

## Задача А. Ярый коллекционер бабочек

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 64 мегабайта

Как известно, Андрей Сергеевич — ярый коллекционер бабочек. Он имеет огромную коллекцию, экспонаты которой собраны со всего мира. Будем считать, что в мире существует 2 000 000 000 видов бабочек.

Чтобы не запутаться, Андрей Сергеевич присвоил каждому виду уникальный номер. Нумерация видов бабочек начинается с единицы.

Теперь он хочет знать, есть ли бабочка с видом  $K$  в его коллекции, или же её придётся добывать, затрачивая уйму сил и денег.

### Формат входных данных

В первой строке входного файла содержится единственное число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество видов бабочек в коллекции Андрея Сергеевича.

В следующей строке через пробел находятся  $N$  упорядоченных по возрастанию чисел — номера видов бабочек в коллекции.

Все виды бабочек в коллекции имеют различные номера.

В третьей строке файла записано число  $M$  ( $1 \leq M \leq 100\,000$ ) — количество видов бабочек, про которых Андрей Сергеевич хочет узнать, есть ли они у него в коллекции или же нет. В последней строке входного файла содержатся через пробел  $M$  чисел — номера видов бабочек, наличие которых необходимо проверить.

### Формат выходных данных

Выходной файл должен содержать  $M$  строчек. Для каждого запроса выведите “YES”, если бабочка с данным номером содержится в коллекции, и “NO” — в противном случае.

### Примеры

стандартный ввод	стандартный вывод
7	YES
23 27 36 45 54 72 98	YES
4	YES
54 36 23 45	YES

## Задача В. Корень кубического уравнения

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

Дано кубическое уравнение  $ax^3 + bx^2 + cx + d = 0$  ( $a \neq 0$ ). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

### Формат входных данных

Во входном файле через пробел записаны четыре целых числа:  $-1000 \leq a, b, c, d \leq 1000$ .

### Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 5 знаков после десятичной точки.

### Примеры

стандартный ввод	стандартный вывод
1 -3 3 -1	1
-1 -6 -12 -7	-1.000000

## Задача С. Сортировка выбором

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

Отсортируйте массив целых чисел в порядке неубывания, **используя сортировку выбором минимума**.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  ( $1 \leq N \leq 2500$ ), вторая строка —  $N$  целых чисел, по модулю не превышающих  $2 \cdot 10^9$ .

### Формат выходных данных

Данные числа следует вывести в порядке неубывания.

### Примеры

<code>sort.in</code>	<code>sort.out</code>
5 9 2 7 1 2	1 2 2 7 9

## Задача D. Сортировка вставками

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

Отсортируйте массив целых чисел в порядке невозрастания, **используя сортировку вставками**.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  ( $1 \leq N \leq 1000$ ), вторая строка —  $N$  целых чисел, по модулю не превышающих  $2 \cdot 10^9$ .

### Формат выходных данных

Данные числа следует вывести в порядке невозрастания.

### Примеры

<code>sort.in</code>	<code>sort.out</code>
5 9 2 7 1 2	9 7 2 2 1
5 5 4 3 2 1	5 4 3 2 1

### Замечание

В этой задаче нельзя пользоваться методом `.sort()` и функцией `sorted()`, а также использовать метод `.reverse()`

## Задача Е. Сортировка подсчетом

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

Вам дан массив. Требуется его отсортировать.

### Формат входных данных

В первой строке входного файла находится одно целое число  $n$  — количество элементов массива ( $1 \leq n \leq 200000$ ). Во второй строчке находятся  $n$  натуральных чисел — элементы массива. Все элементы массива не превосходят  $10^4$ .

### Формат выходных данных

В единственную строку выходного файла выведите отсортированный массив.

