

Задача А. Атомы

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

В лаборатории аномальных материалов антинаучно-исследовательского комплекса «Black Mesa» проводят эксперименты с недавно разработанным графитовым наностержнем. Графитовый наностержень представляет собой n последовательно соединенных атомов углерода, находящихся на одной прямой. Каждый атом имеет определенный заряд.

Для проведения эксперимента, стержень располагают вертикально. Пронумеруем атомы от 1 до n снизу вверх. Между двумя атомами образуется сильная связь, если это соседние атомы и верхний из них имеет заряд ровно на один больше, чем нижний. Иными словами, атомы a и b соединены сильной связью, если $a = b + 1$ и $q_a = q_b + 1$, где q_i — заряд i -го атома. Цепочкой атомов назовем несколько последовательных атомов, соединенных сильными связями.

Вчера был проведен очередной эксперимент. Перед началом эксперимента каждому атому установили определенный заряд: i -му атому установили заряд q_i .

Во время эксперимента ученые проводили действия двух типов:

- у всех атомов с номерами от l_i до r_i , включительно, заряд изменяли на величину d_i ;
- временно разрушали все сильные связи атомов, кроме тех, которые соединяют атомы с номерами от l_i до r_i , включительно, и измеряли длину самой длинной цепочки атомов среди оставшихся сильных связей. Затем восстанавливали все временно разрушенные связи.

Было произведено m действий, однако выяснилось, что в результате побочного эффекта эксперимента запись результатов измерений оказалась утеряна. Для продолжения работы с графитовым наностержнем необходимо восстановить результаты вчерашних измерений. К счастью, сохранился план действий, произведенных во время эксперимента. Помогите ученым продолжить исследования, восстановите результаты измерений.

Формат входных данных

В первой строке находится одно целое число n ($1 \leq n \leq 100\,000$) — количество атомов в наностержне. Во второй строке находятся n чисел q_i ($|q_i| \leq 10^9$) — начальный заряд i -го атома. В третьей строке находится одно целое число m ($0 \leq m \leq 100\,000$) — количество действий в эксперименте. В следующих m строках содержится описание эксперимента.

Если строка начинается с символа «+», очередное действие — изменение заряда атомов. В таком случае, далее в этой строке находятся три целых числа: l_i , r_i и d_i ($1 \leq l_i \leq r_i \leq n$, $|d_i| \leq 10^9$), которые характеризуют это действие.

Если строка начинается с символа «?», очередное действие — второго типа. В таком случае, далее в этой строке находятся два целых числа: l_i и r_i ($1 \leq l_i \leq r_i \leq n$), которые характеризуют это действие.

Формат выходных данных

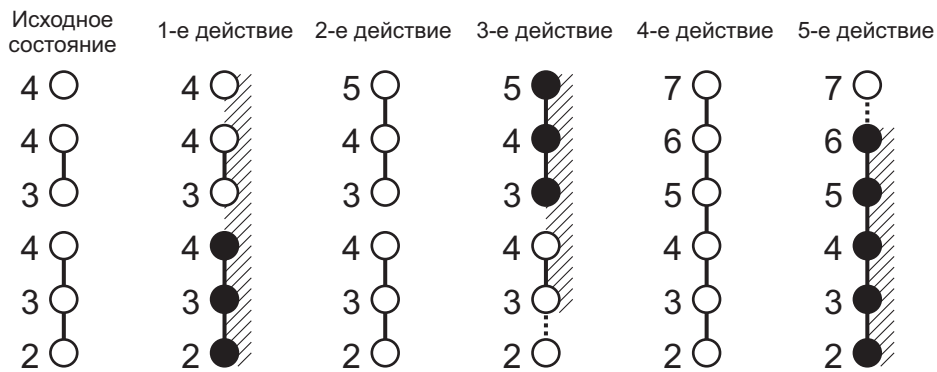
Для каждого действия второго типа выведите в новой строке одно число — длину наибольшей цепочки.

Пример

atoms.in	atoms.out
6	3
2 3 4 3 4 4	3
5	5
? 1 6	
+ 6 6 1	
? 2 6	
+ 4 6 2	
? 1 5	

Пояснение к примеру

Иллюстрация к примеру. Пунктиром выделены сильные связи, которые разрушаются на время действия второго типа. Для каждого действия второго типа выделены отрезок запроса и самая длинная цепочка.



