

Задача А. Реверс

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

Переверните массив чисел. Без циклов. Без списков.

Формат входных данных

В первой строке — число N . Далее в N строках указаны N чисел ($1 \leq N \leq 10000$), по одному числу на строке. Все числа по модулю не превышают 10^5 .

Формат выходных данных

Выведите N чисел — перевернутый массив.

Примеры

<code>reverse.in</code>	<code>reverse.out</code>
4 92846 96841 33582 25998	25998 33582 96841 92846

Задача В. Ханойские башни

Имя входного файла: `hanoi.in`
Имя выходного файла: `hanoi.out`
Ограничение по времени: 6 секунды
Ограничение по памяти: 64 мегабайта

Даны три стержня. На первом стержне находятся несколько дисков сверху вниз по возрастанию размера диска. Два другие пустые. Требуется перенести все диски с первого стержня на второй. Переносить диски разрешается только по одному. Не разрешается класть больший диск на меньший.

Формат входных данных

Вводится одно число n ($1 \leq n \leq 19$) — количество дисков на первом стержне.

Формат выходных данных

Выведите по два числа в строке — номера стержней, откуда и куда переносится диск. Решение должно быть оптимальным по количеству действий.

Примеры

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

Замечание

В программе нельзя пользоваться циклами.

Задача С. Цифровой корень

Имя входного файла: `dig-root.in`
Имя выходного файла: `dig-root.out`
Ограничение по времени: 2 секунды
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

При решении задачи запрещается пользоваться встроенными функциями `sort` и `sorted`, а также теми квадратичными сортировками, которые вы уже написали в предыдущий день. Кроме того, требуется в решении написать вспомогательную функцию `digital-root(number)`, вычисляющую и возвращающую цифровой корень числа. Эту функцию необходимо использовать.

Задача D. Возведение в степень

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

Формат входных данных

Во входном файле даны три натуральных числа A, B, M ($1 \leq A, B \leq 10^9, 2 \leq M \leq 10^9$), записанные на одной строке через пробел.

Формат выходных данных

В выходной файл выведите одно число, равное $A^B \bmod M$ (`mod` означает взятие остатка при делении).

Примеры

<code>power.in</code>	<code>power.out</code>
2 3 100	8