

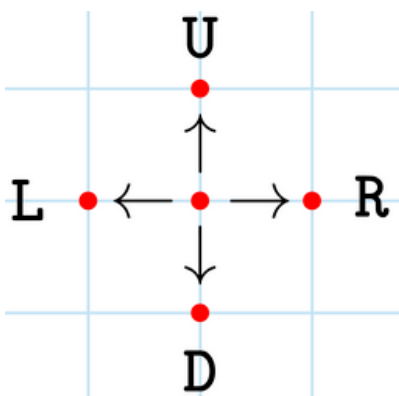
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Следование направлениям

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

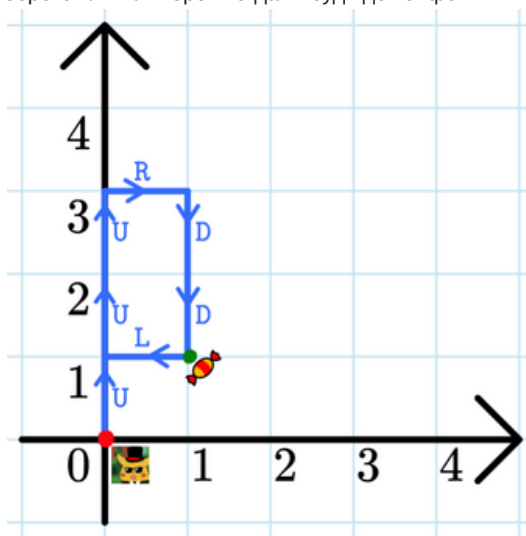
Альперен стоит в точке $(0, 0)$. Ему дана строка s длины n и он производит n перемещений: i -е перемещение происходит так:

- если $s_i = \text{L}$, он перемещается на одну клетку влево;
- если $s_i = \text{R}$, он перемещается на одну клетку вправо;
- если $s_i = \text{U}$, он перемещается на одну клетку вверх;
- если $s_i = \text{D}$, он перемещается на одну клетку вниз.



Если Альперен начинает в точке в центре, он может сделать четыре различных перемещения, как показано на рисунке.

В точке $(1, 1)$ находится конфета (то есть на одну клетку вверх и на одну клетку вправо от начальной точки, в которой находится Альперен). Определите, доберется ли Альперен когда-нибудь до конфеты.



Alperen's path in the first test case.

Входные данные

Первая строка содержит целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Первая строка каждого набора содержит целое число n ($1 \leq n \leq 50$) — длину строки.

Вторая строка каждого набора содержит строку s длины n состоящую из символов **L**, **R**, **D**, и **U**, обозначающую перемещения Альперена.

Выходные данные

Для каждого набора входных данных выведите «YES» (без кавычек), если Альперен доберется до конфеты, в противном случае, выведите «NO» (без кавычек).

Вы можете выводить ответ в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ).

Пример

входные данные	Скопировать
7 7 UUURDDL 2 UR 8 RRRUUDDD 3 LLL 4 DUUR 5 RUDLL 11 LLLLDDRUDRD	
выходные данные	Скопировать
YES YES NO NO YES YES NO	

Примечание

В первом наборе входных данных Альперен следует по пути

$$(0, 0) \overset{U}{\rightarrow} (0, 1) \overset{U}{\rightarrow} (0, 2) \overset{U}{\rightarrow} (0, 3) \overset{R}{\rightarrow} (1, 3) \overset{D}{\rightarrow} (1, 2) \overset{D}{\rightarrow} (1, 1) \overset{L}{\rightarrow} (0, 1).$$

Обратите внимание, что Альперену не обязательно находится в точке (1, 1) в конце своего пути, ему достаточно оказаться там хотя бы раз за весь путь.

Во втором наборе Альперен следует по пути

$$(0, 0) \overset{U}{\rightarrow} (0, 1) \overset{R}{\rightarrow} (1, 1).$$

В третьем наборе Альперен следует по пути

$$(0, 0) \overset{R}{\rightarrow} (1, 0) \overset{R}{\rightarrow} (2, 0) \overset{R}{\rightarrow} (3, 0) \overset{U}{\rightarrow} (3, 1) \overset{U}{\rightarrow} (3, 2) \overset{D}{\rightarrow} (3, 1) \overset{D}{\rightarrow} (3, 0) \overset{D}{\rightarrow} (3, -1).$$

В четвертом наборе Альперен следует по пути

$$(0, 0) \overset{L}{\rightarrow} (-1, 0) \overset{L}{\rightarrow} (-2, 0) \overset{L}{\rightarrow} (-3, 0).$$

При поддержке



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Идеальная точка

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам дано n одномерных отрезков (каждый отрезок представляется двумя числами — координатами его концов).
Давайте определим функцию $f(x)$ как количество отрезков, покрывающих точку x (отрезок покрывает точку x , если $l \leq x \leq r$, где l — левый конец отрезка, а r — правый).
Целочисленная точка x называется *идеальной*, если она принадлежит большему количеству отрезков, чем любая другая целочисленная точка, т.е. $f(y) < f(x)$ верно для любой другой целочисленной точки y .
Вам дано целое число k . Ваша задача состоит в том, чтобы определить, возможно ли удалить несколько (возможно, ни одного) отрезков, чтобы данная точка k стала *идеальной*.

Входные данные
Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.
Первая строка каждого набора содержит два целых числа n и k ($1 \leq n, k \leq 50$).

Далее следует n строк, i -я из них содержит два целых числа l_i и r_i ($1 \leq l_i, r_i \leq 50; l_i \leq r_i$) — концы i -го отрезка.

Выходные данные
Для каждого набора входных данных, выведите YES, если возможно удалить несколько (возможно, ни одного) отрезков, чтобы данная точка k стала *идеальной*, в противном случае выведите NO.
Каждую букву можно выводить в любом регистре (например, YES, yes, Yes будут распознаны как положительный ответ, NO, no и nO будут распознаны как отрицательный ответ).

Пример

входные данные	Скопировать
4 4 3 1 3 7 9 2 5 3 6 2 9 1 4 3 7 1 3 2 4 3 5 1 4 6 7 5 5	
выходные данные	Скопировать
YES NO NO YES	

Примечание
В первом примере точка 3 уже идеальна (она покрыта тремя отрезками), поэтому не нужно ничего удалять.
В четвертом примере можно удалить все, кроме отрезка [5, 5].



[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

В. Махмуд и треугольник

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

У Махмуда есть n отрезков, i -й из них имеет длину a_i . Ехаб поспорил с ним, сможет ли он использовать **ровно 3** отрезка для того, чтобы составить из них невырожденный треугольник. Махмуд никогда не спорит, если не уверен, что может выиграть, поэтому он спросил вас, должен ли он принять этот вызов или нет. Вам даны длины отрезков, проверьте, можно ли выбрать ровно 3 из них, чтобы составить невырожденный треугольник.

Махмуд должен использовать ровно 3 отрезка, он не может составить отрезок из двух или изменять какие-либо длины. Невырожденным считается треугольник с положительной площадью.

Входные данные

Первая строка содержит целое число n ($3 \leq n \leq 10^5$) — число отрезков.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — длины отрезков.

