

Задача А. Лучший подарок — массив!

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

Завтра у Дим Димыча день рождения, и Симка решила порадовать его подарком. Но под рукой ничего не оказалось, поэтому она решила подарить ему массив натуральных чисел. Чтобы Дим Димыч не обнаружил подарок раньше времени, Симка спрятала массив в системном блоке его компьютера. Однако как всегда прибежал Нолик и все испортил. Он перемешал все числа в массиве, и теперь он выглядит ужасно, на день рождения дарить его нельзя. К счастью, Нолик запомнил исходный массив и готов восстановить его, если Симка решит для него одну задачу. А именно, он хочет, чтобы она по данному набору чисел p нашла подотрезок q такой же длины в массиве-подарке, что p и q «практически совпадают». Два набора чисел практически совпадают, если в них можно переставить числа таким образом, чтобы они совпали точно. Например, наборы чисел $\{1, 3, 2, 1\}$ и $\{2, 3, 1, 1\}$ практически совпадают, а $\{1, 2, 3\}$ и $\{2, 3, 4\}$ — нет.

Нолик написал на бумажке набор чисел p и теперь просит вас решить придуманную им задачу. Помогите Симке восстановить подарок Дим Димыча на день рождения!

Формат входных данных

В первой строке входного файла содержится единственное число n ($1 \leq n \leq 100\,000$) — количество чисел в наборе p .

Во второй строке содержится n целых чисел p_i ($1 \leq p_i \leq 100\,000$) — набор чисел p .

В третьей строке дано число m ($n \leq m \leq 100\,000$) — длина массива-подарка.

В четвертой строке содержится m чисел a_i ($1 \leq a_i \leq 100\,000$) — числа из массива.

Формат выходных данных

В первой строке выходного файла выведите «YES», если в массиве-подарке существует подотрезок q , удовлетворяющий условиям Нолика. В противном случае выведите «NO». В случае положительного ответа на второй строке также выведите позицию начала подотрезка. Если ответов несколько, можно вывести любой.

Примеры

<code>anagrams.in</code>	<code>anagrams.out</code>
3 2 3 4 4 1 4 2 3	YES 2
3 1 2 3 4 2 3 4 5	NO

Задача В. Игрек и скобочное дерево

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

Все знают, что Игрек самый умный в фиксик в школе. Он, как и мы с вами, любит решать сложные задачи. Недавно он заинтересовался деревьями, а конкретно *скобочными деревьями*, но они не так просты, как ему показалось сначала. Вот одна из задачек, которые он не смог решить.

Скобочное дерево — это такое подвешенное дерево, в каждой вершине которого записана открывающая или закрывающая скобка, дети каждой вершины упорядочены, и каждый ребенок является скобочным деревом.

Обозначим корень скобочного дерева T как $root_T$. За a_v обозначим скобку, которая записана в вершине v .

Введем также отношение равенства двух скобочных деревьев: два скобочных дерева A и B являются *равными*, если скобки, записанные в $root_A$ и $root_B$ равны, количество детей $root_A$ и $root_B$ одинаково и, наконец, i -й ребенок $root_A$ равен i -му ребенку $root_B$.

Введем рекурсивную функцию $f(v)$ от вершины скобочного дерева, которая будет вычисляться следующим образом:

$$f(v) = \begin{cases} a_v, & \text{если } v \text{ лист} \\ f(c_1) + f(c_2) + \dots + f(c_k) + a_v, & \text{где } c_i \text{ — дети вершины } v \text{ в упорядоченном порядке} \end{cases}$$

Знаком $+$ обозначена операция конкатенации. Результатом этой функции является строка из открывающих и закрывающих скобок.

Сколько существует различных *скобочных деревьев* T , таких что $f(root_T)$ является правильной скобочной последовательностью с n открывающими скобками?

А вы сможете помочь Игреку с этой задачей?

Формат входных данных

В первой строке дано натуральное число n ($1 \leq n \leq 10^5$) — количество открывающих скобок в скобочной последовательности, полученной в результате применения функции f .

