Задача A. Пожарная часть на прямой

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

Город Энск расположен на координатной оси. В городе n домов с координатами $x_1, x_2, \ldots, x_n$.

Мэр города хочет выбрать место для пожарной части так, чтобы расстояние от неё до самого далёкого дома было как можно меньше. Помогите ему найти такое место.

Формат входных данных

В первой строке задано целое число n — количество домов ($1 \leqslant n \leqslant 10^5$).

Во второй строке заданы n целых чисел $x_1, x_2, \ldots, x_n$ — координаты домов ($|x_i| \leqslant 10^9$).

Формат выходных данных

В первой строке выведите одно вещественное число d — расстояние от пожарной части до самого далёкого дома.

Во второй строке выведите одно вещественное число x_0 — координату пожарной части.

Ответ будет считаться верным, если:

- расстояние от x_0 до самого далёкого дома отличается от d не больше чем на 0.1;
- расстояние от x_0 до самого далёкого дома отличается от оптимального не больше чем на 0.1.

Примеры

стандартный ввод	*стандартный вывод*
5 1 3 3 2 5	2.0 3.0
4 9 3 2 6	3.5 5.5

Задача B. Магазин на прямой

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

Город Энск расположен на координатной оси. В городе n домов с координатами $x_1, x_2, \ldots, x_n$.

Мэр города хочет выбрать место для магазина так, чтобы сумма расстояний от него до всех домов была как можно меньше. Помогите ему найти такое место.

Формат входных данных

В первой строке задано целое число n — количество домов ($1 \leqslant n \leqslant 10^5$).

Во второй строке заданы n целых чисел $x_1, x_2, \ldots, x_n$ — координаты домов ($|x_i| \leqslant 10^9$).

Формат выходных данных

В первой строке выведите одно вещественное число d — сумму расстояний от магазина до всех домов.

Во второй строке выведите одно вещественное число x_0 — координату магазина.

Ответ будет считаться верным, если:

- сумма расстояний от x_0 до всех домов отличается от d не больше чем на 0.1;
- сумма расстояний от x_0 до всех домов отличается от оптимальной не больше чем на 0.1.

Примеры

стандартный ввод	***стандартный вывод***
5 1 3 3 2 5	5.0 3.0
4 9 3 2 6	10.0 5.5

Задача C. Магазин на плоскости

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

Город Энск расположен на координатной плоскости. В городе n домов с координатами (x_1, y_1), (x_2, y_2), …, (x_n, y_n).

Город состоит из улиц, параллельных осям координат. Поэтому расстоянием между двумя точками $A = (x_A, y_A)$ и $B = (x_B, y_B)$ на плоскости считается величина $|x_B - x_A| + |y_B - y_A|$.

Мэр города хочет выбрать место для магазина так, чтобы сумма расстояний от него до всех домов была как можно меньше. Помогите ему найти такое место.

Формат входных данных

В первой строке задано целое число n — количество домов ($1 \leqslant n \leqslant 10^5$).

В следующих строках заданы координаты домов: i-я из этих строк содержит два целых числа x_i и y_i ($|x_i|, |y_i| \leqslant 10^9$).

Формат выходных данных

В первой строке выведите одно вещественное число d — сумму расстояний от магазина до всех домов.

Во второй строке выведите два вещественных числа x_0 и y_0 — координаты магазина.

Ответ будет считаться верным, если:

- сумма расстояний от (x_0, y_0) до всех домов отличается от d не больше чем на 0.1;
- сумма расстояний от (x_0, y_0) до всех домов отличается от оптимальной не больше чем на 0.1.

Примеры

стандартный ввод	**стандартный вывод**
5 1 4 3 6 3 4 2 2 5 3	10.0 3.0 4.0
4 9 1 3 6 2 4 6 7	18.0 5.5 4.0

Задача D. Пожарная часть на плоскости

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

Город Энск расположен на координатной плоскости. В городе n домов с координатами (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n).

