

Задача А. Amphitheatre

Имя входного файла: `amphitheatre.in`
Имя выходного файла: `amphitheatre.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Здесь можно копать!

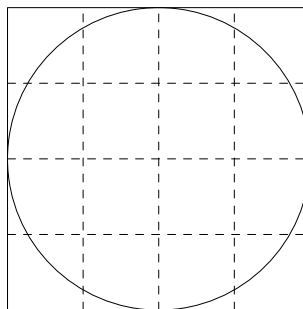
НоММ III

Однажды вечером Ааз заявил Скиву, что устал от совершенно ненаучных методов ведения работы и от репутации всемогущих шарлатанов от магии. «Тебе нужно научное имя. Пусть не такое, как у академика Я.П. Невструева или академика К. Хунты, но хотя бы такое, как у профессора Я.П.А. Невструева, доктора Э. Амперяна или, в крайнем случае, профессора С. Снейпа.» Скив напомнил, что его познания в теоретической магии как минимум обрывочны. Ааз сказал, что это не имеет значения и пообещал что-то придумать.

Было раннее утро, но в штаб-квартире корпорации «МИФ» уже сидел какой-то посетитель. Более того, только что проснувшийся Скив увидел, что беседа между посетителем и Аазом в самом разгаре — очевидно, для них это был всего лишь поздний вечер.

Посетитель был психологом и сотрудничал с ИАИА (Институтом Археологии и Антиквариата). Сотрудники этого института недавно нашли в одном измерении древний амфитеатр и готовились изучить его более детально. Посетитель же, в свою очередь, исследовал такой тонкий вопрос, как количественное измерение удовольствия от работы, получаемого археологами.

Область раскопок — квадрат со сторонами, параллельным направлениям с севера на юг и с запада на восток. Область выбрана так, чтобы она содержала амфитеатр (имеющий форму круга радиуса r) и при этом имела наименьшую возможную площадь. После этого область раскопок была разбита на участки k прямыми, идущими с севера на юг и l прямыми, идущими с запада на восток, при этом все расстояния между двумя соседними параллельными линиями, а также между стороной квадрата и ближайшей параллельной к ней линией, идущими в одном направлении, одинаковы.



Но как численно определить понятие удовольствия от научной работы? Ааз предложил определить *удовольствие* от процесса раскопок для настоящего археолога как площадь, которую занимает амфитеатр на участке, на котором работает данный археолог.

Идея была принята, более того, учёный согласился отдать статью в только что основанный Аазом научный журнал «Междисциплинарный исследовательский форум» (сокращённо — МИФ). Главным редактором журнала значился Скив. «Это самый быстрый способ сделать научное имя. Скоро ты окажешься соавтором таких работ, о которых ты даже не догадывался,» — ответил Ааз удивлённому Скиву.

Для завершения статьи, однако, требуется подсчитать для заданного разбиения зоны раскопок на участки наивысшее удовольствие от работы на одном участке. Вас, принятого в штат журнала на должность программиста-рецензента, попросили написать программу для такого подсчёта.

Формат входного файла

Единственная строка входного файла содержит три целых числа k , l и r ($1 \leq k, l, r \leq 1000$).

Формат выходного файла

Выведите одно целое число — наивысшее удовольствие от работы археолога, которое можно получить на одном из заданных участков. Ответ считается правильным, если абсолютная или относительная ошибка не превышает 10^{-6} .

Примеры

amphitheatre.in	amphitheatre.out
3 3 5	6.250000

Задача В. Brackets

Имя входного файла: `brackets.in`
Имя выходного файла: `brackets.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Именная работа с офрографией

Из рекламы MS Word 2010

Институт круглых скобок (ИКС) передал для публикации в новый журнал статью о регулярных скобочных последовательностях.

