

256 мегабайт

Задача А. Ассоциативное мышление

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Профессор $\hat{H}$ часто совершает важные открытия. Особенно хорошо у него это получается, когда он гуляет по бульвару. Вдоль всего бульвара установлены лавочки, и среди них нет ни одной пары одинаковых.

Профессор $\hat{H}$ находит это обстоятельство совершенно замечательным, и вот почему. Он давно заметил, что время, необходимое для совершения очередного открытия, удивительным образом совпадает со временем, в течение которого он проходит мимо k лавочек. Поскольку у профессора $\hat{H}$ чрезвычайно развито ассоциативное мышление, то ему достаточно посмотреть некоторое время на лавочку — и он уже не позабудет сделанное открытие.

После этого он отправляется гулять дальше, и, пройдя очередные k лавочек, запоминает очередное сделанное открытие. Каждый раз, когда он доходит до конца бульвара, он разворачивается и продолжает гулять и совершать открытия дальше. Однако профессор $\hat{H}$ не может связывать с одной лавочкой более одного открытия. И если он, сделав очередное открытие, оказался возле лавочки, с которой уже связано другое открытие, он прекращает прогулку и отправляется домой, чтобы не забыть совершённое открытие.

Будем считать лавочки занумерованными от 1 до n . Первая лавочка расположена в самом начале бульвара, а лавочка n — в самом конце бульвара. Профессор начал свою прогулку от лавочки m и сначала отправился по направлению к лавочке n . Лавочку m в начале прогулки профессор не считает пройденной. Лавочки, возле которых он разворачивается (в начале и в конце бульвара), он считает пройденными единожды, а не дважды.

Ваша задача — определить, сколько открытий совершит профессор, прежде чем закончит прогулку.

Формат входного файла

В первой строке содержатся целые числа n, m, k ($2 \leq n \leq 5000, 1 \leq m \leq n, 1 \leq k \leq 5000$) — количество лавочек на бульваре, номер лавочки, возле которой начинается прогулка профессора, количество лавочек, которые должен миновать профессор для совершения очередного открытия.

Формат выходного файла

В первой строке — количество открытий, которые совершит профессор, прежде чем завершит прогулку.

Примеры

<code>input.txt</code>	<code>output.txt</code>
5 3 4	2
6 6 3	6

Note

В первом примере профессор сначала миновывает четвёртую лавочку, затем пятую, вернется к четвёртой, а возле третьей совершит открытие. Затем он пройдет мимо второй, мимо первой, вновь мимо второй и вернется к третьей. Увы, она уже занята предыдущим открытием, и профессору придется пойти домой, совершив два открытия.

Во втором примере профессор сразу развернется и пойдет к началу бульвара. Он совершит открытия возле лавочек с номерами 3, 2, 5, 4, 1, и после этого вновь придет к лавочке 4. Таким образом, сделав 6 открытий, профессор отправится домой.

256 мегабайт

Задача В. Точка, точка, запятая...

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Совершив очередное открытие, профессор $\hat{H}$, конечно же, пишет очередную статью в журнал. К сожалению, в последнее время на его любимой клавиатуре клавиша пробела работает не так хорошо, как хотелось бы: иногда её нажатие приводит к появлению в тексте нескольких пробелов подряд, иногда, напротив, пробел вовсе не печатается.

Профессор $\hat{H}$ написал много статей, прежде чем обратил на это внимание. И теперь он хочет автоматизировать исправление опечаток, связанных с пробелами. В его статьях встречаются следующие знаки: точки (.), запятые (,), восклицательные знаки (!), вопросительные знаки (?), двоеточия (:), многоточия (...), три идущие подряд точки (.), точки с запятой (;), тире (--) и дефисы (-).

Согласно правилам, перед этими знаками, за исключением тире, пробел не ставится. После этих знаков, за исключением дефиса, пробел ставится.

В тексте статей, кроме перечисленных выше знаков, могут присутствовать строчные и заглавные латинские буквы. Переводы строк в тексте отсутствуют. Никакие два знака не могут идти подряд (в том числе через пробел(ы)). Текст начинается с буквы (перед которой не нужен пробел) и завершается точкой или многоточием (после которых пробел не ставится).

