

## Задача А. Бактерии

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

У юного биолога Антона в красивой стеклянной колбе живут  $n$  бактерий.

Добавляя различные реактивы в колбу, Антон может контролировать число бактерий. Так, если  $p$  — некоторое простое число, то Антон умеет в домашних условиях получать вещество  $C_pH_{2p+1}OH$ , которое, будучи добавленным в колбу, уменьшает число бактерий ровно в  $p$  раз. Если же число бактерий не делилось на  $p$ , то результат действия вещества неопределен, и эксперимент теряет научную точность. Этого Антон допустить не желает, поэтому он применяет вещество  $C_pH_{2p+1}OH$  только когда число бактерий делится на  $p$ .

Кроме того, у Антона на кухне есть неограниченный запас диэтиламида лизергиновой кислоты ( $C_{20}H_{25}N_3O$ ). При добавлении в колбу с бактериями диэтиламида лизергиновой кислоты, число бактерий возводится в квадрат.

Антон хочет, чтобы в колбе стало  $m$  бактерий. При этом он хочет добавлять какие-либо вещества в колбу наименьшее возможное число раз. Помогите ему сделать это.

### Формат входного файла

Во входном файле содержатся два натуральных числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^9$ ) — изначальное и желаемое число бактерий в колбе у Антона.

### Формат выходного файла

Если получить ровно  $m$  бактерий невозможно, выведите в выходной файл слово «Impossible».

Если же искомый результат достижим, выведите кратчайшую последовательность добавлений веществ, которая позволяет его достичь, в следующем формате: добавление вещества  $C_pH_{2p+1}OH$  кодируется числом  $p$ , добавление вещества  $C_{20}H_{25}N_3O$  — числом 0. Числа должны быть разделены пробелами и/или переводами строк.

Если существует несколько кратчайших последовательностей добавлений веществ, оставляющих  $m$  бактерий, выведите любую из них.

### Примеры

<code>bacteria.in</code>	<code>bacteria.out</code>
12 18	2 0 2
56 6	Impossible

В первом примере Антону требуется добавлять в колбу вещества три раза: сначала добавить  $C_2H_5OH$ , уменьшая число бактерий в два раза, то есть оставляя 6 бактерий; затем добавить  $C_{20}H_{25}N_3O$ , возводя число бактерий в квадрат, то есть увеличивая его до 36; и наконец, добавить снова  $C_2H_5OH$ , деля число бактерий на два и делая его равным 18.

## Задача В. Шахматная головоломка

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

Борис очень любит различные шахматные головоломки. У него есть младший брат Вова. Борис очень любит задавать простые головоломки Вове, а в награду, если тот их решит, давать ему конфету. Но Вова, к сожалению, не очень любит шахматы, зато любит программирование.

В этот раз Борис задал Вове следующую головоломку: на шахматном поле размером  $8 \times 8$  клеток стоит одна шахматная фигура — конь. Необходимо расположить на поле еще две шахматные фигуры — ладью и слона, таким образом, чтобы они били коня, но не били друг друга, и конь не бил их. Так как Вова еще не очень силен в программировании, он попросил вас помочь ему с решением данной головоломки.

Напомним, что конь бьет те клетки, которые отстоят от его текущего положения на две клетки по горизонтали и одну клетку по вертикали, или на две клетки по вертикали и одну по горизонтали.

Ладья бьет те клетки, которые находятся на той же горизонтали или вертикали, что и она. Слон бьет те клетки, которые находятся на той же диагонали, что и он.

### Формат входного файла

Первая строка входного файла содержит положение коня в следующем формате. Сначала буква от «a» до «h», обозначающая номер столбца в котором находится конь, потом цифра от «1» до «8», обозначающая номер строки.

### Формат выходного файла

В первую строку выведите положение ладьи в аналогичном формате, во вторую строку выведите положение слона.

