price_i \leq 100$) — количество роботов i -го типа, которые уже построены, количество, необходимое формирования одного отряда, и цена одного робота.

Выходные данные

В единственной строке выведите единственное число — максимальное количество отрядов, которое можно получить. Гарантируется, что ответ — не бесконечность.

Пример

Входные данные	Результат
3 4 6 3 1 4 2 2 1 1 3	2
3 21 1 2 1 10 0 10 1 1 1	7
3 1000000000000 1 1 1 1 1 1 1 1 3	2000000000001

Е. Многоугольник по серединам сторон 1

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной файл: стандартный поток ввода

Выходной файл: стандартный поток вывода

Заданы N точек на плоскости: A_1, A_2, \dots, A_N . Построить многоугольник, имеющий N сторон, с первой вершиной в начале координат, у которого N подряд идущих сторон имеют точки A_1, A_2, \dots, A_N в качестве середин (i -ая сторона должна иметь точку A_i в качестве середины). Допускается, чтобы построенный многоугольник имел самопересечения. Если это сделать невозможно, вывести «-1», иначе вывести координаты вершин многоугольника в порядке обхода, начиная с точки $(0;0)$.

Гарантируется, что середины отрезков заданы таким образом, чтобы не получалось трёх и более подряд идущих вершин, располагающихся на одной прямой.

Входные данные

В первой строке находится число N ($3 \leq N \leq 15$) — количество исходных точек. В следующих N строках находится по два числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) через пробел — целочисленные координаты точек A_1, A_2, \dots, A_N .

Выходные данные

Координаты вершин многоугольника, по одной точке в строке, либо одно число «-1», если нельзя построить.

Пример

Входные данные	Результат
3	0 0
1 1	2 2
3 2	4 2
2 1	

Ф. Бинарный поиск

Ограничение по времени: 3 секунды
Ограничение по памяти: 256МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

Сегодня у студентов группы PA-101 университета X-OSPU лекция по Теории алгоритмов. Их преподаватель, Николай Алгоритмович, рассказывает им про бинарный поиск. Вообще говоря, бинарный поиск — довольно абстрактный алгоритм, опытные олимпиадники даже умеют применять технику, называемую «бинарный поиск по ответу»... Но, начнём с простого. В частности, бинарный поиск позволяет довольно быстро (за $O(\log_2(n))$) проверить, есть ли в отсортированном массиве заданное число.

Николай Алгоритмович привёл ребятам вот такой пример реализации бинарного поиска:

```
bool check(const int* a, int n, int x) {
    int l = 0, r = n - 1;
    while (l <= r) {
        int m = (l + r) / 2;
        if (a[m] == x) return true;
        if (a[m] < x) l = m + 1;
        else r = m - 1;
    }
    return false;
}
```

Разумеется, сначала массив нужно отсортировать, но студенты PA-101 уже знают, что в библиотеке `<algorithm>` их любимого языка C++ уже есть готовая функция сортировки, так что отсортировать массив можно одной строчкой кода:

```
std::sort(a, a + n);
```

Для того, чтобы проверить, как студенты усвоили материал, Николай Алгоритмович задал им написать дома следующую программу. Программа получает на вход массив A , состоящий из N чисел (первое число имеет индекс 1, последнее — N), и Q запросов, каждый из которых состоит из одного числа X_i . Программа должна ответить на каждый запрос. Если в заданном массиве есть число с индексом X_i , то программа должна вывести индекс этого числа в массиве A , иначе — вывести «-1».

Помогите студентам справиться с домашним заданием. Разумеется, вы можете воспользоваться их конспектом.

Входные данные

В первой строке находится число N ($1 \leq N \leq 10^5$) — количество элементов в массиве A . Во второй строке находится N целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — элементы массива. В третьей строке находится число Q ($1 \leq Q \leq 10^5$) — количество запросов. В четвёртой строке находится Q целых чисел X_i ($-10^9 \leq X_i \leq 10^9$) — сами запросы.

Выходные данные

Для каждого из Q запросов выведите в отдельной строке одно число — ответ на соответствующий запрос.

Пример

Входные данные	Результат
3 1 2 3 5 3 1 2 5 42	3 1 2 -1 -1

Г. О девочках

Ограничение по времени: 3 секунды

Ограничение по памяти: 256МБ

Входной файл: стандартный поток ввода

Выходной файл: стандартный поток вывода

Саша и Таня — девочки. А девочки, как вы все знаете, любят играть в игры. И даже сейчас, надо же, пришла им в голову идея сыграть в игру.

У девочек есть набор из N целых положительных чисел. Так как набор этот был найден в интернете (а наборам чисел из Интернета верить нельзя ни в коем случае!), он странный — в нём точно найдётся как минимум одна пара чисел отличающихся в точности на единицу. Если Таня правильно назовёт наибольший общий делитель (GCD) всех этих чисел, то победит Саша, а если не назовет или назовет неправильно — победит Таня.

Таня очень любит свою подругу, поэтому хочет, чтобы она победила. Помогите ей это сделать — найти наибольший общий делитель данного набора.

Входные данные

В первой строке входных данных указано число N ($2 \leq N \leq 10^6$) — количество чисел в наборе. В следующей строке указаны числа из набора, разделенные пробелом. Каждое число в наборе находится в промежутке $[1; 10^9]$. О наборе точно известно, что в нём найдётся как минимум одна пара чисел, разница между которыми составляет в точности 1.

Выходные данные

Одно число — ответ на вопрос задачи.

Пример

Входные данные	Результат
2 1 2	1

Н. Счастливым студент

Ограничение по времени: 1 секунда
Ограничение по памяти: 256МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

Студент очень обрадовался сдаче экзамена. На радостях он стал бегать вокруг одного из корпусов своего ВУЗа. Этот корпус можно считать прямоугольником, левая нижняя координата которого лежит в точке $(0;0)$. Известно, что студент начал бегать из точки $(0;0)$. Он пробежал вокруг корпуса против часовой стрелки D метров. Теперь студент хочет знать координату, в которой он находится. Размеры корпуса — $N \times M$ метров. Корпус лежит стороной длины N на оси абсцисс (по оси OX) и стороной длины M на оси ординат (OY).

Входные данные

В единственной строке даны три целых числа: N , M и D ($1 \leq N, M \leq 10^6$, $0 \leq D \leq 10^9$).

Выходные данные

Выведите через пробел два числа: X и Y — координаты точки, в которой находится студент, пробежав D метров.

Пример

Входные данные	Результат
5 10 16	4 10
4 4 3	3 0

I. Железная дорога 2

Ограничение по времени: 2 секунды

Ограничение по памяти: 256МБ

Входной файл: стандартный поток ввода

Выходной файл: стандартный поток вывода

Вы продолжаете работу над новой информационной системой для China Railways. Первая итерация прошла успешно и теперь вам предстоит наращивать функционал системы. На второй итерации вы должны реализовать самую важную для компании функцию — подсчёт стоимости проезда.

Имеется некоторое количество станций. Между ними проложены маршруты. Каждый i -ый из R (Route) маршрутов представляет из себя последовательность из S_i (Station) станций, через которые он проходит. Стоимости проезда между разными станциями различны. Для каждого маршрута задана последовательность из (S_i-1) чисел P_{ij} (Price). Число P_{ij} определяет стоимость проезда между j -ой и $(j+1)$ -ой станциями i -ого маршрута. Пассажир может купить билет как от начальной до конечной, так и между двумя промежуточными станциями. Стоимость билета от a -ой до b -ой станции в r -ом маршруте равна сумме стоимостей всех проездов на пути от станции a до станции b в r -ом маршруте. Считается, что доехать от станции до самой себя можно бесплатно.

Системе будут поступать запросы вида (r, a, b) . Система должна подсчитать стоимость проезда между станциями a и b по маршруту r . Здесь a и b — порядковые номера станций в маршруте (маршруты нумеруются с 1 до R , станции в маршруте нумеруются с 1 до S_r). Ваша программа должна ответить на последовательность из Q (Query) таких запросов.

Директор China Railways Че Фа просит вас не забывать, что вы «решаете задачу для Китая», поэтому вам следует ожидать огромного количества потенциальных пассажиров, а следовательно и запросов к вашей системе.

Входные данные

В первой строке теста указано число R ($1 \leq R \leq 100$) — количество маршрутов. Далее следует R строк, описывающих маршруты. Каждая строка начинается с числа S_i ($2 \leq S_i \leq 10^5$, $S_1 + S_2 + \dots + S_r \leq 10^5$) — количества станций в i -ом маршруте, за которым следует (S_i-1) целых чисел P_{ij} ($1 \leq P_{ij} \leq 100$) — стоимости проезда между j -ой и $(j+1)$ -ой станциями данного маршрута.

Затем, в $(R+2)$ -ой строке теста, указано число Q ($1 \leq Q \leq 10^5$) — количество запросов. Далее следует Q строк, описывающих запросы. Каждая строка содержит числа r, a и b — номер маршрута и номера станций (маршруты нумеруются с 1 до R , станции нумеруются с 1 до S_r).

