

## 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 (встроенную сортировку использовать запрещено).

### 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

### Замечание

Реализуйте в этой задаче `Qsort`.

## Задача С. Количество инверсий

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайта

Напишите программу, которая для заданного массива  $A = \langle a_1, a_2, \dots, a_n \rangle$  находит количество пар  $(i, j)$  таких, что  $i < j$  и  $a_i > a_j$ . Обратите внимание на то, что ответ может не влезать в `int`.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество элементов массива. Вторая строка содержит  $n$  попарно различных элементов массива  $A$  — целых неотрицательных чисел, не превосходящих  $10^9$ .

### Формат выходных данных

В выходной файл выведите одно число — количество инверсий в массиве.

### Примеры

<code>stdin</code>	<code>stdout</code>
5 6 11 18 28 31	0
8 999994 999989 999982 999972 999969 999961 999954 999950	28

## Задача D. 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 ( $q = A[(1 + r) / 2]$ ):

```
void qsort(vector<int> & a, int left, int right)
// Сортировка A[left...right] включительно
{
    if (right <= left)
        return;
    int q = A[(1 + 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 Е. 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 F. Кассы

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 64 мегабайта

На одном из московских вокзалов билеты продают  $N$  касс. Каждая касса работает без перерыва определенный промежуток времени по фиксированному расписанию (одному и тому же каждый день). Требуется определить, на протяжении какого времени в течение суток работают все кассы одновременно.

### Формат входных данных

Сначала вводится одно целое число  $N$  ( $0 < N \leq 10000$ ).

В каждой из следующих  $N$  строк через пробел расположены 6 целых чисел, первые три из которых обозначают время открытия кассы в часах, минутах и секундах (часы — целое число от 0 до 23, минуты и секунды — целые числа от 0 до 59), оставшиеся три — время закрытия в том же формате. Числа разделены пробелами.

Время открытия означает, что в соответствующую ему секунду касса уже работает, а время закрытия — что в соответствующую секунду касса уже не работает. Например, касса, открытая с 10 ч 30 мин 30 с до 10 ч 35 мин 30 с, ежедневно работает 300 секунд.

Если время открытия совпадает с временем закрытия, то касса работает круглосуточно. Если первое время больше второго, то касса начинает работу до полуночи, а заканчивает — на следующий день.

### Формат выходных данных

Требуется вывести одно число — суммарное время за сутки (в секундах), на протяжении которого работают все  $N$  касс.

### Примеры

<code>stdin</code>	<code>stdout</code>
3 1 0 0 23 0 0 12 0 0 12 0 0 22 0 0 2 0 0	7200
2 9 30 0 14 0 0 14 15 0 21 0 0	0
2 14 0 0 18 0 0 10 0 0 14 0 1	1

## Задача G. Дорешивание

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Как известно, после обеда в ЛКШ проходит много интересных мероприятий, но все равно каждый ЛКШОнок старается хотя бы ненадолго заглянуть в комповник, чтобы дорешать задачи, не сделанные во время практики.

В этом году погода стоит особо жаркая, поэтому в комповнике очень душно и важно следить за тем, чтобы в комповнике не находилось одновременно очень много школьников. Поэтому завуч записал время прихода и ухода из комповника каждого ЛКШОнка.

Теперь завуч хочет узнать, сколько ЛКШАт встретил в комповнике каждый ЛКШОнок.

### Формат входных данных

В первой строке записано количество ЛКШАт  $N$  ( $1 \leq N \leq 10^5$ ). В  $i$ -й из следующих  $N$  строк через пробел записаны целые числа  $S_i$  и  $T_i$  ( $0 \leq S_i \leq T_i \leq 10^9$ ) — время прихода в комповник и ухода из него  $i$ -го ЛКШОнка.

### Формат выходных данных

Программа должна вывести  $N$  целых чисел,  $i$ -е число должно быть равно количеству ЛКШАт, которых встретил в комповнике  $i$ -й ЛКШОнок.

Если в некоторый момент времени один ЛКШОнок приходит в комповник, а другой уходит из него, то они встречаются друг с другом.

### Примеры

<code>stdin</code>	<code>stdout</code>
4	3
1 10	3
2 5	2
5 6	2
1 4	

### Замечание

Ограничение времени работы для языка Python — 4 секунды.

## Задача Н. Точки и отрезки

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

Дано  $n$  отрезков на числовой прямой и  $m$  точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка  $x$  считается принадлежащей отрезку с концами  $a$  и  $b$ , если выполняется двойное неравенство  $\min(a, b) \leq x \leq \max(a, b)$ .

### Формат входных данных

Первая строка содержит два целых числа  $n$  ( $1 \leq n \leq 10^5$ ) — число отрезков и  $m$  ( $1 \leq m \leq 10^5$ ) — число точек. В следующих  $n$  строках записаны по два целых числа  $a_i$  и  $b_i$  — координаты концов соответствующего отрезка. В последней строке записаны  $m$  целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю  $10^9$ .

### Формат выходных данных

В выходной файл выведите  $m$  чисел — для каждой точки выведите количество отрезков, в которых она содержится.

### Примеры

<code>segments.in</code>	<code>segments.out</code>
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1