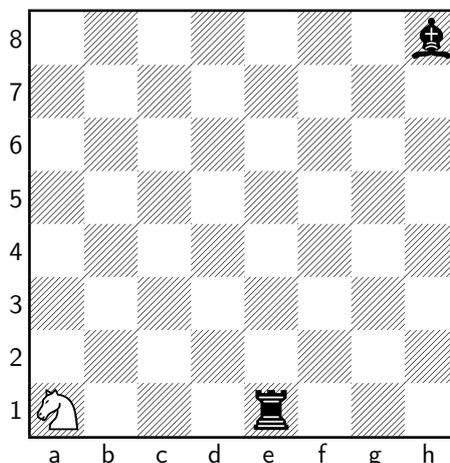
Гарантируется, что требуемая расстановка всегда существует.

### Примеры

<code>chess.in</code>	<code>chess.out</code>
a1	e1 h8

### Примечание

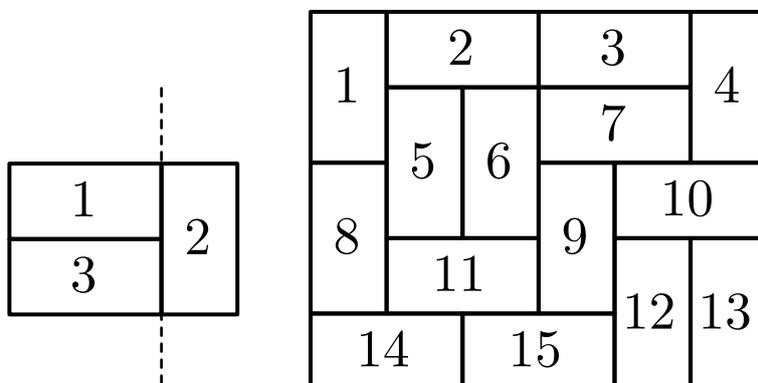
Примеру соответствует следующее расположение фигур:



## Задача С. Шоколад

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

Петя очень любит шоколад. И Маша очень любит шоколад. Недавно Петя купил шоколадку и теперь хочет поделиться ею с Машей. Шоколадка представляет собой прямоугольник  $n \times m$ , который полностью состоит из маленьких шоколадных долек — прямоугольников  $2 \times 1$ .



Петя делит шоколадку на две части, разламывая ее вдоль некоторой прямой, параллельной одному из краев шоколадки. Ни Петя, ни Маша не любят ломаные дольки, поэтому Петя хочет разломать шоколадку так, чтобы ни одна долька не была повреждена.

Помогите Пете поделиться шоколадкой с Машей.

### Формат входного файла

В первой строке входного файла два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 20$ ; хотя бы одно из чисел  $n$  и  $m$  — четно). Далее следуют  $n$  строк по  $m$  чисел в каждой — номера долек, в которые входят соответствующие кусочки шоколадки. Дольки имеют номера от 1 до  $\frac{n \cdot m}{2}$ , и никакие две дольки не имеют одинаковые номера.

### Формат выходного файла

В выходной файл выведите «Yes», если Петя может разломать шоколадку, не повредив дольки. Иначе выведите «No».

### Примеры

<code>chocolate.in</code>	<code>chocolate.out</code>
2 3 1 1 2 3 3 2	Yes
5 6 1 2 2 3 3 4 1 5 6 7 7 4 8 5 6 9 10 10 8 11 11 9 12 13 14 14 15 15 12 13	No

## Задача D. Луна

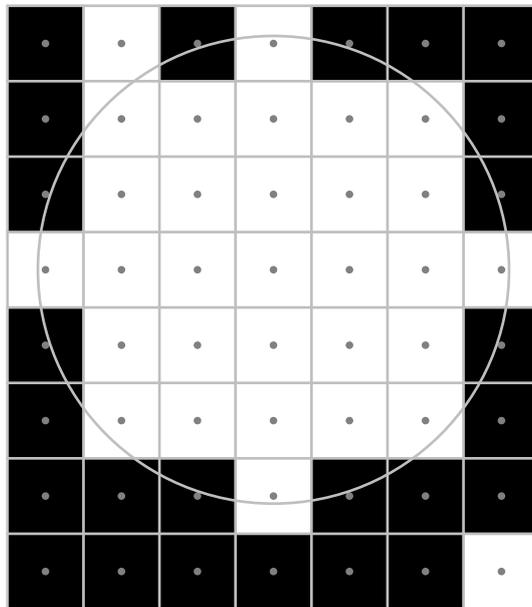
Имя входного файла: moon.in  
Имя выходного файла: moon.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Начинающий астроном Даша наконец-то обзавелась цифровым фотоаппаратом. Конечно, фиксировать звездное небо нажатием кнопки, а уже затем производить исследования — гораздо удобней.