Выходные данные

Для каждого из Q запросов выведите в отдельной строке стоимость проезда между станциями a и b по маршруту r .

Пример

Входные данные	Результат
3	15
5 10 5 6 8	19
6 6 8 10 3 1	24
3 5 3	22
5	5
1 1 3	
1 2 5	
2 1 4	
2 6 2	
3 1 2	

Примечание

В этой задаче нет сквозной нумерации станций. Станции нумеруются отдельно в каждом маршруте. В этой задаче нет пересадок.

J. Dumbness of the Ancients 2

Ограничение по времени: 2 секунды
Ограничение по памяти: 256МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

На клетчатом поле $M \times M$ находятся герой и N врагов. Герой изначально находится в клетке с координатами $(1;1)$. Для каждого врага известно его положение на карте и его запас здоровья H . Герой умеет перемещаться по карте — за одну секунду он может переместиться в любую клетку, имеющую смежное ребро с той, в которой он находится. Когда герой приходит в клетку с врагом, враг моментально погибает от меча героя и герой получает G единиц золота. В самом начале герой единожды накладывает на всех врагов заклинание «Порча», из-за чего каждый враг каждую секунду теряет по одной единице здоровья. Если из-за «порчи» здоровье врага опускается до нуля, враг умирает, и герой не получает золота за его смерть. Каждую секунду сначала срабатывает «Порча», и лишь после этого происходит шаг героя.

Герой очень любит золото и очень хочет получить его как можно больше. Ваша задача — выяснить, какое максимальное количество золота герой может получить.

Входные данные

В первой строке указано целое положительное число N ($1 \leq N \leq 13$) — количество врагов. Во второй строке указан размер стороны поля M ($2 \leq M \leq 50$).

Далее следует N строк, содержащих описание врагов. Для каждого врага через пробел указаны координаты X и Y ($2 \leq X, Y \leq M$) его положения на карте и количество его здоровья H ($1 \leq H \leq 350$).

В последней строке написано целое положительное число G ($1 \leq G \leq 42$) — количество золота, которое герой получает за убийство одного врага.

Все числа во входных данных являются целыми. Никакие два врага не находятся в одной клетке. Ни один враг не находится в клетке $(1;1)$.

Выходные данные

Единственное целое число — максимальное количество золота, которое герой может получить за убийство врагов.

Пример

Входные данные	Результат
4 11 9 9 4 3 4 6 6 8 20 8 3 12 13	39

К. В + А

Ограничение по времени: 8 секунд
Ограничение по памяти: 15МБ
Ограничение по памяти для Java: 25МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

Думаете, что $A + B$ — простая задача? Да, пожалуй вы правы. Тогда попробуйте решить задачу $B + A$. Для строк. Под плюсом понимается конкатенация. Думаете всё-равно просто? Как бы не так...

Строго говоря, вам даны строка $A = a_1 a_2 a_3 \dots a_{n-1} a_n$ и строка $B = b_1 b_2 b_3 \dots b_{m-1} b_m$. Вам нужно вывести строку $C = B + A = b_1 b_2 b_3 \dots b_{m-1} b_m a_1 a_2 a_3 \dots a_{n-1} a_n$.

Входные данные

В первой строке входного потока данных задана строка A . Во второй строке входного потока данных задана строка B . Длины обеих строк находятся в диапазоне от 0 до $2 \cdot 10^7$ символов. Обе строки состоят исключительно из маленьких английских букв — это очень важно.

Учтите, что входной поток данных можно прочесть только один раз.

Выходные данные

Выведите в выходной поток строку, состоящую из всех символов строки B , за которыми следуют все символы строки A , т.е. конкатенацию строк $B + A$.

Пример

Входные данные	Результат
bar foo	foobar
world hello	helloworld
two forty	fortytwo

Подсказка

Если вы всё ещё не видите, где в этой задаче подвох, вы можете легко узнать это за один фэйловый сабмит ;-)

L. Железная дорога 3

Ограничение по времени: 1 секунда
Ограничение по памяти: 256МБ
Входной файл: стандартный поток ввода
Выходной файл: стандартный поток вывода

С момента начала разработки вашей системы обслуживания продажи железнодорожных билетов прошло почти 3 года. Система уже полностью реализована и отлично работает. Но в один день случилась неприятность: вся информация о маршрутах движения поездов исчезла из базы данных. «Хакерская атака!» — подумали вы, и начали изучать логи:

```
Fri Mar 13 13:13:09 2015: out of disk space
Fri Mar 13 13:13:10 2015: deleting unneeded data
Fri Mar 13 13:13:12 2015: still not enough disk space
Fri Mar 13 13:13:13 2015: oh, well... deleting NEEDED data then
```

«Нет времени обламывать руки тому, кто это накодил!» — подумали вы. Нужно срочно восстанавливать резервные копии. Стоп, вот ещё одна строчка:

```
Fri Mar 13 13:13:11 2015: backups? such a waste! deleting...
```

«What a Terrible Failure condition» — подумали вы. Похоже, кто-то из ваших коллег скоро возьмёт блокнот и ручку и отправится в пешую прогулку вдоль рельсов по всему Китаю.