Выходные данные

В единственной строке выведите «YES», если можно выбрать ровно три отрезка и составить из них невырожденный треугольник, и «NO» иначе.

Примеры

входные данные	Скопировать
5 1 5 3 2 4	
выходные данные	Скопировать
YES	

входные данные	Скопировать
3 4 1 2	
выходные данные	Скопировать
NO	

Примечание

В первом примере Махмуд может выбрать отрезки с длинами 2, 4 и 5 и составить из них невырожденный треугольник.





[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

А. Завидный забор

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Эмускальд у надо построить забор вокруг фермы, но вот самому это делать ему лень. Так что юноша приобрел робота-заборостройщика.

Эмускальду хочется, чтобы забор представлял собой правильный многоугольник. К сожалению, его робот строит забор, двигаясь по единственному пути. При этом робот может соединять стороны забора только под заданным углом a .

Сможет ли робот построить забор, который хочет Эмускальд? Иными словами, существует ли правильный многоугольник с углами, равными a ?

Входные данные

Первая строка входных данных содержит целое число t ($0 < t < 180$) — количество тестов. Каждая из последующих t строк содержит единственное целое число a ($0 < a < 180$) — угол, под которым робот может соединять стороны, в градусах.

Выходные данные

Для каждого теста выведите в единственной строке «YES» (без кавычек), если робот может построить тот забор, который хочет Эмускальд. Если это невозможно, выведите «NO» (без кавычек).

Примеры

входные данные	Скопировать
3 30 60 90	
выходные данные	Скопировать
NO YES YES	

Примечание

В первом тесте невозможно построить забор, так как не существует правильного многоугольника с углами 30° .

Во втором тесте забор представляет собой правильный треугольник, а в последнем тесте — квадрат.

При поддержке




[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Рудольф и елочка

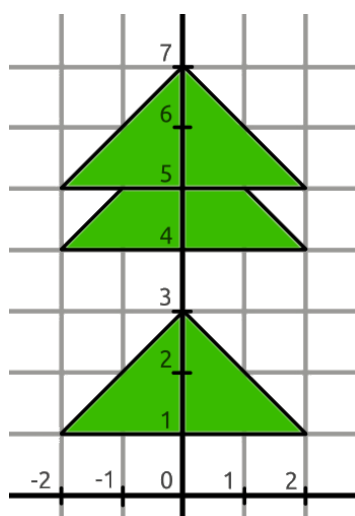
ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Рудольф нарисовал красивую елочку и решил распечатать рисунок. Но краска в картридже часто заканчивается в самый неподходящий момент. Поэтому Рудольф хочет заранее посчитать, сколько зеленой краски ему понадобится.

Елочка представляет собой вертикальный ствол, на котором на разной высоте располагаются **одинаковые** треугольные ветки. Толщина ствола пренебрежимо мала.

Каждая ветка представляет собой равнобедренный треугольник с основанием d и высотой h , основание которого перпендикулярно стволу. Треугольники располагаются углом вверх, и ствол проходит ровно посередине. Основание i -го треугольника находится на высоте y_i .

На рисунке изображен пример елочки с $d = 4$, $h = 2$ и тремя ветками с основаниями в координатах $[1, 4, 5]$.



Помогите Рудольфу вычислить суммарную площадь веток елочки.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Далее следуют описания наборов.

Первая строка набора содержит три целых числа n, d, h ($1 \leq n, d, h \leq 2 \cdot 10^5$) — количество веток елочки, длина основания и высота веток, соответственно.

Вторая строка набора содержит n целых чисел y_i ($1 \leq y_i \leq 10^9, y_1 < y_2 < \dots < y_n$) — координаты оснований веток.

Сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных в отдельной строке выведите одно вещественное число — суммарная площадь веток елочки. Ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-6} .

Пример

входные данные	Скопировать
<pre> 5 3 4 2 1 4 5 1 5 1 3 4 6 6 1 2 3 4 2 1 200000 1 200000 2 4 3 9 11 </pre>	

выходные данные	Скопировать
11 2.5 34.5 199999.9999975 11.333333	

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 16:04:12^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке





ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

А. Добытчик алмазов

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

«Добытчик алмазов» — игра, похожая на «Добытчик золота», но в этой игре n шахтеров, а не 1.

Шахта может быть представлена как плоскость. Будем рассматривать n шахтеров как n точек, расположенных на оси y . В шахте расположены n алмазов, будем рассматривать их как n точек на оси x . По каким-то причинам ни шахтеры, ни алмазы не могут быть расположены в начале координат (точке $(0, 0)$).

Каждый шахтер должен добыть **ровно** один алмаз. У каждого шахтера есть крюк, с помощью которого он может достать алмаз. Если шахтер, находящийся в точке (a, b) использует свой крюк для добычи алмаза, расположенного в точке (c, d) , он потратит $\sqrt{(a - c)^2 + (b - d)^2}$ (расстояние между точками) единиц энергии. Шахтеры не могут перемещаться и помогать друг другу.

Цель игры — распределить алмазы по шахтерам так, чтобы минимизировать **сумму** затраченной энергии. Сможете ли вы найти этот возможный минимум?

Входные данные

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка каждого набора входных данных содержит единственное целое число n ($1 \leq n \leq 10^5$) — количество шахтеров и алмазов.

Каждая из следующих $2n$ строк содержит два целых числа x ($-10^8 \leq x \leq 10^8$) и y ($-10^8 \leq y \leq 10^8$), описывающие некоторую точку (x, y) , в которой находится **шахтер или алмаз**. Гарантируется, что либо $x = 0$, что значит, что в точке $(0, y)$ находится шахтер, либо $y = 0$, что значит, что в точке $(x, 0)$ находится алмаз. В одной точке могут находиться несколько шахтеров или алмазов.

Гарантируется, что никакая точка не находится в начале координат. Также гарантируется, что на оси x ровно n точек, и на оси y тоже ровно n точек.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 10^5 .

Выходные данные

Для каждого набора входных данных выведите одно вещественное число — минимально возможную сумму затраченных энергий.

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 $\frac{|a - b|}{\max(1, |b|)} \leq 10^{-9}$.