Разрешающая способность матрицы фотоаппарата оказалась не слишком высокой, да и на фотографиях ночного неба можно различить только два цвета: черный и белый. Впрочем, в этом есть и свои плюсы: Даша сделала уже огромное число снимков, а память все еще не закончилась. Теперь Дашу интересует положение Луны на каждом из снимков.

Будем считать, что Луна на снимке выглядит как круг с центром в точке изображения  $C$  и целым неотрицательным радиусом  $r$ , то есть как множество белых точек, расстояние от центров которых до точки  $C$  не больше  $r$ . Луна полностью поместилась на снимке.

Также некоторые достаточно яркие звезды могут присутствовать на снимке в виде отдельных белых точек. Таких точек не больше 25. Объектов, отличных от Луны и звезд, на снимке не изображено.



Напишите программу, которая по изображению найдет наибольший возможный радиус круга, который соответствует Луне, а также ее возможное положение.

### Формат входного файла

В первой строке ввода записаны целые числа  $w$  и  $h$  — горизонтальное и вертикальное разрешение снимка, соответственно ( $1 \leq w, h \leq 50$ ). В следующих  $h$  строках записано по  $w$  символов «.» (черная точка) или «\*» (белая точка).

### Формат выходного файла

В первой строке выведите натуральное число — максимальный радиус изображения Луны. Во второй строке выведите координаты (столбец, затем строку) центра изображения Луны с данным радиусом. Столбцы и строки нумеруются с единицы, слева направо и сверху вниз, соответственно.

Если центров может быть несколько, выведите любой. Гарантируется, что корректный ответ существует.

## Примеры

moon.in	moon.out
7 8 .*.*... .*****. .*****. ***** .*****. .*****. ...*... .....*	3 4 4
5 4 ..... ..... ..*.. .*...	0 2 4

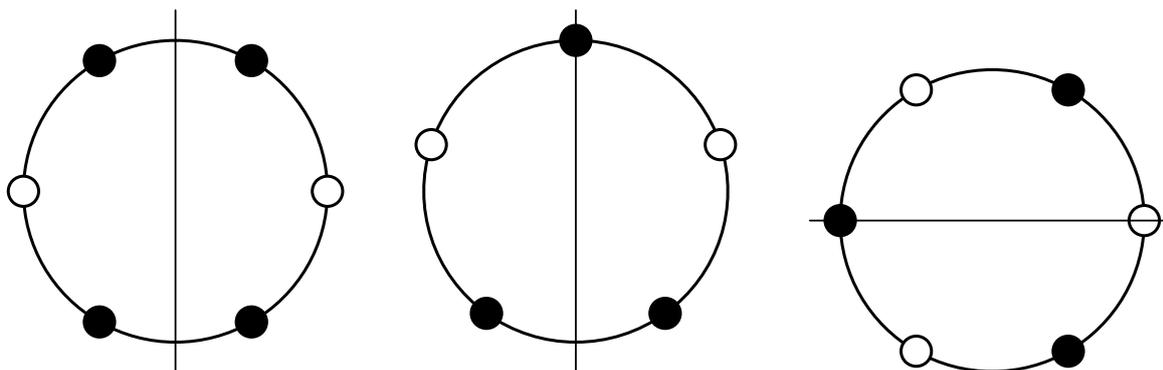
## Задача Е. Ожерелье

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

Юная любительница ювелирных изделий Октябрина к празднику 4-го ноября хочет подарить своей лучшей подруге Тракторине ожерелье из  $n$  черных и розовых жемчужин.