Город состоит из улиц, параллельных осям координат. Поэтому расстоянием между двумя точками $A = (x_A, y_A)$ и $B = (x_B, y_B)$ на плоскости считается величина $|x_B - x_A| + |y_B - y_A|$.

Мэр города хочет выбрать место для пожарной части так, чтобы расстояние от неё до самого далёкого дома было как можно меньше. Помогите ему найти такое место.

Формат входных данных

В первой строке задано целое число n — количество домов ($1 \leqslant n \leqslant 10^5$).

В следующих строках заданы координаты домов: i-я из этих строк содержит два целых числа x_i и y_i ($|x_i|, |y_i| \leqslant 10^9$).

Формат выходных данных

В первой строке выведите одно вещественное число d — расстояние от пожарной части до самого далёкого дома.

Во второй строке выведите два вещественных числа x_0 и y_0 — координаты пожарной части.

Ответ будет считаться верным, если:

- расстояние от (x_0, y_0) до самого далёкого дома отличается от d не больше чем на 0.1;
- расстояние от (x_0, y_0) до самого далёкого дома отличается от оптимального не больше чем на 0.1.

Примеры

стандартный ввод	**стандартный вывод**
5 1 4 3 6 3 4 2 2 5 3	2.5 3.0 3.5
4 9 1 3 6 2 4 6 7	5.5 5.0 2.5

Задача E. Анти-расстояние

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

Рассмотрим плоскость, разбитую на квадраты со стороной 1. Выберем один квадрат и проведём из его центра оси координат параллельно его сторонам.

В каждом пятом квадрате находится препятствие: более точно, препятствиями заняты все квадраты с координатами центров $(2i + j, i - 2j)$ для всевозможных целых i и j. Иллюстрацию расстановки препятствий можно увидеть в пояснениях к примерам. Все остальные квадраты свободны.

Амелия находится в свободном квадрате A и хочет попасть в свободный квадрат B. За один шаг она может из квадрата перейти в один из четырёх соседних, имеющих с ним общую сторону, но только если этот соседний квадрат свободен. Какое минимальное количество шагов должна сделать Амелия, чтобы оказаться в квадрате B?

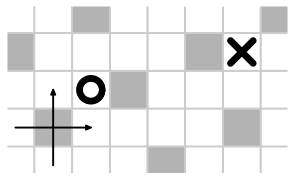
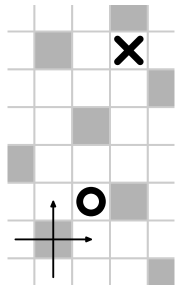
Формат входных данных

В первой строке записаны два целых числа x_1 и y_1 — координаты центра исходного квадрата A. Во второй строке записаны два целых числа x_2 и y_2 — координаты центра целевого квадрата B. Все заданные координаты лежат в пределах от 1 до 10^9. Гарантируется, что оба заданных квадрата свободны.

Формат выходных данных

Выведите одно целое число — минимальное количество шагов от исходного до целевого квадрата.

Примеры

стандартный ввод	стандартный вывод	иллюстрация
1 1 5 2	7	
1 1 2 5	5	

Задача F. Расстояние в крестах

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

Рассмотрим плоскость, разбитую на квадраты со стороной 1. Выберем один квадрат и проведём из его центра оси координат параллельно его сторонам.

Обведём крест из пяти квадратов: центрального и четырёх его соседей — квадратов, имеющих с ним общую сторону. Затем возьмём квадрат с центром в точке $(2, 1)$, добавим к нему четырёх его соседей и обведём ещё один крест из пяти квадратов. Замостим всю плоскость такими крестами: их центры будут находиться в точках с координатами $(2i + j, i - 2j)$ для всевозможных целых i и j. Иллюстрации замощения можно увидеть в пояснениях к примерам.