Профессору $\hat{H}$ сейчас очень важно знать, какое количество символов будет содержать его статья, если в неё добавить недостающие и удалить лишние пробелы.

Ваша задача — написать программу, которая подсчитает количество символов в исправленной статье профессора $\hat{H}$.

Формат входного файла

Входными данными является текст статьи профессора $\hat{H}$. Текст содержит не более 5000 символов, описанных в условии.

Формат выходного файла

Целое число — количество символов в статье профессора после внесения исправлений.

Примеры

<code>input.txt</code>	<code>output.txt</code>
My , name-- is H!I am:professor .Problem - B ;is very easy ...	61
Punctuation is a very important.	32

256 мегабайт

Задача С. В поисках непротиворечивости

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Недавно профессора $\hat{H}$ пригласили поработать с группой археологов. Среди находок оказался папирус, в котором, помимо прочего, описывались состояния самых богатых горожан.

В папирусе встречались записи вида:

«Фенуку владеет имуществом стоимостью не менее, чем имущество Шушу, имущество Эбо, имущество Табита и еще раз имущество Табита, вместе взятых»

или

«Хорус владеет имуществом стоимостью более, чем имущество Табита, Фенуку, Джаха и Бакари, вместе взятых».

Профессор $\hat{H}$ понял, что древние не умели обращаться с большими числами и заинтересовался, могут ли все утверждения папируса одновременно быть верными. Он построил математическую модель этих записей, состоящую из n неравенств (по числу записей в папирусе), используя в них вместо имён горожан переменные вида x_j , где j — целое положительное число, не превосходящее 100.

Таким образом, у него получились записи вида:

$$x_1 \geq x_2 + x_2 + x_3 + x_5,$$

$$x_6 > x_2 + x_1 + x_8 + x_{11}.$$

Ваша задача — найти набор чисел x_j ($1 \leq x_j \leq 10^{16}$), который удовлетворяет системе неравенств. В случае, если решение не существует, следует вывести единственное число 0.

Формат входного файла

В первой строке содержатся целые числа n и m ($1 \leq n \leq 10, 1 \leq m \leq 100$) — количество неравенств и наибольший номер переменной, которая в них используется (нумерация переменных с единицы).

В каждой из последующих n строк содержится по одному неравенству, подобному описанному в условии. В левой части каждого неравенства записана единственная переменная, в правой части — сумма переменных. В неравенствах используются только знаки $>$ и $\geq$. В правой части неравенства сумма состоит из не более чем десяти слагаемых.

Гарантируется, что если система имеет решение в целых положительных числах, то она имеет и решение в ограничениях ($1 \leq x_j \leq 10^{16}$).

Формат выходного файла

В каждой из m строк содержится возможное значение переменной с соответствующим номером, если система неравенств допускает решение.

В противном случае в единственной строке содержится единственное число 0.

Примеры

input.txt	output.txt
2 11 x1 >= x2 + x2 + x3 + x5 x6 > x1 + x2 + x8 + x11	4 1 1 1 1 8 1 1 1 1 1
2 2 x1 > x2 x2 > x1	0
2 2 x1 >= x2 x2 >= x1	1 1
5 5 x2 >= x3 + x3 + x3 + x3 + x3 + x3 + x3 + x3 + x3 + x3 x1 >= x3 x3 > x5 + x4 x1 > x2 + x5 x4 > x5	42 40 4 2 1

256 мегабайт

Задача D. Надпись на стене

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ещё одной находкой археологов оказалась надпись на городской стене. Эта надпись содержит n символов.

Профессор $\hat{H}$ предполагает, что надпись имела некое «охраняющее» значение и содержала в том числе многократно повторённое имя покровителя города. Также профессор абсолютно уверен в том, что имя покровителя состоит ровно из 2^k символов. Наконец, он совершенно убеждён, что среди всех подстрок длиной 2^k символов имя покровителя города — наиболее часто встречающаяся в надписи подстрока.

Ваша задача — по заданной надписи найти подстроку длины 2^k , которая встречается в ней чаще всего. Если таких подстрок несколько, выберите среди них лексикографически минимальную (т.е. ту, которая окажется первой в словаре, если внести в него все найденные варианты).