Чтобы ожерелье не было скучным, Октябрина хочет расположить жемчужины таким образом, чтобы как его ни повернуть, левая половина ожерелья не была симметрична правой. Более формально — у него не должно быть оси симметрии.

Ось симметрии делит ожерелье на непрерывные части, содержащие одинаковое число жемчужин. При этом, если ось проходит через какую-то жемчужину, то она относится к обеим частям, если же ось проходит между двух жемчужин, то эти жемчужины находятся в разных частях. Таким образом, следующие ожерелья имеют ось симметрии:



Ваша задача — помочь Октябрине найти необходимую расстановку жемчужин.

### Формат входного файла

Первая и единственная строка входного файла содержит единственное число  $n$  ( $2 \leq n \leq 1000$ ) — требуемое количество жемчужин в ожерелье.

### Формат выходного файла

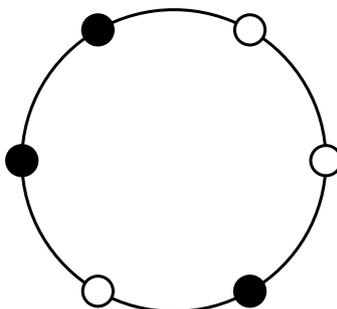
Если искомой расстановки не существует, выведите в выходной файл единственное число  $-1$ , иначе выведите  $n$  целых чисел — расстановку жемчужин. Розовой жемчужине соответствует число 0, черной — число 1.

### Примеры

<code>necklace.in</code>	<code>necklace.out</code>
6	0 0 1 0 1 1
3	-1

### Примечание

Расстановка жемчужин в первом примере



## Задача F. Гонки

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

Недавно руководством одной известной автомобильной телепередачи «Верхняя шестерня» было решено провести обзор автомобилей на солнечных батареях. Для этого были выбраны две модели.

К сожалению, современные технологии еще далеки от совершенства, поэтому автомобили не могут ехать непрерывно. Руководство по эксплуатации первого автомобиля гласит, что при передвижении на большие дистанции нужно действовать следующим образом: в течение  $t_1$  часов ехать со скоростью  $v_1$  километров в час, после чего такое же время заряжать батареи. Стратегия по оптимальному использованию второго автомобиля аналогична, но числа  $t_2$  и  $v_2$  для него могут отличаться.

Для демонстрации работы автомобилей было решено устроить соревнование — заезд по прямой трассе длиной  $X$  километров, придерживаясь стратегии из руководства.

Вам поручено предсказать результат этого заезда.

### Формат входного файла

Первая строка входного файла содержит целые числа  $t_1$ ,  $v_1$ ,  $t_2$  и  $v_2$ , разделенные пробелами ( $1 \leq t_i, v_i \leq 1000$ ). Вторая строка содержит одно целое число  $x$  — длину трассы ( $1 \leq x \leq 1000\,000$ ).

### Формат выходного файла

Если первый автомобиль финиширует первым, выведите «**First**». Если второй автомобиль окажется на финише раньше, выведите «**Second**». Если же обе машины преодолеют трассу за одинаковое время, выведите «**Draw**».

### Примеры

<code>race.in</code>	<code>race.out</code>
1 1 2 1 4	Second

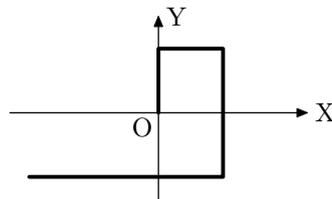
## Задача G. Робот

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

Робот-марсоход «ТцТцПетя» двигается по поверхности Марса как ему вздумается, отправляя на Землю информацию о своих передвижениях.

«ТцТцПетя» пользуется следующей системой координат: начало координат совпадает с его начальным положением, ось  $OY$  направлена в сторону, в которую он направлен в начальный момент времени (при высадке на Марс).

Передвигается «ТцТцПетя» следующим образом: после высадки на Марс он проезжает вперед какое-то целое число сантиметров, от 1 до  $10^6$ ; затем поворачивает на 90 градусов либо налево, либо направо; после чего снова проезжает вперед от 1 до  $10^6$  сантиметров; и снова поворачивает на 90 градусов либо налево, либо направо; и так далее. Наконец, проехав последний отрезок (также длиной от 1 до  $10^6$  сантиметров), он останавливается и начинает передавать на Землю описание своего маршрута.



