

Задача А. Конфетки

Имя входного файла: `candies.in`
Имя выходного файла: `candies.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

У Кролика день рождения! Он пригласил в гости n гостей. Чтобы гостям не было грустно и скучно, Кролик купил n коробок конфет. Кролик любит разнообразие, поэтому конфеты были разные. В i -й коробке лежало a_i конфет.

В назначенный день с самого утра к Кролику начали приходить гости. Каждый гость характеризуется своей наглостью b_i . Это означает, что, зайдя домой к Кролику и увидев коробки конфет, он брал из каждой коробки, в которой не меньше, чем b_i , конфет, по одной и съедал её. Например, у Винни-Пуха вполне могла быть наглость один. Это значит, что он бы съел по конфете из каждой коробки.

Вечером, когда гости разошлись, Кролику стало интересно, кто съел сколько конфет. Помогите ему определить это.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — количество коробок конфет. В следующей строке задано n натуральных чисел a_i ($1 \leq a_i \leq 10^9$) — сколько конфет в каждой коробке.

Далее, в следующей строке задано число m ($1 \leq m \leq 100\,000$) — количество гостей. В четвёртой и последней строке задано m чисел b_i ($1 \leq b_i \leq 10^9$) — наглости гостей.

Формат выходных данных

В выходной файл выведите n строк, i -ая из которых должна содержать количество конфет съеденных i -ым гостем.

Примеры

candies.in	candies.out
3	3
3 1 1	1
2	
1 1	

Задача В. Выборы

Имя входного файла: `elections.in`
Имя выходного файла: `elections.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Недавно в стране прошли выборы, при проведении которых были использованы современные технологии: про каждый бюллетень известно, когда он был положен в урну и кандидат, за которого проголосовал человек, опустивший в урну этот бюллетень. Бюллетени пронумерованы в хронологическом порядке. После подсчета голосов все, кому не лень, захотели проанализировать ход выборов. Ведь не зря же вводилась вся эта система?

Многим стала интересна следующая информация: «сколько голосов было отдано за заданный промежуток времени за кандидата, выигрывавшего на этом промежутке?». Ведь, если за какого-то кандидата в какой-то небольшой промежуток времени голосовали активно, а в остальное — менее активно, то, возможно, в этот промежуток времени была совершена фальсификация. Для ответа на эти запросы даже был создан специальный сайт, на котором каждый посетитель, которому небезразлична судьба страны, мог узнать интересующую его информацию. Однако, количество посетителей оказалось неожиданно большим, и сайт перестал справляться с нагрузками.

Руководством решено было переписать все так, чтобы обслуживать одновременно m пользователей. Вам поручено реализовать программу, отвечающую на вопрос, который ставят пользователи: «Сколько голосов за определенный промежуток времени набрал кандидат, который набрал за этот промежуток больше, чем все остальные?».

Формат входных данных

В первой строке задано количество обработанных бюллетеней n ($1 \leq n \leq 100\,000$). Во второй строке через пробел заданы n целых чисел a_i ($1 \leq a_i \leq 10^9$) — номер кандидата, за которого проголосовал человек, опустивший в урну бюллетень номер i . В третьей строке задано число m ($1 \leq m \leq 100\,000$) — количество запросов. В следующих m строках заданы сами запросы в формате $l_i r_i$ ($1 \leq l_i \leq r_i \leq n$) — наименьший и наибольший номера бюллетеня, которые следует рассматривать при обработке этого запроса.

Формат выходных данных

Для каждого запроса в отдельной строке выведите одно число — количество бюллетеней с номерами не меньше l_i и не больше r_i , голоса на которых отданы самому популярному кандидату на этом отрезке. Самым популярным называется любой такой кандидат, за которого среди бюллетеней с подходящими номерами отдано количество голосов не меньшее, чем за любого другого кандидата среди этих же бюллетеней.

Примеры

elections.in	elections.out
6	4
1 1 2 2 1 1	2
3	2
1 6	
2 4	
2 5	

Примечание

Примечание

Первая группа тестов проверяется в момент сдачи задачи на проверку и стоит 20 баллов. Баллы за эту группу начисляются только при прохождении всех тестов группы. Для всех тестов этой группы выполнено условие $n \leq 100$ и $m \leq 100$.

Вторая группа тестов проверяется в момент сдачи задачи на проверку и стоит 20 баллов. Баллы за эту группу начисляются только при прохождении всех тестов группы. Для всех тестов этой группы выполнено условие $n \leq 3\,000$ и $m \leq 3\,000$.

