

Задача 1. Кинотеатр

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.4 секунд
Ограничение по памяти:	512 мегабайт

В кинотеатре места часто расставляют со сдвигом соседних рядов для удобства зрителей. Пусть в таком кинотеатре N мест в 1-м, 3-м, 5-м и всех нечётных рядах и $N + 1$ место во 2-м, 4-м и всех чётных рядах. Места в рядах нумеруются от 1 до N в нечётных рядах и от 1 до $N + 1$ в чётных рядах.

Касса продаёт билеты подряд: сначала в 1-й ряд на места с 1-го по N -е, потом — во 2-й ряд на места с 1-го по $N + 1$ -е, затем в 3-й ряд с 1-го места и т.д.

Определите номер ряда и номер места для K -го проданного билета.

Формат входных данных

Программа получает на вход два целых числа. В первой строке записано число N ($1 \leq N \leq 10^9$) — количество мест в 1-м ряду кинотеатра. Во второй строке записано число K — порядковый номер проданного билета ($1 \leq K \leq 2 \times 10^9$).

Формат выходных данных

Программа должна вывести два числа: номер ряда и номер места K -го проданного билета.

Система оценки

Решения, правильно работающие, когда входные числа не превосходят 1000, будут оцениваться в 60 баллов.

Пример

стандартный ввод	стандартный вывод
10 25	3 4

Замечание

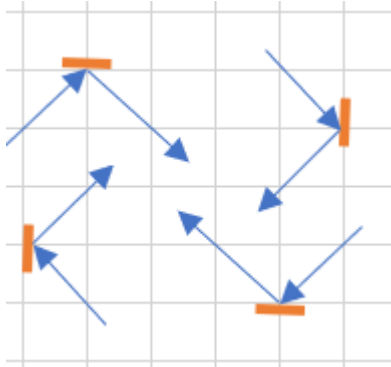
Пояснение к примеру из условия. Билеты с 1 по 10 будут проданы в первый ряд. Билеты с 11 по 21 будут проданы во второй ряд. В третий ряд будут проданы билеты, начиная с 22-го, 25-й билет окажется на 4-м месте 3-го ряда.

Задача 2. Лазерная пушка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

У игрока в космической стрелялке есть очень мощная лазерная пушка. Но она неподвижна и может стрелять только в одном направлении. Игрок может расставить на игровом поле двусторонние зеркала, меняющие ход луча, чтобы поражать врагов.

Введём декартову систему координат с центром, где расположена пушка, то есть пушка имеет координаты $(0; 0)$. Пушка стреляет в направлении точки $(1; 1)$. Игрок может поставить зеркала **в точках с целочисленными координатами**. Зеркала могут быть горизонтальными или вертикальными, попадание луча в зеркало меняет траекторию луча по законам отражения света. Некоторые возможные варианты отражения луча от зеркала изображены на рисунке.



Вам необходимо расставить **минимальное количество** зеркал так, чтобы лазерный луч поразил цель.

Формат входных данных

Программа получает на вход два целых числа X и Y , не превосходящих по модулю 10000, записанные в разных строках — координаты цели. Точка $(X; Y)$ не совпадает с началом координат.

Формат выходных данных

Программа должна вывести в первой строке число N — необходимое количество зеркал.

Следующие N строк должны содержать информацию о каждом зеркале. В i -й строке должны быть записаны через пробелы два целых числа x_i и y_i и один символ t_i , обозначающие координаты $(x_i; y_i)$ точки, в которых установлено i -е зеркало, и тип этого зеркала t_i . Если t_i является символом «V», то i -е зеркало размещено вертикально, если же t_i является символом «H», то зеркало размещено горизонтально. Например, строка «-2 5 H» обозначает горизонтальное зеркало в точке $(-2; 5)$. Зеркала можно выводить в любом порядке. Зеркало нельзя размещать в точке $(0; 0)$, также нельзя размещать два зеркала в одной точке. Значения x_i и y_i не должны по модулю превосходить 100 000. Также, разумеется, нельзя допустить, чтобы отражённый луч попал в пушку.

Если вариантов ответа несколько, выведите любой из них.

Если поразить цель в соответствии с условиями задачи невозможно, программа должна вывести одно число «-1».

Если для поражения цели зеркала не нужны, программа должна вывести одно число «0».