Формат выходных данных

В единственной строке выведите ответ на задачу по модулю $10^9 + 7$.

Примеры

bracetreel.in	bracetreel.out
1	1
2	10
3	210

Задача С. Нолик и игра

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

Нолик обнаружил на телефоне Дим Димыча одну занимательную игру. Эта игра по задумке ее создателей должна тренировать реакцию и внимательность. Но для Дим Димыча она слишком сложна.

Игра состоит в следующем. На экране телефона расположены n цветных кубиков, которые стоят в один ряд. В некоторые моменты времени цвета некоторых кубиков меняются. А также, иногда игра задает вопрос играющему: она говорит ему некоторый отрезок $[l \dots r]$ ($1 \leq l \leq r \leq n$), а играющий должен ответить — какое максимальное количество различных цветов находится на некотором отрезке длины k , содержащемся в $[l \dots r]$. Число k генерируется в начале игры и остается постоянным на всем ее протяжении. Цель игры заключается в том, чтобы правильно ответить на все вопросы игры.

Вам предлагается побыть в роли программиста этой игры и посчитать правильные ответы на все вопросы.

Формат входных данных

В первой строке входного файла находятся три числа n, m, k ($1 \leq n \leq 10^5, 1 \leq m \leq 10^5, 1 \leq k \leq n$) — количество кубиков, расположенных в ряд, количество событий в игре и длина отрезка. В следующей строке находятся n чисел a_i ($1 \leq a_i \leq 10^6$) — цвета кубиков в начальный момент времени.

В следующих m строках находятся описания событий игры — по одному в строке:

- $1 \ i \ x$ — изменить цвет i -го ($1 \leq i \leq n$) кубика на цвет x ($1 \leq x \leq 10^6$);
- $2 \ l \ r$ — ответить на вопрос: какое максимальное количество различных цветов находится на некотором отрезке длины k , содержащемся в $[l \dots r]$, $r - l + 1 \geq k$.

Формат выходных данных

Выведите ответы на вопросы игры по одному в строке в том порядке, в котором они задаются игрой.

Примеры

<code>different.in</code>	<code>different.out</code>
5 3 3	2
1 2 1 3 3	3
2 1 3	
1 3 3	
2 1 5	
6 6 3	3
2 3 1 1 1 9	1
2 1 3	2
2 3 5	3
1 3 3	
2 1 5	
1 2 1	
2 1 6	

Задача D. Сладкая жизнь

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

Несмотря на то, что фиксика питаются энергией, они так же очень любят сладости. И вот сегодня Папус, гуляя по квартире Дим Димыча вместе с Масей, нашел на столе упаковку конфет.

Недолго думая, они достали все конфеты и пересчитали их. Конфет оказалось ровно n штук. Оба фиксика хотят забрать себе как можно больше конфет, а поделить их пополам — слишком сложно. Поэтому они придумали другой способ поделить конфеты — игровой. Папус и Маса по очереди будут брать конфеты из кучки, но каждый раз не более k конфет, так как руки у маленьких фиксиков не очень большие. Также запрещается брать количество конфет, которое было взято соперником на предыдущем ходу, потому что повторение — это скучно.

Проигрывает тот, кто не может сделать ход. Папус, как истинный джентельмен, отдал право первого хода Маса. Выигравший фиксик оставляет за собой право поделить конфеты так, как он хочет. Помогите им, скажите, кто выиграет при правильной игре и будет делить все конфеты!

Формат входных данных

В первой и единственной строке входного файла даны два числа n, k ($1 \leq n \leq 10^9, 1 \leq k \leq 100$) — начальное количество конфет в упаковке и максимальное количество конфет, которое можно взять за один ход.

Формат выходных данных

В единственной строке выходного файла выведите «Masya», если при правильной игре выиграет Муса и «Papus» в противном случае.

Примеры

<code>game.in</code>	<code>game.out</code>
4 2	Masya
5 3	Masya

Задача Е. Симка и прыжки

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

Все знают, что Симка активный фиксик. Сегодня она решила устроить день физкультуры, поэтому она будет заниматься спортом, пока не устанет. Для начала она попрыгает в длину.

