

Задача А. Битва за Хогвартс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Приспешники Тёмного Лорда уже подошли совсем близко к Хогвартсу.

Хогвартс окружен системой защитных башен. Профессор Флитвик накладывает защитные заклятия на замок. Заключаются они в следующем: вокруг выпуклой оболочки башен создается защитный барьер.

Тёмный Лорд, пользуясь Бузинной палочкой, может разрушить защитный барьер за минуту, при этом все башни на выпуклой оболочке тоже разрушаются.

После того, как башни уничтожены, Флитвик мгновенно восстанавливает защитный барьер на выпуклой оболочке оставшихся башен, а Вола-де-Морт их снова разрушает через минуту. Так продолжается, пока все башни не падут.

У Гарри и его друзей мало времени — они ищут и уничтожают очередной крестраж. Поэтому их очень интересует, сколько времени у них осталось.

Рассчитайте для каждой башни момент времени, когда она будет уничтожена.

Формат входных данных

В первой строке вводится одно число n ($1 \leq n \leq 2 \cdot 10^4$) — количество башен.

В следующих n строках вводятся целочисленные координаты башен — x_i, y_i ($|x_i|, |y_i| \leq 10^4$).

Гарантируется, что башни расположены так, что каждый следующий защитный барьер будет лежать строго внутри предыдущего (то есть, они не пересекаются и не имеют точек касания).

Формат выходных данных

Выведите n строк, каждая из которой содержит одно число — момент времени (в минутах), к которому падёт каждая башня, начиная с первой.

Пример

стандартный ввод	стандартный вывод
5	1
0 0	1
4 4	1
0 4	2
1 1	1
4 0	

Замечание

Задача стоит 5 баллов.

Задача В. Замок

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Есть замок — точка $(0, 0)$. Замок окружен несколькими непересекающимися заборами, каждый представляет из себя выпуклый многоугольник.

Есть также m захватчиков, известны их координаты. Захватчики не умеют перелезть через заборы. Захватчика будем считать опасным, если он находится внутри внешнего забора. Требуется вычислить суммарную площадь области, куда опасные захватчики могут добраться без пересечения заборов.

Формат входных данных

В первой строке задано число n — количество заборов ($1 \leq n \leq 100\,000$). Далее следуют описания n заборов. Каждое описание начинается с числа k , далее следуют k строк, содержащих по два числа x и y — координаты вершин ($|x|, |y| \leq 2 \cdot 10^6$). Вершины каждого многоугольника перечисляются в порядке обхода против часовой стрелки. Гарантируется, что точка $(0, 0)$ лежит внутри каждого забора.

Далее следует число m ($0 \leq m \leq 100\,000$) — количество захватчиков. В следующих m строках заданы координаты захватчиков.

Суммарное число вершин во всех многоугольниках не превосходит 100 000.

Формат выходных данных

Выведите единственное число — общую захваченную площадь с шестью знаками после десятичной точки.

Пример

стандартный ввод	стандартный вывод
3	2400.000000
4	
-10 -10	
10 -10	
10 10	
-10 10	
4	
20 20	
-20 20	
-20 -20	
20 -20	
4	
30 -30	
30 30	
-30 30	
-30 -30	
3	
1 1	
22 23	
111 123	

Задача С. Сыр

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Крешо купил вкуснейший сыр с перцем, но Степану перец не нравится, поэтому он хочет отрезать кусок, на котором не было бы перца. Сыр имеет форму выпуклого многоугольника, а каждая перчинка является точкой внутри него. Степан режет сыр только 1 раз. Он выбирает две вершины многоугольника, не являющиеся смежными, и режет по диагонали, соединяющей их. Затем Степан забирает ту из получившихся частей, на которой нет перца (ни внутри, ни на границе).

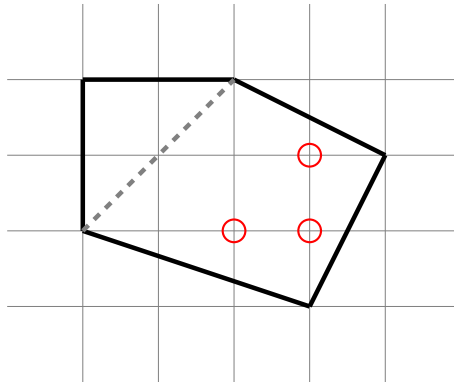


Рис. 1: Рисунок соответствует первому тесту. Пунктирной линией показан разрез Степана.

Напишите программу, которая определит, может ли Степан отрезать кусок без перца. Если он может это сделать, выведите максимальную площадь куска, который может отрезать Степан.

Формат входных данных

Первая строка входного файла содержит одно целое число N ($3 \leq N \leq 3 \cdot 10^5$) – количество вершин в многоугольнике. Каждая из следующих N строк содержит два числа x_i и y_i – координаты i -й вершины. Следующая строка содержит одно число M ($1 \leq M \leq 3 \cdot 10^5$) – количество перчинок. Каждая из следующих M строк содержит два числа x_i и y_i – координаты i -й перчинки.

Вершины многоугольника заданы в порядке обхода против часовой стрелки и образуют выпуклый многоугольник. Никакие две подряд идущие стороны не параллельны.

Все перчинки расположены в различных точках и внутри многоугольника (они не расположены на стороне или снаружи многоугольника).

Все входные координаты по модулю не превосходят 10^9 .

Формат выходных данных

Выведите одно число – удвоенную максимальную площадь (это число всегда целое). Если отрезать кусок без перца невозможно, выведите 0.

Пример

стандартный ввод	стандартный вывод
5	4
0 1	
3 0	
4 2	
2 3	
0 3	
3	
2 1	
3 1	
3 2	

Задача D. Выпуклая оболочка

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

Дано N точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

Формат входных данных

На первой строке число N ($3 \leq N \leq 10^5$). Следующие N строк содержат пары целых чисел x и y ($-10^9 \leq x, y \leq 10^9$) — точки.

Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

Формат выходных данных

В первой строке выведите N число вершин выпуклой оболочки. Следующие N строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой. Кроме того, в последней строке выведите площадь получившейся выпуклой оболочки. Площадь необходимо вывести абсолютно точно.

Примеры

<code>hull.in</code>	<code>hull.out</code>
5 0 0 2 0 0 2 1 1 2 2	4 0 0 0 2 2 2 2 0 4.0
3 0 0 0 1 1 1	3 0 0 0 1 1 1 0.5

Задача Е. Разрезание торта

Имя входного файла: `cut.in`
Имя выходного файла: `cut.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Мама купила Пете на день рождения торт в виде выпуклого многоугольника. Торт большой и вкусный, и Петя хочет разделить его с мамой поровну. Для этого он хочет сделать один прямой разрез, причем ему будет удобнее, если этот разрез будет параллелен оси Oy . Помогите Пете определить, как ему разрезать торт.

Формат входных данных

В первой строке записано целое число N ($1 \leq N \leq 10000$) — количество вершин многоугольника. В последующих N строках записаны координаты вершин многоугольника в порядке обхода. Гарантируется, что все координаты — целые числа, не превосходящие по модулю 10^3 .

Формат выходных данных

Выведите x -координату точки, через которую необходимо провести разрез, с точностью не менее 10^{-6} .

Пример

<code>cut.in</code>	<code>cut.out</code>
4 0 0 0 2 2 2 2 0	1.000000000

Задача F. Платные дороги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Мэр одного большого города решил ввести плату за проезд по шоссе, проходящим в районе города, чтобы снизить объем транзитного транспорта. В районе города проходит n шоссе.

Но руководство области, в которой расположен город, воспротивилось планам мэра. Действительно — дальнбойщики представляют собой неплохой источник доходов для большого количества кафе и гостиниц в небольших городках.

В результате решили, что плата будет введена только на шоссе, которые проходят через город.

В городе используется развитая система метрополитена, всего в городе есть m станций метро. Решено было, что шоссе проходит через город, если либо одна из станций метро расположена непосредственно на шоссе, либо есть хотя бы одна станция с каждой стороны от шоссе.

Помогите теперь мэру определить, какие шоссе проходят через город.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество шоссе и количество станций метро, соответственно ($1 \leq n, m \leq 100\,000$).

Следующие n строк описывают шоссе. Каждое шоссе описывается тремя целыми числами a , b и c и представляет собой прямую на плоскости, задаваемую уравнением $ax + by + c = 0$ ($|a|, |b|, |c| \leq 10^9$).

Следующие m строк входного файла описывают станции метро. Каждая станция описывается двумя целыми числами x и y и представляет собой точку на плоскости с координатами (x, y) ($|x|, |y| \leq 10^9$).

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — количество шоссе, которые проходят через город. Вторая строка должна содержать номера этих шоссе в возрастающем порядке. Шоссе нумеруются от 1 до n в порядке, в котором они описаны во входном файле.

Пример

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

Задача G. Место встречи изменить нельзя

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.3 секунд
Ограничение по памяти: 256 мегабайт

Даны N точек. Найдите такие две из них, что расстояние между ними минимально.

Формат входных данных

Первая строка входного файла содержит целое число N ($2 \leq N \leq 100\,000$) — количество точек. Каждая из следующих N строк содержит пару целых чисел X и Y , разделённых пробелом, — координаты ($-1\,000\,000\,000 \leq X, Y \leq 1\,000\,000\,000$). Все точки различны.

Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

Пример

стандартный ввод	стандартный вывод
4	0 0
0 0	0 1
0 1	
1 1	
1 0	

Задача Н. Две самые далёкие точки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Формат входных данных

Первая строка содержит количество точек N , ($1 \leq N \leq 10^5$). Каждая из последующих N строк содержит два целых числа — координаты x_i и y_i . Координаты по модулю не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл расстояние между двумя наиболее удалёнными точками с максимально возможной точностью.

Примеры

стандартный ввод	стандартный вывод
7 0 0 1 1 2 2 0 2 1 3 0 1 2 0	3.16227766016837952279
8 0 0 38 85 40 90 20 96 0 100 -20 96 -40 90 -38 85	100.00000000000000000000

Задача I. Минимальный полигон

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На этот раз Даздраперма и Оюшминальд были кратки-супекратки: они попросили вас выбрать из данных N точек ровно K таким образом, чтоб они образовывали выпуклый многоугольник и его площадь была минимальна.

Формат входных данных

В первой строке входного файла находятся два целых числа N и K ($3 \leq N \leq 100$, $3 \leq K \leq 20$, $K \leq N$). В следующих N строках записано по два целых числа x_i, y_i ($-10^4 \leq x_i, y_i \leq 10^4$) – координаты точек.

Гарантируется, что во входных данных все точки различны и не существует 3 точек, лежащих на одной прямой.

Формат выходных данных

В выходной файл выведите одно вещественное число – площадь искомого многоугольника. Ваш ответ будет засчитан, если его относительная или абсолютная погрешность не будет превосходить 10^{-4} . Если невозможно выбрать ровно K точек, чтобы они являлись вершинами выпуклого многоугольника, то считается, что искомая площадь равна нулю.

Примеры

стандартный ввод	стандартный вывод
4 3 0 0 10 0 10 2 0 1	5.0
4 4 0 0 10 0 4 4 0 10	0

Задача J. Тигры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В заповеднике живут q тигров. Чтобы следить за положением тигров на территории заповедника, используются ошейники с радиомаяком. Ошейник у каждого тигра имеет радиомаяк с уникальным сигналом. Система обнаружения настраивается на приём сигнала радиомаяка от i -го тигра последовательно для i от 1 до q .

Для приёма сигнала на территории заповедника установлено n приёмников в точках с координатами $(x_1, y_1), \dots, (x_n, y_n)$. Система обнаружения позволяет сотруднику заповедника за один запрос выбрать любые m ($3 \leq m \leq n$) приёмников. Выбранные приёмники должны являться вершинами выпуклого многоугольника. Система определяет, находится ли радиомаяк i -го тигра внутри этого многоугольника.

Сотрудник заповедника должен локализовать положение каждого тигра. Положение i -го тигра считается локализованным, если удалось определить такое множество приёмников, являющихся вершинами выпуклого многоугольника, что внутри этого многоугольника находится тигр, но нет других приёмников.

Для того, чтобы локализовать положение каждого из тигров, сотруднику разрешается сделать не более k запросов.

После того как положение i -го тигра локализовано, система автоматически переходит к приёму сигналов от следующего тигра, пока положение всех q тигров не будет локализовано.

Гарантируется, что никакие три приёмника не лежат на одной прямой, и ни один тигр не находится на прямой, проходящей через два приёмника. Гарантируется, что существует хотя бы один выпуклый многоугольник с вершинами в приёмниках, внутри которого находится тигр.

Требуется написать программу, которая взаимодействует с программой жюри и локализует положение каждого тигра.

Формат входных данных

Это интерактивная задача.

Сначала на вход подаётся информация об установленных в заповеднике приёмниках и количестве тигров.

Первая строка входных данных содержит целое число n — количество приёмников ($3 \leq n \leq 5000$). Последующие n строк описывают координаты приёмников, j -я из этих строк содержит два целых числа x_j и y_j — координаты j -го приёмника ($-10^9 \leq x_j, y_j \leq 10^9$). Следующая строка содержит число целое число q — количество тигров ($1 \leq q \leq 2000$).

Для локализации положения тигров необходимо выполнять запросы к системе обнаружения, роль которой выполняет программа жюри.

Для каждого теста зафиксировано число k — максимальное количество запросов к системе обнаружения для локализации положения одного тигра. Гарантируется, что k запросов достаточно, чтобы решить задачу для соответствующих данных. Это число не сообщается программе-решению, но ограничения на него в различных подзадачах приведены в таблице системы оценивания. Если программа-решение делает более k запросов для определения местоположения одного из тигров, на этом тесте она получает в качестве результата тестирования «Неверный ответ».

Запрос к системе обнаружения начинается с символа «?», за которым следует целое число m — количество выбранных в запросе приёмников ($3 \leq m \leq n$), и m различных целых чисел p_i — номера приёмников, перечисленные в порядке обхода многоугольника по или против часовой стрелки ($1 \leq p_i \leq n$).

В ответ программа получает строку «Yes», если тигр находится внутри многоугольника, образованного приёмниками с номерами p_1, \dots, p_m , и строку «No» в противном случае.

После того, как положение тигра локализовано, программа-решение должна вывести строку, начинающуюся с символа «!», за которым следует целое число m — количество выбранных приёмников ($3 \leq m \leq n$), и m различных целых чисел p_i — номера приёмников, перечисленные в порядке

обхода многоугольника по или против часовой стрелки ($1 \leq p_i \leq n$). Эта строка означает, что внутри выпуклого многоугольника, образованного приёмниками с номерами p_1, \dots, p_m , находится тигр и нет других приёмников.

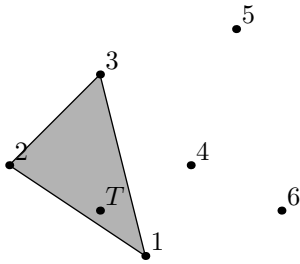
Ответное сообщение от программы жюри отсутствует, и программа-решение должна немедленно приступить к поиску следующего тигра. Локализовав положение тигра с номером q , программа-решение должна завершить работу.

Тигры не перемещаются во время работы системы обнаружения. Координаты тигров в каждом тесте фиксированы и не меняются в процессе тестирования.

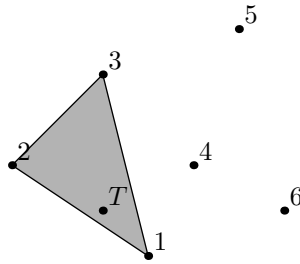
Если существует несколько правильных многоугольников, локализующих положение тигра, можно вывести любой из них.

На рисунке продемонстрирована процедура локализации положения каждого из тигров из приведенного ниже примера.

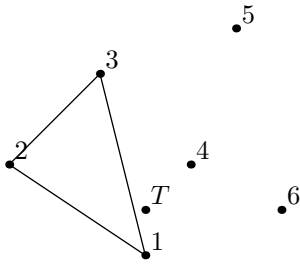
Тигр 1, запрос 1:



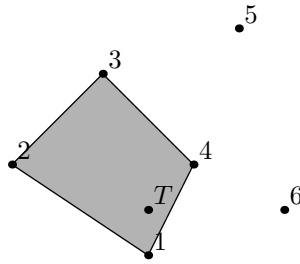
Тигр 1, ответ:



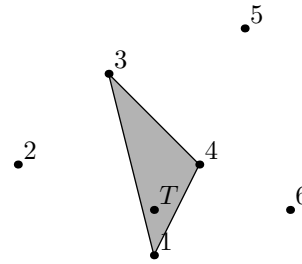
Тигр 2, запрос 1:



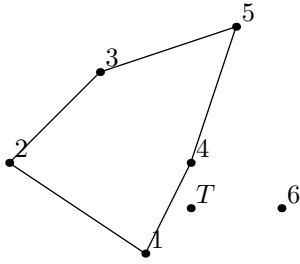
Тигр 2, запрос 2:



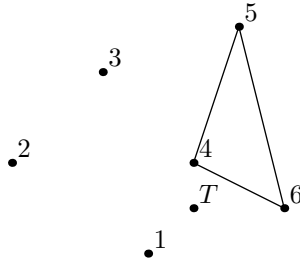
Тигр 2, ответ:



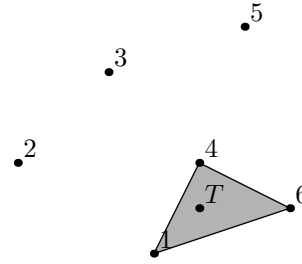
Тигр 3, запрос 1:



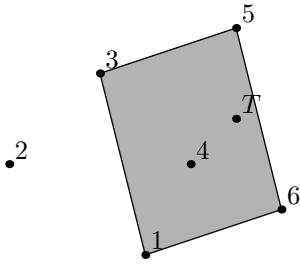
Тигр 3, запрос 2:



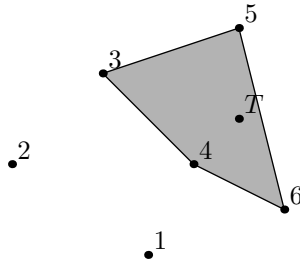
Тигр 3, ответ:



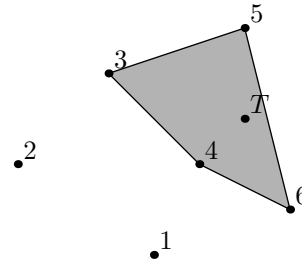
Тигр 4, запрос 1:



Тигр 4, запрос 2:



Тигр 4, ответ:



Пример

стандартный ввод	стандартный вывод
6	
3 0	
0 2	
2 4	
4 2	
5 5	
6 1	
4	
Yes	? 3 1 2 3
	! 3 1 2 3
	? 3 1 2 3
No	
	? 4 1 2 3 4
Yes	
	! 3 4 3 1
	? 5 2 3 5 4 1
No	
	? 3 4 5 6
No	
	! 3 1 4 6
	? 4 1 3 5 6
Yes	
	? 4 4 3 5 6
Yes	
	! 4 4 3 5 6

Замечание

Приведённые примеры иллюстрируют взаимодействие программы-решения с программой жюри «по шагам», для чего в них добавлены дополнительные пустые строки. При реальном тестировании лишние пустые строки вводиться не будут, выводить пустые строки также не требуется.

Задача К. Стена

Имя входного файла: `wall.in`
Имя выходного файла: `wall.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стеной. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стеной, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстояние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.

Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

Формат входных данных

Первая строка входного файла содержит числа N и L , разделенные пробелом. N ($3 \leq N \leq 1000$) — это количество вершин в королевском замке, а L ($1 \leq L \leq 1000$) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие N строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа x_i и y_i , разделенные пробелом ($-10000 \leq x_i, y_i \leq 10000$), которые обозначают координаты i -ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.

Формат выходных данных

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

Пример

wall.in	wall.out
9 100	1628.3185307180
200 400	
300 400	
300 300	
400 300	
400 400	
500 400	
500 200	
350 200	
200 200	