Система оценки

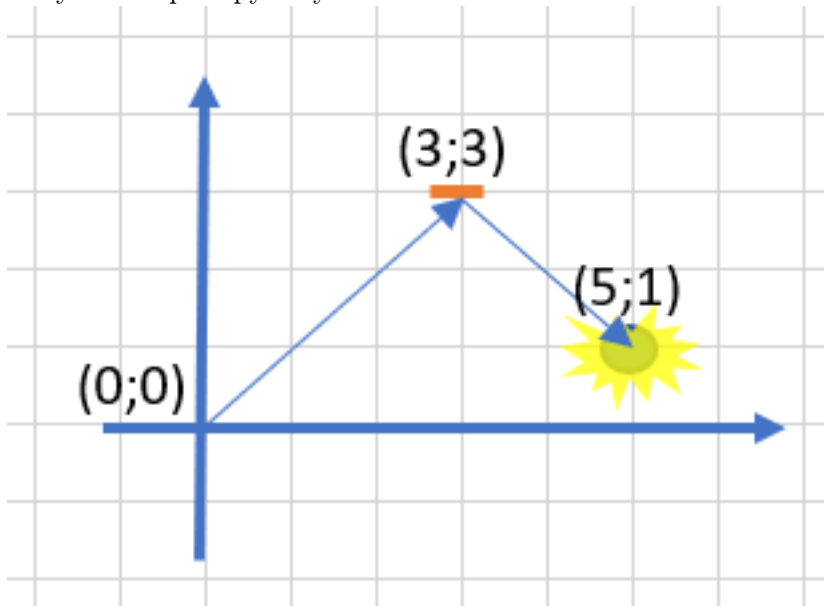
Решения, правильно работающие только для $X \geq 0$ и $Y \geq 0$, будут оцениваться в 35 баллов.

Пример

стандартный ввод	стандартный вывод
5	1
1	3 3 H

Замечание

Рисунок к примеру из условия.



Задача 3. Не был предателем...

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Гриша уже несколько недель отработывает свои навыки в новомодной онлайн-игре про команду космического корабля, вычисляющую предателей среди них. Так как игра очень популярна, появились игроки, которые договариваются между собой о каких-то способах коммуницировать заранее. Таких людей называют заговорщиками.

Заговорщики действуют по следующему алгоритму. В начале игры каждый из заговорщиков пишет в общий чат строку T — ключ шифрования. Далее в течение игры игрок придумывает строку S , записывает её N раз подряд и отправляет в чат. Для того, чтобы получить зашифрованное сообщение, остальным заговорщикам нужно посчитать, сколько раз в этой повторённой N раз строке S встречается ключ шифрования T . Чат обновляется слишком быстро и Гриша не успевает это сделать руками. Помогите Грише решить эту задачу.

Формат входных данных

В первой строке входных данных записана строка T , содержащая не более 300 символов — ключ шифрования.

Во второй строке записана строка S , её длина также не превосходит 300.

В третьей строке записано целое число N , $1 \leq N \leq 5 \times 10^6$ — количество повторений строки S .

Все строки состоят только из заглавных английских букв.

Формат выходных данных

Программа должна вывести единственное целое число — количество вхождений строки T в строку S , повторённую N раз. Под одним вхождением подразумевается один способ выбрать подстроку, то есть несколько подряд идущих символов строки, совпадающих со строкой T , не меняя порядок следования этих символов.

Система оценки

Решения, правильно работающие, когда длины строк T и S , а также число N не превосходят 100, будут оцениваться в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
MON AMONGUS 3	3
ABA BABA 3	5

Замечание

Пояснение ко второму примеру. Если строку BABA повторить 3 раза, получится BABA BABA BABA. В полученной строке подстрока ABA встречается 5 раз: BABA BABA BABA, BABABA BABA BABA, BABA BABA BABA, BABA BABA BABA, BABA BABA BABA.

Задача 4. Железная дорога

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Вите на день рождения подарили Очень Большую Игрушечную Железную Дорогу. Она представляет собой два параллельных пути, по которым движутся поезда во встречных направлениях. В центре находится станция, а железная дорога очень большая, поэтому можно считать пути бесконечными в обе стороны.

Витя расставил поезда на железной дороге и одновременно запустил их, включив электропитание. Все поезда движутся с одинаковыми скоростями в одном из двух возможных направлений. Но когда-нибудь поезда придётся остановить и убрать игру, а поскольку поезд не может развернуться и начать движение в противоположном направлении, Вите придётся самому собирать поезда руками и переносить их на станцию. Витя хочет выбрать такой момент остановки всех поездов, чтобы ему пришлось потратить минимальное число усилий для того, чтобы собрать после этого все поезда на станции вместе, то есть в этот момент времени сумма расстояний всех поездов до станции была бы минимальной.

Формат входных данных

Введём на железной дороге координаты, считая, что станция находится в начале координат, а все поезда первоначально находятся в целочисленных точках координатной прямой. Сами поезда также будем считать точками. За одну секунду координаты всех поездов изменяются на $+1$ или на -1 . Поезда движутся с равными скоростями, первоначально никакие два поезда, движущиеся в одном направлении, не находятся в одной точке.

Первая строка входных данных содержит целое число N — количество поездов, движущихся в положительном направлении. Вторая строка входных данных содержит целое число M — количество поездов, движущихся в отрицательном направлении. Ограничения: $N \geq 0$, $M \geq 0$, $1 \leq N + M \leq 10^5$.

Следующие N строк содержат N чисел a_i ($|a_i| \leq 10^9$) — первоначальные координаты поездов, движущихся в положительном направлении. Все числа a_i различны и заданы в порядке возрастания.

Следующие M строк содержат M чисел b_j ($|b_j| \leq 10^9$) — первоначальные координаты поездов, движущихся в отрицательном направлении. Все числа b_j различны и заданы в порядке возрастания.