Так как статья была передана в редакцию в формате Word, то при подготовке статьи в печать Гвидо, получивший в журнале должность главного корректора, заметил странный эффект — в двух регулярных скобочных последовательностях длины $2N$ Word переставил между собой символы, находившиеся в этих последовательностях на одинаковых позициях. Например, из последовательностей $A = (())$ и $B = ()()$ в результате подобного сбоя могли быть получены четыре пары последовательностей:

$$\begin{aligned} A' &= (()), & B' &= ()() \\ A' &= (((), & B' &= ()) \\ A' &= ()) , & B' &= (((\\ A' &= () (), & B' &= (() \end{aligned}$$

Здесь в первом случае перестановок не было. Во втором случае были переставлены скобки в позиции 3. В третьем случае были переставлены скобки в позиции 2. В четвёртом случае были переставлены скобки как в позиции 2, так и в позиции 3 — итого 2 перестановки. Естественно, что получившиеся после сбоя последовательности вовсе не обязаны быть регулярными.

Проблема заключается в том, что Гвидо не знает, как изначально выглядели последовательности A и B . Например, если в результате получились $A' = ((($ и $B' = ())$, изначально последовательность могли быть $A = (()$ и $B = ()()$ (Word переставил скобки в позиции 3), или же они могли быть $A = ()()$ и $B = (()$ — тогда были переставлены скобки в позиции 2.

О замеченном сбое было доложено Аазу. Тот распорядился, во-первых, отныне принимать статьи только в формате TeX, а во-вторых, поручил Вам найти минимальное количество перестановок между строками, которые мог сделать Word для того, чтобы получить такой результат.

Формат входного файла

Входной файл содержит две строки, каждая из которых содержит последовательность, составленную из знаков '(' и ')'. Длина каждой из последовательностей равна $2N$ ($1 \leq N \leq 1000$).

Формат выходного файла

Выведите одно целое число — ответ к задаче. Если заданные строки не могли быть получены из регулярных скобочных выражений подобным образом, выведите -1 .

Примеры

<code>brackets.in</code>	<code>brackets.out</code>
<code>(((</code> <code>)))</code>	1
<code>((((</code> <code>)))</code>	-1

Задача C. Circle

Имя входного файла: `circle.in`
Имя выходного файла: `circle.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Даже в аду я обойду его на круг!

Из интервью автогонщика

Одной из идей, которая, по замыслу Ааза, должна была привлечь к журналу широкие массы думающей публики, являлась обязательная публикация в каждом номере журнала статьи по какой-то новой игре, сопровождаемой задачей для читателей.

«Читатели устали от кроссвордов и sudoku. Мы должны разрушить стандарты в задачах на нестандартное мышление,» — заявил Ааз на совещании редакции.

В первом номере журнала была опубликована следующая игра.

N ячеек с номерами от 1 до N расположены по кругу по часовой стрелке. В эти ячейки произвольным образом помещают N фишек, также занумерованных от 1 до N , по одной фишке в ячейку. Каждая из ячеек соединена с двумя соседними с ней по кругу (то есть любые две ячейки с номерами, отличающимися на 1, соединены, ячейки с номерами 1 и N также соединены). Также имеется центральная ячейка, соединённая с 4 ячейками с номерами 1, $N/2$, $N/2 + 1$, N . Изначально центральная ячейка пуста. Можно передвигать фишку в пустую ячейку, соединённую с той, в которой в данный момент находится фишка.

Требуется, передвигая фишки по этим правилам, добиться того, что каждая фишка будет находиться в ячейке со своим номером. Вам поручено написать программу, которая находит решение указанной головоломки за не более, чем 5×10^5 ходов — иначе ответ не поместится в отведённую для него во втором номере журнала площадь.

Формат входного файла

В первой строке входного файла записано одно чётное число N ($4 \leq N \leq 100$). Во второй строке записаны N целых чисел, при этом i -е число обозначает номер фишки в ячейке с номером i .

Формат выходного файла

В выходной файл выведите последовательность номеров фишек, которые вы должны двигать для решения задачи. Если последовательность будет содержать более, чем 5×10^5 чисел, Ваше решение будет признано неверным.

Примеры

<code>circle.in</code>	<code>circle.out</code>
4 1 2 4 3	3 4 3
4 4 3 2 1	4 1 4 3 2 3

Задача D. Computer

Имя входного файла: `computer.in`
Имя выходного файла: `computer.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

В мире нет ничего абсолютного, кроме абсолютной ошибки.

проф. Н.Н. Налейпиво

Учёные Института Шестимерной Архитектуры Компьютеров (ИШАК) недавно разработали новый компьютер. Это поистине фантастическая машина — у неё бесконечно много оперативной памяти и все операции выполняются мгновенно. Благодаря своим связям Ааз добился того, что первая публикация по этому проекту будет сделана в журнале «МИФ».