Формат входного файла

В первой строке содержатся целые числа n и k ($1 \leq n \leq 10^5$, $1 \leq 2^k \leq n$).

Во второй строке содержится строка из n строчных латинских букв.

Формат выходного файла

В единственной строке — найденная подстрока.

Примеры

input.txt	output.txt
35 2 sitametconsecteturacommodoconsequat	cons
26 0 zyxwvutsrqponmlkjihgfedcba	a

Note

В первом примере существуют две подстроки длины 4, которые входят в строку дважды: это «cons» и «onse».

Для лексикографического сравнения необходимо найти первую позицию, в которой эти подстроки различаются. Та подстрока, в которой буква в этой позиции предшествует в алфавите букве в этой же позиции в другой подстроке, и будет меньше. В данном случае это позиция 1 (при нумерации с единицы), буква «с» предшествует в алфавите букве «о», так что ответом будет подстрока «cons».

256 мегабайт

Задача Е. Весомые аргументы

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Профессор $\hat{H}$ успешно работает в самых разных областях науки. Недавно он получил приглашения на две конференции — по археологии и по криптографии. К сожалению, обе они проходят в одно и то же время в разных городах. Это значит, что профессору придется выбирать, куда же он поедет.

Профессор $\hat{H}$ решил подойти к выбору очень тщательно. «Конференция по археологии проходит в очень красивом городе, и это плюс». — рассуждал он. — «Но на конференцию по криптографии планирует приехать профессор $\hat{L}$ с докладом о новом методе построения шифров. Это тоже плюс. К тому же с ним можно обсудить мой новый результат, а это большой плюс. Даже два плюса».

По мере возникновения тех или иных доводов в пользу поездки на ту или иную конференцию профессор записывал количество плюсов, которые дают эти доводы. Правда, чтобы не запутаться, перед количеством плюсов, которые дает конференция по криптографии, он записывал минус.

Он решил, что если количество плюсов какой-либо конференции окажется хотя бы на p больше, чем количество плюсов другой конференции, то он поедет на неё. В противном же случае он запросит у организаторов обеих конференций дополнительную информацию.

Ваша задача — определить, какое решение примет профессор $\hat{H}$.

Формат входного файла

В первой строке содержится целые числа n и p ($1 \leq n \leq 1000, 1 \leq p \leq 1000$) — общее количество доводов, сформулированных профессором $\hat{H}$, и разница в количестве плюсов, которая позволит профессору сделать выбор.

Во второй строке содержится n положительных и отрицательных чисел — количество плюсов, которые дают конференции. Каждое из этих чисел не превосходит по абсолютной величине 1000.

Формат выходного файла

Выведите единственное целое число:

- 0 — если профессор запросит дополнительную информацию,
- 1 — если профессор поедет на конференцию по археологии,
- 2 — если профессор поедет на конференцию по криптографии.

Примеры

input.txt	output.txt
5 3 1 -3 2 -1 -1	0
6 2 -2 -2 4 1 -1 -3	2

256 мегабайт

Задача F. Продуктовая корзина

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Иногда профессору $\hat{H}$ хочется хотя бы на некоторое время сменить область своих исследований. И тогда он отправляется в гости к своим друзьям, чтобы за чашкой чая поговорить об актуальных проблемах разных областей науки. Профессор $\hat{H}$ не любит ходить в гости с пустыми руками, и непременно заходит в супермаркет, чтобы приобрести что-нибудь вкусное.

В супермаркете каждый покупатель сам взвешивает приобретенный товар на весах и лишь потом отправляется на кассу. Весов в супермаркете двое, но есть небольшая проблема — каждый товар можно взвесить лишь на каких-то одних весах, поскольку информация об этом товаре в памяти других весов отсутствует.

У профессора $\hat{H}$ в корзинке n товаров. Он подходит к произвольно выбранным весам и действует следующим образом.

Профессор достаёт произвольно выбранный товар из корзинки, кладёт его на весы и нажимает соответствующую клавишу. Если в памяти весов есть информация об этом товаре, то в течение 1 единицы времени происходит печать этикетки. Профессор $\hat{H}$ наклеивает этикетку на товар, достаёт из корзинки (также случайным образом) следующий товар и кладёт его на эти же весы.