Пример

входные данные

Скопировать

3
2
0 1
1 0
0 -1
-2 0
4
1 0
3 0
-5 0
6 0
0 3
0 1
0 2
0 4
5
3 0
0 4
0 -3
4 0
2 0
1 0
-3 0

0 -10
0 -2
0 -10

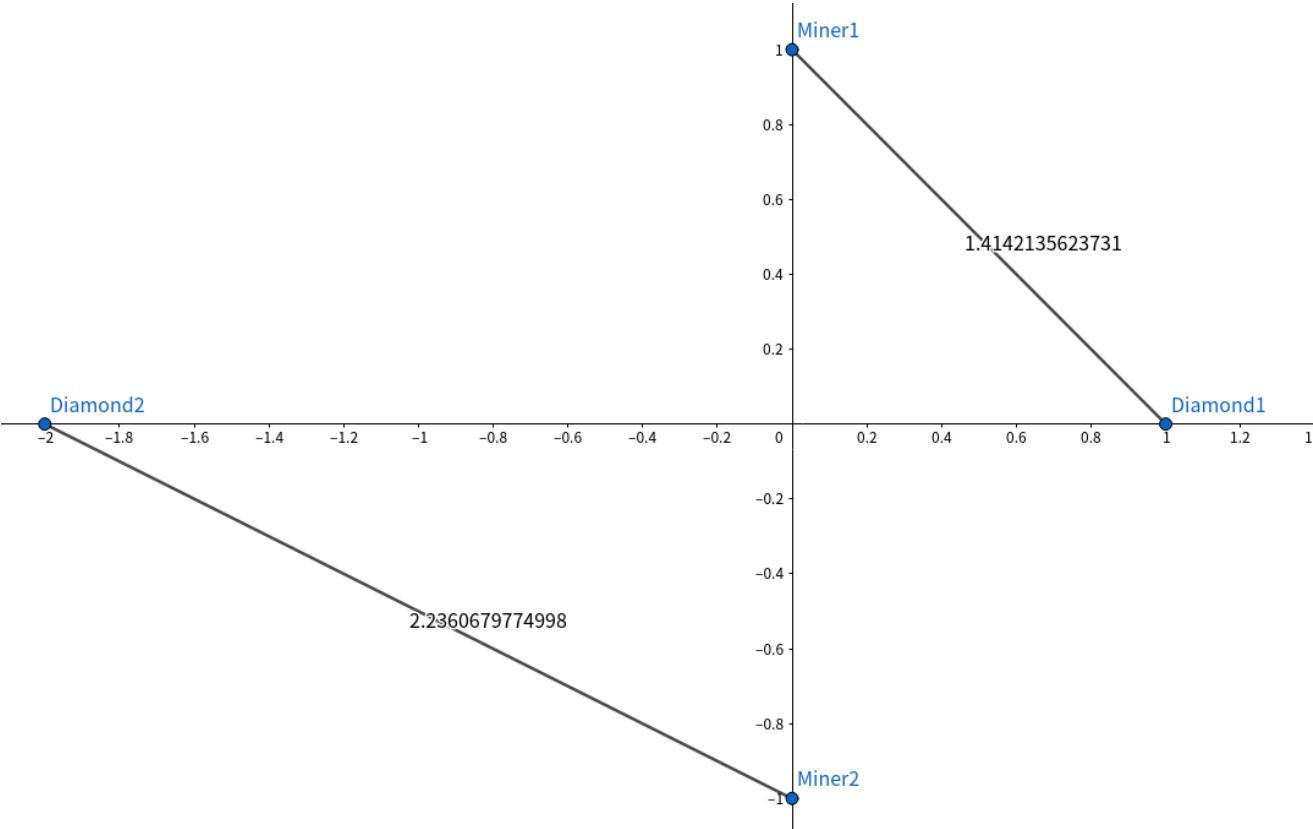
выходные данные

Скопировать

3.650281539872885
18.061819283610362
32.052255376143336

Примечание

В первом примере есть два шахтера в точках $(0, 1)$ и $(0, -1)$, а алмазы расположены в точках $(1, 0)$ и $(-2, 0)$. Если распределить алмазы по шахтерам, как показано на рисунке, то потратится $\sqrt{2} + \sqrt{5}$ единиц энергии.



[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 16:04:14^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Ларек с шаурмой

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

План столицы Берляндии представляет собой плоскость, причем все строения расположены в точках с целочисленными координатами, а пешеходными улицами являются отрезки, соединяющие каждую точку с четырьмя соседними. Все пешеходные улицы параллельны осям координат.

Известно, что школа в столице Берляндии расположена в точке (s_x, s_y) . В школе учатся n учеников, причем i -й ученик живет в доме, который расположен в точке (x_i, y_i) . Допустимо, что несколько учеников живут в одном и том же доме, но никакой ученик не живет в точке (s_x, s_y) .

После занятий каждый ученик идет из школы до своего дома по улицам, причем каждый ученик обязательно идет до своего дома по какому-то из кратчайших путей. Таким образом, расстояние, которое преодолет i -й ученик, чтобы добраться из школы до дома, равно $|s_x - x_i| + |s_y - y_i|$.

Министерство питания решило установить ларек с шаурмой в столице Берляндии в точке с целочисленными координатами. Считается, что i -й школьник купит шаурму в ларьке, если ларек находится на одном из кратчайших путей i -го школьника до дома. Запрещено ставить ларек с шаурмой в той же точке, в которой расположена школа, но разрешено ставить ларек в той точке, в которой расположен дом кого-то из учеников.

Определите максимальное количество школьников, которые смогут купить шаурму в ларьке, а также сообщите координаты для постройки ларька.

Входные данные

В первой строке следует три целых числа n, s_x, s_y ($1 \leq n \leq 200\,000, 0 \leq s_x, s_y \leq 10^9$) — количество школьников и координаты школы.

В i -й из следующих n строк следуют по два целых числа x_i, y_i ($0 \leq x_i, y_i \leq 10^9$) — координаты дома i -го школьника. Координаты домов у некоторых школьников могут совпадать. Гарантируется, что дом никакого школьника не расположен в той же точке, что и школа.

Выходные данные

В первую строку выведите целое число c — максимальное количество школьников, которые смогут купить шаурму в ларьке.

Во вторую строку выведите два целых числа p_x и p_y — координаты для постройки ларька. Если подходящих ответов несколько, разрешается вывести любой из них. Обратите внимание, что каждое из чисел p_x и p_y должно быть не менее 0 и не более 10^9 .

Примеры

<div>входные данные</div> <div>4 3 2 1 3 4 2 5 1 4 1</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>3 4 2</div>	<div>Скопировать</div>
<div>входные данные</div> <div>3 100 100 0 0 0 0 100 200</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>2 99 100</div>	<div>Скопировать</div>
<div>входные данные</div> <div>7 10 12 5 6 20 23</div>	<div>Скопировать</div>

```
15 4
16 5
4 54
12 1
4 15
```

выходные данные

Скопировать

```
4
10 11
```

Примечание

В первом примере нужно построить ларек с шаурмой в точке $(4, 2)$. Тогда три школьника его посетят. Это школьники, чьи дома расположены в точках $(4, 2)$, $(4, 1)$ и $(5, 1)$.

Во втором примере можно, например, построить ларек с шаурмой в точке $(1, 1)$. В таком случае оба школьника, дом которых расположен в точке $(0, 0)$, его посетят.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 16:04:16^{UTC+5} (11).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке

**ИТМО**



ЗАДАЧИ
 ОТОСЛАТЬ
 СТАТУС
 ПОЛОЖЕНИЕ
 ЗАПУСК

В. Новый год и северный полюс

ограничение по времени на тест: 2 секунды
 ограничение по памяти на тест: 256 мегабайт

В данной задаче мы будем полагать, что планета Земля является идеальным шаром, а её поверхность — идеальной сферой. Длину экватора и любого меридиана будем считать равной в точности 40 000 километров. Таким образом, путешествие от Северного полюса к Южному полюсу или наоборот занимает 20 000 километров.

