

Задача А. Веревочки

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

С утра шел дождь, и ничего не предвещало беды. Но к обеду выглянуло солнце, и в лагерь заглянула СЭС. Пройдя по всем домикам и корпусам, СЭС вынесла следующий вердикт: бельевые веревки в жилых домиках не удовлетворяют нормам СЭС. Как выяснилось, в каждом домике должно быть ровно по одной бельевой веревке, и все веревки должны иметь одинаковую длину. В лагере имеется N бельевых веревок и K домиков. Чтобы лагерь не закрыли, требуется так нарезать данные веревки, чтобы среди получившихся веревочек было K одинаковой длины. Размер штрафа обратно пропорционален длине бельевых веревок, которые будут развешены в домиках. Поэтому начальство лагеря стремится максимизировать длину этих веревочек.

Формат входных данных

В первой строке заданы два числа — N ($1 \leq N \leq 10001$) и K ($1 \leq K \leq 10001$). Далее в каждой из последующих N строк записано по одному числу — длине очередной бельевой веревки. Длина веревки задана в сантиметрах. Все длины лежат в интервале от 1 сантиметра до 100 километров включительно.

Формат выходных данных

В выходной файл следует вывести одно целое число — максимальную длину веревочек, удовлетворяющую условию, в сантиметрах. В случае, если лагерь закроют, выведите 0.

Примеры

<code>ropes.in</code>	<code>ropes.out</code>
4 11 802 743 457 539	200

Задача В. Космическое поселение

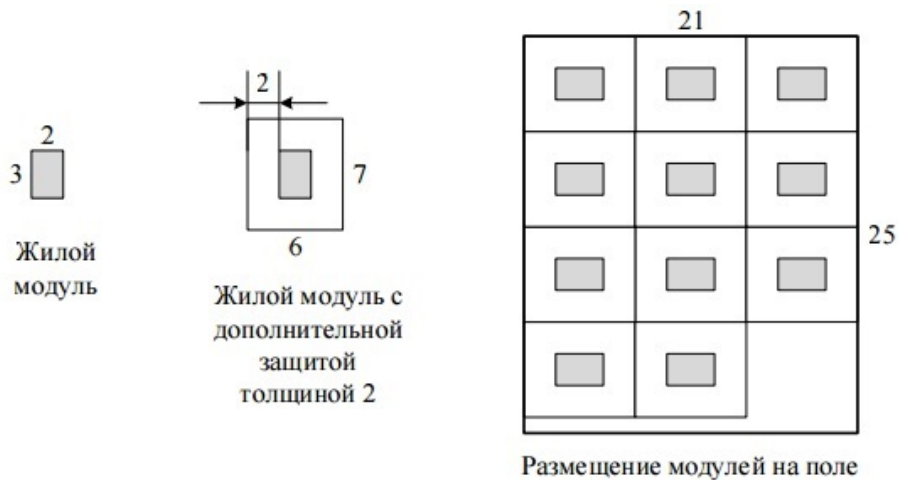
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Для освоения Марса требуется построить исследовательскую базу. База должна состоять из n одинаковых модулей, каждый из которых представляет собой прямоугольник.

Каждый модуль представляет собой жилой отсек, который имеет форму прямоугольника размером ab метров. Для повышения надежности модулей инженеры могут добавить вокруг каждого модуля слой дополнительной защиты. Толщина этого слоя должна составлять целое число метров, и все модули должны иметь одинаковую толщину дополнительной защиты. Модуль с защитой, толщина которой равна d метрам, будет иметь форму прямоугольника размером $(a+2d)(b+2d)$ метров.

Все модули должны быть расположены на заранее подготовленном прямоугольном поле размером wh метров. При этом они должны быть организованы в виде регулярной сетки: их стороны должны быть параллельны сторонам поля, и модули должны быть ориентированы одинаково.

Требуется написать программу, которая по заданному количеству и размеру модулей, а также размеру поля для их размещения, определяет максимальную толщину слоя дополнительной защиты, который можно добавить к каждому модулю.



Формат входных данных

Входной файл содержит пять разделенных пробелами целых чисел: n , a , b , w и h ($1 \leq n \leq 10^{18}$). Гарантируется, что без дополнительной защиты все модули можно разместить в поселении описанным образом.

Формат выходных данных

Выходной файл должен содержать одно целое число: максимальную возможную толщину дополнительной защиты. Если дополнительную защиту установить не удастся, требуется вывести число 0.

Примеры

стандартный ввод	стандартный вывод
11 2 3 21 25	2
1 5 5 6 6	0

Задача С. Уравнение

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

Мальчик Рома очень любит использовать бинарный поиск где попало, поэтому сейчас он бьется над решением очередной задачи. А именно дано уравнение $\cos x = ax$. Помогите Роме решить эту задачу.

Формат входных данных

Вам дано вещественное неотрицательное число a , не превышающее 1 000, с точностью до четырех знаков после запятой.

Формат выходных данных

Выведите искомый ответ — положительное число x , при котором выражение $\cos x - ax$ по абсолютному значению не превышает 0.000001.

Примеры

	<code>equatation.in</code>	<code>equatation.out</code>
1	1	0.7390851332
0.5	0.5	1.0298665293

Задача D. Дремучий лес

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

Чтобы помешать появлению СЭС в лагере, администрация ЛКШ перекопала единственную дорогу, соединяющую “Берендеевы поляны” с Судиславлем, теперь проехать по ней невозможно. Однако, трудности не остановили инспекцию, хотя для СЭС остается только одна возможность — дойти до лагеря пешком. Как известно, Судиславль находится в поле, а “Берендеевы поляны” — в лесу.