Эмилия стоит в центре какого-то квадрата на плоскости. За один шаг она может из квадрата перейти в один из четырёх соседних. Если при этом она переходит в другой крест замощения, то должна заплатить за этот шаг одну монетку. Перемещения же внутри креста бесплатны.

Назовём *расстоянием в крестах* между двумя квадратами A и B наименьшее возможное количество монеток, которое Эмилия должна заплатить, чтобы попасть из A в B. Заданы координаты двух точек на плоскости — центр исходного квадрата и центр целевого квадрата. Найдите расстояние в крестах между ними.

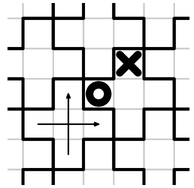
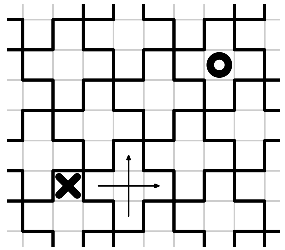
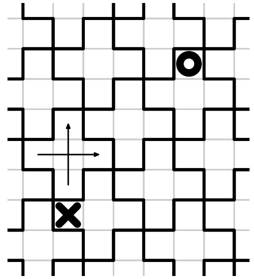
Формат входных данных

В первой строке записаны два целых числа x_1 и y_1 — координаты исходного квадрата. Во второй строке записаны два целых числа x_2 и y_2 — координаты целевого квадрата. Все заданные координаты не превосходят 10^9 по абсолютной величине.

Формат выходных данных

Выведите одно целое число — расстояние в крестах от исходного до целевого квадрата.

Примеры

стандартный ввод	*стандартный вывод*	*иллюстрация*
1 1 2 2	0	
3 4 -2 0	4	
4 3 0 -2	3	

Задача G. Кирпичный червь

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

Кирпичный червь Эдгар живёт в кирпичной стене. Стену можно упрощённо представить в виде клетчатой плоскости, бесконечной во все стороны. Каждый кирпич состоит из двух соседних клеток плоскости.

Введём систему координат так, чтобы каждая клетка имела целые координаты (x, y), а координаты соседних клеток отличались на единицу. В чётных рядах, где $y = 2j$ для какого-то целого числа j, каждому целому числу i соответствует кирпич из клеток $(2i, 2j)$ и $(2i + 1, 2j)$. В нечётных рядах, где $y = 2j+1$ для какого-то целого числа j, каждому целому числу i соответствует кирпич из клеток $(2i - 1, 2j + 1)$ и $(2i, 2j + 1)$.

Сегодня Эдгар хочет из середины клетки (x_A, y_A) попасть в середину клетки (x_B, y_B). Но он не любит проходить сквозь цемент между кирпичами. Какое наименьшее количество раз Эдгару всё же придётся переместиться между соседними кирпичами, чтобы добраться до цели?

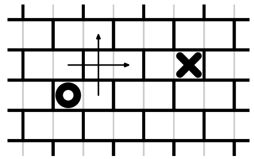
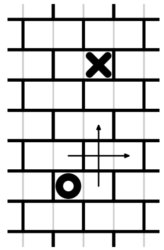
Формат входных данных

В первой строке записаны два целых числа x_A и y_A — координаты начальной клетки. Во второй строке записаны два целых числа x_B и y_B — координаты конечной клетки. Все эти числа не превосходят $1\,000\,000\,000$ по абсолютной величине. Гарантируется, что начальная и конечная клетки не совпадают.

Формат выходных данных

В первой строке выведите одно целое число — наименьшее количество раз, которое Эдгару придётся переместиться между соседними кирпичами на пути к цели.

Примеры

стандартный ввод	*стандартный вывод*	*иллюстрация*
`-1 -1` `3 0`	`2`	
`-1 -1` `0 3`	`4`	

Задача H. Проективное расстояние

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

Рассмотрим прямоугольное поле, состоящее из $w \times h$ квадратных клеток. В клетке A стоит фишка. За один ход можно подвинуть фишку в любую соседнюю по стороне клетку. Сколько ходов потребуется, чтобы добраться до клетки B?