### Примеры

<code>countsort.in</code>	<code>countsort.out</code>
3 1 2 3	1 2 3
3 3 2 1	1 2 3

## Задача F. Космическое поселение

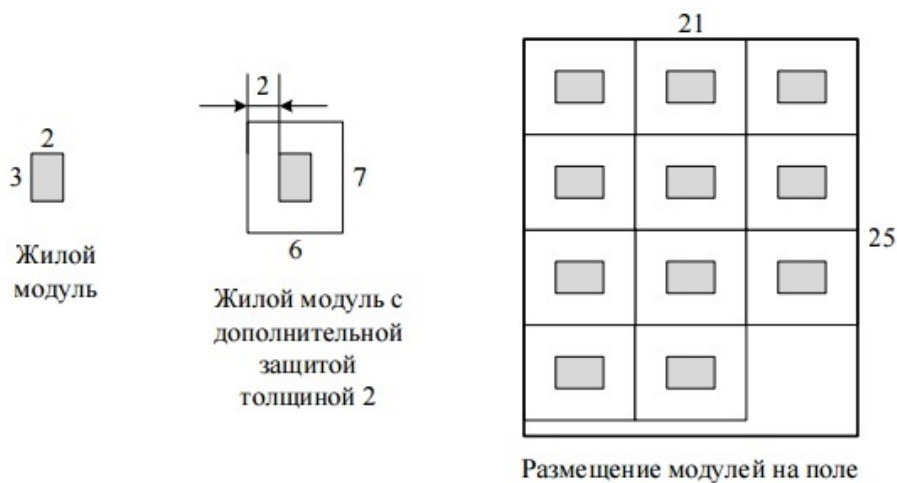
Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

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  $a \times b$  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  $w \times h$  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 G. 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 Н. Имперский марш

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

На этот раз Император нагрязнул с ревизией не в какой-то там ангар, а в казармы 501-го легиона имперских штурмовиков. В связи с этим каждого штурмовика постригли «под ежика». Несмотря на развитие нанотехнологий, постригли плохо — в результате из-за различной длины волос штурмовики могут отличаться друг от друга по росту, но незначительно — разница не превышает 137 нанометров. Ваша задача — выстроить штурмовиков по росту.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  — количество штурмовиков ( $1 \leq N \leq 100\,000$ ), вторая строка  $N$  — натуральных чисел, не превышающих  $2 \cdot 10^9$  каждое — рост штурмовика в нанометрах. Никакие два роста не различаются более, чем на 137 нм.

### Формат выходных данных

Выведите роста штурмовиков в порядке неубывания.

### Примеры

march.in	march.out
5 12 1 2 1 13	1 1 2 12 13
1 1000000000	1000000000

### Замечание

При решении этой задачи нельзя пользоваться стандартными функциями и методами `min`, `index`, `sort`, `sorted` и т. д.

Естественно, можно пользоваться функциями `min`, `max`, которые принимают два числа.



## Задача I. Вырубка леса

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

Фермер Николай нанял двух лесорубов: Дмитрия и Фёдора, чтобы вырубить лес, на месте которого должно быть кукурузное поле. В лесу растут  $X$  деревьев.

Дмитрий срубает по  $A$  деревьев в день, но каждый  $K$ -й день он отдыхает и не срубает ни одного дерева. Таким образом, Дмитрий отдыхает в  $K$ -й,  $2K$ -й,  $3K$ -й день, и т. д.

Фёдор срубает по  $B$  деревьев в день, но каждый  $M$ -й день он отдыхает и не срубает ни одного дерева. Таким образом, Фёдор отдыхает в  $M$ -й,  $2M$ -й,  $3M$ -й день, и т. д.

Лесорубы работают параллельно и, таким образом, в дни, когда никто из них не отдыхает, они срубают  $A + B$  деревьев, в дни, когда отдыхает только Фёдор —  $A$  деревьев, а в дни, когда отдыхает только Дмитрий —  $B$  деревьев. В дни, когда оба лесоруба отдыхают, ни одно дерево не срубается.

Фермер Николай хочет понять, за сколько дней лесорубы срубят все деревья, и он сможет засеять кукурузное поле.

Требуется написать программу, которая по заданным целым числам  $A$ ,  $K$ ,  $B$ ,  $M$  и  $X$  определяет, за сколько дней все деревья в лесу будут вырублены.

### Формат входных данных

Программа получает на вход пять целых чисел, разделённых пробелами:  $A$ ,  $K$ ,  $B$ ,  $M$  и  $X$  ( $1 \leq A, B \leq 10^9$ ,  $2 \leq K, M \leq 10^{18}$ ,  $1 \leq X \leq 10^{18}$ ).

### Формат выходных данных

Программа должна вывести одно целое число — искомое количество дней.

### Примеры

<code>forest.in</code>	<code>forest.out</code>
2 4 3 3 25	7