Симка может прыгнуть на любое целое количество метров из интервала $[A \dots B]$. Чтобы немного разнообразить свои занятия, она хочет выбрать a , $A \leq a \leq B$, и каждый раз прыгать ровно на a метров. Она собирается сделать от C до D прыжков. В результате своих занятий она хочет пропрыгать от X до Y метров.

Формально говоря, вам даны интервалы $[A \dots B]$, $[C \dots D]$, $[X \dots Y]$, от вас требуется выбрать целые a и b ($A \leq a \leq B$, $C \leq b \leq D$), такие что $X \leq a \times b \leq Y$.

Формат входных данных

В единственной строке входного файла находятся шесть натуральных чисел A, B, C, D, X, Y ($A \leq B \leq 10^{12}$, $C \leq D \leq 10^{12}$, $X \leq Y \leq 10^{12}$).

Формат выходных данных

В первой строке выведите YES, если решение существует, и в следующей строке два натуральных числа a и b ($A \leq a \leq B$, $C \leq b \leq D$). В противном случае, выведите NO.

Если существует несколько ответов — выведите любой из них.

Примеры

jumps.in	jumps.out
1 10 5 15 20 30	YES 2 10
3 9 5 15 23 50	YES 7 7
6 7 8 9 100 200	NO

Задача F. Счастливые билеты

Имя входного файла: `lucky.in`
Имя выходного файла: `lucky.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дим Димыч едет в автобусе.

Каждому пассажиру в автобусе на входе выдают билет, а на каждом билете написан номер — число от 1 до 10^9 . Дим Димыч заметил, что до него в автобус вошли M пассажиров. Номер билета который выдают очередному пассажиру всегда на единицу больше, чем номер билета, который выдали предыдущему пассажиру. Счастливым билетом называется билет, в номере которого все цифры одинаковые. Дим Димычу стало интересно, у скольких пассажиров в автобусе счастливые номера. Помогите ему выяснить это.

Формат входных данных

В единственной строке входного файла через пробел перечислено два числа N ($1 \leq N \leq 10^9$) — номер билета, который выдали Дим Димычу, и M ($0 \leq M \leq 10^9$) — количество пассажиров, вошедших до него.

Формат выходных данных

Выведите единственное число — количество счастливых билетов.

Примеры

<code>lucky.in</code>	<code>lucky.out</code>
1000 999	27

Задача G. Грибы

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

Дим Димыч вместе с одноклассниками отправился собирать грибы.

Всего (включая Дим Димыча) собралось 5 человек. Для удобства они посчитались и каждому ученику присвоили номер от одного до пяти. Ребята решили устроить соревнование, чтобы определить, кто лучше всех собирает грибы. Соревнования устроены следующим образом: всего будет десять забегов, в каждом забеге участвуют два ученика:

В первом забеге — первый и второй участники.

Во втором — первый и третий.

В третьем — первый и четвертый.

В четвертом — первый и пятый.

В пятом — второй и третий.

В шестом — второй и четвертый.

В седьмом — второй и пятый.

В восьмом — третий и четвертый.

В девятом — третий и пятый.

В десятом — четвертый и пятый.

Дим Димыч записал, сколько грибов собрали суммарно в каждом забеге. А также он знает, что каждый участник в любом из своих забегов собирал одинаковое (целое, больше либо равное нулю) количество грибов. Теперь Дим Димыч хочет подвести итоги соревнования, но боится, что допустил в записях ошибку. Помогите ему выяснить, была ли допущена ошибка.

Формат входных данных

В единственной строке входного файла через пробел перечислены 10 чисел (от 1 до 10^5) — количество грибов собранных, в каждом забеге.

Формат выходных данных

Выведите «NO» (без кавычек), если в результатах соревнования допущена ошибка, и выведите «YES» в противном случае.

Примеры

mushrooms.in	mushrooms.out
9 13 7 8 10 4 5 8 9 3	YES
1 1 1 1 1 1 1 1 1 1	NO
2 2 2 2 2 2 2 2 2 2	YES

Замечание

В первом примере подходящие количества грибов, собранных каждым участником: 6, 3, 7, 1, 2.