Важное дополнение: края поля склеены по принципу проективной плоскости. А именно, i-я сверху клетка левого края — соседняя с i-й снизу клеткой правого края, а j-я слева клетка верхнего края — соседняя с j-й справа клеткой нижнего края (для всех i и j, для которых существуют эти клетки).

Формат входных данных

В первой строке заданы два целых числа w и h — ширина и высота поля ($1 \leqslant w, h \leqslant 10^8$).

Во второй строке заданы два целых числа x_A и y_A — столбец и строка клетки A.

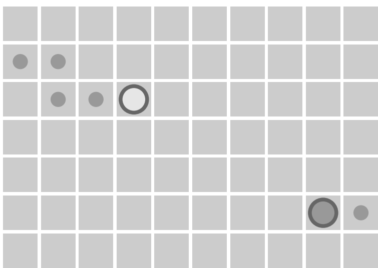
В третьей строке заданы два целых числа x_B и y_B — столбец и строка клетки B.

Столбцы пронумерованы от 1 до w слева направо, а строки — от 1 до h сверху вниз.

Формат выходных данных

Выведите одно целое число: минимальное количество ходов, за которое можно добраться из клетки A в клетку B.

Пример

стандартный ввод	стандартный вывод	иллюстрация
10 7 9 6 4 3	6	

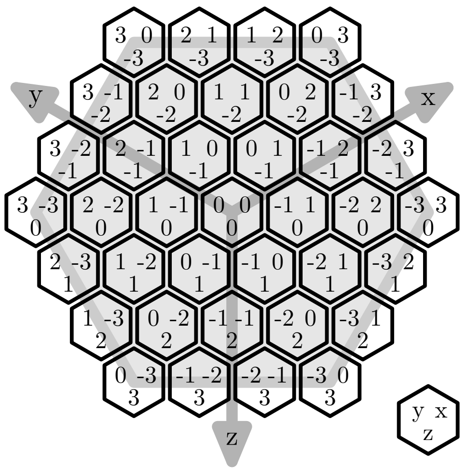
Задача I. Гексагональный тор

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

Рассмотрим плоскую доску, состоящую из равных правильных шестиугольников. Будем называть шестиугольники клетками. *Расстоянием* между двумя клетками назовём минимальное количество шагов между ними, если за один шаг можно переместиться в соседнюю по стороне клетку.

Выберем один шестиугольник и назовём его *центральной* клеткой. Из его центра — точки O — проведём три луча через середины сторон, под углом 120 градусов друг к другу. Назовём эти лучи *осями* координат Ox, Oy и Oz. Каждая ось делит все клетки на полосы, перпендикулярные оси. Для каждой оси пронумеруем эти полосы: центральная клетка будет на полосе 0, следующая полоса в направлении луча получит номер 1, следующая в обратном направлении — номер −1, и так далее. Получится система координат, показанная на картинке ниже. Каждой клетке сопоставлены три числа (x, y, z) — номера полос в нумерации вдоль соответствующих осей. Заметим, что для любой клетки выполняется свойство $x + y + z = 0$.

Рассмотрим теперь шестиугольную доску со склеенными краями. Она состоит из клеток, у которых все координаты не превосходят n по абсолютному значению. Кроме того, клетки на краях доски отождествлены:



- клетки полосы $x = n$ сверху вниз — те же, что клетки полосы $x = -n$ сверху вниз;
- клетки полосы $y = n$ снизу вверх — те же, что клетки полосы $y = -n$ снизу вверх;
- клетки полосы $z = n$ слева направо — те же, что клетки полосы $z = -n$ слева направо.