Если же информация о товаре отсутствует, то на экране весов высвечивается сообщение об ошибке, и профессор переходит к другим весам. Переход также занимает 1 единицу времени.

Для каждого товара известно, на каких весах его можно взвесить. Ваша задача — найти среднее значение времени, которое профессор может потратить на взвешивание всего содержимого своей корзинки.

Пояснение

Для случайной величины t , которая может принимать дискретные значения $t_1, t_2, \dots, t_m$ с вероятностью $p_1, p_2, \dots, p_m$ ($0 \leq p_i \leq 1, i = 1, 2, \dots, m$) соответственно, среднее значение (математическое ожидание) может быть вычислено как $\sum_{i=1}^m t_i \cdot p_i$.

Формат входного файла

В первой строке содержится целое число n ($1 \leq n \leq 100$) — количество товаров в корзинке профессора $\hat{H}$.

В каждой из следующих n строк содержится число 1, если товар может быть взвешен только на первых весах, и 2, если товар может быть взвешен только на вторых весах.

Формат выходного файла

Вещественное число с абсолютной или относительной точностью не менее 4 знаков после запятой — среднее значение времени, которое профессор $\hat{H}$ может потратить на взвешивание всех товаров из корзинки.

Примеры

input.txt	output.txt
3 1 2 1	4.83333
4 1 1 2 2	6.50000

Note

В первом примере доставать товары из корзинки можно 6 способами. Кроме того, необходимо учесть, что для каждого способа нужно рассмотреть две ситуации: когда профессор сначала подходит к первым весам и когда профессор сначала подходит ко вторым весам. В приведённой ниже таблице указано время, которое потребуется для взвешивания каждой последовательности товаров на обоих весах.

Последовательность номеров товаров	Весы 1	Весы 2
1 2 3	5	6
1 3 2	4	5
2 1 3	5	4
2 3 1	5	4
3 1 2	4	5
3 2 1	5	6

Отсюда легко видеть, что среднее время взвешивания всех товаров из корзинки составит $58/12$ или $4.833333(3)$.

256 мегабайт

Задача G. Зеленый и черный

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Профессор $\hat{H}$ собирается в гости к своим друзьям, чтобы побеседовать с ними за чашечкой чая о главном вопросе Вселенной, жизни и всего остального. Нужно сказать, он старается не злоупотреблять вниманием своих друзей, поэтому за день успевает заглянуть в гости к нескольким друзьям.

Среди друзей профессора есть как любители чёрного чая, так и любители зелёного чая. Когда профессор $\hat{H}$ приходит к ним в гости, они предлагают ему именно тот чай, который любят сами.

Профессор $\hat{H}$ полагает, что чёрный чай не слишком способствует творческому настрою, и потому следит, чтобы в любой момент времени (в течение дня) количество выпитых им чашечек чёрного чая превосходило количество выпитых им чашечек зелёного чая не более чем на b .

Конечно, может случиться так, что профессор $\hat{H}$ придёт в гости к любителю чёрного чая, и окажется, что очередная чашечка чёрного чая сделает эту разницу неприемлемой с точки зрения профессора. В этом случае он поблагодарит друга и предложит ему выпить с ним зелёного чая, который (разумеется, чисто случайно) есть у профессора в портфеле. Друг также поблагодарит профессора, но предпочтёт пить чёрный чай.

Профессор $\hat{H}$ уже решил, в какой последовательности он посетит сегодня n своих друзей. Ему известно, какой чай любит каждый из его друзей. Он записал строку из n символов и поставил на позиции j букву B , если j -ый друг любит чёрный чай, и букву G , если j -ый друг любит зелёный чай. Теперь он хочет знать, какой запас зелёного чая ему потребуется.

Ваша задача — определить, сколько раз профессор будет заваривать свой чай.

Формат входного файла

В первой строке содержатся целые числа n ($1 \leq n \leq 100000$) и b ($1 \leq b \leq 100000$) — количество друзей, которых посетил профессор $\hat{H}$, и допустимая разница между количеством чашечек чёрного и чашечек зелёного чая.