Полярный медвежонок Лимак живёт на Северном полюсе. Незадолго до нового года он помогает кое-кому развозить подарки по всему миру. К сожалению, вместо точных координат мест, которые Лимак должен посетить, ему выдали лишь описание маршрута движения, в предположении, что он начинает на Северном полюсе. Описание маршрута состоит из n частей. На i -м участке Лимак должен двигаться t_i километров в направлении, определяемом строкой dir_i , которая является одной из строк «North», «South», «West», «East». Данные направления означают движение на север, юг, запад и восток соответственно.

Лимак не уверен, что описание маршрута корректно. Помогите ему проверить выполнение следующих условий:

- Если в некоторый момент времени (перед выполнением какой-либо инструкции или во время выполнения одной из них) Лимак находится на Северном полюсе, то он может двигаться только на юг.
- Если в некоторый момент времени (перед выполнением какой-либо инструкции или во время выполнения одной из них) Лимак находится на Южном полюсе, то он может двигаться только на север.
- Путешествие обязательно должно завершаться на Северном полюсе.

Проверьте, что все три условия выше выполняются и выведите «YES» или «NO».

Входные данные

В первой строке входных данных записано целое число n ($1 \leq n \leq 50$).

В i -й из последующих n строк записано целое число t_i и строка dir_i ($1 \leq t_i \leq 10^6, dir_i \in \{\text{North, South, West, East}\}$) — длина i -го участка маршрута и направление, в котором должен двигаться Лимак.

Выходные данные

Выведите «YES», если данный маршрут удовлетворяет всем трём условиям. В противном случае выведите «NO».

Примеры

входные данные	Скопировать
5 7500 South 10000 East 3500 North 4444 West 4000 North	
выходные данные	Скопировать
YES	

входные данные	Скопировать
2 15000 South 4000 East	
выходные данные	Скопировать
NO	

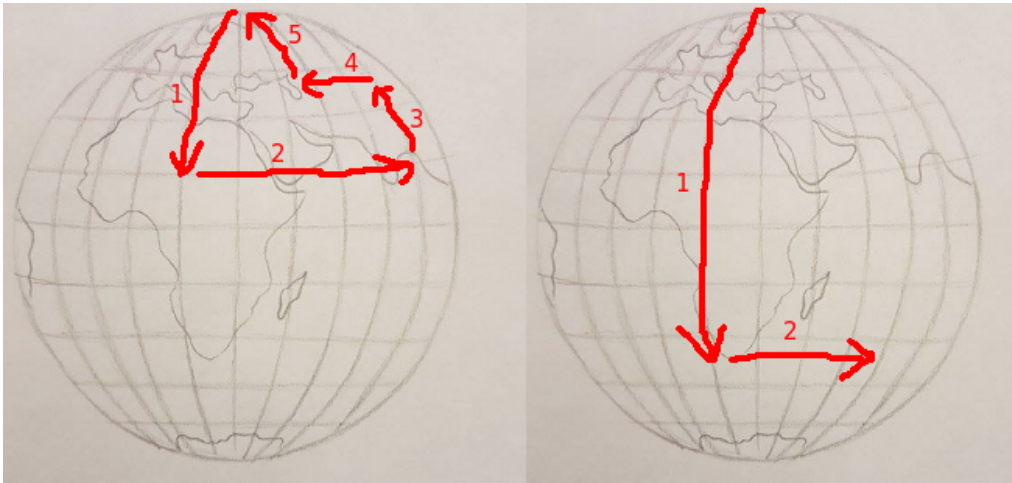
входные данные	Скопировать
5 20000 South 1000 North 1000000 West 9000 North 10000 North	
выходные данные	Скопировать
YES	

входные данные	Скопировать
3 20000 South 10 East 20000 North	
выходные данные	Скопировать
NO	

входные данные	Скопировать
2 1000 North 1000 South	
выходные данные	Скопировать
NO	

входные данные	Скопировать
4 50 South 50 North 15000 South 15000 North	
выходные данные	Скопировать
YES	

Примечание
Рисунок ниже демонстрирует как будет выглядеть путешествие Лимака в первых двух примерах. Во втором примере ответ «NO», поскольку путешествие не заканчивается на Северном полюсе.



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C1. Просто вписываем многоугольник

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Условие этой задачи почти полностью совпадает с условием задачи C2. Единственное отличие в следующем: в задаче C1 n всегда чётно, а в задаче C2 n всегда нечётно.

Вам задан правильный многоугольник из $2 \cdot n$ вершин (то есть он выпуклый и имеет равные стороны и углы) и все его стороны имеют длину 1. Назовем этот многоугольник $2n$ -угольником.

Ваша задача — найти квадрат минимального размера, такой что в него можно вписать $2n$ -угольник. $2n$ -угольник можно вписать в квадрат, если $2n$ -угольник можно разместить таким образом, что каждая точка, лежащая внутри или на границе $2n$ -угольника также будет лежать внутри или на границе квадрата.

Вы можете вращать $2n$ -угольник и/или квадрат.

Входные данные

В первой строке задано целое число T ($1 \leq T \leq 200$) — количество наборов входных данных.

В следующих T строках заданы описания наборов — по одной строке на набор. В каждой строке задано единственное чётное целое число n ($2 \leq n \leq 200$). Не забывайте, что вам нужно вписать $2n$ -угольник, а не n -угольник.

Выходные данные

Выведите T действительных чисел — по одному на набор входных данных. Для каждого набора, выведите минимальную длину стороны квадрата, в который можно вписать $2n$ -угольник. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не превосходит 10^{-6} .

Пример

входные данные	Скопировать
3 2 4 200	
выходные данные	Скопировать
1.000000000 2.414213562 127.321336469	





[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

А. Хранители

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

Хранители в опасности, и Доктор Манхэттен со своим другом Дэниелом Драйбергом должны срочно их предупредить. Всего в команде хранителей n человек, i -й из которых находится в точке плоскости с координатами (x_i, y_i) .

Как всем известно, доктор Манхэттен вычисляет расстояние между двумя хранителями i и j по формуле $|x_i - x_j| + |y_i - y_j|$. Дэниел, как обычный человек, считает, что расстояние равно $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Сейчас успех операции зависит от того, сколько существует пар (i, j) ($1 \leq i < j \leq n$), таких что расстояние между хранителем i и хранителем j , вычисленное Доктором Манхэттенем, равняется расстоянию между ними, вычисленному Дэниелом. Вычислить эту величину попросили именно вас.

Входные данные

В первой строке входных данных записано число n ($1 \leq n \leq 200\,000$) — количество хранителей.

В каждой из следующих n строк записаны два целых числа x_i и y_i ($|x_i|, |y_i| \leq 10^9$).

Некоторые позиции могут совпадать.

Выходные данные

Выведите количество пар хранителей, таких что расстояние между ними, вычисленное доктором Манхэттенем, равно расстоянию, вычисленному Дэниелом.

