

Задача А. Сортировка

Имя входного файла: `sort.in`
Имя выходного файла: `sort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

ан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 100000$) — количество элементов в массиве. Во второй строке находятся n целых чисел, по модулю не превосходящих 10^9 .

Формат выходного файла

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Пример

sort.in	sort.out
10 1 8 2 1 4 7 3 2 3 6	1 1 2 2 3 3 4 6 7 8

Задача В. Мерлин

Имя входного файла: `merlin.in`
Имя выходного файла: `merlin.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Однажды, вернувшись в свою башню, Мерлин обнаружил, что Моргана наложила проклятие на все его сосуды с эликсиром мудрости.

Мерлин знает, как снять проклятие, но соответствующее заклинание требует, чтобы во всех сосудах, к которым оно применяется, было равное количество эликсира.

Чтобы добиться этого, Мерлин решил действовать следующим образом. Он выбирает несколько сосудов и переливает весь эликсир из выбранных сосудов в оставшиеся. Он может распределить переливаемый эликсир между оставшимися сосудами произвольным образом. После того, как весь эликсир из выбранных сосудов перелит, Мерлин разбивает опустошенные сосуды (с них проклятие уже не снять), выбрасывает осколки и применяет заклинание снятия проклятия к оставшимся сосудам.

Помогите волшебнику узнать, какое наименьшее количество сосудов ему придется разбить, чтобы снять проклятие Морганы.

Формат входного файла

В первой строке входного файла находится число n ($2 \leq n \leq 10^5$) — количество сосудов. Во второй строке содержатся n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количество литров эликсира мудрости в каждом сосуде.

Формат выходного файла

Выведите в выходной файл минимальное количество сосудов, которые Мерлину придется разбить.

Примеры

<code>merlin.in</code>	<code>merlin.out</code>
3 2 3 2	1
4 4 4 4 4	0
5 1 2 3 4 5	2

В первом примере можно, например, перелить 0.5 литра эликсира из первого сосуда во второй и 1.5 литра в третий, после чего разбить первый сосуд.

Во втором сосудах исходно содержат равное количество эликсира, можно ничего не переливать.

В третьем примере можно, например, перелить 1 литр эликсира из первого сосуда во второй, по 2 литра из пятого во второй и третий, 1 литр из пятого в четвертый, после чего разбить первый и пятый сосуды.

Задача С. Призы

Имя входного файла: `prizes.in`
Имя выходного файла: `prizes.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Алиса и Боб стали победителями телевикторины, и теперь им предстоит выбрать себе призы. На выбор предлагается n призов, пронумерованных от 1 до n .

Распределение призов происходит следующим образом. Организаторы телевикторины сообщают победителям целое положительное число k ($1 \leq k \leq \frac{n}{3}$). Сначала Алиса выбирает себе любые k подряд идущих номеров призов. Потом Боб выбирает себе k подряд идущих номеров призов, при этом он не может выбирать номера, которые уже выбрала Алиса. После этого победители забирают выбранные ими призы.

Алиса хорошо знает Боба, и для каждого приза выяснила его ценность для Боба, которая является целым положительным числом. Алиса обижена на Боба и хочет выбрать свои призы так, чтобы суммарная ценность призов, которые достанутся Бобу, была как можно меньше. При этом Алису не волнует, какие призы достанутся ей.

Требуется написать программу, которая по информации о ценности призов и значению k определит, для какого минимального значения x Алиса сможет добиться того, чтобы Боб не смог выбрать призы с суммарной ценностью больше x .

Формат входного файла

Первая строка входного файла содержит два целых числа: n — общее количество призов и k — количество подряд идущих номеров призов, которое должен выбрать каждый из победителей ($3 \leq n \leq 100\,000, 1 \leq k \leq \frac{n}{3}$).

Вторая строка содержит n целых положительных чисел: a_1, a_2, \dots, a_n . Для каждого приза указана его ценность для Боба ($1 \leq a_i \leq 10^9$).

Формат выходного файла

Выходной файл должен содержать одно число — минимальное значение x , для которого Алиса сможет добиться того, чтобы Боб не смог выбрать призы с суммарной ценностью больше x .

Пример

<code>prizes.in</code>	<code>prizes.out</code>
10 2 1 2 4 5 2 4 2 2 1 6	7

Задача D. Тестирующая система

Имя входного файла: `ejudge.in`
Имя выходного файла: `ejudge.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный контест.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам ACM:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;
- при прочих равных — по номеру команды в порядке возрастания.

Формат входного файла

Первая строка содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество команд, участвующих в контесте. В i -й из следующих n строк записано количество решённых задач S ($1 \leq S \leq 100$) и штрафное время T ($1 \leq T \leq 100\,000$) команды с номером i .

Формат выходного файла

В выходной файл выведите n чисел — номера команд в отсортированном порядке.

Пример

<code>ejudge.in</code>	<code>ejudge.out</code>
5	5 2 1 3 4
3 50	
5 720	
1 7	
0 0	
8 500	

Задача Е. Двоичный поиск

Имя входного файла: `binsearch.in`
Имя выходного файла: `binsearch.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

ан массив из n элементов, упорядоченный в порядке неубывания и m запросов: найти первое и последнее вхождение числа в массив.