Вторая строка — строка из n символов B и G (все символы — заглавные латинские буквы).

Формат выходного файла

В первой строке выведите единственное целое число — сколько раз профессор $\hat{H}$ заварит свой чай.

Примеры

<code>input.txt</code>	<code>output.txt</code>
6 1 BGBBGG	1
10 3 BGBBBBGBBG	1

256 мегабайт

Задача Н. Восстановление пути

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

После посещения друзей профессор $\hat{H}$ пребывал в замечательном настроении: он снова был готов совершать открытия. И эти открытия он начал совершать уже по дороге домой.

Конечно, Вы помните, что у профессора $\hat{H}$ чрезвычайно развито ассоциативное мышление. К сожалению, его путь пролегал слишком далеко от набережной с лавочками, и профессору пришлось выбирать другие объекты в качестве «пометок» для своих открытий. По счастью, в городе, где живёт профессор $\hat{H}$, установлено очень много больших и не очень скульптур, изображающих литературных героев, или же посвящённых какому-нибудь событию в жизни города. Именно они и стали такими «пометками».

Наутро профессор $\hat{H}$ рассматривал карту города: ему нужно было восстановить в памяти все сделанные открытия. Однако это оказалось не так-то просто: профессор не был уверен, что помнит все сделанные им «пометки». Более того, он не был уверен, от кого именно из друзей он отправился домой. В чём он был уверен точно — так это в том, что домой он шёл кратчайшим путём.

Конечно, если профессор окажется в том месте, где установлена скульптура, посмотрит на неё, то он вспомнит, связано ли с ней открытие (и если связано, то какое). Поэтому сейчас он хочет выяснить, на какие скульптуры ему нужно сходить посмотреть.

Отметим один интересный факт: возле домов всех друзей профессора $\hat{H}$ есть скульптуры.

Формат входного файла

В первой строке содержатся целые числа n , m и s ($1 \leq n \leq m$, $1 \leq m \leq 100000$, $1 \leq s \leq m$) — количество друзей профессора $\hat{H}$, количество скульптур в городе и количество скульптур, про которые профессор помнит, что использовал их в качестве «пометок». Скульптуры, находящиеся возле домов друзей профессора, занумерованы числами от 1 до n . Профессор $\hat{H}$ живёт в доме 0.

Во второй строке содержится s целых чисел — номера скульптур, использованных в качестве «пометок».

В третьей строке содержится целое число p ($1 \leq p \leq 100000$) — количество описаний дорог между скульптурами.

В каждой из следующих p строк содержится по одному описанию дороги. Описание состоит из двух целых чисел — номеров соответствующих скульптур. Если между скульптурами с номерами i и j существует дорога, то дорога существует и между скульптурами j и i .

Гарантируется, что профессор может добраться от своего дома до дома любого своего друга.

Формат выходного файла

Если не существует ни одного такого друга профессора, что кратчайший путь от его дома до дома профессора проходит через все уже существующие «пометки», выведите в качестве ответа единственную строку **BAD**.

Иначе выведите в качестве ответа m строк. Если скульптура k является или может являться «пометкой», выведите в строке с номером k **YES**, в противном случае выведите **NO**.

Примеры

input.txt	output.txt
2 5 2 1 2 5 0 1 1 2 2 3 3 4 4 5	YES YES NO NO NO
2 5 3 1 2 3 5 0 1 1 2 2 3 3 4 4 5	BAD
2 5 2 1 2 5 0 1 0 2 2 3 3 4 4 5	BAD
4 8 1 6 10 5 0 0 4 5 1 5 6 4 6 4 7 6 7 6 2 2 8 7 3	NO YES NO YES YES YES NO NO NO

256 мегабайт

Задача I. Грант

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Просматривая рабочую почту, профессор $\hat{H}$ нашёл объявление о конкурсе грантов. Это было очень кстати — одно из совершённых им вчера открытий отлично вписывалось в нужную тематику. Единственная проблема состояла в том, что до подачи заявки оставалось совсем немного времени, и нужно было срочно собирать рабочую группу, договариваться по некоторым организационным вопросам ...