Примеры

входные данные	Скопировать
3 1 1 7 5 1 5	
выходные данные	Скопировать
2	
входные данные	Скопировать
6 0 0 0 1 0 2 -1 1 0 1 1 1	
выходные данные	Скопировать
11	

Примечание

В первом примере расстояние между хранителем 1 и хранителем 2 равняется $|1 - 7| + |1 - 5| = 10$ в понимании Доктора Манхэттена и $\sqrt{(1 - 7)^2 + (1 - 5)^2} = 2 \cdot \sqrt{13}$ в понимании Дэниела. Для пар (1, 1), (1, 5) и (7, 5), (1, 5) расстояния, вычисленные Доктором Манхэттенем и Дэниелом, совпадают.



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Сатьям и подсчет

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Сатьям даны n различных точек на двумерной координатной плоскости. **Гарантируется, что $0 \leq y_i \leq 1$ для всех заданных точек (x_i, y_i) .** Сколько различных невырожденных прямоугольных треугольников* можно сформировать, выбрав три различные точки в качестве его вершин?

Два треугольника a и b различны, если существует точка v , такая что v является вершиной a , но не является вершиной b .

*Невырожденный прямоугольный треугольник имеет положительную площадь и внутренний угол 90° .

Входные данные

Первая строка содержит целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит целое число n ($3 \leq n \leq 2 \cdot 10^5$) — количество точек.

Следующие n строк содержат по два целых числа x_i и y_i ($0 \leq x_i \leq n, 0 \leq y_i \leq 1$) — i -я точка, которую Сатьям может выбрать. Гарантируется, что все (x_i, y_i) попарно различны.

Гарантируется, что сумма n по всем наборам входных данных не превышает $2 \cdot 10^5$.

Выходные данные

Выведите целое число для каждого набора входных данных — количество различных невырожденных прямоугольных треугольников, которые можно сформировать, выбрав три точки.

Пример

входные данные	Скопировать
3 5 1 0 1 1 3 0 5 0 2 1 3 0 0 1 0 3 0 9 1 0 2 0 3 0 4 0 5 0 2 1 7 1 8 1 9 1	
выходные данные	Скопировать
4 0 8	

Примечание

Четыре треугольника, о которых идет речь в первом наборе входных данных:



[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 16:04:27^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке





[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

A. Странная сумма

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

У Егора есть табличка $n \times m$, где строки пронумерованы от 1 до n сверху вниз, а столбцы пронумерованы с 1 до m слева направо. Каждая клетка таблички покрашена в некоторый цвет, где цвета пронумерованы целыми числами от 1 до 10^5 .

Будем обозначать клетку, которая находится на пересечении r -й строки и c -го столбца, как (r, c) . Определим манхэттенское расстояние между клетками (r_1, c_1) и (r_2, c_2) как длину кратчайшего пути между этими клетками, в котором любые две соседние клетки имеют общую сторону. Например, в таблице 3×4 манхэттенское расстояние между клетками $(1, 2)$ и $(3, 3)$ равно 3, и один из кратчайших путей имеет вид $(1, 2) \rightarrow (2, 2) \rightarrow (2, 3) \rightarrow (3, 3)$. Обратите внимание, что путь может проходить по клеткам любого цвета.

У Егора возникло желание посчитать сумму манхэттенских расстояний по всем парам клеток одного цвета. Помогите Егору — вычислите эту сумму.

Входные данные

Первая строка входного файла содержит два целых числа n и m ($1 \leq n \leq m, n \cdot m \leq 100\,000$) — число строк и столбцов таблички.

Следующие n строк описывают соответствующие строки таблицы. i -я строка содержит m чисел $c_{i1}, c_{i2}, \dots, c_{im}$ ($1 \leq c_{ij} \leq 100\,000$) — цвета клеток в i -м ряду таблицы.

Выходные данные

Выведите одно число — искомую сумму.

Примеры

<div>входные данные</div> <div> 2 3 1 2 3 3 2 1 </div>	<div>Скопировать</div>
<div>выходные данные</div> <div>7</div>	<div>Скопировать</div>
<div>входные данные</div> <div> 3 4 1 1 2 2 2 1 1 2 2 2 1 1 </div>	<div>Скопировать</div>
<div>выходные данные</div> <div>76</div>	<div>Скопировать</div>
<div>входные данные</div> <div> 4 4 1 1 2 3 2 1 1 2 3 1 2 1 1 1 2 1 </div>	<div>Скопировать</div>
<div>выходные данные</div> <div>129</div>	<div>Скопировать</div>

Примечание

В первом примере есть три пары клеток с одинаковым цветом: в координатах $(1, 1)$ и $(2, 3)$, в координатах $(1, 2)$ и $(2, 2)$, в координатах $(1, 3)$ и $(2, 1)$. Соответствующие манхэттенские расстояния равны 3, 1 и 3, их сумма равна 7.



[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

В. Восточная выставка

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Вы и ваши друзья живете в n домах. Каждый дом расположен на 2D плоскости в точке с целыми координатами. Несколько домов могут быть расположены в одной точке. Мэр города хочет узнать у вас возможные места для постройки здания Восточной выставки. Найдите количество мест (точек с целыми координатами) таких, что суммарное расстояние от всех домов до выставки будет минимальным. Здание выставки может быть построено в той же точке, что и какой-то дом. Расстояние между двумя точками (x_1, y_1) и (x_2, y_2) считается по формуле $|x_1 - x_2| + |y_1 - y_2|$, где $|x|$ — модуль величины x .

Входные данные
В первой строке находится единственное целое число t ($1 \leq t \leq 1000$) — число наборов входных данных.

В первой строке каждого теста находится единственное целое число n ($1 \leq n \leq 1000$). В следующих n строках находятся позиции домов (x_i, y_i) ($0 \leq x_i, y_i \leq 10^9$).

Гарантируется, что сумма n не превосходит 1000.

Выходные данные
Для каждого набора входных данных выведите единственное число — количество различных позиций для здания выставки. Здание выставки может быть построено в той же точке, что и какой-то дом.

Пример

входные данные	Скопировать
<pre> 6 3 0 0 2 0 1 2 4 1 0 0 2 2 3 3 1 4 0 0 0 1 1 0 1 1 2 0 0 1 1 2 0 0 2 0 2 0 0 0 0 </pre>	
выходные данные	Скопировать
<pre> 1 4 4 4 3 1 </pre>	

Примечание
Снизу находятся картинки поясняющие примеры. Голубые точки обозначают дома, а зеленые - возможные позиции для выставки.

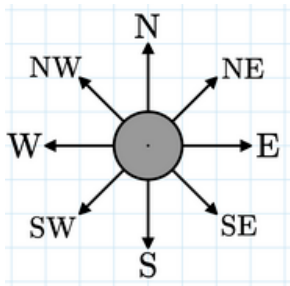


[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

G. The Morning Star

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Компас указывает прямо на утреннюю звезду. Он может указывать только в одном из восьми направлений: четыре основных направления (N, S, E, W) или их комбинации (NW, NE, SW, SE). В противном случае, он сломается.



Направления, в которых может указывать компас.

На плоскости находится n различных точек с целочисленными координатами. Сколько существует способов разместить компас в одной точке и утреннюю звезду в другой, чтобы компас не сломался?

