

Задача А. Генеральная уборка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Миша живёт один в большой комнате в общежитии, представляющей собой клетчатое поле размера $n \times m$. Строки поля пронумерованы от 1 до n сверху вниз, столбцы — от 1 до m слева направо.

Он очень чистолюбивый, но слегка ленивый, поэтому купил себе несколько роботов-пылесосов для уборки комнаты. В очередной раз после уборки роботы остановились в разных клетках комнаты. Для каждой клетки комнаты известно, сколько роботов в ней остановилось.

Миша хочет собрать всех роботов в одной клетке комнаты. Чтобы это сделать, он использует пульт управления: за одно нажатие на кнопку на пульте он может сдвинуть одного робота в любую соседнюю по стороне клетку по своему усмотрению.

Определите минимальное число нажатий, которые придётся сделать Мише.

Формат входных данных

В первой строке входных данных через пробел задано два целых числа n, m ($1 \leq n, m \leq 1000$) — количество строк и столбцов поля.

В следующих n строках задано m целых чисел a_{ij} — количество роботов, находящихся в клетке i, j ($0 \leq a_{ij} \leq 10^9$).

Формат выходных данных

Выведите одно целое число — минимальное количество нажатий на кнопки, чтобы собрать всех роботов в одной клетке.

Примеры

стандартный ввод	стандартный вывод
3 3 1 0 1 0 0 0 1 0 1	8
2 2 3 2 2 0	4

Замечание

В первом тестовом примере все роботы могут собраться в клетке $(2, 2)$, причём для каждого робота потребуется два нажатия на кнопку.

Во втором тестовом примере можно подвинуть роботов из клеток $(1, 2)$ и $(2, 1)$ в клетку $(1, 1)$, потратив на каждого из 4-х роботов по два нажатия.

Задача В. Прибыльный старт

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

k друзей организовали стартап по производству укулеле для кошек. На сегодняшний день прибыль составила n рублей. Вы, как главный бухгалтер компании, хотите в каждый из ближайших d дней приписывать по одной цифре в конец числа, выражающего прибыль. При этом в каждый из дней прибыль должна делиться на k .

Формат входных данных

В единственной строке входных данных через пробел записаны три числа: n, k, d — изначальная прибыль, количество учредителей компании и количество дней, которое вы собираетесь следить за прибылью ($1 \leq n, k \leq 10^9, 1 \leq d \leq 10^5$). **НЕ** гарантируется, что n делится на k .

Формат выходных данных

Выведите одно целое число x — прибыль компании через d дней. Первые цифры числа x должны совпадать с числом n . Все префиксы числа x , которые длиннее числа n на $1, 2, \dots, d$ цифр, должны делиться на k . Если возможных ответов несколько, выведите любой из них. Если ответа не существует, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
21 108 1	216
5 12 4	-1

Замечание

В первом тестовом примере всего один день нужно следить за прибылью. Можно дописать цифру 6 в конец числа 21 и получить прибыль, делящуюся на 108.

Во втором тестовом примере мы в первый же день не можем получить прибыль, делящуюся на k , какая бы цифра не была дописана у числу n , поэтому ответа не существует.

Задача С. Тайное заклинание

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Хината и Кенма играют в компьютерную игру. Для победы одному из них необходимо одолеть всех магических существ другого.

Игру можно представить себе так: n существ выстроены в ряд, i -е существо обладает силой a_i . Изначально все существа нейтральны, то есть не принадлежат ни Хинате, ни Кенме.

Хината может выбрать **ровно** k существ и уничтожить их тайным заклинанием. После этого из оставшихся $n - k$ существ первая половина (стоящие в начале ряда существа) переходит на сторону Хинаты, а вторая — на сторону Кенмы, после чего начинается бой.

Хината очень хочет победить. Для этого ему необходимо максимизировать разность суммарной силы своих существ и существ Кенме, то есть величину $D = (a_{i_1} + a_{i_2} + \dots + a_{i_{(n-k)/2}}) - (a_{i_{(n-k)/2+1}} + \dots + a_{i_{n-k}})$, где $i_1, \dots, i_{n-k}$ — позиции существ, оставшихся после использования Хинатой своей способности.

Хината и Кенма собираются сыграть t игр. Помогите Хинате — для каждой игры найдите максимальное D , которого он может добиться.

Формат входных данных

В первой строке входных данных задано единственное целое число t ($1 \leq t \leq 10$) — количество игр.

Далее следуют t тестовых примеров, разделённых переводом строки.

В первой строке каждого тестового примера через пробел заданы два целых числа n ($2 \leq n \leq 3 \cdot 10^5$) и k ($0 \leq k < n$) — число существ и количество существ, которых Хината уничтожит. Обратите внимание, что Хината **обязан** использовать свою способность. Гарантируется, что $n - k$ — чётное число.