Университетский городок состоит из n корпусов, между которыми пролегает m тропинок. Из любого корпуса по тропинкам можно пройти в любой другой. И за сегодняшний день профессору $\hat{H}$ пришлось изрядно походить по этим тропинкам.

Вечером, оказавшись в своём кабинете (который размещается в корпусе 1), профессор задумался, каким же был его сегодняшний «маршрут». Он помнил, что в каждом корпусе побывал нечётное количество раз.

Ваша задача — найти какой-либо возможный «маршрут» профессора $\hat{H}$. Профессор абсолютно уверен, что этот маршрут не может быть длиннее, чем $10 \cdot n + 42$.

Формат входного файла

В первой строке содержатся целые числа n и m ($2 \leq n \leq 100, 1 \leq m \leq 1000$) — количество корпусов в университетском городке и количество тропинок между ними.

В каждой из следующих m строк содержится по паре целых чисел — номеров корпусов, которые связывает соответствующая тропинка.

Формат выходного файла

В первой строке выведите целое число — количество номеров корпусов, которые образуют «маршрут» профессора $\hat{H}$.

Во второй строке выведите эти номера корпусов в порядке посещения их профессором. «Маршрут» должен начинаться и заканчиваться в корпусе 1. Между любыми соседними в «маршруте» корпусами должна быть тропинка. Профессор не считает, что посетил корпус 1 в начале «маршрута», и считает, что посетил его в конце. Таким образом, в этой строке любое число от 2 до n должно встретиться нечётное число раз, а число 1 — чётное число раз. Обратите внимание, что 0 — чётное число.

Если искомого маршрута не существует, в единственной строке выведите -1 .

Примеры

input.txt	output.txt
2 1 1 2	7 1 2 1 2 1 2 1
3 2 1 2 2 3	-1
3 3 1 2 2 3 1 3	12 1 2 1 2 1 2 3 2 3 2 3 1
9 10 9 4 9 5 5 1 5 6 4 6 4 7 6 7 6 2 7 3 2 8	50 1 5 1 5 1 5 6 2 6 2 8 2 8 2 8 2 6 4 6 4 6 5 6 5 6 7 3 7 3 7 3 7 4 7 4 7 4 7 4 9 4 9 4 9 4 9 5 9 5 1

256 мегабайт

Задача J. Вечер трудного дня

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Профессор $\hat{H}$ дописал заявку на грант. Завтра утром он перечитает её и отправит. А на сегодня, пожалуй, работа закончена. Бумаги сложены, компьютер выключен. Осталось только убрать на полку шкафа стаканы: профессор не любит оставлять их на столе.

Стаканов у профессора два: один имеет форму цилиндра (из него профессор пьёт зелёный чай), другой — усечённого конуса, основания которого перпендикулярны оси симметрии конуса (он предназначен для чёрного чая). Поскольку на полке в шкафу места не очень много, профессор помещает один стакан в другой, в точности совмещая их оси симметрии. Делает он это таким образом, чтобы высота получившейся «конструкции» оказалась минимально возможной.

Ваша задача — определить эту минимальную высоту.

Считайте, что толщина стенок стаканов пренебрежимо мала. Разумеется, профессор ставит стаканы на полку таким образом, чтобы дно «внешнего» стакана полностью соприкасалось с полкой — иначе он не считает «конструкцию» достаточно устойчивой.

Формат входного файла

В первой строке содержатся целые числа r и h ($1 \leq r \leq 100$, $1 \leq h \leq 100$) — радиус основания и высота цилиндрического стакана.

Во второй строке содержатся целые числа b , t , l ($1 \leq b < t \leq 100$, $1 \leq l \leq 100$) — радиус нижнего основания, радиус верхнего основания и высота стакана, имеющего форму усечённого конуса.

Формат выходного файла

Выведите единственное вещественное число, вычисленное с абсолютной или относительной точностью не менее 4 знаков после запятой, — минимально возможную высоту описанной «конструкции» из двух стаканов.

Примеры

<code>input.txt</code>	<code>output.txt</code>
5 10 5 7 8	10.000000
5 10 3 11 12	13.000000