В третьем примере каждый участник собрал по одному грибу.

Задача Н. Последовательность

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

Участвуя в раскопках гробницы Тутонхамона, фиксики нашли последовательность из n k -битных чисел и руководство к действию. Чтобы открыть таинственную дверь, нужно выполнить последовательность из m операций одного из двух типов:

- 1 $x y$ — поменять число в позиции x на число y ;
- 2 $l r$ — посчитать значение функции $f(l, r)$.

Функция $f(l, r)$ определяется так:

- $f(l, l) = a_l$;
- $f(l, r) = \neg(f(l, r - 1) \& a_r)$, где $\neg$ — операция побитового отрицания числа, а $\&$ — операция побитового И двух чисел.

Помогите фиксикам найти значения, полученные в ходе выполнения всех операций второго типа.

Формат входных данных

В первой строке даны два числа n, m, k ($1 \leq n, m \leq 2 \cdot 10^5, 1 \leq k \leq 31$) — количество чисел в последовательности, число операций и длина числа.

Во второй строке даны n чисел a_i ($0 \leq a_i \leq 2^k - 1$) — исходное состояние последовательности.

В следующих m строках даны запросы.

Для запроса первого типа записаны три числа 1 $x y$ ($1 \leq x \leq n, 0 \leq y \leq 2^k - 1$) — позиция в последовательности и новое значение.

Для запроса второго типа записаны три числа 2 $l r$ ($1 \leq l \leq r \leq n$) — левая и правая граница подотрезка, на котором нужно посчитать значение функции.

Формат выходных данных

Для каждого запроса второго типа выведите одно целое число — результат применения операции на этом отрезке.

Примеры

nand.in	nand.out
4 4 5	31
31 0 12 4	31
2 1 4	27
2 2 3	
1 3 19	
2 2 4	

Задача I. Починка транзистора

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

Папус учит Нолика чинить транзисторы. Транзисторы состоят из основы и трех ножек. Для каждой ножки сопротивление выражается целым числом и известно. На основе написаны два числа A и B . Транзистор будет работать, если сумма сопротивлений ножек равна A , а произведение соответственно B .

Когда Папус ушел, Нолик обнаружил основу от транзистора. Чтобы показать свою самостоятельность, он хочет починить этот транзистор.

У Нолика есть n ножек в ремонтном наборе. Чинить транзисторы он умеет, однако выбрать нужные ножки непростая задача. Помогите ему найти три ножки, которые будут совместимы с данной основой.

Формат входных данных

В первой строке входного файла даны три числа n , A , B ($3 \leq n \leq 10^5, 1 \leq A, B \leq 10^5$) — количество ножек в ремонтном наборе, сумма и произведение искомых ножек.

Во второй строке даны n чисел a_i ($1 \leq a_i \leq 10^5$) — сопротивления ножек, которые есть в наборе.

Формат выходных данных

В выходной файл выведите три числа — сопротивление подходящих ножек. Если ответов несколько, можно вывести любой. Если ответа не существует выведите `-1`.

Примеры

<code>repair.in</code>	<code>repair.out</code>
5 11 40 1 2 3 4 5	2 4 5
3 9 27 3 3 3	3 3 3
3 9 27 3 2 3	-1

Задача J. Дорога домой

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

Возвращаясь с работы Гений Евгеньевич Чудаков обратил внимание, что вдоль дороги расставлены разные дорожные знаки. Дорожные знаки на данной дороге представляют из себя таблички, с написанными на них числами. В любой момент времени действующим правилом на дороге является число, написанное на последнем не отмененном из всех введенных до этого знаков. Если на знаке написано число x , то:

- $x > 0$: Вводит правило x .
- $x < 0$: Отменяет правило $-x$, если таковое было введено на дороге.
- $x = 0$: Отменяет действующее правило. Если действующих правил нет, то ничего не происходит.