Вам поручено заняться разработкой программного обеспечения для этого компьютера. Однако довольно скоро Вы заметили, что в проекте имеются некоторые недостатки.

1. Процессор этого компьютера имеет только две операции — умножение или сложение двух целых чисел, записанных в памяти, с последующей записью результата в память.
2. Несмотря на то, что операции вычисляются корректно, при записи результата в память может возникнуть ошибка, и записываемое число произвольно уменьшится или увеличится на 1. Например, при вычислении $2 \cdot 3$ результат будет равен 6, но в память при этом может быть записано любое из трёх чисел 5, 6 или 7.
3. Пока что компьютер не имеет устройств ввода-вывода, и сразу после загрузки в его памяти находятся только два целых положительных числа x и y , которые при последующих вычислениях не могут быть перезаписаны.

Результатом выполнения программы является последнее число, записанное в память.

Авторы статьи хотят, чтобы Вы написали программу, для которой наибольшее возможное отклонение результата её выполнения от заданного числа z было бы минимально.

Формат входного файла

Входной файл содержит три целых числа $1 \leq x, y, z \leq 1000$. При этом z не может быть равно ни x , ни y .

Формат выходного файла

Выведите наибольшее расстояние от результата выполнения заказанной авторами программы до z .

Примеры

<code>computer.in</code>	<code>computer.out</code>
2 3 6	1
2 3 7	2

Задача E. Guards in the Chess Kingdom (Division 1 Only!)

Имя входного файла: `guards.in`
Имя выходного файла: `guards.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

For great justice.

Zero Wing

- Каждый номер журнала должна открывать редакторская статья, — объяснял Скиву Ааз.
- Но я не знаю, о чём писать...
- Напиши статью по экономике — о том, как мы помогли навести порядок в Шахматном Королевстве. В соавторы возьми кого-нибудь из Института Экономических Химер (ИЭХ).

Шахматное Королевство состоит из n городов, соединённых m двунаправленными дорогами. Каждый город окрашен в один из двух цветов — белый или чёрный. При этом никакие два города одного цвета не соединены дорогой.

Для того, чтобы обеспечить спокойное передвижение по дорогам, король по совету консультантов из корпорации «МИФ» принял решение о размещении в некоторых городах гарнизонов. Гарнизон в городе a обеспечивает порядок на всех дорогах, идущих в этот город. Королю требуется, чтобы на *всех* дорогах в королевстве был порядок, так что для каждой дороги хотя бы в одном из городов, которые она соединяет, должен присутствовать гарнизон. Содержание гарнизона обходится казне в изрядную сумму денег, так что король хочет обойтись минимальным количеством гарнизонов.

Кроме того, размещение гарнизона в городе обеспечивает порядок и в самом этом городе. Так что король хочет, чтобы, при выполнении уже объявленных условий, суммарное население городов, в которых будут размещены гарнизоны, было наибольшим.

Таким образом, руководство корпорации «МИФ» поручило Вам написать программу, которая выбирает подмножество городов для размещения в них гарнизонов. Количество городов в этом подмножестве должно быть минимально возможным. Если таких подмножеств несколько, выберите то из них, в котором проживает максимальное количество жителей. Если и таких подмножеств несколько — выведите любое из них.

Формат входного файла

В первой строке входного файла содержатся два целых числа — n и m ($(1 \leq n \leq 300)$, $(1 \leq m \leq 10^4)$). n — количество городов в королевстве, m — количество дорог.

Вторая строка содержит n целых чисел — население городов $1, 2, \dots, n$, соответственно. Население каждого из городов положительно и не превосходит 1000.

Следующие m строк содержат описание дорог. Строка с номером $i+2$ содержит 2 целых числа — a_i и b_i — это значит, что i -я дорога соединяет города a_i и b_i (нумерация городов идёт с единицы).

Формат выходного файла

Первая строка выходного файла должна содержать два целых числа — P и K . P — суммарное население городов, в которых должны быть размещены гарнизоны, K — количество городов. Вторая (и последняя) строка должна содержать K целых чисел — номера соответствующих городов, отсортированные по возрастанию.