Задача В. День рождения

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

Ваня сомневается, хочет ли он праздновать свой день рождения. Он решил сообщить друзьям частичную информацию о дате своего рождения, и если они вычислят точную дату, то так тому и быть — праздник состоится.

Подруге Ане он личным сообщением в социальной сети сообщил число внутри месяца, а другу Боре — месяц своего рождения. Затем он опубликовал на своей странице список дней года, один из которых — день его рождения, а также фразу о том, что Аня знает число, но не знает месяц, а Боря — знает месяц, но не число.

Затем в комментариях состоялся следующий диалог:

Аня: Я не знаю точную дату рождения Вани, но я уверена, что и Боря её тоже не знает.

Боря: Я сначала не знал дату Ваниногo рождения, но теперь, после твоего комментария, знаю точно!

Аня: Ого! Теперь и я тоже знаю точную дату!

Когда же у Вани день рождения?

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 366$) — количество возможных дат, опубликованных Ваней на своей странице.

В каждой из следующих n строк содержится описание одной даты: число и номер месяца. Месяцы пронумерованы от 1 до 12.

Даты приведены в хронологическом порядке внутри года.

Гарантируется, что ситуация корректна, и существует единственная возможная дата рождения Вани, которая приводит к описанному диалогу.

Формат выходных данных

Выведите дату рождения Вани в таком же формате: сначала число, а затем месяц.

Пример

<code>birthday.in</code>	<code>birthday.out</code>
11	17 6
29 2	
5 5	
16 5	
31 5	
17 6	
18 6	
14 7	
16 7	
5 12	
14 12	
17 12	

Замечание

Напомним число дней в каждом месяце. Поскольку Ваня мог родиться и в високосном году, будем считать, что в феврале 29 дней.

Номер месяца	1	2	3	4	5	6	7	8	9	10	11	12
Число дней	31	29	31	30	31	30	31	31	30	31	30	31

Задача С. Шушпанчики и кинотеатр

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

В кинотеатре «Дружба» через неделю пройдет премьера мирового хита «Осторожный Макс», и шушпанчики обязательно хотят попасть на первый сеанс. Зал в кинотеатре состоит из n рядов по n мест в каждом. Ряды пронумерованы от 1 до n , места в каждом ряду также пронумерованы от 1 до n . Обозначим место с номером c в ряду r как (r, c) .

Шушпанчики точно знают, что лучшее место в зале — место (r_b, c_b) . Для любого места (r, c) можно посчитать *неудачность* этого места как $|r - r_b| + |c - c_b|$, и чем неудачность меньше, тем лучше.

Шушпанчики пойдут на сеанс группой из k шушпанчиков. Они хотят сидеть рядом друг с другом, поэтому договорились купить k мест подряд в одном ряду. Таким образом, шушпанчики купят билеты на места $(r_a, c_a), (r_a, c_a + 1), \dots, (r_a, c_a + k - 1)$ для некоторых r_a и c_a .

К сожалению, некоторые места уже забронированы, и их выкупить не получится. Помогите шушпанчикам выбрать k соседних мест на одном ряду так, чтобы суммарная неудачность выбранных ими мест была минимальна.

Формат входных данных

В первой строке заданы три числа n , m и k ($1 \leq n \leq 10^9$, $0 \leq m \leq \min(n^2, 10^5)$, $1 \leq k \leq n$) — размер зала, число проданных мест и число шушпанчиков.

В следующих m строках описаны занятые места. Каждое место описывается двумя числами: r_i, c_i ($1 \leq r_i, c_i \leq n$, все описанные места различны).

В следующей строке даны два числа r_b, c_b ($1 \leq r_b, c_b \leq n$) — оптимальное место, как можно ближе к которому хотят оказаться шушпанчики.

Формат выходных данных

Если все шушпанчики не смогут купить билеты на k мест подряд в одном ряду, выведите -1 . Иначе, выведите минимальную суммарную неудачность, которую можно получить.