Входные данные

Каждый тест содержит несколько наборов входных данных. Первая строка содержит количество наборов входных данных t ($1 \leq t \leq 10^4$). Затем следуют описания наборов.

Первая строка каждого набора содержит одно целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество точек.

Затем следуют n строк, каждая строка содержит два целых числа x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты каждой точки.

Гарантируется, что сумма n по всем тестовым случаям не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число — количество пар точек, которые не сломают компас.

Пример

входные данные	Скопировать
<pre> 5 3 0 0 -1 -1 1 1 4 4 5 5 7 6 9 10 13 3 -1000000000 1000000000 0 0 1000000000 -1000000000 5 0 0 2 2 -1 5 -1 10 2 11 3 0 0 -1 2 1 -2 </pre>	
выходные данные	Скопировать
<pre> 6 2 6 8 0 </pre>	

Примечание

В первом примере, любая пара точек не ломает компас:

- Компас находится в $(0, 0)$, утренняя звезда находится в $(-1, -1)$: компас будет указывать SW.
- Компас находится в $(0, 0)$, утренняя звезда находится в $(1, 1)$: компас будет указывать NE.
- Компас находится в $(-1, -1)$, утренняя звезда находится в $(0, 0)$: компас будет указывать NE.
- Компас находится в $(-1, -1)$, утренняя звезда находится в $(1, 1)$: компас будет указывать NE.
- Компас находится в $(1, 1)$, утренняя звезда находится в $(0, 0)$: компас будет указывать SW.
- Компас находится в $(1, 1)$, утренняя звезда находится в $(-1, -1)$: компас будет указывать SW.

Во втором примере, только две пары точек не сломают компас:

- Компас находится в $(6, 9)$, утренняя звезда находится в $(10, 13)$: компас будет указывать NE.
- Компас находится в $(10, 13)$, утренняя звезда находится в $(6, 9)$: компас будет указывать SW.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 16:04:39^{UTC+5} (11).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ІІТМО

С. Репрезентативные края

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Массив a_1, a_2, \dots, a_n называется *хорошим*, если и только если для любого подотрезка $1 \leq l \leq r \leq n$, выполняется следующее: $a_l + a_{l+1} + \dots + a_r = \frac{1}{2}(a_l + a_r) \cdot (r - l + 1)$.

Вам дан массив целых чисел a_1, a_2, \dots, a_n . За одну операцию вы можете заменить один любой элемент массива на любое вещественное число. Определите минимальное число операций, необходимое, чтобы сделать массив хорошим.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 100$): количество наборов входных данных.

Далее следуют описания t наборов входных данных, по две строки на набор.

Первая строка содержит одно целое число n ($1 \leq n \leq 70$): количество чисел в массиве.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-100 \leq a_i \leq 100$): изначальный массив.

Выходные данные

Для каждого набора входных данных выведите одно целое число: минимальное число элементов, которое нужно заменить, чтобы массив стал хорошим.

Пример

входные данные	Скопировать
5 4 1 2 3 4 4 1 1 2 2 2 0 -1 6 3 -2 4 -1 -4 0 1 -100	
выходные данные	Скопировать
0 2 0 3 0	

Примечание

В первом примере массив хороший изначально.

Во втором примере один из возможных хороших массивов — $[1, 1, \underline{1}, \underline{1}]$ (замененные элементы подчеркнуты).

В третьем примере массив хороший изначально.

Во четвертом примере один из возможных хороших массивов — $[-2.5, -2, \underline{-1.5}, -1, \underline{-0.5}, 0]$.



CODEFORCES
Sponsored by TON

|  

[Кар6502](#) | [Выйти](#)

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Новый год и керлинг

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Кэрол играет в керлинг.

У нее есть n дисков радиуса r на плоскости.

Изначально все диски находятся выше линии $y = 10^{100}$.

Кэрол пускает диски по направлению к прямой $y = 0$ по одному в порядке от 1 до n .

Когда она пускает i -й диск, она ставит его центр в точку $(x_i, 10^{100})$, а затем толкает его так, что его y координата начинает уменьшаться, а координата x остается постоянной. Диск останавливается, когда он касается прямой $y = 0$ или любого предыдущего диска. Обратите внимание, как только диск остановился, он больше не движется, даже от удара следующих дисков.

Вычислите координату y центров всех дисков после того, как они остановятся.

Входные данные

Первая строка содержит два целых числа n и r ($1 \leq n, r \leq 1\,000$) — количество дисков и радиус каждого диска, соответственно.

Следующая строка содержит n целых чисел x_1, x_2, \dots, x_n ($1 \leq x_i \leq 1\,000$) — x -координаты дисков.

Выходные данные

Выведите единственную строку с n числами. i -е из этих чисел должно быть равно итоговой y -координате центра i -го диска. Ваш ответ будет зачтен, если его абсолютная или относительная ошибка не превосходит 10^{-6} .

А именно, пусть ваш ответ для определенного диска равен a , а ответ жюри — b . Ваш ответ будет зачтен, если $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ для всех дисков.

Пример

входные данные	Скопировать
6 2 5 5 6 8 3 12	
выходные данные	Скопировать
2 6.0 9.87298334621 13.3370849613 12.5187346573 13.3370849613	

Примечание

В первом примере конечные позиции дисков показаны на рисунке:



ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

В. Собрание на прямой

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

n людей живут на координатной прямой, i -й человек живет в точке x_i ($1 \leq i \leq n$). Они хотят выбрать точку x_0 для встречи. i -й человек потратит $|x_i - x_0|$ минут, чтобы добраться до места встречи. Также i -му человеку требуется t_i минут чтобы одеться, поэтому суммарно ему нужно $t_i + |x_i - x_0|$ минут чтобы добраться до места встречи.

Здесь $|y|$ обозначает модуль числа y .

Эти люди просят вас выбрать позицию x_0 , которая минимизирует время, через которое все n людей доберутся до места встречи.

Входные данные

В первой строке задано единственное целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных. Затем следуют сами наборы входных данных.

Каждый набор входных данных состоит из трех строк.

В первой строке задано единственное целое число n ($1 \leq n \leq 10^5$) — количество людей.

Во второй строке заданы n целых чисел x_1, x_2, \dots, x_n ($0 \leq x_i \leq 10^8$) — позиции людей.

В третьей строке заданы n целых чисел t_1, t_2, \dots, t_n ($0 \leq t_i \leq 10^8$), где t_i это время, которое нужно i -му человеку, чтобы одеться.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите единственное вещественное число — оптимальная позиция x_0 . Можно показать, что существует единственная оптимальная позиция x_0 .

Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-6} . Формально, пусть ваш ответ равен a , а ответ жюри равен b . Ваш ответ будет зачтен, если $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-6}$.