В итоге Центр Управления получил от «ТцТцПети» следующее сообщение: «Я сделал  $n$  передвижений. Сообщаю  $n - 1$  поворот, который я совершил: *последовательность поворотов*. В итоге я оказался в точке с координатами  $(x, y)$ . Мне тут нравится. Конец связи.»

И тут-то создатели «ТцТцПети» поняли, что забыли запрограммировать его, чтобы он сообщал длины своих передвижений!

Теперь их интересует хоть какой-нибудь вариант пути «ТцТцПети», который удовлетворяет полученным от него данным. Помогите им.

### Формат входного файла

В первой строке входного файла содержатся три целых числа  $x, y, n$  ( $-100\,000 \leq x, y \leq 100\,000$ ;  $1 \leq n \leq 100\,000$ ) — конечные координаты «ТцТцПети» и количество передвижений, которые он совершил.

Вторая строка имеет длину  $n - 1$  и состоит из символов «L» и «R» — это последовательность поворотов, которые совершил «ТцТцПетя». Символ «L» обозначает поворот налево на 90 градусов, символ «R» — направо на 90 градусов.

### Формат выходного файла

Если информация противоречива и двигаться подобным образом робот не мог, выведите в выходной файл слово «Impossible».

В противном случае выведите  $n$  целых чисел от 1 до  $10^6$  — длины передвижений «ТцТцПети» в сантиметрах, такие что с учетом указанных им поворотов, «ТцТцПетя» заканчивает движение в точке  $(x, y)$ . Числа должны быть разделены пробелами и/или переводами строк.

### Примеры

<code>robot.in</code>	<code>robot.out</code>
-2 -1 4 RRR	1 1 2 3
4 1 5 LRRL	Impossible
0 10 1	10

Примечание: поверхность Марса в рамках данной задачи считается плоской.

## Задача Н. Санта Клаус

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

Санта Клаус готовится к Рождеству. В этот праздник он хочет вручить подарки  $n$  детям. Его помощники Эльфы уже собрали два мешка, с которыми он отправится в новогоднее путешествие по всем странам мира. И чтобы Санта не запутался, Эльфы составили список детей, чьи подарки уже лежат в каждом из мешков. Санта хочет помочь Эльфам, и поэтому решил положить в третий мешок подарки для тех детей, которым они еще не подготовлены.

Помогите Санте, составьте список детей, чьи подарки надо положить в третий мешок.

### Формат входного файла

Первая строка входного файла содержит три целых числа:  $n$  — число детей,  $m$  и  $k$  — число подарков в первом и втором мешке соответственно ( $1 \leq n, m, k \leq 100$ ;  $m + k \leq n$ ). Вторая строка входного файла содержит  $m$  целых чисел — номера детей, подарки для которых лежат в первом мешке. Третья строка входного файла содержит  $k$  целых чисел — номера детей, подарки для которых лежат во втором мешке.

Гарантируется что Эльфы положили для каждого ребенка не более одного подарка. Номера всех детей являются целыми положительными числами не превосходящими  $n$ . Все дети должны получить подарок на Рождество, иначе Санта расстроится.

### Формат выходного файла

В первой строке выведите одно число  $a$  — сколько подарков должно быть в третьем мешке. Во второй строке выведите в произвольном порядке  $a$  чисел — номера детей, которым эти подарки должны быть доставлены.

### Примеры

<code>santa.in</code>	<code>santa.out</code>
7 2 1	4
7 3	2 6 4 5
1	

## Задача I. Подстрока

Имя входного файла: `substring.in`  
Имя выходного файла: `substring.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Недавно разведка перехватила зашифрованное сообщение — строку  $s$ . Все ресурсы аналитического центра, в котором вы работаете, были брошены на его декодирование.