Пример

<code>cinema.in</code>	<code>cinema.out</code>
3 1 2 1 2 1 1	3
3 3 2 1 2 2 2 3 2 2 2	-1

Задача D. Доставка футболок

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

Завершилась Командная Интернет-олимпиада Новой Зеландии по Алгоритмам (КИНЗА). После олимпиады оргкомитет принял решение отправить футболки всем призерам олимпиады. Доставка футболок была поручена компании, в которой работает курьер Билл.

Биллу требуется доставить n футболок. Он должен доставлять посылки в том порядке, в котором они значатся в его обходном листе. При этом, если посылку доставить не удалось, то в обходном листе ставится отказ, и Билл едет по следующему адресу.

Билл начинает свой рабочий день в офисе компании в момент 0. Приезжая к адресату посылки, он ждет не более k минут, и если за это время адресат открывает дверь, то Билл отдает ему посылку, процесс передачи посылки без учета времени ожидания занимает t минут. Если же в течение k минут дверь не открывается и получатель не появляется, то Билл едет дальше. При этом, если получатель появился ровно через k минут после приезда Билла, то Билл успевает это заметить, и посылка оказывается доставлена. Рабочий день Билла заканчивается, когда он заканчивает передавать последнюю посылку, либо отмечает в обходном листе отказ от ее получения.

Начальник Билла Джон хочет проверить, насколько добросовестно тот работает. Джону известно, сколько времени ему потребуется на то, чтобы доехать от офиса до первого адресата, а также время на переезд от очередного адресата до следующего. Кроме того, он провел телефонный опрос и знает время, начиная с которого каждый из получателей футболок будет дома. Теперь Джон хочет узнать, в какой момент Билл закончит свой рабочий день.

Формат входных данных

В первой строке входного файла находятся три целых числа n, k, t ($1 \leq n \leq 50\,000, 1 \leq k, t \leq 10^4$).

Во второй строке находятся n натуральных чисел z_0, z_1, \dots, z_{n-1} , где z_0 — это время, которое требуется Биллу, чтобы доехать от офиса до первого адресата, а z_i — это время, которое требуется Биллу для преодоления пути от i -го до $i+1$ -го адресата. Все z_i не превосходят 10^4 .

В третьей строке находятся n неотрицательных целых чисел s_1, s_2, \dots, s_n ($0 \leq s_i \leq 10^9$), где s_i — это момент, начиная с которого i -й адресат готов принять посылку. Билл начинает свой путь в момент времени 0.

Формат выходных данных

Выведите одно число — минимальное время, за которое Билл сможет выполнить свое задание.

Примеры

<code>delivery.in</code>	<code>delivery.out</code>
3 3 1 1 5 4 1 11 7	15

Пояснение к примеру

В примере Билл приезжает к первому адресату в момент времени 1 и сразу же передает ему посылку. В момент времени 2 он выезжает от первого адресата и в момент времени 7 приезжает ко второму. Он ждет его 3 минуты, до момента времени 10, но тот не появляется, поэтому Билл отправляется к третьему адресату и приезжает к нему в момент 14. Тот дома и принимает посылку. Таким образом, Билл освобождается в момент времени 15.

Задача E. Деление

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

Сегодня Леша проходил в школе деление столбиком. На дом ему задали вычислить частное двух больших чисел: n и m . Леша уже попробовал решить пример, но, неожиданно для себя, понял, что n на m не делится. Он уверен, что учитель задавал пример, в котором результат не имеет остатка, поэтому он предположил, что допустил ошибку при переписывании примера с доски.

Теперь он хочет исправить несколько цифр в числе n , чтобы оно стало делиться на m . При этом, Леша хочет, чтобы новое число отличалось от записанного у него в минимальном количестве позиций.

Числа, записанные Лешей, не имеют ведущих нулей, он уверен, что числа, записанные на доске, также не имели ведущих нулей, поэтому и в новом числе их не должно быть. Само число 0 при этом является допустимым.

Помогите Леше.

Формат входных данных

В единственной строке входного файла находятся два целых числа n , m ($0 \leq n \leq 10^{11}$, $1 \leq m \leq 10^{11}$).

Формат выходных данных

В единственной строке выходного файла выведите одно целое число — результат изменения минимального числа цифр в числе n , чтобы полученное число не имело ведущих нулей и делилось на m .

Если ответов несколько, можете вывести любой. Если ответа не существует, выведите -1 .