Пример

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

Задача F. Integral (Division 1 Only!)

Имя входного файла: `integral.in`
Имя выходного файла: `integral.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

`submit.exe? head.think!`

ИШАК (Институт Шестнадцатеричной Архитектуры Компьютеров) опубликовал статью об ещё одной разработке. Новый суперкомпьютер может мгновенно вычислять любой определённый интеграл заданного полинома. Однако этот компьютер работает только с полиномами с целыми коэффициентами и выдаёт правильный ответ только в случае, когда определённый интеграл является целым числом.

$$I_i = \int_{u_i}^{v_i} \sum_{k=0}^n a_k x^k dx$$

Ааз решил устроить дискуссию по этой разработке на страницах журнала. Для этого статья была отправлена на рецензию известному специалисту по нестандартным компьютерам проф. А. Привалову.

Профессор Привалов усомнился в полезности подобного компьютера и предложил по заданным T отрезкам $[u_i, v_i]$ вычислить, для каких из них значение I_i может быть корректно вычислено с помощью новой разработки института.

Формат входного файла

В первой строке входного файла содержатся два целых числа n и T ($1 \leq n \leq 30, 1 \leq T \leq 10^5$), n — степень полинома. Вторая строка содержит $n + 1$ целых чисел a_k ($|a_k| \leq 10^9$) — коэффициенты полинома. Каждая из последующих T строк содержит два целых числа u_i и v_i ($|u_i| \leq 10^9, |v_i| \leq 10^9$) — границы i -ого отрезка.

Формат выходного файла

Для каждого отрезка ваша программа должна вывести **Yes**, если I_i — целое, и **No** в обратном случае.

Примеры

<code>integral.in</code>	<code>integral.out</code>
2 2	No
0 0 1	Yes
0 1	
0 3	

Задача G. Minimal Path

Имя входного файла:	<code>minimal.in</code>
Имя выходного файла:	<code>minimal.out</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Умение проходить сквозь стены
атрофировалось с развитием логики.

Полная История Лабиринтов

Институт лабиринтов (ИЛ) предоставил в журнал «МИФ» для публикации работу по строковому представлению лабиринтов.

Лабиринт представлен в виде двумерной карты, состоящая из $w \times h$ квадратных клеток. Каждая клетка может быть или свободна, или занята стеной; сам лабиринт считается окружённым стенами. Игрок делает *ход* в лабиринте из некоторой клетки, выбирая одно из четырёх направлений (вниз, влево, вправо или вверх) и двигаясь в соседнюю с данной клеткой в этом направлении. Обе клетки должны быть свободны для того, чтобы ход был возможен. *Путь* между двумя клетками — это конечная последовательность ходов, которая начинается в одной из клеток и заканчивается в другой, при этом разрешается проходить через одну клетку любое количество раз.

Сотрудники Института предложили ввести несколько дополнительных терминов. *Строковое представление* пути — это строка, составленная из строчных латинских букв ‘d’, ‘l’, ‘r’ и ‘u’, описывающая путь. Каждой буква обозначает ход с текущей клетки в соответствующем направлении. Здесь ‘d’ обозначает ход вниз (увеличение номера строки на 1), ‘l’ обозначает ход влево (уменьшение номера столбца на 1), ‘r’ обозначает ход вправо (увеличение номера столбца на 1) и ‘u’ обозначает ход вверх (увеличение номера строки на 1). *Минимальный путь* между двумя клетками — это путь, имеющий лексикографически наименьшее строковое представление.

Требуется написать программу, которая будет вычислять строковое представление минимального пути для заданного лабиринта.

Формат входного файла

Первая строка входного файла содержит два целых числа w и h , разделённых пробелом ($1 \leq w, h \leq 10$). Вторая строка содержит 4 целых числа c_s, r_s, c_f и r_f , разделённых пробелами. Здесь, (c_s, r_s) — координаты начальной клетки пути и (c_f, r_f) — координаты конечной клетки пути ($1 \leq c_s, c_f \leq w, 1 \leq r_s, r_f \leq h$). Последующие h строк содержат w символов каждая; s -й символ на r -й строке является ‘.’ (точкой), если клетка (c, r) свободна, или ‘X’ (заглавная латинская буква X), если она занята стеной. Начальная и конечная клетки различны и обе являются свободными.