Ваш отдел занимается шифрами нового поколения. На данный момент известно всего  $n$  таких шифров. Для каждого из них есть три характерных параметра — целые числа  $l$ ,  $r$  и строка  $t$ . Пусть строка  $g$  была получена в результате применения этого метода. Тогда строка  $g_l g_{l+1} \dots g_{r-1} g_r$  (здесь  $g_i$  — это  $i$ -й символ строки  $g$ ) содержит  $t$  как подстроку.

Вам поручено определить для каждого типа шифрования, могло ли сообщение  $s$  быть получено в результате его применения.

### Формат входного файла

Первая строка входного файла содержит строку  $s$  ( $1 \leq |s| \leq 100\,000$ , где  $|s|$  — длина строки  $s$ ).

Вторая строка входного файла содержит целое число  $n$  — количество типов шифрования ( $1 \leq n \leq 100\,000$ ). Последующие  $n$  строк содержат по два целых числа  $l_i$ ,  $r_i$  и строку  $t_i$ , разделенные пробелами — характерные параметры  $i$ -го метода шифрования ( $1 \leq l_i \leq r_i \leq |s|$ ).

Все строки состоят из строчных букв латинского алфавита. Суммарная длина всех  $t_i$  не превосходит 100 000.

### Формат выходного файла

Выведите одну строку — для каждого типа шифрования «+», если сообщение  $s$  могло быть получено в результате его применения, или «-» в противном случае.

### Примеры

<code>substring.in</code>	<code>substring.out</code>
<code>frommarsiam</code>	<code>++-</code>
<code>3</code>	
<code>6 10 i</code>	
<code>2 11 am</code>	
<code>1 9 human</code>	

## Задача J. Вода

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

Кто возьмет билетов пачку,  
Тот получит водокачку!

---

из к/ф «Бриллиантовая рука»

Виталий является директором небольшой водопроводной компании. Он работает на водокачке. Водокачка представляет собой цилиндр высотой  $H$  метров, изначально целиком заполненный водой.

Водокачка очень старая, поэтому в ней постоянно появляются дыры. Периодически Виталий устраняет некоторые течи. Виталия как директора очень интересует уровень воды в водонапорной башне, но каждый раз лезть по лестнице, чтобы заглянуть в башню, ему не хочется.

Напишите программу, которая могла бы обрабатывать следующие запросы:

1. В момент времени  $t$  появилась течь на высоте  $h$  со скоростью вытекания воды  $v$ .
2. В момент времени  $t$  Виталий устранил все течи на высоте  $h$ .
3. В момент времени  $t$  Виталий хотел бы узнать уровень воды.

Вода вытекает из отверстия, только если уровень воды выше высоты отверстия. Изначально водокачка полна. Скорость вытекания воды  $v$  означает, что за единицу времени через отверстие вытекает объем воды, соответствующий  $v$  метрам воды в водокачке.

### Формат входного файла

В первой строке входного содержится два целых числа  $n$  и  $H$  ( $1 \leq n \leq 10^5$ ;  $1 \leq H \leq 10^9$ ) — количество запросов и высота водонапорной башни. Далее следуют  $n$  строк описывающих запросы. В каждой строке первое число  $k$  ( $1 \leq k \leq 3$ ) — номер типа запроса. Далее в строке в зависимости от запроса идут от одного до трех целых чисел из  $t$ ,  $h$  и  $v$  ( $0 \leq t \leq 10^9$ ;  $0 \leq h \leq H$ ;  $1 \leq v \leq 10^9$ ). Запросы перечислены в порядке увеличения времени и никакие два запроса не имеют одинаковое время.

### Формат выходного файла

На каждый запрос третьего типа выведите одно число — уровень воды в водонапорной башне в соответствующий момент времени.

Ответ должен иметь относительную или абсолютную точность  $10^{-9}$ . Это означает, что если правильный ответ равен  $a$ , а выведено  $p$ , то значение будет признано правильным, если  $|a - p| / \max(a, 1) \leq 10^{-9}$ .

### Примеры

<code>water.in</code>	<code>water.out</code>
6 10	8.0
1 0 1 2	6.0
3 1	4.0
3 2	4.0
2 3 1	
3 4	
3 5	