Во второй строке каждого тестового примера задано n целых чисел a_i ($1 \leq a_i \leq 10^9$) — силы существ.

Гарантируется, что сумма n по всем t не превосходит $3 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера на отдельной строке выведите одно целое число D — максимальную достижимую разность суммарной силы существ Хинаты и Кенмы.

Пример

стандартный ввод	стандартный вывод
3	1
6 2	2
3 2 4 1 5 10	-1
4 0	
1 5 2 2	
3 1	
1 2 3	

Замечание

В первом тестовом примере из условия Хината может уничтожить существ с силами 2 и 10. Тогда D будет равно $(3 + 4) - (1 + 5) = 1$.

Во втором тестовом примере Хината не уничтожает никого, так что $D = (1 + 5) - (2 + 2) = 1$.

В третьем тестовом примере Хината может убрать существо с силой 1 или 3, чтобы добиться $D = -1$.

Задача D. Бегущий вор

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Один охотник за сокровищами решил ограбить музей. Музей представляет собой n залов, пронумерованные от 1 до n и расположенные в ряд друг за другом. В каждом зале находится a_i сокровищ. Изначально в вестибюле (зал номер ноль) находятся вор и охранник. В момент времени ноль вор начинает бежать в первый зал, тратя 1 минуту на перемещение по коридору, соединяющему этот зал с вестибюлем. Охранник, заметив это, сразу же бросается в погоню - у него аналогичное перемещение по коридору занимает 2 минуты. Все передвижения внутри одного зала происходят мгновенно у обоих.

Забежав в очередной зал, вор может провести в нём x минут. При этом, с каждой минутой, проведённой в одном зале, он всё больше входит во вкус воровства и в за первую минуту собирает 1 сокровище, за вторую ещё 2, за третью 3 и т.д. Конечно, он не может взять больше сокровищ, чем было в зале изначально. Например, если в зале было 8 сокровищ, то за первые 3 минуты вор возьмёт $1 + 2 + 3 = 6$ сокровищ, а за четвертую оставшиеся 2. Забежав в следующий зал, он снова начнёт со скорости грабежа, равной одному сокровищу в минуту. Вор так же может покинуть зал сразу же, задержавшись в нём на $x = 0$ минут, тогда он не возьмёт из него ничего.

При этом охранник всё это время бежит за воров. Если в какой-то момент он вбегает в зал, в котором находится вор, то последний должен сразу же покинуть этот зал и бежать дальше, иначе его схватят. Более формально, по прошествии ровно t минут вор обязан находиться хотя бы в зале номер $\lceil \frac{t}{2} \rceil$.

Вам нужно определить, какое максимальное количество сокровищ может унести с собой вор. Выбежав из последнего зала, он покидает музей со всеми награбленными сокровищами.

Формат входных данных

В первой строке задаётся t — количество тестовых случаев ($1 \leq t \leq 100$)

На каждый тестовый случай в первой строке задаётся n - количество залов ($1 \leq n \leq 10^3$), на следующей строке вводятся n чисел a_i — количество сокровищ в i -м зале. ($0 \leq a_i \leq 10^3$)

Гарантируется, что сумма n по всем тестовым случаям не превосходит 10^3

Формат выходных данных

На каждый тестовый случай выведите одно число — максимальное количество сокровищ, которое может унести вор.

Пример

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

Замечание

В третьем примере вор может потратить по одной минуте в каждом зале, украв из каждого по одному сокровищу, или провести 1 минуту в первом зале и 2 в третьем. В четвертом примере вор не может провести больше одной минуты в первом зале, поэтому может унести только 1 сокровище.

Задача Е. Прямоугольники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана прямоугольная таблица A размера $N \times M$. В ней требуется посчитать, сколько связанных фигур из закрашенных клеток образуют прямоугольники. Связная фигура — фигура из соединенных между собой закрашенных клеток, соседних по сторонам. Прямоугольник должен быть со всех сторон окружен пустыми клетками или границами поля.

Формат входных данных

В первой строке входных данных записаны два числа N и M ($1 \leq N, M \leq 2000$) — размеры матрицы A . В каждой из следующих N строк записано M символов $A_{i,j}$, равных '#' (ASCII код 35) или '.' (ASCII код 46), символ '#' означает закрашенную клетку, '.' — пустую.

Формат выходных данных

Выведите одно число: число закрашенных прямоугольников в исходной матрице.