Формат выходного файла

Если минимального пути между двумя заданными клетками не существует, выведите “None.” в первой строке выходного файла. В ином случае выведите строковое представление минимального пути.

Примеры

minimal.in	minimal.out
2 2 2 1 1 2	d1
5 1 1 1 5 1 ..X..	None.
3 1 2 1 3 1 ...	None.

Задача H. Cutting a Pie

Имя входного файла: pie.in
Имя выходного файла: pie.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Разделяй и властвуй.

Основной закон биологии одноклеточных

- В серьёзном научном журнале должны быть статьи, посвящённые образованию, заявил Ааз.
- Но у нас нету ни одной статьи на эту тему...
- А если посмотреть внимательнее?

Сотрудники Института Воспитательной Аналитики (ИВА) в своей статье рассматривали следующую ситуацию.

N детей хотят разделить один пирог. У каждого из детей есть своё мнение, какой кусок пирога является «хорошим». Это значит, что для каждого ребёнка определены два рациональных числа a и b , такие, что он считает «хорошим» кусок пирога размером s , если $a \leq s \leq b$.

Задача — разделить пирог между детьми так, чтобы каждому из них достался «хороший», по его мнению, кусок пирога.

Пирог делится следующим образом:

1. Присваиваем детям номера от 1 до N .
2. Для всех i определяем целое число $k_i (1 \leq k_i \leq K)$ для ребёнка с номером i .
3. Для всех i даём ребёнку с номером i долю пирога размера $\frac{k_i}{\sum_{i=1}^N k_i}$

Вам предложено написать программу, решающую эту задачу — тогда из этой ситуации вполне можно сделать статью в журнал.

Формат входного файла

Первая строка входного файла содержит два целых числа $N (1 \leq N \leq 1000)$ и $K (1 \leq K \leq 200)$. Далее следуют N строк. $(i + 1)$ -я строка входного файла содержит два рациональных числа a_i и b_i , разделённых ровно одним пробелом. Каждое из этих чисел задано как отношение двух целых чисел, лежащих между 0 и 10^9 для числителя и 1 и 10^9 для знаменателя.

Формат выходного файла

Если не существует требуемого способа деления пирога, выведите строку «No solution». Если решение существует, выведите единственную строку, содержащую N целых чисел k_i , в соответствии с которыми будет производиться деление. Если решений более одного, выведите решение с минимальной $\sum_{i=1}^N k_i$. Если и таких решений несколько, сравните векторы k_i лексикографически и выведите наименьший из них.

Примеры

pie.in	pie.out
1 2 0/1 1/2	No solution
3 2 0/1 1/1 0/1 1/1 0/1 1/4	1 2 1

Задача I. Pisces (Division 1 Only!)

Имя входного файла: `pisces.in`
Имя выходного файла: `pisces.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

It's not easy to find a black fish in a black room.
Especially if a Black Cat came there first.

man cat

Очередная статья была посвящена особенностям индивидуального поведения аквариумных рыбок. Впрочем, у автора статьи были определённые проблемы — например, для него все рыбки в аквариуме выглядели одинаково и он не мог отличить их одну от другой иначе, как по цвету. В других журналах после такого комментария автора отправляли восвояси. Но Ааз, заметивший, что работа поддержана грантом RFFI (Research Foundation of Fish Individuality), предложил исправить положение. Он предложил пометить рыбок цветными колечками на плавниках.

Следующий алгоритм должен повторяться до тех пор, пока любую пару рыбок будет возможно различить между собой:

- выбирается цвет, который до сих пор не был использован,
- выбираются k случайных рыбок из аквариума (каждое множество мощности k имеет равные шансы быть выбранным),
- каждая из выбранных рыбок помечается кольцом выбранного цвета.

Каждая процедура (три шага) занимает ровно одну минуту.

Автор статьи может различить двух рыбок, если их изначальный цвет различен или если множества цветов колец на их плавниках различны.

Напишите программу, которая вычислит ожидаемое время, которое потребуется для того, чтобы сделать всех рыбок попарно различимыми.