Тем не менее, к вашему серверу продолжают поступать запросы вида «Каким маршрутом можно проехать от станции A до станции B без пересадок?». К счастью, в базе данных сохранилось общее количество станций S (Station) и маршрутов R (Route). Поскольку не известно, какие маршруты проходят через какие станции, было решено, что сервер будет возвращать случайное целое число в диапазоне от 1 до R в качестве номера предлагаемого маршрута. «В крайнем случае — прогуляются» — подумали вы.

Теперь, предположим, у вас уже есть информация обо всех маршрутах и обо всех запросах, поступивших к серверу. Определите, с какой вероятностью случайный ответ сервера мог быть правильным. Обратите внимание, что каждая пара станций может соединяться несколькими маршрутами, а поезда по каждому маршруту ходят в обоих направлениях.

Входные данные

В первой строке находится три целых числа: S , R и Q ($2 \leq S \leq 1000$, $1 \leq R \leq 1000$, $1 \leq Q \leq 2000$) — общее количество станций, маршрутов движения поездов и запросов, поступивших к серверу. В следующих R строках следуют описания маршрутов. Каждая строка начинается с числа C_i ($2 \leq C_i \leq S$) — количества станций в i -ом маршруте. За ним следуют C_i чисел P_{ij} ($1 \leq P_{ij} \leq S$) — номера станций, через которые проходит i -ый маршрут. Номера станций в пределах одного маршрута не повторяются.

В следующих Q строках находятся описания запросов, поступивших к серверу. Каждый запрос описывается двумя числами: A_i и B_i ($1 \leq A_i, B_i \leq S$, $A_i \neq B_i$) — номерами станций, для которых сервер должен был найти номер маршрута, соединяющего их.

Выходные данные

Для каждого из Q запросов выведите в отдельной строке вероятность, с которой случайный ответ сервера мог быть верным, в виде несократимой обыкновенной дроби P/Q ($P \geq 0$, $Q > 0$). Несоблюдение требований к виду дроби приведёт к вердикту Presentation Error.

Пример

Входные данные	Результат
5 3 4 3 1 5 4 4 5 2 4 1 3 5 3 1 4 5 5 1 2 3 4 2	2/3 1/1 0/1 1/3
10 4 1 6 1 2 3 4 5 6 6 7 2 8 9 5 10 3 7 4 6 5 1 8 10 3 6 2 5	1/2

М. Карусель

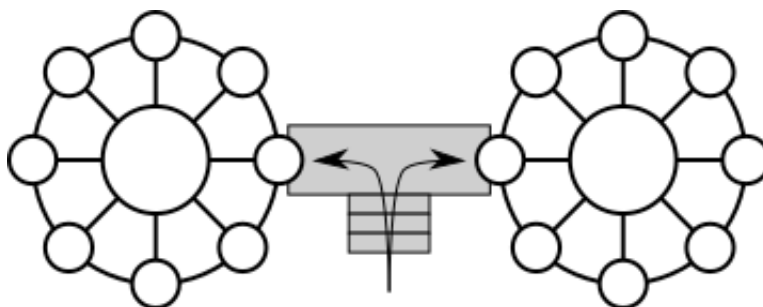
Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Входной файл: стандартный поток ввода

Выходной файл: стандартный поток вывода

Два друга решили покататься на каруселях. В парке были две карусели, соединенные общей посадочной площадкой, и мальчики решили кататься одновременно на разных.



Время их катания ограничено 1 часом. Заданы скорости вращения каруселей V_1 и V_2 в виде целого числа оборотов в час. Известно, что мальчики начали катание одновременно. Определить, через какое минимальное время после начала движения они снова окажутся на минимальном расстоянии друг от друга.

Входные данные

В единственной строке через пробел записаны два целых положительных числа V_1 и V_2 ($1 \leq V_1, V_2 \leq 30$).

Выходные данные

Ответ на вопрос задачи в виде несократимой обыкновенной дроби P/Q без незначащих нулей в числителе и знаменателе.

Пример

Входные данные	Результат
2 3	1/1
6 8	1/2
1 1	1/1
30 30	1/30