

## Задача А. На грани открытия

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

Экспериментатор Глюк нашёл на свалке старый прибор и решил исследовать его. Для этого он сделал серию измерений. Прибор оказался хорошим, а экспериментатор талантливым, поэтому все показания прибора не убывали от измерения к измерению. Но Глюк не только талантливый, но и ленивый, поэтому серию своих измерений он решил дополнить исследованием одного известного ученого из Берляндии, чьё имя слишком известно, чтобы произносить его. Берляндский ученый исследовал совершенно другой объект, но Глюку было важно, что у учёного получились тоже неубывающие значения. Теперь Глюк хочет объединить две серии, чтобы прославиться на весь мир. Как известно, он слишком ленив, поэтому поручает эту задачу вам.

### Формат входных данных

В первой строке входного файла даны два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100000$ ). В следующих двух строках находятся  $N$  и  $M$  целых чисел, по модулю не превосходящие  $10^9$  - серии Глюка и слишком известного ученого соответственно.

### Формат выходных данных

В выходной файл необходимо вывести полученную серию, между любыми двумя числами – пробел.

### Примеры

<code>breakthrough.in</code>	<code>breakthrough.out</code>
4 3	1 2 3 7 9 21 34
3 9 21 34	
1 2 7	

### Замечание

В этой задаче нельзя использовать встроенную сортировку.

## Задача В. Сортировка

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2.5 секунды  
Ограничение по памяти: 128 мегабайт

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью написанной самостоятельно сортировки слиянием.

### Формат входных данных

В первой строке входного файла содержится число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество элементов в массиве. Во второй строке находятся  $N$  целых чисел, по модулю не превосходящих  $10^9$ .

### Формат выходных данных

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

### Примеры

<code>stdin</code>	<code>stdout</code>
10 1 8 2 1 4 7 3 2 3 6	1 1 2 2 3 3 4 6 7 8

## Задача С. Быстрая сортировка

Имя входного файла: `qsort.in`  
Имя выходного файла: `qsort.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Отсортируйте данную последовательность используя алгоритм быстрой сортировки Хоара.

### Формат входных данных

В единственной строке входного файла содержится последовательность, содержащая не более чем 100000 целых чисел.

### Формат выходных данных

В единственной строке выходного файла выведите последовательность в неубывающем порядке.

### Примеры

<code>qsort.in</code>	<code>qsort.out</code>
4 1 4 8 6 6 5	1 4 4 5 6 6 8

## Задача D. Свадьбы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Одна очень предприимчивая и симпатичная девушка решила собрать себе денег на роскошную жизнь. У нее есть  $N$  поклонников, про каждого из них она узнала размер его состояния  $P_i$ . Девушка намерена выйти замуж и сразу же развестись с некоторыми из своих поклонников. По законам страны в случае развода каждый из супругов получает ровно половину их общего состояния.

Девушка хочет путем заключения браков и разводов получить со своих поклонников как можно больше денег.

### Формат входных данных

Первая строка входных данных содержит число  $N$  — количество поклонников ( $1 \leq N \leq 10^5$ ). Вторая строка содержит  $N$  целых чисел  $P_1, \dots, P_N$  — размеры состояний поклонников ( $0 \leq P_i \leq 10^6$ ). Третья строка содержит одно число  $Y$  — состояние девушки ( $0 \leq Y \leq 10^6 + 1$ ).

### Формат выходных данных

Программа должна вывести одно действительное число — максимальную сумму денег, которая может оказаться у девушки в результате махинаций.

### Примеры

стандартный ввод	стандартный вывод
2 5 10 5	7.5
3 1 3 2 0	2.125
1 1 2	2.0

## Задача E. Anti-qsort test

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

Рассмотрим алгоритм быстрой сортировки Хоара, с выбором в качестве барьерного элемента среднего элемента на отрезке ( $q = a[(l + r) / 2]$ ):

```
void qsort(vector<int> & a, int left, int right)
// Сортировка a[left...right] включительно
{
    if (right <= left)
        return;
    int q = a[(l + r) / 2];
    int i = left;
    int j = right;
    while (i <= j) {
        while (a[i] < q)
            ++i;
        while (q < a[j])
            --j;
        if (i <= j) {
            swap(a[i], a[j]);
            ++i;
            --j;
        }
    }
    qsort(a, left, j);
    qsort(a, i, right);
}
```

По данному числу  $n$  составьте тест, являющийся перестановкой чисел от 1 до  $n$ , на котором этот алгоритм выполняет наибольшее число сравнений (подсчитываются сравнения  $a[i] < q$  и  $q < a[j]$ ).

### Формат входных данных

Программа получает на вход одно целое число  $n$ ,  $1 \leq n \leq 70\,000$ .

### Формат выходных данных

Программа должна вывести перестановку чисел от 1 до  $n$ , на которой данная реализация алгоритма быстрой сортировки Хоара будет выполнять наибольшее число сравнений.

Можно вывести любой из возможных ответов.

### Примеры

стандартный ввод	стандартный вывод
3	1 3 2

## Задача F. Цифровой корень

Имя входного файла: `dig-root.in`  
Имя выходного файла: `dig-root.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 64 мегабайта

Цифровым корнем числа  $n$  называется следующее число: берется сумма цифр числа  $n$ , затем сумма цифр у получившегося числа и так далее пока не получится однозначное число.

