

Задача А. Разница последовательностей

Имя входного файла: `setdiff.in`
Имя выходного файла: `setdiff.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вам заданы две строго возрастающие последовательности A и B . Найдите все элементы последовательности A , не содержащиеся в последовательности B .

Формат входных данных

В первой строке входного файла записано число n – длина последовательности A . Во второй строке записаны через пробелы элементы последовательности A . В третьей строке записано число m – длина последовательности B . Во четвертой строке записаны через пробелы элементы последовательности B . Элементы обеих последовательностей записаны в возрастающем порядке, элементы внутри каждой последовательности различны (но могут быть совпадающие элементы из разных, иначе задача была бы странной :) и не превосходят 10^9 по модулю. $1 \leq n, m \leq 5 \cdot 10^5$

Формат выходных данных

Выведите через пробелы все элементы последовательности A , не содержащиеся в последовательности B .

Примеры

setdiff.in	setdiff.out
5 1 2 3 4 5 5 0 3 5 6 7	1 2 4
3 1 2 3 3 1 2 3	

Задача В. Установка Козырька

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вадим работает в ЖКХ и сегодня он крайне озабочен вопросов сосулек. А именно он наблюдает за домом по адресу — проспект Программистов, дом 404. В доме n этажей, на каждом этаже по m окон, включая первый. Окна на каждом этаже пронумерованы слева направо от 1 до m . i -е окно j -го этажа находится ровно под i -м окном $j+1$ -го этажа. Под некоторыми окнами свисают сосульки.

Вадим руководит установкой на дом козырька, который будет установлен ниже первого этажа и проходить под d окнами. То есть, если начало козырька находится под окном с номером i , то он проходит под всеми окнами с номерами от i до $i+d-1$. Вадим хочет выбрать такую позицию для козырька, чтобы над ним находилось наибольшее возможное количество сосулек и из всех подходящих вариантов выбрать такой, что начало козырька будет находиться как можно левее.

Помогите Вадиму выбрать нужное окно.

Формат входных данных

В первой строке входного файла `input.txt` содержатся числа n, m, d, k — количество этажей, количество окон на каждом этаже, длина козырька и количество окон, под которыми есть сосульки, соответственно ($1 \leq n, m \leq 100, 1 \leq d \leq m, 0 \leq k \leq n \cdot m$).

В следующих k строках заданы тройки чисел x, y, z — номер этажа, номер окна на этаже и количество сосулек под ним ($1 \leq x \leq n, 1 \leq y \leq m, 1 \leq z \leq 10$).

Гарантируется, что каждая пара x, y встречается не более одного раза.

Формат выходных данных

В единственной строке выходного файла `output.txt` выведите одно число — номер окна, под которым следует разместить начало козырька.

Примеры

input.txt	output.txt
2 2 2 3 1 1 1 1 2 3 2 1 1	1
1 3 1 2 1 3 1 1 2 1	2
3 3 2 5 1 3 1 1 2 2 2 3 1 3 3 1 3 2 1	2

Замечание

В первом примере длина козырька равна количеству окон на этаже, поэтому существует всего одна позиция, куда его можно поставить.

Во втором примере козырек имеет длину один и если его поставить под окнами с номерами 1, 2 или 3, над ним будет соответственно 0, 1 или 1 сосулька. Таким образом подходит две позиции начала козырька, выбираем более левую.

В третьем примере возможны две позиции для козырька, но так как под окнами с 1-м, в отличие от окон 3-м номером, нет сосулек, а окна с номерами 2 будут над козырьком в любом случае, существует единственный оптимальный вариант разместить козырек.

Задача С. 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 в том случае, если они будут на расстоянии не более r метров.

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

Формат входных данных

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 ($1 \leq d_1 < d_2 < \dots < d_n \leq 10^9$).

Формат выходных данных

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

Задача D. Коньки

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

В ЛКШ-Зима школьники любят ходить на каток. В прокате коньков есть много коньков самых разных размеров. Школьник может надеть коньки любого размера, который не меньше размера его ноги. Известны размеры всех коньков и размеры ног школьников. Определите, какое наибольшее число школьников сможет одновременно пойти покататься.

Формат входных данных

Первая строка входных данных содержит число N — количество коньков в прокате ($1 \leq N \leq 10^5$). Во второй строке записано N чисел — размеры коньков. В третьей строке содержится число M — количество школьников в ЛКШ ($1 \leq M \leq 10^5$), четвертая строка содержит размеры их ног. Размеры коньков и ног — натуральные числа, не превосходящие 100.

Формат выходных данных

Выведите единственное число — наибольшее количество школьников, которое сможет пойти на каток.

Пример

<code>skates.in</code>	<code>skates.out</code>
4 41 40 39 42 3 42 41 42	2

Задача Е. Дюбели и сверла

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

Петя хочет повесить картину на стену. Для этого ему нужно просверлить в стене дырку, вбить в нее дюбель и вкрутить в него саморез. Петя покопался в кладовке и нашел n сверел и m дюбелей. Петя хочет найти сверло и дюбель одного радиуса. Однако, таких может не быть, в этом случае он хочет подобрать сверло и дюбель так, чтобы разность их диаметров была как можно меньше. Помогите Пете.