Примеры

<code>division.in</code>	<code>division.out</code>
123 10	120
123 141	423
9 123	0
12 123	-1

Задача F. Милый общий делитель

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

Вася очень силен в подсчёте ворон за окном на уроках математики. Сегодня выдался необычный день, ведь ворон было очень много. К тому же, их было целых два вида: белые и чёрные. К середине урока Вася закончил подсчёт — за окном оказалось a белых и b чёрных ворон.

Поскольку до конца урока было невыносимо много времени, Вася решил послушать, что говорит учитель. Как раз в этот момент учитель рассказывал, что такое наибольший общий делитель двух чисел. Вася очень талантливый мальчик и сразу понял, что к чему. Он моментально посчитал наибольший общий делитель чисел a и b .

После этого он придумал новый термин — милый общий делитель. Милым общим делителем чисел x и y Вася решил назвать такое положительное целое число d , что x делится на d , y делится на d , а сумма цифр числа d максимальна.

Помогите Васе найти милый общий делитель чисел a и b .

Формат входных данных

В единственной строке входного файла находятся два целых числа a, b ($1 \leq a, b \leq 10^9$).

Формат выходных данных

В единственной строке выходного файла выведите милый общий делитель чисел a и b . Если ответов несколько, можете вывести любой.

Примеры

<code>mcd.in</code>	<code>mcd.out</code>
220 440	55

Задача G. Робот

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

Компания «Филипп индастриз» отправила на Марс новый робот-марсоход. Целью робота является исследование поверхности Марса.

Для исследования Марса робот будет перемещаться по поверхности планеты на север и на юг вдоль прямой. Программа робота состоит из n команд, каждая из которых описывается целым числом a_i . Каждое число a_i задаёт количество шагов, которое необходимо сделать роботу. Если $a_i > 0$, то робот совершает $|a_i|$ шагов на север, если $a_i < 0$, то $|a_i|$ шагов на юг. Робот исполняет команды последовательно, начиная с первой.

Однако по пути на Марс робот подвергся космическому излучению и его программа могла повредиться. Запустив процедуру тестирования памяти, ученые выяснили, что в программу было внесено от 0 до k ошибок следующего вида: число a_i оказалось заменено на $-a_i$. Тем не менее, приземлившись на Марс, робот выполнил свою, возможно поврежденную, программу.

Теперь для организации спасения робота ученые хотят выяснить, насколько далеко от точки, в которой он начал выполнение программы, робот мог оказаться в результате. Помогите им это выяснить.

Формат входных данных

В первой строке входного файла находятся два числа n, k ($1 \leq k \leq n \leq 10^5$) — количество чисел в программе робота и максимальное количество ошибок.

Во второй строке входного файла находятся n чисел a_i ($-10^4 \leq a_i \leq 10^4, a_i \neq 0$) — программа робота.

Формат выходных данных

В единственной строке выходного файла выведите максимальное расстояние в шагах, на которое мог удалиться робот, выполнив все команды и совершив не более k ошибок.

Примеры

<code>robot.in</code>	<code>robot.out</code>
4 1 1 2 -1 -3	5
7 2 5 -3 7 9 -2 -8 -1	29

Пояснение к примеру

В первом примере робот мог, например, выполнить программу 1, 2, -1, 3 и в результате удалиться на 5 шагов на север.

Задача Н. Эстафета

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

Каждый год в честь дня города в Южно-Берляндске проводится открытая эстафета. В конкурсе участвуют команды из k человек, в процессе эстафеты участники команды должны посетить n контрольных пунктов.

Контрольные пункты пронумерованы от 1 до n , место старта обозначим как пункт 0. Соревнование проходит следующим образом: первый участник из команды стартует из пункта 0, пробегает по некоторым a_1 ранее не посещенным контрольным пунктам, возвращается в пункт 0 и передает эстафету второму участнику. После этого второй участник пробегает какие-либо a_2 ранее не посещенных контрольных пунктов, возвращается и передает эстафету следующему. Эстафета продолжается, пока последний участник не посетит a_k ранее не посещенных контрольных пунктов и не вернется на старт. Передача эстафеты происходит мгновенно. Цель команды — пробежать эстафету как можно быстрее.