Пример

входные данные	Скопировать
7 1 0 3 2 3 1 0 0 2 1 4 0 0 3 1 2 3 0 0 0 3 1 2 3 4 1 2 3 3 3 3 5 3 3 6 5 4 7 2 10 4 3 2 5 1 4 6	
выходные данные	Скопировать
0 2 2.5 2 1	

3

6

Примечание

- В 1-м наборе входных данных есть только один человек, поэтому целесообразно выбрать место встречи в его позиции. Тогда он доберется до него за 3 минуты, которые нужны ему, чтобы одеться.
- Во 2-м наборе входных данных есть 2 человека, которым не нужно время, чтобы одеться. Чтобы добраться до позиции 2, каждому из них потребуется по одной минуте.
- В 5-м наборе входных данных 1-му человеку нужно 4 минуты, чтобы добраться до позиции 1 (4 минуты, чтобы одеться, и 0 минут на сам путь); 2-му человеку нужно 2 минуты, чтобы добраться до позиции 1 (1 минута, чтобы одеться, и 1 минута на сам путь); 3-му человеку нужно 4 минуты, чтобы добраться до позиции 1 (2 минуты, чтобы одеться, и 2 минуты на сам путь).

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 16:04:58^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке

**IITMO**

ЗАДАЧИ

ОТОСЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

А. Шестиугольник Геральда

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Геральду на день рождения подарили очень любопытный шестиугольник. Именинник выяснил, что все углы шестиугольника равны 120° . Затем он измерил длины его сторон и обнаружил, что каждая из них равна целому числу сантиметров. На этом свойства самого шестиугольника закончились и Геральд решил на нём порисовать.

Он нарисовал несколько отрезков, параллельных сторонам этого шестиугольника, которые разбили шестиугольник на правильные треугольники со стороной в 1 сантиметр. Теперь Геральду интересно, сколько у него получилось треугольников. Но их так много, что Геральд всё время сбивается со счёта. Помогите имениннику посчитать треугольники.

Входные данные

В первой и единственной строке входных данных через пробел перечислены 6 целых чисел a_1, a_2, a_3, a_4, a_5 и a_6 ($1 \leq a_i \leq 1000$) — длины сторон шестиугольника в сантиметрах в порядке обхода по часовой стрелке. Гарантируется, что шестиугольник с указанными свойствами и именно такими сторонами существует.

Выходные данные

Выведите единственное целое число — количество треугольников со стороной 1 сантиметр, на которые оказался разбит шестиугольник.

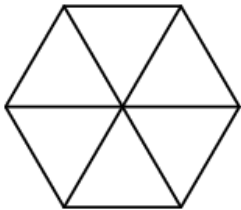
Примеры

входные данные	Скопировать
1 1 1 1 1 1	
выходные данные	Скопировать
6	

входные данные	Скопировать
1 2 1 2 1 2	
выходные данные	Скопировать
13	

Примечание

Вот так выглядит шестиугольник Геральда в первом примере:



А вот так — во втором:



[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

F. Периметр круга

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Дано целое число r , найдите количество решетчатых точек, Евклидово расстояние от которых до $(0, 0)$ **больше или равно** r , но **строго меньше** $r + 1$.

Решетчатая точка — это точка с целыми координатами. Евклидово расстояние от $(0, 0)$ до точки (x, y) равно $\sqrt{x^2 + y^2}$.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Единственная строка каждого набора входных данных содержит одно целое число r ($1 \leq r \leq 10^5$).

Сумма r по всем наборам входных данных не превышает 10^5 .

Выходные данные

Для каждого набора входных данных выведите одно целое число — количество решетчатых точек, расстояние от которых до $(0, 0)$ равно d , где $r \leq d < r + 1$.

Пример

входные данные	Скопировать
<div>6</div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>1984</div>	
выходные данные	Скопировать
<div>8</div> <div>16</div> <div>20</div> <div>24</div> <div>40</div> <div>12504</div>	

Примечание

Точки для первых трех наборов входных данных показаны ниже.

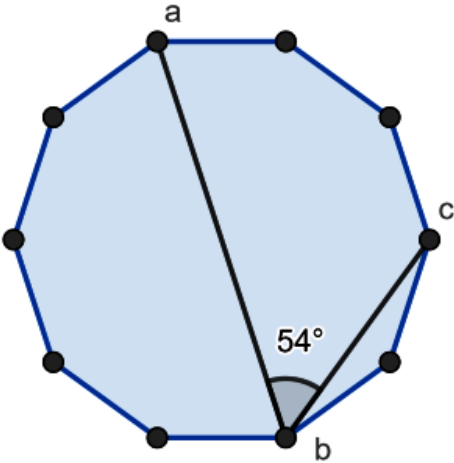
ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

С. Многоугольник для угла

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам задан угол ang .

Жюри просит Вас найти такой **правильный** n -угольник (правильный многоугольник из n вершин), что у него найдутся три вершины a, b и c (не обязательно последовательные) с углом $\angle abc = \text{ang}$ или сообщить, что такого n -угольника не существует.



Если существует несколько ответов, выведите **наименьший** из них. Гарантируется, что если ответ существует, то он не превосходит 998244353.

Входные данные

Первая строка содержит единственное целое число T ($1 \leq T \leq 180$) — количество запросов.

Следующие T строк содержат по одному целому числу ang ($1 \leq \text{ang} < 180$) — заданный угол в градусах.

Выходные данные

Для каждого запроса выведите по одному целому числу n ($3 \leq n \leq 998244353$) — наименьшее возможное число вершин в подходящем правильном n -угольнике либо -1 , не такого n не существует.

Пример

входные данные	Скопировать
4 54 50 2 178	
выходные данные	Скопировать
10 18 90 180	

Примечание

Ответ для первого запроса изображен на изображении выше.

Ответ для второго запроса достигается на правильном 18-угольнике. Например, $\angle v_2 v_1 v_6 = 50^\circ$.

Возможный угол для третьего запроса — $\angle v_{11} v_{10} v_{12} = 2^\circ$.

В четвертом запросе, минимальный возможный n равен 180 (но не 90).



[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

А. Треугольник

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Есть прямоугольный треугольник с катетами длины a и b . Требуется расположить его на плоскости так, что ни одна из его сторон не параллельна осям координат, а вершины находятся в точках с целочисленными координатами, либо определить, что это невозможно.

Входные данные

В первой и единственной строке находятся два целых числа a, b ($1 \leq a, b \leq 1000$), разделенные одним пробелом.

Выходные данные

В первой строке выведите «YES» или «NO» (без кавычек) в зависимости от того, существует ли искомое расположение. Если оно существует, то в трех следующих строках выведите три пары целых чисел — координаты вершин треугольника, по одной паре в строке. Координаты должны быть целыми числами, не превышающими 10^9 по модулю.

Примеры

входные данные	Скопировать
1 1	
выходные данные	Скопировать
NO	

входные данные	Скопировать
5 5	
выходные данные	Скопировать
YES 2 1 5 5 -2 4	

входные данные	Скопировать
5 10	
выходные данные	Скопировать
YES -10 4 -2 -2 1 2	

При поддержке





[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

С. Фифа и Фафа

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Фифа и Фафа живут в одной квартире. Фифа любит компьютерные игры и хочет скачать одну новую игру про футбол. К сожалению, Фафа уже израсходовал всю их месячную квоту интернета, однако, Фифа может подключиться к интернету через свою Wi-Fi точку доступа. К точке доступа можно подключиться в некотором радиусе r метров от места ее установки, Фифа может выбирать этот радиус. Фифа должен поставить точку доступа внутри квартиры, которая имеет круглую форму радиуса R . Фифа хочет минимизировать площади квартиры, из которой нельзя подключиться к точке доступа, при этом он не хочет, чтобы Фафа или кто-либо снаружи квартиры смог подключиться к точке доступа.