Формат входного файла

Входной файл состоит из двух строк. Первая из них содержит два целых числа n и k ($1 \leq k < n \leq 30$). Вторая строка содержит n целых чисел. i -е число описывает первоначальный цвет i -й рыбки.

Формат выходного файла

Выведите одно целое число — ожидаемое время (в минутах), которое потребуется, чтобы все рыбки были попарно различимы. Ответ считается правильным, если абсолютная или относительная ошибка не превышает 10^{-6} .

Примеры

<code>pisces.in</code>	<code>pisces.out</code>
2 1 1 1	1.000000
2 1 30 239	0.000000

Задача J. Screening a String

Имя входного файла: `screening.in`
Имя выходного файла: `screening.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Краткость — сестра таланта.

Прокруст

Замысел Ааза удался — дискуссия о результатах, достигнутых в Институте Шестнадцатеричной Архитектуры Компьютеров (ИШАК), вызвали большой интерес читателей.

Так что очередная статья, описывающая последние результаты сотрудников этого института — алгоритм сжатия строки перед её выводом на экран — была поставлена в ближайший номер.

Сжатие строки S происходит следующим образом. Из двух первых символов выбирается только один и выводится в результирующую строку первым. Аналогично из третьего и четвёртого символа выбирается только один и выводится следующим, и так далее. В случае, если длина строки нечётна, последний символ строки выводится всегда, в противном же случае процесс завершается выбором одного из $N - 1$ -го и N -го символов.

Например, из строки `abacaba` можно в результате сжатия получить `aaaa` или `baba`, однако получить из неё в результате сжатия, например, `baca` или `acb` невозможно.

Для проверки утверждений, указанных в статье, Вам поручено по заданной строке вычислить, какое минимальное количество последовательных операций сжатия надо применить к ней для того, чтобы получить строку, составленную из одинаковых символов.

Формат входного файла

Единственная строка входного файла содержит одну строку, состоящую из строчных латинских букв и содержащую не более 5×10^5 символов.

Формат выходного файла

Выведите одно число — необходимое количество сжатий.

Примеры

<code>screening.in</code>	<code>screening.out</code>
<code>abacaba</code>	1
<code>littlekitten</code>	3

Задача К. Tree

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

You're thinking in Japanese aren't you? If you *must* think, do it in German!

Neon Genesis Evangelion

В редакцию журнала «МИФ» поступила статья ведущего научного сотрудника Института Общей Вивисекции (ИОВ), профессора А.А. Выбегалло. Статья была посвящена внедрению архитектуры локальных сетей и идей Neon Genesis в мичуринскую биологию.

Несмотря на малую научную ценность статьи, Ааз поручил редактору опубликовать хотя бы краткую заметку по теме работы Выбегалло в разделе «История науки» как пример эволюции взглядов современных сторонников академика Лысенко. Но для начала требовалось разобраться в терминологии.

Автор называет *сетью* множество деревьев, соединённых несколькими «биопроводами». Каждый биопровод соединяет ровно два различных дерева.

Подсеть — это подмножество биопроводов. Оно называется «стабильным», если

- от каждого дерева можно добраться до другого, используя один или несколько биопроводов и
- оно не содержит никакого другого подмножества с аналогичным свойством.

Время от времени профессору требуется сравнить две подсети лексикографически. Для этого он записывает номера составляющих подсеть биопроводов в возрастающем порядке и сравнивает лексикографически полученные массивы.

Напоминаем, что массив a_1, a_2, \dots, a_p является *лексикографически меньшим*, чем массив b_1, b_2, \dots, b_p , если для наименьшего i , для которого $a_i \neq b_i$, верно, что $a_i < b_i$.

В работе часто упоминается подсеть с наименьшей возможной суммарной длиной биопроводов, которая была бы лексикографически наименьшей из всех таких подсетей. Редактор поручил вам явно построить такую подсеть.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($2 \leq n \leq 10^5$, $m \leq 10^5$) — количество деревьев и биопроводов между ними соответственно. В каждой из следующих m строк содержатся три целых числа — номера деревьев, соединённых соответствующим биопроводом, а затем — длина провода. Деревья занумерованы, начиная с единицы. Длина каждого биопровода неотрицательна и не превосходит 10^5 .

