

Задача А. Скейтборд

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

Начинающий скейтбордист Дэн хочет прокатиться на своём скейте. К несчастью, выбранная Дэном дорога отнюдь не ровная, а сам он, в силу неопытности, не умеет преодолевать бугры — на пике такого он обязательно падает и разбивает колени. Зная это, его подружка Кэт, вздыхая, запасается зелёной и пытается отговорить Дэна от дальнейших подвигов во имя её, ибо не трудно догадаться, для чего он вообще затеял эту прогулку ;) Но упрямый Дэн не желает сдаваться и собирается проехать всю дорогу. Кэт интересно, сколько же раз за время этой злополучной прогулки ей придётся работать скорой медицинской помощью.

Формат входного файла

В первой строке дано число N — число изломов дороги ($3 \leq N \leq 1000$). В следующей строке N чисел a_i ($-1000 \leq a_i \leq 1000$), содержащих y -координаты изломов слева направо. Дорога между двумя соседними изломами представляет собой отрезок, соединяющий их. Никакие два соседних числа не равны между собой ($a_1 < a_2$, $a_{N-1} > a_N$)

Формат выходного файла

Выведите одно число — количество падений Дэна.

Примеры

<code>stdin</code>	<code>stdout</code>
5 -1 2 7 1 0	1

Задача В. Калькулятор

Имя входного файла: `calculator.in`
Имя выходного файла: `calculator.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Имеется калькулятор, который выполняет три операции:

- Прибавить к числу X единицу.
- Умножить число X на 2.
- Умножить число X на 3.

Определите, какое наименьшее число операций необходимо для того, чтобы получить из числа 1 заданное число N .

Формат входного файла

Программа получает на вход одно число, не превосходящее 10^6 .

Формат выходного файла

Одно число — наименьшее количество искомых операций.

Примеры

<code>calculator.in</code>	<code>calculator.out</code>
1	0
5	3

Задача С. Кафе «Хоботания»

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

В городе Е. открылось новое кафе «Хоботания», рассчитанное на слоников. Все клиенты приходят в кафе в момент времени 0, а владелец выбирает в каком порядке их обслуживать. При этом, каждую секунду обслуживается один слоник (первый обслуживается в момент времени 0).

Владельцу известно, что если слоник i будет обслужен в момент времени t , то он заплатит $tips_i - t$ чаевых. Если число $tips_i - t$ отрицательно, то он ничего не платит.

Помогите владельцу найти такой порядок обслуживания слонов, который принесет ему максимальную прибыль.

Формат входного файла

Первая строка входного файла содержит целое число n ($0 \leq n \leq 100$) — количество слоников, пришедших в кафе. Следующая строка содержит n чисел $tips_1, tips_2 \dots tips_n$ ($0 \leq tips_i \leq 10^5$).

Формат выходного файла

В выходной файл выведите максимальную выручку владельца кафе.

Примеры

<code>i.in</code>	<code>i.out</code>
3 3 2 3	5

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

Note

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

Note

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.