Мир можно представить как бесконечную плоскость. Центр квартиры расположен в точке (x_1, y_1) , квартира имеет радиус R , ноутбук Фафы расположен в точке (x_2, y_2) , не обязательно внутри квартиры. Можно представлять ноутбук Фафы как точку. Найдите точку установки и радиус, которые должен выбрать Фифа для своей точки доступа, чтобы минимизировать непокрытую площадь квартиры.

Входные данные

Единственная строка содержит пять целых чисел R, x_1, y_1, x_2, y_2 ($1 \leq R \leq 10^5, |x_1|, |y_1|, |x_2|, |y_2| \leq 10^5$).

Выходные данные

Выведите три числа x_{ap}, y_{ap}, r где (x_{ap}, y_{ap}) — точка, в которую Фифа должен поставить точку доступа, а r — ее радиус действия.

Ваш ответ будет считаться правильным, если радиус отличается от оптимального не более, чем на 10^{-6} абсолютно или относительно, и радиус может быть изменен не более, чем на 10^{-6} (абсолютно или относительно) так, что все точки снаружи квартиры, а также позиция ноутбука Фафы окажутся снаружи области действия точки доступа.

Примеры

входные данные	Скопировать
5 3 3 1 1	
выходные данные	Скопировать
3.7677669529663684 3.7677669529663684 3.914213562373095	

входные данные	Скопировать
10 5 5 5 15	
выходные данные	Скопировать
5.0 5.0 10.0	

При поддержке




[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Прямоугольники

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На плоскости даны n прямоугольников, которые заданы координатами своих левого нижнего и правого верхнего углов. Известно, что некоторые $(n - 1)$ из этих n прямоугольников имеют хотя бы одну общую точку. Точка принадлежит прямоугольнику, если она находится строго внутри прямоугольника или на его границе.

Найдите любую точку с целыми координатами, которая принадлежит хотя бы $(n - 1)$ прямоугольнику из заданных.

Входные данные

В первой строке задано число n ($2 \leq n \leq 132\,674$) — количество прямоугольников.

В следующих n строках заданы прямоугольники четверками целых чисел: x_1, y_1, x_2, y_2 ($-10^9 \leq x_1 < x_2 \leq 10^9$, $-10^9 \leq y_1 < y_2 \leq 10^9$).

Выходные данные

Выведите два целых числа x, y — координаты произвольной точки на плоскости, принадлежащей хотя бы $(n - 1)$ прямоугольнику.

Примеры

входные данные	Скопировать
3 0 0 1 1 1 1 2 2 3 0 4 1	
выходные данные	Скопировать
1 1	

входные данные	Скопировать
3 0 0 1 1 0 1 1 2 1 0 2 1	
выходные данные	Скопировать
1 1	

входные данные	Скопировать
4 0 0 5 5 0 0 4 4 1 1 4 4 1 1 4 4	
выходные данные	Скопировать
1 1	

входные данные	Скопировать
5 0 0 10 8 1 2 6 7 2 3 5 6 3 4 4 5 8 1 9 2	
выходные данные	Скопировать
3 4	

Примечание

Рисунок ниже показывает расположение прямоугольников в первом и втором тестах из примера. Возможные ответы выделены.

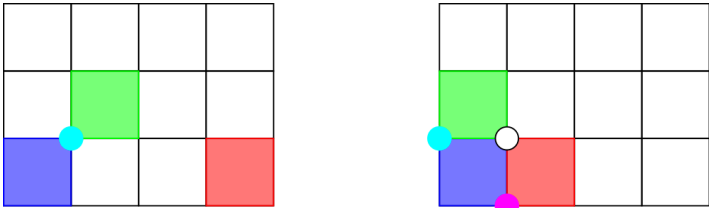
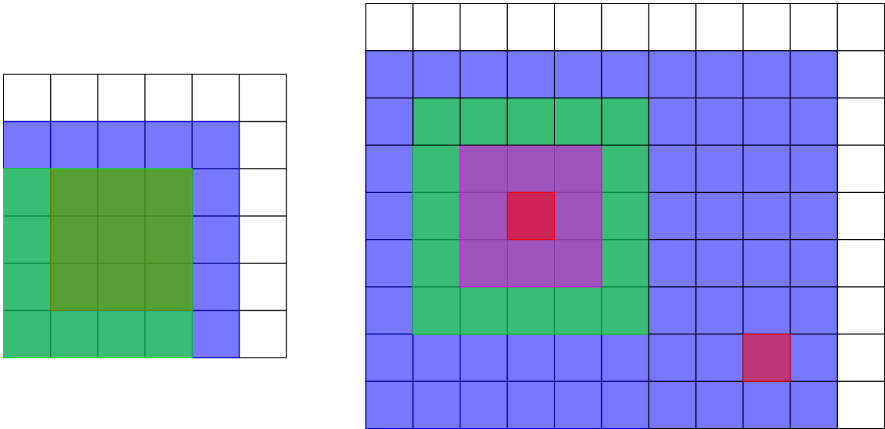


Рисунок ниже показывает расположение прямоугольников в третьем и четвертом примерах.





[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Созвездие

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Кот Ноку заполучил карту ночного неба. На этой карте он нашёл созвездие, которое состоит из n звёзд, пронумерованных от 1 до n . Известно, что i -я звезда расположена в точке с координатами (x_i, y_i) . Координаты всех звёзд различны.

Вечером Ноку собирается пойти смотреть на ночное небо. Он хотел бы найти три различные звезды, которые образуют треугольник положительной площади. Кроме того, все остальные звезды должны лежать строго вне этого треугольника. У kota возникли проблемы с поиском таких трёх звёзд, и он не отказался бы от вашей помощи. Ваша задача — найти индексы трёх звезд, которые образуют треугольник, удовлетворяющий всем условиям.

Гарантируется, что все звёзды не лежат на одной прямой. Можно доказать, что если предыдущее условие выполняется, то существует хотя бы одно решение задачи.

Входные данные

В первой строке входных данных записано единственное целое число n ($3 \leq n \leq 100\,000$) — количество звёзд.

Каждая из следующих n строк содержит два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$).

Гарантируется, что никакие две звезды не лежат в одной точке и что не существует прямой, на которой лежат все звёзды.

Выходные данные

Выведите три целых числа — индексы трёх точек, которые образуют треугольник, удовлетворяющий всем требованиям Ноку.

Если правильных ответов несколько, то разрешается вывести любой.

Примеры

входные данные	Скопировать
3 0 1 1 0 1 1	
выходные данные	Скопировать
1 2 3	

входные данные	Скопировать
5 0 0 0 2 2 0 2 2 1 1	
выходные данные	Скопировать
1 3 5	

Примечание

В первом примере можно вывести три индекса в любом порядке.

Второму примеру соответствует следующая картинка: