

## Задача А. Сортировка подсчетом

Имя входного файла: `countsort.in`  
 Имя выходного файла: `countsort.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 4 мегабайта

Вам дан массив. Требуется его отсортировать.

### Формат входного файла

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 выходного файла

В единственную строку выходного файла выведите отсортированный массив.

### Примеры

countsort.in	countsort.out
3 1 2 3	1 2 3
3 3 2 1	1 2 3

### Note

Используйте силу, силу сортировки подсчетом.

## Задача В. Хипуй!

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

В этой задаче вам необходимо организовать структуру данных **Heap** для хранения целых чисел, над которой определены следующие операции:

- **Insert( $X$ )** — добавить в **Heap** число  $X$ ;
- **Extract** — достать из **Heap** наибольшее число (удалив его при этом).

### Формат входного файла

Во входном файле записано количество команд  $N$  ( $1 \leq N \leq 100000$ ), потом последовательность из  $N$  команд, каждая в своей строке.

Каждая команда имеет такой формат: „0 <число>“ или „1“, что означает соответственно операции **Insert**(<число>) и **Extract**. Добавляемые числа находятся в интервале от 1 до  $10^7$  включительно.

Гарантируется, что при выполнении команды **Extract** в структуре находится по крайней мере один элемент.

## Формат выходного файла

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды **Extract**.

### Примеры

heap.in	heap.out
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

## Задача С. Количество различных элементов в массиве

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

Дан массив целых чисел. Определите, сколько в нем содержится различных элементов.

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ). Вторая строка содержит  $n$  целых чисел, по модулю не превосходящих  $2 \cdot 10^9$ .

### Формат выходного файла

Выведите единственное число — количество различных элементов в данном массиве.

### Примеры

differ.in	differ.out
5 1 0 1 2 0	3
5 1 5 2 3 4	5
5 0 0 0 0 0	1

## Задача D. Коммерческий калькулятор

Имя входного файла: `stdin`  
 Имя выходного файла: `stdout`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

Фирма OISAC выпустила новую версию калькулятора. Этот калькулятор берет с поль-

зователя деньги за совершаемые арифметические операции. Стоимость каждой операции в долларах равна 5% от числа, которое является результатом операции.

На этом калькуляторе требуется вычислить сумму  $N$  натуральных чисел (числа известны). Нетрудно заметить, что от того, в каком порядке мы будем складывать эти числа, иногда зависит, в какую сумму денег нам обойдется вычисление суммы чисел (тем самым, оказывается нарушен классический принцип *от перестановки мест слагаемых сумма не меняется* :-)).

Например, пусть нам нужно сложить числа 10, 11, 12 и 13. Тогда если мы сначала сложим 10 и 11 (это обойдется нам в \$1.05), потом результат - с 12 (\$1.65), и затем - с 13 (\$2.3), то всего мы заплатим \$5, если же сначала отдельно сложить 10 и 11 (\$1.05), потом - 12 и 13 (\$1.25) и, наконец, сложить между собой два полученных числа (\$2.3), то в итоге мы заплатим лишь \$4.6.

Напишите программу, которая будет определять, за какую минимальную сумму денег можно найти сумму данных  $N$  чисел.

### Формат входного файла

Во входном файле записано число  $N$  ( $2 \leq N \leq 100000$ ). Далее идет  $N$  натуральных чисел, которые нужно сложить, каждое из них не превышает 10000.

### Формат выходного файла

В выходной файл выведите, сколько денег нам потребуется на нахождение суммы этих  $N$  чисел с точностью не менее  $10^{-6}$ .

### Примеры

stdin	stdout
4 10 11 12 13	4.600000
2 1 1	0.100000

## Задача E. Медиана объединений

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

Дано  $N$  упорядоченных по неубыванию последовательностей целых чисел (т.е. каждый следующий элемент больше либо равен предыдущему), в каждой из последовательностей ровно  $L$  элементов. Для каждой двух последовательностей выполняют следующую операцию: объединяют их элементы (в объединенной последовательности каждое число будет идти столько раз, сколько раз оно встречалось суммарно в объединяемых последовательностях), упорядочивают их по неубыванию и смотрят, какой элемент в этой последовательности из  $2L$  элементов окажется на месте номер  $L$  (этот элемент называют левой медианой).

Напишите программу, которая для каждой пары последовательностей выведет левую медиану их объединения.

### Формат входного файла

Сначала вводятся числа  $N$  и  $L$  ( $2 \leq N \leq 200$ ,  $1 \leq L \leq 50000$ ). В следующих  $N$  строках задаются параметры, определяющие последовательности.

Каждая последовательность определяется пятью целочисленными параметрами:  $x_1$ ,  $d_1$ ,  $a$ ,  $c$ ,  $m$ . Элементы последовательности вычисляются по следующим формулам:  $x_1$  нам задано, а для всех  $i$  от 2 до  $L$ :  $x_i = x_{i-1} + d_{i-1}$ . Последовательность  $d_i$  определяется следующим образом:  $d_1$  нам задано, а для  $i \geq 2$   $d_i = ((a \cdot d_{i-1} + c) \bmod m)$ , где  $\bmod$  - операция получения остатка от деления  $(a \cdot d_{i-1} + c)$  на  $m$ .

Для всех последовательностей выполнены следующие ограничения:  $1 \leq m \leq 40000$ ,  $0 \leq a < m$ ,  $0 \leq c < m$ ,  $0 \leq d_1 < m$ . Гарантируется, что все члены всех последовательностей по модулю не превышают  $10^9$ .

### Формат выходного файла

В первой строке выведите медиану объединения 1-й и 2-й последовательностей, во второй строке — объединения 1-й и 3-й, и так далее, в  $(N - 1)$ -ой строке — объединения 1-й и  $N$ -ой последовательностей, далее медиану объединения 2-й и 3-й, 2-й и 4-й, и т.д. до 2-й и  $N$ -ой, затем 3-й и 4-й и так далее. В последней строке должна быть выведена медиана объединения  $(N - 1)$ -й и  $N$ -ой последовательностей.

### Примеры

mergemedian.in	mergemedian.out
3 6	7
1 3 1 0 5	10
0 2 1 1 100	9
1 6 8 5 11	

### Note

Метод слияния массивов в этой задаче не поможет. Догадайтесь почему ;)