- Судиславль находится в точке с координатами $(0, 1)$.
- “Берендеевы поляны” находятся в точке с координатами $(1, 0)$.
- Граница между лесом и полем — горизонтальная прямая $y = a$, где a — некоторое число ($0 \leq a \leq 1$).
- Скорость передвижения СЭС по полю составляет V_p , скорость передвижения по лесу — V_f .
Вдоль границы можно двигаться как по лесу, так и по полю.

Администрация ЛКШ хочет узнать, сколько времени у нее осталось для подготовки к визиту СЭС. Она попросила вас выяснить, в какой точке инспекция СЭС должна войти в лес, чтобы дойти до “Берендеевых полей” как можно быстрее.

Формат входных данных

В первой строке входного файла содержатся два положительных целых числа — V_p и V_f ($1 \leq V_p, V_f \leq 10^5$). Во второй строке содержится единственное вещественное число — координата по оси Oy границы между лесом и полем a ($0 \leq a \leq 1$)

Формат выходных данных

В единственной строке выходного файла выведите вещественное число с точностью не менее 7 знаков после запятой — координата по оси Ox точки, в которой инспекция СЭС должна войти в лес.

Примеры

<code>forest.in</code>	<code>forest.out</code>
5 3 0.4	0.783310604
5 5 0.5	0.500000000

Задача Е. Изгнание

Имя входного файла: `exile.in`
Имя выходного файла: `exile.out`
Ограничение по времени: 6 секунды
Ограничение по памяти: 256 мегабайт

Борьба между титанами Кроноса и окружением Зевса породила потоп, охвативший всю Землю, и громогласное сотрясение небес. Обитель богов — Олимп, под натиском бессмертных, тоже пережил сотрясение до самого основания. Битва уравнивала силы противоборствующих сторон, и в этот момент в дело вмешался Зевс.

Он спустился с Олимпа и бросился в бой! В итоге Кронос и его титаны были повержены, изгнаны с Олимпа и отправлены в подземное царство Тартар. Среди изгнанных титанов были братья Атлас, Прометей, Эпиметеус, Меноитиос и другие.

Сразу после случившегося изгнанные титаны начали собирать свои вещи для отправления в изгнание. У каждого из N покидающих Олимп Богов есть ровно по одной ценной вещи, и они хотят взять их с собой в коробках. Для этого они хотят заказать в компании «Поставка коробок на Олимп» ровно M одинаковых коробок. При этом у каждой вещи и у каждой коробки есть свой размер, а суммарный размер всех вещей в коробке не должен превышать размера этой самой коробки. По старым стандартам Олимпа даже Богам запрещено класть в одну коробку более двух предметов, и наши изгнанники решили соблюдать это правило.

Боги хотят потратить на заказ коробок как можно меньше денег, и потому хотят заказать коробки как можно меньшего размера. Однако, они хотят, чтобы все их вещи уместились. Помогите им определить размер коробок.

Формат входных данных

В первой строке входного файла указаны целые числа N и M , $1 \leq N, M \leq 10^5$. В следующей строке через пробел записаны N чисел a_i — размеры вещей. Это неотрицательные целые числа, не превосходящие 10^9 .

Формат выходных данных

В единственной строке выходного файла укажите одно число — минимальный возможный размер коробок. Если разложить предметы по коробкам невозможно, выведите -1 .

Примеры

<code>exile.in</code>	<code>exile.out</code>
5 4 3 5 3 7 5	7

Задача F. Сообщение

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

Параллель S изучает упрощённую модель передачи сообщений в сети. Каждый пакет сообщения выглядит следующим образом:

- сначала идёт целое число n — длина сообщения в битах, записанное в двоичной системе счисления;
- далее следует последовательность из n бит — текст сообщения.

Например, если мы отправляем сообщение `0100`, то отправленный пакет будет выглядеть следующим образом: `1000100` (жирным выделен текст сообщения). Итоговый размер пакета составляет 7 бит. Размер каждого из пакетов, отправляемых по сети, не должен превосходить m .

В качестве домашнего задания параллели S задали отправить сообщения длиной n . Чтобы задание не выглядело слишком простым, требуется разбить сообщение на пакеты так, чтобы минимальный размер пакета был как можно больше. Помогите параллели S справиться с домашним заданием.

Формат входных данных

В единственной строке входных данных находятся два натуральных числа n и m ($1 \leq n \leq 10^{18}$, $2 \leq m \leq 10^{18}$) — длина сообщения и максимальный размер пакета в битах соответственно.

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

<code>send-message.in</code>	<code>send-message.out</code>
5 3	2
179 57	50

Замечание

В первом примере сообщение разбивается на 5 пакетов, в каждом из которых передаётся по одному биту сообщения.

Во втором примере одно из оптимальных разбиений на пакеты выглядит следующим образом:
 $53 = 6 + 47$, отправляется кусок сообщения длины 47, размер пакета — 53;
 $50 = 6 + 44$, отправляется кусок сообщения длины 44, размер пакета — 50;
 $50 = 6 + 44$, отправляется кусок сообщения длины 44, размер пакета — 50;
 $50 = 6 + 44$, отправляется кусок сообщения длины 44, размер пакета — 50.