При этом число $x \neq 0$ может встретиться на дороге не более одного раза. Например, Чудаков встретил следующую последовательность из четырех знаков: 1, 2, -2, 0. После знака -2, действующим правилом будет, соответственно, правило 1.

Гений Евгеньевич хочет после проезда каждого нового знака знать, какое правило действует сейчас, но из-за обучения фиксиков у него нет времени, чтобы решить эту задачу, поэтому он просит вас о помощи в этом нелегком деле.

Формат входных данных

В первой строке входного файла дано одно число n ($1 \leq n \leq 2 \cdot 10^5$) — количество знаков на дороге.

Далее следуют n строк. В каждой строке записано одно число a_i ($-10^5 \leq a_i \leq 10^5$) — число, написанное на знаке, встреченном Гению Евгеньевичу i -м.

Формат выходных данных

Для каждого знака выведите на новой строке число — действующее правило после исполнения инструкции текущего знака. Если действующих правил нет, выведите 0.

Примеры

road.in	road.out
7	1
1	0
-1	4
4	2
2	3
3	3
-2	4
0	

Задача К. Дерево

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

Гений Евгеньевич давно мечтал о дереве. Сейчас в моде деревья с n вершинами, на каждой из которых написано число x_i . Долгожданный день настал, и Гений Евгеньевич смог себе приобрести такое дерево. Вернувшись из магазина после покупки, он начал считать различные величины на этом дереве. Все ему давалось очень просто, пока он не решил найти путь в этом дереве, значение величины $l(p(u, v)) \cdot \min(x_u, x_v)$ для которого максимально. Здесь $p(u, v)$ — путь между вершинами u и v , $l(p)$ — длина пути p в ребрах.

Уже который день Гений Евгеньевич не выходит на улицу. Его друзья, естественно, начали волноваться. Они просят Вас помочь Гению Евгеньевичу справиться с задачей.

Формат входных данных

В первой строке входных данных дано число n ($1 \leq n \leq 10^5$) — количество вершин в дереве.

Во второй строке даны n чисел x_i ($1 \leq x_i \leq 10^9$) — числа написанные на вершинах. На i -й вершине написано число x_i .

В следующих $n - 1$ -й строке идут пары чисел a_i, b_i ($1 \leq a_i, b_i \leq n$) — рёбра в дереве.

Формат выходных данных

Выведите максимальное значение величины, описанной в условии.

Примеры

tree.in	tree.out
8 9 2 8 8 7 2 1 2 2 1 3 1 4 1 5 2 6 4 7 5 8 5	21

Задача L. Великий Флатландский Забор

Имя входного файла: `fence.in`
Имя выходного файла: `fence.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 МБ

С незапамятных времен граница страны Флатландия имеет форму N -угольника без самопересечений и самокасаний (необязательно выпуклого). В каждой из вершин этого многоугольника король построил по башне.

В целях увеличения обороноспособности государства на его территории король задумал построить Великий Флатландский Забор. По причине многолетней войны ресурсов хватает на строительство ровно K стен (неважно, какой длины). Каждая стена должна соединять ровно две башни по прямой и не должна даже частично выходить за пределы Флатландии. К тому же, Забор должен представлять собой не более чем K -угольник также без самопересечений и самокасаний (углов может оказаться и меньше, чем K , так как некоторые соседние стены могут лежать на одной прямой). Военный советник настаивает на том, что площадь защищенной области должна быть как можно больше.

Ваша задача — помочь спроектировать такой Забор.

Формат входного файла

В первой строке записаны два целых числа N и K ($3 \leq K \leq N \leq 230$). В следующих N строках содержатся пары целых чисел — координаты башен в порядке обхода границы против часовой стрелки. Гарантируется, что никакие три последовательные башни не лежат на одной прямой. Все координаты не превосходят 1000 по абсолютной величине.

Формат выходного файла

В первую строку запишите максимальную площадь, которую можно защитить Забором, с точностью до пяти знаков после десятичной точки. Во второй строке должно быть Q — количество выбранных башен.