Например, на картинке $n = 3$. Клетки полосы с $x = 3$ имеют координаты $(3, 0, -3)$, $(3, -1, -2)$, $(3, -2, -1)$, $(3, -3, 0)$. Клетки полосы с $x = -3$ имеют координаты $(-3, 3, 0)$, $(-3, 2, 1)$, $(-3, 1, 2)$, $(-3, 0, 3)$. Они отождествляются в порядке перечисления: например, $(3, -1, -2)$ — это та же клетка, что и $(-3, 2, 1)$.

Заданы размер доски и координаты двух клеток на ней. Найдите расстояние между этими клетками.

Формат входных данных

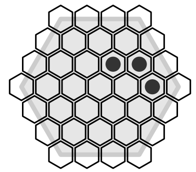
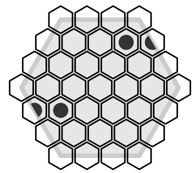
В первой строке задано целое число n — размер доски ($1 \leqslant n \leqslant 10^8$).

В следующих двух строках заданы координаты клеток: в каждой строке — по три целых числа x_i, y_i и z_i ($|x_i|, |y_i|, |z_i| \leqslant n$; $x_i + y_i + z_i = 0$).

Формат выходных данных

Выведите одно целое число: расстояние между заданными клетками.

Примеры

стандартный ввод	стандартный вывод	иллюстрация
3 1 0 -1 2 -2 0	2	
3 -2 1 1 2 0 -2	2	

Задача J. Правый поворот

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

В этой задаче требуется найти оптимальный путь движения в городе, где разрешён только правый поворот.

Мы едем по городу в машине. Карта города представляет собой бесконечную прямоугольную сетку: улицы соответствуют прямым $y = c$ для целых чисел c, а авеню соответствуют прямым $x = c$ для целых чисел c.

В городе интенсивное дорожное движение, поэтому правила движения довольно строгие. Когда машина подъезжает к перекрёстку, она может лишь остановиться, двигаться прямо или повернуть направо, изменив направление движения на 90 градусов. Так, например, если машина подъезжает к перекрёстку $(0, 1)$ со стороны перекрёстка $(0, 0)$ и собирается продолжить движение, она может двигаться к перекрёстку $(0, 2)$ или повернуть направо и двигаться к перекрёстку $(1, 1)$, но не может сразу поехать в направлении перекрёстка $(-1, 1)$.

Мы начинаем на перекрёстке $(0, 0)$ и можем поехать к перекрёстку $(0, 1)$. Найдите минимальное расстояние, которое нам потребуется проехать, чтобы добраться до перекрёстка (x, y). Направление, по которому мы приедем в (x, y), не имеет значения.

Формат входных данных

Первая строка ввода содержит одно целое число t: количество тестовых случаев ($1 \leqslant t \leqslant 10\,000$). Каждая из следующих t строк описывает один тестовый случай. Каждое описание состоит из двух целых чисел x и y, разделённых пробелом: координат пункта назначения ($|x|, |y| \leqslant 10^9$).

Формат выходных данных

Для каждого тестового случая выведите на отдельной строке одно число: минимальное расстояние, которое нам потребуется проехать, чтобы добраться до пункта назначения по правилам, описанным выше. Выводите ответы в том порядке, в котором тестовые случаи заданы во вводе.

Пример

стандартный ввод	**стандартный вывод**

`3`
`0 0`
`3 2`
`0 -1`
	`0`
	`5`
	`5`

Пояснение к примеру

В первом тестовом случае мы уже находимся в пункте назначения.

Во втором тестовом случае мы можем проехать от $(0, 0)$ до $(0, 2)$, повернуть направо и проехать от $(0, 2)$ до $(3, 2)$.

В третьем тестовом случае мы можем двигаться следующим образом:

$$(0, 0) \to (0, 1) \xrightarrow{\text{поворот}} (1, 1) \xrightarrow{\text{поворот}} (1, 0) \to (1, -1) \xrightarrow{\text{поворот}} (0, -1).$$

Отметим, что мы не можем повернуть до того, как поедем из $(0, 0)$, а после этого не можем повернуть два раза подряд, находясь в $(0, 1)$.