Третья группа тестов проверяется в момент сдачи задачи на проверку и стоит 20 баллов. Баллы за эту группу начисляются только при прохождении всех тестов группы. Для всех тестов этой группы выполнено условие $\forall a_i : a_i \leq 5$

Четвертая группа тестов проверяется после окончания олимпиады и стоит 40 баллов. Баллы за тесты этой группы начисляются пропорционально количеству пройденных тестов.

Задача С. Покраска забора

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

Юному Энакину Скайуокеру магистр Йода поручил, на первый взгляд, простое задание. Ему необходимо было покрасить забор специальной краской. Задание состояло из n подзадач. Каждая из подзадач заключалась в покраске некоторого участка забора краской заданного цвета. Каждый цвет обозначался целым числом. Краска, которая была дана Энакину, как и любая другая, после того, как высыхала, меняла свой цвет. Но это происходило немного необычным образом.

Будем говорить, что два участка забора пересекаются, если они имеют не нулевую площадь пересечения. Разобьем все участки забора, на которые была нанесена краска, на некоторые множества. Вначале каждый отрезок отнесем к отдельному множеству. Если существуют пересекающиеся участки, которые относятся к различным множествам A и B , объединим A и B . Будем делать так до тех пор, пока такие участки не перестанут существовать.

Теперь для каждого множества можно определить его цвет как среднее арифметическое всех цветов входящих в него участков. Зная эту величину для каждого множества, мы легко можем определить, какого цвета будет некоторая точка забора после высыхания краски. Будем говорить, что точка принадлежит множеству, если есть хотя бы один участок в множестве, который содержит в себе эту точку. Если точка забора не принадлежит ни одному множеству, она будет иметь цвет 0, иначе она будет иметь цвет соответствующего множества. Заметим, что точка не может принадлежать сразу двум множествам.

Пока Энакин занимался покраской забора, ему стало интересно, что же случится, если он закончит покраску после нанесения первых i участков. В частности, его заинтересовал такой вопрос «Точку какого наибольшего цвета можно будет найти на заборе после того, как краска высохнет?».

Формат входных данных

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 10^5$) — количество участков забора, которые необходимо покрасить. В следующих n строках находится по три целых числа l , r , c ($0 \leq l < r \leq 10^9$, $1 \leq c \leq 10^9$) — левая и правая границы i -го участка (левая граница включается, правая нет), а также цвет, в который необходимо покрасить соответствующий участок.

Формат выходных данных

В n строках необходимо вывести по одному вещественному числу с не менее чем тремя знаками после запятой. Число в i -ой строке должно обозначать максимальный цвет точки на стене, который можно было бы найти, если бы Энакин закончил покраску забора после выполнения первых i подзадач.

Примеры

painting.in	painting.out
6	1.00000
0 1 1	2.00000
2 5 2	2.50000
3 7 3	3.00000
6 8 4	5.00000
9 11 5	4.00000
7 10 6	

Задача D. Стаканчики

Имя входного файла: `stacks.in`
Имя выходного файла: `stacks.out`
Ограничение по времени: 1 секунда
Ограничение по памяти:

С самого раннего детства Маша мечтала облететь весь земной шар и побывать в разных странах. И вот наконец-то её мечта сбылась — ей предложили работу в одной из крупнейших авиакомпаний мира. Теперь Маша работает стюардессой. В её обязанности входит разносить еду и напитки пассажирам, а также собирать использованную посуду.

Использованные стаканчики Маша собирает следующим образом. Если пассажир не допил напиток, то Маша ставит его стаканчик в новую отдельную стопку. Если же стаканчик пуст, она ставит его в низ самой маленькой стопки (просто вкладывая эту стопку в пустой стаканчик), так как слишком высокую стопку легко уронить (пустой стаканчик ставится в отдельную стопку, только если поднос пуст).

Перед началом сбора использованной посуды Маша знает, сколько напитка в стакане осталось у каждого пассажира. Теперь ее интересует вопрос — сможет ли она донести поднос со стаканчиками, не уронив его. Для этого ей необходимо знать высоту самой большой стопки. Но так как пассажиров очень много, она не может посчитать это сама, поэтому просит вас помочь ей!

Формат входных данных

В первой строке входного файла задано единственное целое число n — количество пассажиров ($1 \leq n \leq 10^6$).

В следующей строке через пробел заданы n целых чисел a_i — количество оставшегося напитка в стаканчиках ($0 \leq a_i \leq 100$) в том порядке, в котором Маша будет их собирать. Пустому i -ому стаканчику соответствует $a_i = 0$.

Формат выходных данных

В выходной файл выведите единственное целое число — количество стаканчиков в самой большой стопке.

Примеры

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

Примечание

Решения, корректно работающие для $n \leq 1000$ будут оцениваться из 40 баллов.

В примере в итоге получится одна стопка из первых трех стаканчиков и стопка из последних двух.