Примеры

стандартный ввод	стандартный вывод
3 3 #.. ..# ##.	3
5 5 ##.## #.## .##.. ###.# .##.#	2

Задача F. Забег по стадиону

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Стадион представляет собой окружность **длиной** L метров, на которой отмечена точка старта. По стадиону бегают Кирилл и Антон. У каждого мальчика есть своя точка старта (она представляет собой расстояние в метрах от старта, отсчитанное по часовой стрелке) и своя скорость в метрах в секунду (положительная скорость означает, что мальчик бежит по часовой стрелке, отрицательная — что бежит против часовой, а нулевая — что он стоит на месте).

Вам нужно сказать, через какое минимальное время мальчики окажутся на одинаковом расстоянии от точки старта. Обратите внимание, что в этот момент они могли находиться в разных точках. Расстоянием от точки A до точки B называется минимальное из расстояний, которое нужно пробежать из точки A по или против часовой стрелки, чтобы оказаться в B .

Формат входных данных

В единственной строке вводится 5 целых чисел L, x_1, v_1, x_2, v_2 ($1 \leq L \leq 10^9$, $0 \leq x_1, x_2 < L$, $|v_1|, |v_2| \leq 10^9$) — длины стадиона в метрах, начальная точка Кирилла, скорость Кирилла, начальная точка Антона, скорость Антона.

Формат выходных данных

В первой строке выведите слово «YES», если случится момент, когда мальчики будут на одинаковом расстоянии от старта, или «NO», если такого момента не произойдёт.

Если ответ «YES», то во второй строке выведите одно вещественное число — через какое минимальное количество времени мальчики окажутся на одинаковом расстоянии от старта.

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-9} .

Формально, пусть ваш ответ равен a , а ответ жюри равен b . Ваш ответ будет зачтен, если и только если $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-9}$.

Примеры

стандартный ввод	стандартный вывод
6 3 1 1 1	YES 1
12 8 10 5 20	YES 0.3
5 0 0 1 2	YES 2
10 7 -3 1 4	YES 0.857142857

Замечание

В первом тесте Кирилл изначально находится в точке 3 и бежит по часовой стрелке со скоростью 1. Антон находится в точке 1 и также бежит по часовой стрелке со скоростью 1. Через 1 секунду мальчики окажутся в точках 4 и 2 соответственно. Обе эти точки расположены на расстоянии 2 метра от старта (точки 0, совпадающей с точкой 6). Можно показать, что до этого они всегда находились на разном расстоянии от старта. Значит, ответ — 1.

Во втором тесте оба мальчика окажутся в точке 11 через 0.3 секунды.

В третьем Антон прибежит к Кириллу в точку 0 за 2 секунды.

Задача G. Тур по Светлогорью

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Страна Светлогорье расположена в горах, поэтому в ней есть много красивых мест. При этом в ней есть n городов, и i -й город расположен на высоте i .

Также в стране есть m **односторонних** дорог, соединяющих города. Дороги проложены под большим уклоном, поэтому по дороге можно проехать только от более высокого города до более низкого (то есть если концами дороги являются города a и b , причём $a > b$, то по этой дороге можно проехать только из a в b).

Турист Илья хочет устроить тур по стране. Посещение города i принесёт ему a_i удовольствия (a_i может быть отрицательным). Путешествие Ильи будет иметь вид простого пути — непустой последовательности городов $p_1, \dots, p_k$ такой, что из города p_i есть дорога в город p_{i+1} для любого $1 \leq i < k$, а также никакой город не должен встречаться на пути дважды.

Помогите Илье найти простой путь с максимальной суммой удовольствий от посещённых городов, и выведите эту сумму.

Формат входных данных

В первой строке даётся два числа n и m ($1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$) — количество городов и количество дорог в стране.

Во второй строке содержатся n чисел — массив a ($|a_i| \leq 10^9$) — удовольствия от посещения городов.

Далее в m строках содержится по 2 числа a, b — концы очередного ребра. При этом выполняется $1 \leq b < a \leq n$, то есть по дороге можно проехать из a в b . Гарантируется, что в графе нет кратных рёбер и петель.

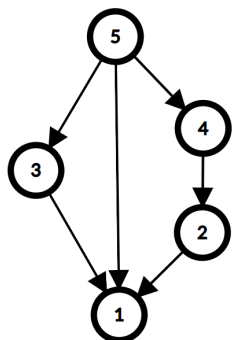
Формат выходных данных

Выведите одно число — максимальную сумму удовольствий посещённых городов для какого-то пути.

Пример

стандартный ввод	стандартный вывод
5 6 1 2 3 2 -1 5 4 5 3 4 2 2 1 3 1 5 1	5

Замечание



В первом тесте оптимальным путём будет $4 \rightarrow 2 \rightarrow 1$ — сумма удовольствий равна $2 + 2 + 1 = 5$.

Задача Н. Количество подмассивов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дан массив с целыми числами и число K . Определите количество *хороших подмассивов* в нём. *Хороший подмассив* — это массив, в котором количество различных целых чисел равно K . Подмассивом называется непрерывная часть массива.