Учащиеся Южно-Берляндского бегового училища решили заранее подготовиться к состязанию. Они раздобыли план соревнования, из которого они узнали числа a_i , а также выяснили про каждую пару пунктов, за какое время можно успеть добежать от одного до другого. Все участники команды перемещаются с одинаковой скоростью, поэтому это время не зависит от того, кто побежит между этими пунктами.

Помогите участникам составить маршрут, в котором команда пробежит эстафету как можно быстрее.

Формат входных данных

В первой строке находятся два целых числа n и k ($1 \leq n \leq 18$, $1 \leq k \leq n$) — число контрольных пунктов и число участников в команде.

Во второй строке находятся k целых чисел a_i ($1 \leq a_i \leq n$, $a_1 + a_2 + \dots + a_k = n$) — число контрольных пунктов, которые должен пробежать i -й участник.

В следующих $n + 1$ строках находится по $n + 1$ целому числу $b_{i,j}$ ($1 \leq b_{i,j} \leq 10^6$, $b_{i,j} = b_{j,i}$, $b_{i,i} = 0$) — время, за которое можно добежать от i -го пункта до j -го для i и j от 0 до n .

Формат выходных данных

В единственной строке выведите одно целое число — минимальное время, за которое команда может пробежать эстафету.

Пример

<code>run.in</code>	<code>run.out</code>
2 2 1 1 0 1 2 1 0 3 2 3 0	6
4 2 2 2 0 1 4 2 5 1 0 2 6 6 4 2 0 6 6 2 6 6 0 2 5 6 6 2 0	16

Задача I. Сапсан

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

Лера часто ездит по работе из Санкт-Петербурга в Москву и обратно. Так как дела у нее всегда срочные, добирается до места назначения она всегда на Сапсане. Как известно, в каждом вагоне Сапсана расположено ровно n мест, а именно $n/2$ рядов по два места в каждом (n четное).

Однажды по пути домой после деловой встречи у Леры не было соседа, и ей стало скучно. Поэтому она задалась вопросом: сколько максимум человек можно посадить в вагон Сапсана, чтобы ровно у половины людей был сосед. Помогите Лере ответить на этот сложный вопрос.

Формат входных данных

В первой и единственной строке входного файла дано число n ($2 \leq n \leq 10^9$) — количество мест в вагоне Сапсана. Гарантируется, что число n четное.

Формат выходных данных

В единственной строке выходного файла выведите максимальное количество человек, которое можно посадить в вагон так, что ровно у половины из них есть сосед.

Примеры

<code>sapsan.in</code>	<code>sapsan.out</code>
20	12

Пояснение к примеру

На рисунке приведено одно из возможных размещений пассажиров в примере. Заштрихованные клетки соответствуют занятым местам.



Задача J. Звезды на погонах

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

В батальоне непонятного назначения действует правило, что у каждого офицера должно быть не менее a и не более b звезд на погоне, при этом ни у каких двух офицеров не должно быть равного числа звезд.

В результате понижения в звании в батальон сослали офицера $\dot{Й}$, у которого на погоне до понижения было c звезд. Теперь командиру батальона положено лишить его части звезд на погоне, в результате чего число звезд на его погоне должно стать строго меньше c .

Командир батальона исследовал вопрос и выяснил, что минимальное положительное число звезд, которое можно удалить с погона офицера $\dot{Й}$, чтобы правило выполнялось, равно d , а максимальное — e . Командир незамедлительно сообщил об этом офицеру $\dot{Й}$.

Теперь офицера $\dot{Й}$ заинтересовал вопрос: какое минимальное и максимальное количество офицеров могло быть в батальоне до его прибытия? При этом командир батальона сам офицером батальона не является, и на его погонах изображены специальные загадочные символы, а не звезды.

Формат входных данных

В первой строке содержатся пять целых чисел a, b, c, d, e ($1 \leq a, b, c, d, e \leq 1000$, $a \leq b$, $a < c$, $d \leq e$).

Гарантируется, что ситуация корректна: офицера $\dot{Й}$ можно понизить так, чтобы приведенное в условии правило выполнялось, а утверждение командира является верным.

Формат выходных данных

Выведите минимальное и максимальное возможное число офицеров в батальоне.