Формат входных данных

В первой строке входного файла заданы целые числа n и m ($1 \leq n, m \leq 10^5$). Во второй строке заданы n целых чисел — диаметры сверел. В следующей строке заданы m целых чисел — диаметры дюбелей. Диаметры заданы в неубывающем порядке, все диаметры — числа от 1 до 10^9 .

Формат выходных данных

Выведите минимальную возможную разницу диаметров сверла и дюбеля.

Примеры

drill.in	drill.out
3 2 1 8 15 5 6	2
3 3 1 3 5 3 4 6	0

Задача F. Красные и чёрные

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

У Демида в руке n карт. Он хочет разложить их на красные и чёрные: чтобы сначала шли карты одного цвета, а затем другого.

За одно действие он может вынуть из руки одну карту и вставить её между какими-то двумя соседними картами или с краю.

Какое минимальное число таких действий необходимо совершить Демиду?

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 50\,000$) — количество карт в руке.

Во второй строке — строчка длины n , состоящая только из символов «R» и «B», означающих красную и чёрную карту соответственно.

Формат выходных данных

Выведите одно число — минимальное число действий для разложения карт по цветам.

Примеры

<code>cards.in</code>	<code>cards.out</code>
4 RBRB	1
4 BRBR	1

Задача G. Новые технологии

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Т'Чалла тестирует новый костюм. Во время очередного теста скорости, которую он может развить в нем, он наткнулся на постройку, возведенную в век, когда в Ваканде еще не были развиты технологии.

На одной из стен данной постройки Т'Чалла обнаружил n плит с числами. Он вспомнил древний обычай Ваканды, о котором ему рассказывал отец: древние вакандцы вели статистику изобретенных новых технологий, записывая в конце каждого года суммарное количество технологий, изобретенных на данный момент. При этом историки выяснили, что каждый год вакандцы изобретали не менее, чем m новых технологий.

Т'Чалла хочет понять, могут ли эти плиты иметь отношение к древнему обычаю. К сожалению, это сложно осуществить, так как числа на некоторых плитах невозможно прочесть. Он просит вас помочь ему проверить это. Чтобы доказать возможность отношения этих плит к обычаю, он хочет увидеть возможную последовательность, показывающую, сколько технологий было изобретено в каждый из годов, которая не противоречила бы надписям на плитах.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество плит и ограничение снизу на количество изобретенных технологий в один год ($1 \leq n \leq 1000$, $0 \leq m \leq 100$).

Во второй строке заданы n чисел $p_1, p_2 \dots p_n$, где p_i означает суммарное число изобретенных технологий в конце i -го года ($-1 \leq p_i \leq 1000$). Если число на плите невозможно прочесть, то $p_i = -1$.

Формат выходных данных

Выведите -1 , если данные плиты не могут иметь отношения к древнему обычаю, иначе выведите n чисел a_i — количество изобретенных технологий в i -й год ($m \leq a_i$).

Если существует несколько ответов, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 1 1 2 3	1 1 1
2 0 -1 -1	3 5
4 2 2 -1 -1 5	-1

Замечание

В первом тестовом примере существует единственная подходящая последовательность количеств изобретенных технологий, так как все плиты известны.

Во втором тестовом примере надписи на обеих плитах невозможно прочесть, поэтому подойдет любая пара неотрицательных чисел.

В третьем тестовом примере нельзя получить сумму, равную 5, из четырех чисел, каждое из которых хотя бы 2.

Задача Н. Перераспределение камней

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

Как известно, у красавицы и чудовища не все сразу было хорошо. Эта история как раз про это. Как только красавица стала пленницей в замке чудовища, он дал ей первое, но сразу же очень ответственное задание.

Перед красавицей стояло бесконечное количество сундуков, выставленных в линию и пронумерованных целыми числами от $-\infty$ до ∞ . В n сундуках лежали волшебные камни, способные как убивать, так и воскрешать кого угодно, остальные же сундуки были пустые. Задание красавицы состояло в перекладывании камней из сундуков так, чтобы они все в конце концов лежали в n различных сундуках с последовательными номерами. За одно перекладывание красавица могла взять камень из любого сундука и переложить его в любой другой не занятый камнем сундук.

Конечно, красавице захотелось как можно быстрее выполнить ее задание, поэтому она решила минимизировать количество перекладываний. С просьбой найти число этих перекладываний она обратилась к великому волшебнику Мерлину, воззвав к нему о помощи. Однако, даже Мерлин не смог справиться с этой задачей, и ему пришлось проделать долгое путешествие сквозь пространство и время, чтобы попросить помощи у вас. Помогите волшебнику решить задачу красавицы!

Формат входных данных

В первой строке содержится число n — количество сундуков с волшебными камнями ($1 \leq n \leq 10^5$).

Во второй строке содержатся n чисел a_i — номера сундуков с камнями ($-10^9 \leq a_i \leq 10^9$). Гарантируется, что в каждом сундуке лежит не более одного камня.

Формат выходных данных

В единственной строке выведите минимальное количество перекладываний, которое требуется, чтобы разместить все n камней в n различных сундуках с последовательными номерами.

Пример

<code>stones.in</code>	<code>stones.out</code>
5 3 1 -2 4 7	2

Замечание

В первом тестовом примере подходит например такой алгоритм действий:

- Переложить камень из сундука -2 в сундук 2
- Переложить камень из сундука 7 в сундук 5

Также можно вторым действием переложить камень из сундука 7 в сундук 0.