

Задача А. Пересечение двух прямых

Имя входного файла: `intersect.in`
Имя выходного файла: `intersect.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Формат входных данных

Шесть чисел — целые коэффициенты A , B и C нормального уравнения двух различных непараллельных прямых (сначала для одной прямой, затем для другой).

Формат выходных данных

Два числа — координаты точки их пересечения с точностью до шестого знака после запятой.

Примеры

<code>intersect.in</code>	<code>intersect.out</code>
<code>1 1 -1 1 -1 0</code>	<code>0.500000 0.5000000000000000000000</code>

Задача В. Две окружности

Имя входного файла: `intersec.in`
Имя выходного файла: `intersec.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вам заданы две окружности. Необходимо выяснить, пересекаются ли заданные окружности и найти точки их пересечения.

Формат входных данных

Первая строка файлов содержит количество тестов — целое число от 1 до 10 000. Далее следуют тесты. Каждый тест имеет вид: две строки, по три числа в каждой — координаты центра и радиус сначала первой, а затем второй окружности.

Все числа — целые, по модулю не превосходящие 10 000. А радиус еще и положительный.

Формат выходных данных

На каждый тест выведите ответ в следующем формате:

На отдельной строке выходного файла выведите количество точек пересечения (0, 1, 2 или 3, если их бесконечно много). В случае одной точки пересечения выведете во второй строке координаты этой точки. В случае двух точек пересечения выведете во второй строке координаты точки H , в третьей строке длины векторов $\overrightarrow{O_1H}$ и \overrightarrow{HP} , в следующих двух строках должны находиться координаты точек пересечения. Эти две точки можно вывести в произвольном порядке

Примеры

<code>intersec.in</code>	<code>intersec.out</code>
4	0
3 4 5	1
11 4 2	8.0000000000 4.0000000000
3 4 5	2
11 4 3	7.5625000000 4.0000000000
3 4 5	4.5625000000 2.0453835215
11 4 4	7.5625000000 6.0453835215
3 4 5	7.5625000000 1.9546164785
3 4 5	3

Задача С. Касательная к окружности

Имя входного файла: `tangent.in`
Имя выходного файла: `tangent.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вам задана окружность и точка. Точка может лежать вне окружности, на ее границе, а также внутри окружности. Необходимо провести касательные к окружности (если это возможно) и найти точки касания.

Формат входных данных

В первой строке входного файла находятся два числа — координаты центра окружности P_0 . Во второй строке записан радиус окружности r ($1 \leq r \leq 10^4$). В третьей строке находятся два числа — координаты точки P_1 . Координаты целые числа не превышающие по модулю 10^4 .

Формат выходных данных

В первой строке выходного файла выведите количество точек касания (0, 1 или 2). В случае одной точки касания выведете во второй строке координаты этой точки. В случае двух точек касания выведите во второй строке координаты точки P_3 , в третьей строке длины векторов $\overrightarrow{P_1P_3}$ и $\overrightarrow{P_3P_2}$, в следующих двух строках должны находиться координаты точек касания.

Примеры

<code>tangent.in</code>	<code>tangent.out</code>
2 2	2
2	2.0000000000 3.3333333333
2 5	1.6666666667 1.4907119850
	0.5092880150 3.3333333333
	3.4907119850 3.3333333333

Задача D. Ловушка для Слонопотама

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

Пятачок и Винни-Пух каждое утро ходят пить чай в гости к Кролику. Естественно, самым коротким путем.

К сожалению, однажды Винни-Пуху пришла в голову идея вырыть ловушку для Слонопотама. Самое обидное, что они с Пятачком ее даже вырыли. Поэтому теперь каждое утро, идя в гости к Кролику, они боятся в нее провалиться.

Напишите программу, которая посчитает длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика.

Ловушка для Слонопотама представляет собой яму абсолютно круглой формы. Путь является безопасным, если он не проходит по ловушке (но может проходить по ее границе).

Формат входных данных

Во входном файле записаны сначала координаты домика Винни-Пуха: X_B, Y_B , затем — координаты домика Кролика: X_R, Y_R , а затем — координаты центра и радиус ловушки: X_T, Y_T, R_T . Все координаты — целые числа из диапазона от -32000 до 32000 . Радиус ловушки — натуральное число, не превышающее 32000 .

Домики Винни-Пуха и Кролика не могут находиться внутри ловушки, но могут находиться на ее границе.

Формат выходных данных

Выведите в выходной файл одно число — длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика с точностью не менее 4 знака после запятой.

Примеры

piglet.in	piglet.out
0 0 0 1 10 10 1	1.000000
5 0 0 5 0 0 5	7.853982
-5 0 5 0 0 0 3	11.861007

Задача Е. Точка в многоугольнике

Имя входного файла: `point.in`
Имя выходного файла: `point.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Формат входных данных

В первой строке содержится три числа — N ($3 \leq N \leq 100\,000$) и координаты точки. Последующие N строк содержат координаты углов многоугольника. Координаты — целые, не превосходят 10^6 по модулю.

Формат выходных данных

Одна строка YES, если заданная точка содержится в приведённом многоугольнике или на его границе, и NO в противном случае.