Ваша задача — отсортировать данный массив по возрастанию цифровых корней его элементов. Если цифровые корни двух чисел равны, то раньше должно идти меньшее число.

### Формат входных данных

В первой строке файла через пробел введены элементы массива. Длина массива не превосходит 200, каждое число положительно и не превосходит  $10^9$ .

### Формат выходных данных

Массив, отсортированный в порядке возрастания цифрового корня.

### Примеры

<code>dig-root.in</code>	<code>dig-root.out</code>
15 14 13 12 11 10 9 8 7	10 11 12 13 14 15 7 8 9
80 61 51 41 22 1	1 22 41 51 61 80

### Замечание

Требуется в решении написать вспомогательную функцию `digital_root(number)`, вычисляющую и возвращающую цифровой корень числа. Эту функцию необходимо использовать в сортировке по ключу в качестве ключа.

## Задача G. Коньки

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

В ЛКШ-Зима школьники любят ходить на каток. В прокате коньков есть много коньков самых разных размеров. Школьник может надеть коньки любого размера, который не меньше размера его ноги. Известны размеры всех коньков и размеры ног школьников. Определите, какое наибольшее число школьников сможет одновременно пойти покататься.

### Формат входных данных

Первая строка входных данных содержит число  $N$  — количество коньков в прокате ( $1 \leq N \leq 10^5$ ). Во второй строке записано  $N$  чисел — размеры коньков. В третьей строке содержится число  $M$  — количество школьников в ЛКШ ( $1 \leq M \leq 10^5$ ), четвертая строка содержит размеры их ног. Размеры коньков и ног — натуральные числа, не превосходящие 100.

### Формат выходных данных

Выведите единственное число — наибольшее количество школьников, которое сможет пойти на каток.

### Примеры

<code>skates.in</code>	<code>skates.out</code>
4 41 40 39 42 3 42 41 42	2

## Задача Н. Проблема свободного места

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Темная ночь. Шумят волны, начинает капать дождь. Челкаш после очередного заплыва приплыл к контрабандистам, чтобы сбыть награбленный товар. Однако не все пошло по плану: выяснилось, что у контрабандистов на корабле уже фактически нет места. Задачу о рюкзаке контрабандисты не знают, и потому они требуют, чтобы Челкаш отдавал им товар в порядке неубывания веса. Челкаш уже очень устал и ничего не соображает, и потому эта задача кажется ему непосильной.

К его счастью, в этот раз он приплыл не один, а вместе с Гаврилой. Но запуганный Гаврила сидит в углу лодки и еле может пошевелить руками, и потому отсортировать товар он также не может. Поэтому Челкаш решает поступить так: он выкладывает весь товар в ряд и  $q$  раз меняет какие-то два элемента ряда местами. После этого он бросается на Гаврилу и требует, что бы он сказал, является ряд отсортированным по неубыванию или нет. Как уже было сказано, Челкаш очень устал, и даже если в какой-то момент товар оказался отсортированным, Челкаш может продолжить задавать вопросы.

В темноте Гаврила видит очень плохо, и потому он не может помочь своему товарищу. Но в то же время он понимает, что если он ошибётся, то на следующий день, когда Челкаш узнает об ошибке, Гавриле несдобровать. Гаврила просит Вас помочь ему справиться со столь непростой задачей.

### Формат входных данных

В первой строке входных данных дано два целых числа  $n$  и  $q$  ( $2 \leq n \leq 10^5$ ,  $1 \leq q \leq 10^5$ ) — количество товара и запросов соответственно. Во второй строке дано  $n$  натуральных чисел, не превосходящих  $10^9$  — веса товаров. Далее следуют  $q$  строк, каждая из которых содержит два различных целых числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ), означающих, что Челкаш поменял элементы ряда с позициями  $a$  и  $b$  местами.

### Формат выходных данных

Выведите  $q$  строк, являющимися ответами на запросы Челкаша. Если после изменения позиций элементов ряд стал отсортированным по неубыванию, выведите "Sorted!". В противном случае выведите "Unsorted!". Кавычки выводить не нужно.

### Примеры

стандартный ввод	стандартный вывод
5 4 1 2 5 3 4 3 4 4 5 1 5 5 1	Unsorted! Sorted! Unsorted! Sorted!
2 3 2 10 1 2 1 2 1 2	Unsorted! Sorted! Unsorted!
6 3 89 76 35 21 11 4 1 6 2 5 3 4	Unsorted! Unsorted! Sorted!

### Замечание

Рассмотрим первый пример из условия:

- После первого запроса ряд выглядит следующим образом: 1, 2, 3, 5, 4. Поскольку он не является отсортированным, ответ на запрос — "Unsorted!"
- После второго запроса ряд становится отсортированным, поэтому ответ — "Sorted!"
- После третьего запроса Челкаш нарушает порядок в отсортированном массиве, и он становится несортированным, поэтому ответ — "Unsorted!"
- После последнего запроса Челкаш восстанавливает отсортированный массив, и потому ответ на запрос — "Sorted!"