

Задача 1. Метро

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

На некоторых кросс-платформенных станциях метро (как, например, «Третьяковская») на разные стороны платформы приходят поезда разных направлений. Таня договорилась встретиться с подругой на такой станции, но поскольку подруга приехала из другого часового пояса, то из-за джетлага сильно проспала, и Тане пришлось долго её ждать. Поезда всегда ходят точно по расписанию, и Таня знает, что поезд стоит на платформе ровно одну минуту, а интервал между поездами (время, в течение которого поезда у платформы нет) составляет a минут для поездов на первом пути и b минут для поездов на втором пути. То есть на первый путь приезжает поезд и стоит одну минуту, затем в течение a минут поезда у платформы нет, затем в течение одной минуты у платформы стоит следующий поезд и т. д.

Пока Таня стояла на платформе, она насчитала n поездов на первом пути и m поездов на втором пути. Определите минимальное и максимальное время, которое Таня могла провести на платформе, или сообщите, что она точно сбилась со счёта.

Все поезда, которые видела Таня, она наблюдала в течение всей минуты, то есть Таня не приходит и не уходит с платформы посередине той минуты, когда поезд стоит на платформе.

Формат входных данных

Первая строка входных данных содержит число a — интервал между поездами на первом пути. Вторая строка содержит число b — интервал между поездами на втором пути. Третья строка содержит число n — количество поездов на первом пути, которые увидела Таня. Четвёртая строка содержит число m — количество поездов на втором пути, которые увидела Таня. Все числа — целые, от 1 до 1000.

Формат выходных данных

Программа должна вывести два числа: минимальное и максимальное время в минутах, которое Таня могла стоять на платформе, или одно число -1 , если Таня точно ошиблась.

Примеры

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

Замечание

В первом примере по первому пути поезда ходят через 1 минуту. По второму — через 3. Стоя на платформе 5, 6 или 7 минут, Таня могла насчитать 3 поезда на первом пути и 2 на втором.

Задача 2. Площадь

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 0,5 секунды |
| Ограничение по памяти: | 512 мегабайт |

Городская площадь имеет размер $n \times m$ и покрыта квадратной плиткой размером 1×1 . При плановой замене плитки выяснилось, что новой плитки недостаточно для покрытия всей площади, поэтому было решено покрыть плиткой только дорожку по краю площади, а в центре площади разбить прямоугольную клумбу (см. рисунок к примеру). При этом дорожка должна иметь одинаковую ширину по всем сторонам площади. Определите максимальную ширину дорожки, которую можно выложить из имеющихся плиток.

Формат входных данных

Первая и вторая строки входных данных содержат по одному числу n и m ($3 \leq n \leq 2 \times 10^9$, $3 \leq m \leq 2 \times 10^9$) — размеры площади.

Третья строка содержит количество имеющихся плиток t , $1 \leq t < nm$.

Обратите внимание, что значение t может быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные числа (тип `int64` в языке Pascal, тип `long long` в C и C++, тип `long` в Java и C#).

Формат выходных данных

Программа должна вывести единственное число — максимальную ширину дорожки, которую можно выложить из имеющихся плиток.

Система оценивания

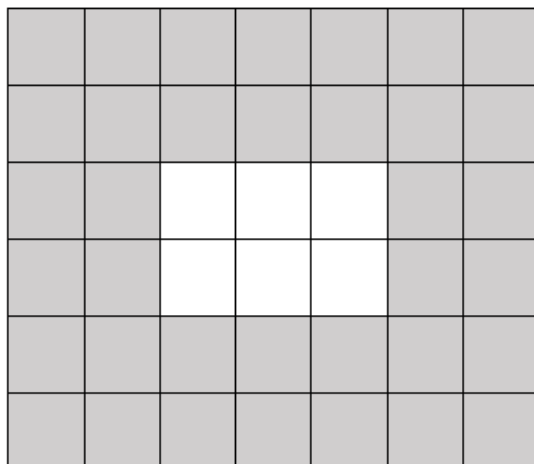
Решение, правильно работающее только для случаев, когда числа n и m не превосходят 1000, будет оцениваться в 60 баллов.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 7 38 | 2 |

Замечание

Пояснение к примеру. Площадь имеет размеры 6×7 , из 38 плиток можно выложить дорожку шириной в 2 плитки.