Примеры

<code>point.in</code>	<code>point.out</code>
3 2 3 1 1 10 2 2 8	YES

Задача F. Теодор Рузвельт

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

«Теодор Рузвельт» — флагман военно-морского флота Кукуляндии. Заклятые враги кукуляндцев, флатландцы, решили уничтожить его. Они узнали, что «Теодор Рузвельт» представляет собой выпуклый многоугольник из n вершин и узнали его координаты. Затем они выпустили m баллистических ракет и определили координаты точек, где эти ракеты взорвались. По расчётам штаба флатландцев, «Теодор Рузвельт» будет уничтожен, если в него попадёт хотя бы k ракет. Вычислите, удалось ли флатландцам уничтожить корабль.

Формат входных данных

В первой строке через пробел записаны целые числа n, m, k ($3 \leq n \leq 10^5, 0 \leq k \leq m \leq 10^5$). В последующих n строках записаны координаты вершин многоугольника в порядке обхода против часовой стрелки. В следующих m строках записаны координаты точек. Гарантируется, что все координаты — целые числа, не превосходящие по модулю 10^9 .

Формат выходных данных

Выведите «YES», если в многоугольнике или на его границе лежит по крайней мере k точек, и «NO» в противном случае.

Примеры

stdin	stdout
5 4 2 1 -1 1 2 0 4 -1 2 -1 -1 -2 -1 1 -1 0 1 2 3	YES

Задача G. Выпуклая оболочка

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

Вам дано множество точек на плоскости. Найдите их выпуклую оболочку.

Формат входных данных

Первая строка входного файла содержит целое число n — количество точек ($3 \leq n \leq 200\,000$). В следующих n строках описываются точки. i -ая строка состоит из двух целых чисел — координат i -ой точки. Координаты не превосходят 10^9 по модулю. Гарантируется, что все точки не лежат на одной прямой. Точки могут совпадать.

Формат выходных данных

В первую строку выходного файла выведите количество вершин в выпуклой оболочке. Во вторую — номера вершин через пробел, которые ее образуют. Выводите вершины в порядке обхода против часовой стрелки. Никакие два ребра выпуклой оболочки не должны лежать на одной прямой.

В третью строку выведите периметр оболочки, в четвертую - ее площадь.

Периметр должен быть выведен с абсолютной или относительной погрешностью не больше 10^{-9} . Площадь должна быть выведена абсолютно точно.

Примеры

<code>convex.in</code>	<code>convex.out</code>
5	4
0 0	3 5 1 4
1 1	6.47213595499958000000
2 2	2.0
1 0	
0 1	

Задача Н. Стена

Имя входного файла: `wall.in`
Имя выходного файла: `wall.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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.

Примеры

<code>wall.in</code>	<code>wall.out</code>
9 100 200 400 300 400 300 300 400 300 400 400 500 400 500 200 350 200 200 200	1628.3185307180

Задача I. Замок

Имя входного файла:	castle.in
Имя выходного файла:	castle.out
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В прошлом существовало большое и могучее королевство, которое называлось Лордаэрон. Его правителем был мудрый король Теренас. В Лордаэроне была могучая армия и бесчисленное множество замков и крепостей. Для управления такой военной машиной, Теренас выбрал несколько генералов из числа наиболее верных ему людей. Как-то раз первый генерал заявил: "Мой король, небезопасно оставлять наш главный замок без хорошей защиты. Позвольте мне построить стену вокруг него". Конечно, король согласился (в конце концов, еще одна стена ничего не ухудшит). Когда стена была закончена, другой генерал заявил: "Мой король, все еще небезопасно оставлять наш главный замок без хорошей защиты. Позвольте мне построить еще одну стену вокруг него". Король согласился. После завершения второй стены еще один генерал сказал то же самое В итоге, было построено N стен вокруг замка и вся площадь между стенами была занята полями пшеницы. Однажды враги Лордаэрона осадили его основной замок. Но что они могут сделать со стенами? Они не могут забраться на них. Но зато у них есть луки и стрелы. И еще врагам известно, что все, что находится между стенами, очень хорошо горит. В течение осады, разведка Лордаэрона выяснила координаты всех точек, в которые попали стрелы. И король Теренас подсчитал ущерб, нанесенный врагами полям — другими словами площадь сгоревших полей. Сможете ли вы повторить это? Вы можете предполагать верными следующие факты. Замок очень мал и может считаться точкой — началом координат. Все стены являются выпуклыми многоугольниками. Они не касаются друг друга и не пересекаются друг с другом. Также, начало координат находится внутри всех многоугольников. Точки выстрелов не находятся на стенах. Если точка выстрела находится вне самой внешней стены, ничего не сгорит. Иначе сгорит вся площадь, до которой огонь может добраться из точки выстрела (а огонь легко распространяется по полям, но не может пересекать стены). Ваша задача — посчитать суммарную площадь того, что сгорит.

Формат входных данных

В первой строке находится число стен N ($1 \leq N \leq 10^5$). Далее следует N блоков. В первой строке каждого блока находится число K — количество вершин в многоугольнике, представляющим стену. Далее в K строках находится по два числа X и Y — координаты вершины многоугольника. Каждый многоугольник будет задан в порядке обхода против часовой стрелки. У каждого многоугольника ненулевая площадь. Вы можете предполагать, что общее количество точек во всех многоугольниках не превосходит 10^5 . Стены будут даны в произвольном порядке. Далее следует количество точек выстрелов, M ($0 \leq M \leq 10^5$). Далее следует M строк с координатами M точек выстрела. Все координаты во входном файле являются целыми и не превосходят 10^6 по абсолютной величине.

Формат выходных данных

Выведите единственное число с не менее чем шестью знаками после десятичной точки — общую площадь сгоревших полей.

Примеры

castle.in	castle.out
3	2400.0
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	