Формат выходных данных

Программа должна вывести единственное целое число t — момент времени в секундах после запуска игры, в который суммарное расстояние всех поездов до станции будет минимальным. Если возможных правильных ответов несколько, то программа должна вывести любой из них. Если после старта игры суммарное расстояние всех поездов до станции будет всегда больше, чем в момент запуска игры, то правильным ответом будет $t = 0$.

Система оценки

Решения, правильно работающие, когда $N \leq 100$, $|a_i| \leq 100$, $|b_i| \leq 100$, будут оцениваться в 30 баллов.

Решения, правильно работающие, когда $N \leq 100$, $|a_i| \leq 10^9$, $|b_i| \leq 10^9$, будут оцениваться в 60 баллов.

Пример

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

Замечание

В примере из условия 3 поезда движутся в положительном направлении, 2 поезда движутся в отрицательном направлении. Начальные координаты поездов, движущихся в положительном направлении равны -5 , -2 , 4 , движущихся в отрицательном направлении равны 1 и 4 . Запишем, что при $t = 0$ координаты поездов $(-5, -2, 4, 1, 4)$. Суммарное расстояние всех поездов до станции равно 16 . В момент $t = 1$ координаты поездов будут $(-4, -1, 5, 0, 3)$, суммарное расстояние до станции равно 13 . В момент $t = 2$ координаты поездов будут $(-3, 0, 6, -1, 2)$, суммарное расстояние до станции равно 12 . В момент $t = 3$ координаты будут $(-2, 1, 7, -2, 1)$, суммарное расстояние до станции равно 13 . Ответ на этот тест $t = 2$.

Задача 5. Джерримендеринг

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Джерримендеринг — разделение территории на избирательные округа неестественным образом с целью искусственного изменения соотношения политических сил в них и, как следствие, в целом на территории проведения выборов. Например, при необходимости обеспечить победу на территории партии X (если от одного избирательного округа избирается один кандидат или один выборщик), нужно всех противников X сосредоточить по округам, где X не сможет выиграть, а всех сторонников X распределить так, чтобы они обеспечивали уверенную победу с небольшим перевесом в нужных округах. Например, в тесте из условия всего за X голосует 10 человек, а против X голосует 15 человек, но, благодаря специальному разделению по округам, X выигрывает в двух избирательных округах из трёх.

В этой задаче избирательная территория представляет собой улицу, на которой в ряд расположены N домов. В i -м доме проживает a_i человек, и все они голосуют одинаково: либо за партию X , либо за другую партию. Улицу необходимо разбить на три избирательных округа, от каждого избирательного округа будет избираться один кандидат, и необходимо произвести такую нарезку улицы на три избирательных округа, чтобы минимум в двух округах из трёх выиграл кандидат от партии X . Кандидат от партии X выигрывает, если за него голосует более половины избирателей, проживающих в домах данного избирательного округа. Но чтобы вас не заподозрили в джерримендеринге, необходимо, чтобы каждый избирательный округ представлял собой непрерывный отрезок из номеров домов, то есть сначала вдоль по улице идут дома первого избирательного округа, затем — второго, затем — третьего. Каждый избирательный округ должен содержать как минимум один дом.

Формат входных данных

Первая строка входных данных содержит целое число N ($3 \leq N \leq 10^5$) — количество домов на улице. Следующие N строк содержат по одному целому числу a_i ($0 < |a_i| \leq 10^4$). Если $a_i > 0$, то в i -м доме проживает a_i избирателей, голосующих за кандидата от партии X . Если $a_i < 0$, то в i -м доме проживает $|a_i|$ избирателей, голосующих против кандидата от партии X .

Формат выходных данных

Если возможно разделить N домов на три округа так, что минимум в двух округах выигрывает кандидат от партии X , программа должна вывести три целых положительных числа N_1, N_2, N_3 , $N_1 + N_2 + N_3 = N$, соответствующих количеству домов в первом, втором и третьем избирательном округе от начала улицы. При таком разбиении минимум в двух округах из трёх должен выигрывать кандидат от партии X . Если возможно несколько таких разбиений, необходимо вывести любое из них.

Если искомое разбиение не существует, программа должна вывести одно число 0.

Система оценки

Решения, правильно работающие при $N \leq 100$, будут оцениваться в 40 баллов.

Решения, правильно работающие при $N \leq 1000$, будут оцениваться в 70 баллов.

Пример

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

Замечание

Пояснение к примеру из условия. На улице расположены 7 домов, избиратели в них распределены так: $(-3, -5, 3, -4, 2, 5, -3)$. Правильный ответ: 4, 1, 2. При таком разбиении в первом округе оказываются 4 дома: $(-3, -5, 3, -4)$. В этом округе за X голосует 3 избирателя, против — 12 избирателей и X разгромно проигрывает. В следующем округе один дом, в котором 2 избирателя голосуют за X , в этом округе X выигрывает. В третьем округе два дома: $(5, -3)$, и в этом округе X тоже выигрывает. Итого X выигрывает в двух округах.