Формат выходного файла

В выходной файл выведите $n - 1$ число — номера биопроводов в лексикографически наименьшей стабильной подсети. Гарантируется, что такая подсеть существует.

Примеры

<code>tree.in</code>	<code>tree.out</code>
4 4 1 2 1 2 3 1 3 4 1 4 1 1	1 2 3

Задача L. Funny Game (Division 2 Only!)

Имя входного файла: `funnygame.in`
Имя выходного файла: `funnygame.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes

Деньги — на «бочку»!

Лозунг общества добровольной помощи
авиации

— Наш журнал не должен отрываться от практики, — заметил Ааз на очередной редколлегии.
— В чём же отрыв от практики? — спросил Гвидо.
— Например, мало места уделяется стратегиям игр, в которых тем или иным способом фигурируют деньги, — уточнил свою мысль практичный Ааз, которому уже давно не терпелось проверить очередную авантюру.

Институт рискованных игровых стратегий (ИРИС) по заказу журнала «МИФ» исследовал игру для двух игроков со следующими правилами.

Игра ведётся на доске размером $m \times m$ квадратных клеток, при этом в клетке с координатами (x, y) размещена монета. Каждый игрок в свою очередь двигает монету в соседнюю клетку (две клетки считаются соседними, если у них есть общая сторона), причём двигать монету на клетку, на которой она уже была, нельзя, равно как и выходить за край доски.

Игрок, который не может сделать ход, проигрывает.

Вам поручено определить, кто выиграет при оптимальной игре обоих участников.

Формат входного файла

В первой строке входного файла содержится число d ($0 \leq d \leq 400$) — количество тестовых примеров.

Каждая из последующих N строк содержит 3 числа: m_i, x_i, y_i ($1 \leq m_i \leq 500, 1 \leq x_i, y_i \leq m_i$).

Формат выходного файла

В выходном файле для каждого тестового примера выведите “**First**”, если выигрывает первый игрок, или “**Second**” — если второй.

Пример

funnygame.in	funnygame.out
2	Second
1 1 1	First
2 2 2	

Задача M. Sacramento Sum (Division 2 Only!)

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes

Зри в корень!

К. Прутков, Теория и практика решения уравнений

В журнал «МИФ» пришла статья из Института Приближений Радикалов (ИПР) о приближении квадратных корней целыми числами.

Сотрудники института определили a_n как целое число, ближайшее к квадратному корню из n .

Ваша задача — найти сумму $S(m) = \sum_{i=1}^m \frac{1}{a_i}$ для заданного M .

Формат входного файла

В первой строке входного файла содержится число d ($0 \leq d \leq 10^5$) — количество тестовых примеров.

Каждая из последующих d строк содержит число m_i ($1 \leq m_i \leq 10^{15}$).

Формат выходного файла

Для каждого тестового примера выведите в отдельной строке значение $S(m_i)$ с точностью до 10^{-4} .

Пример

<code>sum.in</code>	<code>sum.out</code>
2	1.0000
1	2.0000
2	

Задача N. Sophie Germain Primes (Division 2 Only!)

Имя входного файла: `sgprime.in`
Имя выходного файла: `sgprime.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes

Эталон простого числа — выполненая из
специального сплава пареная репа

из экскурсии по Севру

Институту Математической Простоты (ИМП) для продолжения работы над обещанной редколлегии журнала «МИФ» статьёй требуется программа, которая выдаёт список всех простых чисел Софи Жермен в заданном интервале.

Простое число — целое неотрицательное число, которое делится только само на себя и на 1.

Простое число Софи Жермен — простое число p , такое, что число $2p + 1$ — также простое.

Формат входного файла

В первой строке входного файла содержится число d ($0 \leq d \leq 20$) — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки, содержащей два целых числа L и H ($0 \leq L \leq H \leq 10^5$) — начало и конец интервала, соответственно.

Формат выходного файла

Для каждого тестового примера в одной строке выведите значения L и H , разделённые пробелом, после чего выведите двоеточие и пробел. Затем выведите через пробел все простые числа Софи Жермен p_i в порядке возрастания, $L \leq p_i \leq H$.

Пример

<code>sgprime.in</code>	<code>sgprime.out</code>
2	0 4: 2 3
0 4	100 130: 113
100 130	