Формат входных данных

В первой строке содержится два числа: N, K ($1 \leq N \leq 10^5, 1 \leq K \leq N$), где N — количество чисел в массиве. В следующей строке вводится N целых чисел, разделенных пробелами ($1 \leq a_i \leq 10^6$).

Формат выходных данных

Выведите количество хороших подмассивов. Гарантируется, что ответ не превышает 2^{64} .

Примеры

стандартный ввод	стандартный вывод
5 2 1 2 1 2 3	7
5 3 1 2 1 3 4	3

Задача I. НОД по кругу

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

Массив из n натуральных чисел $a_1, a_2, \dots, a_n$ записан по кругу. Следующим для числа a_i назовем число a_{i+1} , если $1 \leq i \leq n-1$. Для числа a_n следующим является число a_1 .

По массиву a можно построить массив b длины n , где b_i равно наибольшему общему делителю a_i и следующего за ним по кругу числа. Более формально, $b_i = \text{НОД}(a_i, a_{(i \bmod n)+1})$. После этого массив a заменяется на массив b .

Какое минимальное число таких операций необходимо совершить, чтобы все числа в массиве стали равны?

Формат входных данных

В первой строке дано натуральное число n ($2 \leq n \leq 200\,000$). В следующей строке через пробел заданы n натуральных чисел: $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 1\,000\,000$).

Формат выходных данных

В единственной строке выведите целое число — сколько операций нужно совершить, чтобы все числа массива стали равными.

Примеры

стандартный ввод	стандартный вывод
3 2 4 3	2
2 1 1	0
5 3 6 1 9 12	4

Замечание

В первом примере после одной операции массив превратится в $(2, 1, 1)$, после следующей в $(1, 1, 1)$.

Во втором примере все числа изначально равны, поэтому не нужно совершать никаких операций. Можно показать, что в третьем примере необходимо совершить ровно 4 операции.

Задача J. Амбициозная улитка

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

Домашний питомец мальчика Васи — улитка Петя. Петя обитает на бесконечном в обе стороны вертикальном столбе, который для удобства можно представить как числовую прямую. Изначально Петя находится в точке 0.

Вася кормит Петю ягодами. У него есть n ягод, каждая в единственном экземпляре. Вася знает, что если утром он даст Пете ягоду с номером i , то поев и набравшись сил, за остаток дня Петя поднимется на a_i единиц вверх по столбу, но при этом за ночь, потяжелев, съедет на b_i единиц вниз. Параметры различных ягод могут совпадать.

Пете стало интересно, а как оно там, наверху, и Вася взялся ему в этом помочь. Ближайшие n дней он будет кормить Петю ягодами из своего запаса таким образом, чтобы максимальная высота, на которой побывал Петя за эти n дней была максимальной. К сожалению, Вася не умеет программировать, поэтому он попросил вас о помощи. Найдите, максимальную высоту, на которой Петя сможет побывать за эти n дней и в каком порядке Вася должен давать Пете ягоды, чтобы Петя смог её достичь!

Формат входных данных

В первой строке входных данных дано число n ($1 \leq n \leq 5 \cdot 10^5$) — количество ягод у Васи. В последующих n строках описываются параметры каждой ягоды. В $i + 1$ строке дано два числа a_i и b_i ($0 \leq a_i, b_i \leq 10^9$) — то, насколько поднимется улитка за день после того, как съест i ягоду и насколько опуститься за ночь.

Формат выходных данных

В первой строке выходных данных выведите единственное число — максимальную высоту, которую сможет достичь Петя, если Вася будет его кормить оптимальным образом. В следующей строке выведите n различных целых чисел от 1 до n — порядок, в котором Вася должен кормить Петю (i число в строке соответствует номеру ягоды, которую Вася должен дать Пете в i день чтобы Петя смог достичь максимальной высоты).

Примеры

стандартный ввод	стандартный вывод
3 1 5 8 2 4 4	10 2 3 1
2 7 6 7 4	10 2 1

Замечание

Во втором примере изначально улитка находится на высоте 0. Пусть сначала Петя накормит её второй ягодой, а затем первой. После того как она съест вторую ягоду, за день она поднимется на 7 (и окажется на высоте 7), а за ночь опустится на 4 (и окажется на высоте 3). После того как она съест первую ягоду, за день она поднимется на 7 (и окажется на высоте 10), а за ночь опустится на 6 (и окажется на высоте 4).

Таким образом, максимальная высота, на которой побывает улитка при данном порядке кормления, равна 10. Нетрудно видеть, что если Петя накормит улитку сначала первой ягодой, а затем второй, то максимальная высота, на которой побывает улитка, будет меньше.