Будем считать, что башни занумерованы числами от 1 до N в порядке перечисления их во входном файле. Во третью строку через пробел выведите номера Q башен, которые будут вершинами Забора, в порядке его обхода против часовой стрелки.

Пример

<code>fence.in</code>	<code>fence.out</code>
3 3 0 0 1 0 0 1	0.50000 3 2 3 1
6 5 0 0 7 0 4 3 4 4 3 4 0 7	24.500000 3 1 2 6
4 3 0 0 -2 -2 1 3 0 2	2.000000 3 1 3 4

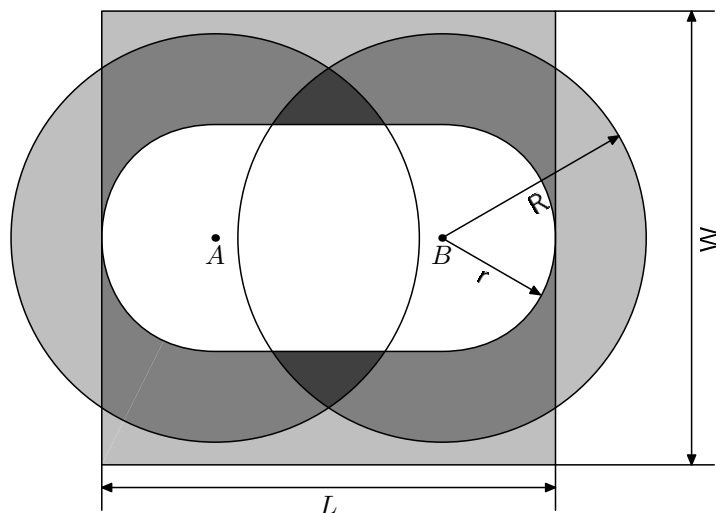
Задача М. Осторожный судья

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

Завтра состоится футбольный матч между двумя знаменитыми командами: Газмясом и Нефтьрыбом. Матч будет проходить на поле длины L и ширины W . Матч будет судить профессиональный футбольный судья в четвертом поколении Вениамин Хлебников.

Быть судьей — ответственное и не всегда безопасное занятие. Поэтому Вениамин решил проработать некоторые игровые эпизоды, которые возникнут в завтрашней игре.

Рассмотрим ситуацию, когда игрок A делает пас игроку B — то есть, передает ему мяч по отрезку, соединяющему точки, в которых находятся игроки. С одной стороны, судья должен хорошо видеть то, что происходит во время паса; с другой стороны, согласно требованиям безопасности, судья не может находиться слишком близко к мячу. Поэтому во время паса судья должен находиться на расстоянии, не меньшем, чем r , и не большем, чем R , от возможного положения мяча. При этом считается, что все то время, в течение которого движется мяч, судья стоит на одном месте. Разумеется, судья должен все время матча находиться на поле.



Так как эти условия достаточно сложны, то даже опытному судье иногда бывает трудно определить, где он должен находиться в момент паса. По этой причине Вениамин хочет перед матчем потренироваться находить те области, где он может находиться, при различных начальных условиях. Для того чтобы сравнить свой ответ с правильным, ему необходима программа, которая по заданным размерам поля, координатам игроков и числам r и R находит площадь тех областей поля, в которых может находиться судья. Помогите ему!

Формат входного файла

В первой строке входного файла даны два целых положительных числа L и W ($1 \leq L, W \leq 100$) — длина и ширина поля.

Во второй строке даны целые числа X_A, Y_A, X_B, Y_B — координаты игроков A и B соответственно. Так как игроки находятся на поле, то $0 \leq X_A, X_B \leq L$, $0 \leq Y_A, Y_B \leq W$.

В третьей строке даны целые числа r и R ($0 < r < R < 100$). Известно, что $R \leq D$, где $D = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$ — расстояние между игроками A и B .

Формат выходного файла

В выходной файл выведите ответ на задачу. Ответ принимается, если он отличается от корректного не более, чем на 10^{-6} .

Примеры

referee.in	referee.out
20 20 5 10 15 10 5 9	13.95667511974284