

Задача А. Реверс

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

Проверните массив чисел. Без циклов. Без списков.

Формат входного файла

В первой строке — число N . Далее в N строках указаны N чисел ($1 \leq N \leq 900$), по одному числу на строке. Все числа по модулю не превышают 10^5 .

Формат выходного файла

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. В программе нельзя пользоваться циклами.

Формат входного файла

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

Формат выходного файла

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

Задача С. Возведение в степень

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

Формат входного файла

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 выходного файла

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

Примеры

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

Задача D. 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 входного файла

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

Формат выходного файла

Массив, отсортированный в порядке возрастания цифрового корня.

Примеры

dig-root.in	dig-root.out
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

Примечание

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 Е. Жребий Крижановского

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

Петя играет с друзьями в игру, которую иногда называют «Жребий Крижановского». Правила игры следующие: в каждом туре каждый игрок загадывает произвольное натуральное число. После этого игрок, загадавший минимальное число, которое не повторяется, выигрывает в этом туре, причем его выигрыш равен этому числу. Например, если играют 6 человек и были загаданы числа 3, 2, 1, 1, 4 и 2, то выиграл первый игрок, причем его выигрыш равен 3. Если все загаданные числа повторяются, то тур считается ничейным и никто баллов не получает.

Общий выигрыш игрока за игру равен сумме баллов за все сыгранные туры.

Петя с друзьями при игре просто называют по очереди загаданные ими числа, а потом определяют, кто выиграл, и подсчитывают баллы. Однако при таком формате игры в принципе можно жульничать, не загадывая число заранее, а, уже зная числа, названные предыдущими игроками, выбрать себе оптимальное «загаданное» число. Этим и пользуется Петя. Он называет число последним и старается выбрать число так, чтобы максимизировать свой выигрыш.

Идет последний тур игры. Известны очки всех игроков перед этим туром и названные игроками числа. Выясните, какое число следует назвать Пете, чтобы по результатам игры у как можно большего числа игроков количество баллов было меньше, чем у него. Если таких чисел несколько, то Петя хочет назвать минимальное возможное.

Формат входного файла

В первой строке вводится число n — количество игроков ($2 \leq n \leq 100$). Вторая строка содержит n чисел — баллы игроков перед последним туром (неотрицательные целые числа, не большие 100). Баллы перечислены в том порядке, в котором игроки обычно

называют числа (то есть Петины баллы указаны последними). В третьей строке задано $(n - 1)$ число — числа, названные игроками в последнем туре (числа не превышают 100), в том порядке, в котором они их называли.

Формат выходного файла

Выведите число, которое следует назвать Пете.

Примеры

gamble.in	gamble.out
6 0 0 0 0 0 0 2 3 4 5 6	1
6 8 3 12 5 0 9 2 1 3 1 4	2

Примечание

Во втором примере Петя не может выиграть в последнем туре. Однако, назвав число 2, Петя не позволяет выиграть первому игроку, и, тем самым, остается вторым по итогам всей игры. У четырех игроков баллы меньше, чем у Пети.