Пример

<code>stars.in</code>	<code>stars.out</code>
10 18 20 5 8	5 7
2 10 5 1 3	0 7

Задача К. Пробки

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

Недавно Женя устроился в компанию «Гуглекс», и сегодня его первый рабочий день. Женя живет на другом берегу реки и для проезда на работу решил воспользоваться паромом. Неожиданно он обнаружил, что съехать с парома на шоссе — не такая уж простая задача.

Машины на съезде с парома стоят в n полос, в i -й из которых стоит c_i машин, а непосредственно перед съездом находится светофор. Раз в минуту он включается зеленым и несколько машин с парома съезжают на берег. Женя заметил, что за один зеленый сигнал светофоров количество машин, которые могут успеть проехать со всех полос, суммарно не превосходит k .

Понимая, что он не один такой, кому не нравится стоять в пробках, Женя ввел понятие «злости водителя». Злость водителя в некоторый момент времени равна количеству машин, которые стоят перед ним в этот момент в той же полосе. Например, если в полосе стоит 3 машины, то у водителя, чья машина находится ближе всего к светофору, злость равна 0, у следующего — 1, а у последнего — 2. Как настоящий программист, Женя сразу же начал думать, как можно оптимизировать процесс съезда с парома.

Оптимальным ему показался следующий вариант: в каждой полосе перед съездом поставить шлагбаум, который будет за один зеленый сигнал светофора пропускать только определенное количество машин. А именно, i -й полосе будет сопоставлено целое число k_i — максимальное количество машин, которые могут съехать за один зеленый сигнал. Сумма всех k_i должна быть равна k .

В качестве параметра оптимизации Женя выбрал суммарную злость всех водителей за все время съезда. Каждую минуту после очередного включения зеленого света просуммируем злость всех еще не покинувших паром водителей. Просуммируем получившиеся величины за все минуты, пока хотя бы один водитель еще находится на пароме. Женя хочет подобрать k_i таким образом, чтобы получившаяся величина была минимальна.

Пока Женя стоит в пробке, помогите ему написать программу, находящую оптимальные k_i , чтобы он смог оправдаться перед начальством за опоздание в первый же рабочий день.

Формат входных данных

В первой строке входного файла содержатся два числа n, k ($1 \leq n \leq k \leq 300$) — количество полос на пароме и ограничение на количество машин, проезжающих за один зеленый сигнал светофора со всех полос.

Во второй строке входного файла находятся n чисел c_i ($1 \leq c_i \leq 100\,000$) — количество машин, стоящих в i -й полосе в начальный момент времени.

Формат выходных данных

В первой строке выходного файла выведите число a — минимальную суммарную злость всех водителей за все время движения машин через светофоры.

В следующей строке выведите n натуральных чисел k_i — ограничения на количество машин, проезжающих за один зеленый сигнал, для каждой полосы. Сумма чисел должна быть равна k . Если оптимальных решений несколько, можно вывести любое.

Примеры

<code>traffic.in</code>	<code>traffic.out</code>
3 4	1
1 2 4	1 1 2
3 4	7
1 2 6	1 1 2

Пояснение к примеру

В первом тестовом примере в оптимальном ответе за первый зеленый сигнал светофоров через первый шлагбаум проедет 1 машина, через второй — 1, через третий — 2. После этого злость

оставшихся водителей будет равна $0 + 1 + 0 = 1$ (в первом ряду не осталось машин, во втором ряду осталась одна машина, злость ее водителя равна 0, в третьем ряду осталось две машины, злость водителей в них равны 1 и 0). За второй зеленый сигнал светофоров проедут все оставшиеся машины. Получаем, что суммарная злость всех водителей за все время движения машин равна 1.

Во втором тестовом примере оптимальные числа k_i такие же, однако суммарная злость водителей другая. После первого зеленого сигнала светофоров на первой полосе не останется машин, на второй полосе останется одна машина, злость ее водителя будет равна 0, а на третьей полосе останется 4 машины, злость водителей будет равна 3, 2, 1 и 0. После второго зеленого сигнала светофора на первой и второй полосах не останется машин, а на третьей полосе останется 2 машины, злость их водителей будет равна 1 и 0. После третьего зеленого сигнала машин на пароме не останется. Итого, суммарная злость водителей за все время равна $0 + 3 + 2 + 1 + 0 + 1 + 0 = 7$.