Формат входного файла

В первую строке входного файла содержится одно число n — размер массива. ($1 \leq n \leq 100000$). Во второй строке находится n чисел в порядке неубывания — элементы массива. В третьей строке находится число m — количество запросов. В следующей строке находится m чисел — запросы.

Формат выходного файла

Для каждого запроса выведите в отдельной строке номер первого и последнего вхождения этого числа в массив. Если числа в массиве нет выведите два раза -1.

Пример

<code>binsearch.in</code>	<code>binsearch.out</code>
5	1 2
1 1 2 2 2	3 5
3	-1 -1
1 2 3	

Задача F. Город Че

Имя входного файла: `che.in`
Имя выходного файла: `che.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В центре города Че есть пешеходная улица — одно из самых популярных мест для прогулок жителей города. По этой улице очень приятно гулять, ведь вдоль улицы расположено n забавных памятников.

Девочке Маше из города Че нравятся два мальчика из ее школы, и она никак не может сделать выбор между ними. Чтобы принять окончательное решение, она решила назначить обоим мальчикам свидание в одно и то же время. Маша хочет выбрать два памятника на пешеходной улице, около которых мальчики будут ее ждать. При этом она хочет выбрать такие памятники, чтобы мальчики не увидели друг друга. Маша знает, что из-за тумана мальчики увидят друг друга только, если они будут на расстоянии не более r метров.

Маше заинтересовалась, а сколько способов есть выбрать два различных памятника для организации свиданий.

Формат входного файла

В первой строке входного файла находятся два целых числа: n и r ($2 \leq n \leq 300\,000$, $1 \leq r \leq 10^9$), количество памятников и максимальное расстояние, на котором мальчики могут увидеть друг друга.

Во второй строке задано n положительных чисел d_1, \dots, d_n , где d_i — расстояние от i -го памятника до начала улицы. Все памятники находятся на разном расстоянии от начала улицы. Памятники приведены в порядке возрастания расстояния от начала улицы ($1 < d_1 < d_2 < \dots < d_n < 10^9$).

Формат выходного файла

Выведите одно число — число способов выбрать два памятника для организации свиданий.

Пример

<code>che.in</code>	<code>che.out</code>
4 4 1 3 5 8	2

Комментарий

В приведенном примере Маша может выбрать памятники 1 и 4 или памятники 2 и 4.

Задача Г. Гирлянда

Имя входного файла: `garland.in`
Имя выходного файла: `garland.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Гирлянда состоит из n лампочек на общем проводе. Один её конец закреплён на заданной высоте A мм ($h_1 = A$). Благодаря силе тяжести гирлянда прогибается: высота каждой неконцевой лампы на 1 мм меньше, чем средняя высота ближайших соседей ($h_i = \frac{(h_{i-1} + h_{i+1})}{2} - 1$ для $1 < i < N$). Требуется найти минимальную высоту второго конца B ($B = h_n$) при условии, что ни одна из лампочек не должна лежать на земле ($h_i > 0$ для $1 \leq i \leq N$).

Формат входного файла

В первую строке входного файла содержится два числа n и A ($3 \leq n \leq 1000$, n — целое, $10 \leq A \leq 1000$, A — вещественное).

Формат выходного файла

Вывести одно вещественное число B с двумя знаками после запятой.

Пример

garland.in	garland.out
8 15	9.75
692 532.81	446113.34

Задача Н. Арсенал

Имя входного файла: `arsenal.in`
Имя выходного файла: `arsenal.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во время очередной тренировки к играм Китнисс решила потренироваться в стрельбе из лука.

В тренировочном зале есть n стрел разной длины, которые находятся в стойках рядом друг с другом вдоль стены. Китнисс хочет выстрелить все стрелы, при этом после каждого выстрела ей придется брать новую стрелу из стойки, а взятые стрелы никогда не возвращаются обратно. Китнисс считает, что стрелу удобно взять, если рядом с ней с обеих сторон стоят стрелы, размеры которых меньше. Отсутствие стрелы с одной из сторон равносильно присутствию стрелы с длиной ноль.

Чтобы процесс тренировки прошел наиболее эффективно, Китнисс хочет взять как можно меньше неудобных стрел. Но так как, ей нужно тренироваться, она просит вас ей помочь подобрать порядок выбора стрел из стоек, чтобы количество неудобно взятых стрел было наименьшим возможным.

Формат входного файла

В первой строке входного файла дано одно натуральное число n ($1 \leq n \leq 10^5$) — количество стоек со стрелами.

Во второй строке дано n чисел a_i ($1 \leq a_i \leq 1000$) — высота стрел.

Формат выходного файла

В первой строке выходного файла выведите наименьшее количество стрел, которые придется неудобно взять.

Во второй строке через пробел выведите n чисел — номера стрел в порядке, в котором Китнисс будет их брать. Если способов взять стрелы несколько, выведите любой.

Пример

<code>arsenal.in</code>	<code>arsenal.out</code>
4 1 2 3 4	0 4 3 2 1
4 1 2 1 1	1 2 1 3 4
6 1 1 2 2 3 3	3 5 6 3 4 1 2