Две плитки
осталось



Задача 3. Много пирожных

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

На кондитерской фабрике есть n видов пирожных, пирожных i -го вида на фабрике a_i штук. Было принято решение отвезти пирожные на продажу на ярмарку, но директор фабрики решил, что кондитерские изделия на ярмарочной витрине должны быть выложены одинаковыми рядами, при этом пирожных каждого вида должно быть одинаковое количество. Необязательно отвозить на ярмарку все виды пирожных, можно выбрать некоторые виды и взять одинаковое число пирожных каждого выбранного вида.

Помогите директору отвезти на ярмарку наибольшее число пирожных — найдите, сколько видов пирожных и сколько пирожных каждого вида нужно отвезти на ярмарку.

Формат входных данных

Первая строка входных данных содержит число n — количество видов пирожных на фабрике, $1 \leq n \leq 10^5$. Следующие n строк содержат по одному числу a_i — количество пирожных i -го вида, $1 \leq a_i \leq 10^5$. Сумма всех значений a_i не превосходит 2×10^9 .

Формат выходных данных

Программа должна вывести два целых числа. Первое число равно количеству видов пирожных, которые необходимо выбрать для ярмарки. Второе число равно количеству пирожных каждого выбранного вида, которые нужно отвезти на ярмарку.

Если возможных ответов несколько, выведите любой из них.

Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 10, будет оцениваться в 30 баллов.

Решение, правильно работающее только для случаев, когда $n \leq 100$, будет оцениваться в 60 баллов.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 | 2 7 |
| 4 | |
| 10 | |
| 7 | |

Замечание

Пояснение к примеру. Имеется 3 вида пирожных количеством 4, 10 и 7 штук. Наилучший ответ будет, если взять по 7 пирожных второго и третьего вида.

Задача 4. Космические шахматы

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

В космические шахматы играют на бесконечной доске, поэтому клетки нумеруют парой чисел (см. пример и рисунок к нему). Фигуры ходят по обычным правилам. Составьте маршрут шахматного коня из клетки $(0; 0)$ в заданную клетку $(x; y)$.

Напомним, что конь за один ход перемещается на одну клетку по одной оси и на две по другой, то есть, например, из клетки $(0; 0)$ он за один ход может попасть в клетки $(1; 2)$, $(2; 1)$, $(-1; 2)$, $(2; -1)$, $(1; -2)$, $(-2; 1)$, $(-1; -2)$ и $(-2; -1)$.

В качестве ответа Вам нужно вывести любой (не обязательно кратчайший) маршрут с началом в $(0; 0)$ и концом в $(x; y)$, длина которого не больше 10^5 ходов.

Формат входных данных

Программа получает на вход два целых числа x и y , записанных в отдельных строках, — координаты конечной клетки маршрута коня. Клетка $(x; y)$ не совпадает с началом координат. $|x| \leq 10^5$, $|y| \leq 10^5$.

Формат выходных данных

Программа должна вывести последовательность ходов, один ход в отдельной строке. В i -й строке должно быть выведено два числа x_i и y_i через пробел — координаты клетки, в которой окажется конь после i -го хода. Количество ходов не должно превышать 10^5 . Последний ход должен вести в заданную клетку.

Система оценивания

Решение, правильно работающее только для случаев, когда $|x| \leq 1$, $|y| \leq 1$, будет оцениваться в 16 баллов.

Решение, правильно работающее только для случаев, когда $|x| \leq 100$, $|y| \leq 100$, будет оцениваться в 40 баллов.

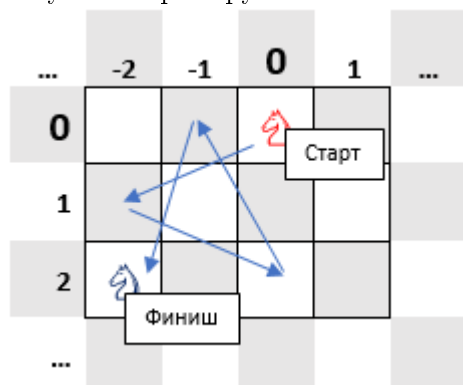
Решение, правильно работающее только для случаев, когда $|x| \leq 15\,000$, $|y| \leq 15\,000$, будет оцениваться в 70 баллов.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| -2 | -2 1 |
| 2 | 0 2 |
| | -1 0 |
| | -2 2 |

Замечание

Рисунок к примеру



Задача 5. Городские центральные диаметры

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

В одном большом городе готовятся к открытию новой ветки наземного метро. Она пролегает между двумя городами в пригороде по разные стороны от самого города, но проходя через него насквозь. Такую модель наземного метро назвали Городские центральные диаметры (ГЦД).

В рамках подготовки к запуску ГЦД было разработано специальное расписание, содержащее n рейсов в одном направлении и m рейсов в обратном. Для каждого рейса определены a_i — время отправления с первой конечной станции и b_i — время прибытия на вторую конечную станцию, для обратных рейсов c_j — время отправления со второй станции и d_j — время прибытия на первую станцию. Времена измеряются в минутах от начала дня. Внутри большого города поезда могут передвигаться по различным маршрутам, поэтому поезд, отправившийся позже какого-то другого поезда, может прибыть раньше него.

Проекты такого масштаба ещё не запускались, а значит, будут происходить непредвиденные события и поезда будут задерживаться. Аналитики компании, обслуживающей ГЦД, посчитали, что при любых обстоятельствах поезд может опоздать на конечную станцию не более чем на t минут. Поезд может отправиться выполнять следующий рейс сразу же, как только закончил предыдущий.

Компании поручено во что бы то ни стало обеспечить выполнение каждого рейса. Определите, какое минимальное количество поездов необходимо иметь, чтобы ни один рейс гарантированно не был отменён даже в случае возможных задержек всех поездов. Поезда обязаны отправляться от начальных станций строго по расписанию.

В начале и в конце дня поезда могут находиться на любой из станций. Перемещаться между станциями во время дня, кроме как по данному расписанию, поезда не могут.

Формат входных данных

Первая строка входных данных содержит максимальное время задержки при выполнении рейса t , $0 \leq t \leq 10^9$.

Следующая строка содержит число n — количество рейсов в расписании в одну сторону, $0 \leq n \leq 100$. Следующие $2n$ строк содержат числа $a_1, b_1, a_2, b_2, \dots, a_n, b_n$ — время отправления поездов от первой станции и время их прибытия на вторую станцию, $0 \leq a_i < b_i \leq 10^9$.

Следующая строка содержит число m — количество поездов в расписании в обратную сторону, $0 \leq m \leq 100$. Следующие $2m$ строк содержат время отправления c_j и прибытия d_j поездов в обратном направлении в аналогичном формате, $0 \leq c_j < d_j \leq 10^9$.

Поезда в расписании перечислены в произвольном порядке, не обязательно в порядке возрастания времени.

Формат выходных данных

Программа должна вывести одно целое число — минимальное количество поездов, необходимое для выполнения данного расписания.

Система оценивания

Решения, правильно работающие для случая, когда значения t, a_i, b_i, c_j, d_j не превосходят 100, будут оцениваться в 40 баллов.

Решения, правильно работающие для случая, когда значения t, a_i, b_i, c_j, d_j не превосходят 10^5 , будут оцениваться в 70 баллов.

Примеры

| стандартный ввод | стандартный вывод |
|---------------------------------------------------------|-------------------|
| 4 2 3 8 5 10 1 11 15 | 3 |
| 1 2 15 18 7 9 2 11 14 1 3 | 1 |

Замечание

В первом тесте $t = 4$, $n = 2$, $m = 1$. Первый поезд уезжает в момент времени $a_1 = 3$ и прибывает в $b_1 = 8$. Второй поезд отправляется в момент времени $a_2 = 5$ и прибывает в $b_2 = 10$. Со второй станции на первую поезд отправляется в $c_1 = 11$ и прибывает в $d_1 = 15$. Первый поезд не успевает выполнить обратный рейс, т.к. он прибывает на вторую станцию по расписанию в 8, но может опоздать на 4 минуты, т.е. он сможет выполнить обратный рейс только при отправлении не ранее 12 минут. Поэтому для реализации расписания необходим третий поезд.

Во втором тесте $t = 1$, $n = 2$, $m = 2$. Поезд поочерёдно выполняет все рейсы. Сначала он выполняет рейс от второй станции к первой — отправлением в $c_2 = 1$, прибытием в $d_2 = 3$. Потом рейс от первой станции ко второй — отправлением $a_2 = 7$, прибытием $b_2 = 9$. Затем он выполняет рейс от второй станции к первой — отправлением $c_1 = 11$, прибытием $d_1 = 14$. Наконец, он выполняет рейс от первой станции ко второй — отправлением в $a_1 = 15$, прибытием в $b_1 = 18$. Максимальное время задержки в пути равно 1, поэтому один поезд сможет выполнить все эти рейсы.