

Выпуклая оболочка

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

На плоскости даны N точек. Вам требуется построить выпуклую оболочку данного множества точек и вывести длину её периметра.

Формат входного файла

Первая строка содержит количество точек N , ($1 \leq N \leq 20\,000$). Каждая из последующих N строк содержит два целых числа — координаты x_i и y_i . Координаты по модулю не превосходят 10 000.

Формат выходного файла

Выведите в выходной файл длину периметра выпуклой оболочки с максимальной возможной точностью. Если в выпуклой оболочке 2 точки, то требуется вывести удвоенную длину отрезка.

Пример

<code>convex.in</code>	<code>convex.out</code>
5 0 0 1 0 0 1 -1 0 0 -1	5.65685

Пересечение отрезков

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

Два отрезка $[A, B]$ и $[C, D]$ на плоскости заданы координатами своих концов — точек A, B, C, D : $(X_a, Y_a), (X_b, Y_b), (X_c, Y_c), (X_d, Y_d)$.

Требуется найти пересечение этих отрезков и вывести:

- слово `Empty`, если эти отрезки не пересекаются;
- координаты точки пересечения, если пересечение состоит из единственной точки;
- координаты точек — начала и конца отрезка пересечения в лексикографическом порядке, если пересечение заданных отрезков — отрезок. То есть первой следует выводить точку с минимальным x , а если координаты x равны, то с минимальным y .

Формат входного файла

Во входном файле содержатся восемь целых чисел, по модулю не превосходящих 1000 — координаты концов точек A, B, C, D . Отрезки могут быть вырожденными.

Формат выходного файла

Числовые значения в ответе следует округлить до шести знаков после десятичной точки.

Пример

<code>segments.in</code>	<code>segments.out</code>
0 0 9 9 9 5 0 5	5.000000 5.000000
0 0 9 9 15 15 7 7	7.000000 7.000000 9.000000 9.000000
0 0 9 9 10 10 10 10	Empty

Максимальный треугольник

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

Посмотрев во время своей поездки на ситуацию в Borderland, Штирлиц понял коварный план верхушки Третьего Рейха. Никакой особой разницы между Y и Z не было. На самом деле вся возня с референдумом была затеяна для того, чтобы население Borderland перестало работать и вымерло: кто с голоду, кто со скуки, кто потому, что его всё это достало. А на освободившееся место планировалось переселить вывозимых из России ёжиков. У Штирлица оставалось всего несколько часов для того, чтобы воспрепятствовать экологическому преступлению. Согласно полученной информации, основным местом обитания породистых ежей был Большой Ежовый лес. Штирлиц раздобыл систему автоматической сигнализации с тремя датчиками, размещаемыми на деревьях. При пересечении извне любой из сторон треугольника, образованных датчиками, срабатывала сигнализация, и перепуганные ёжики внутри треугольника прятались в норы, где их было уже не достать. Считая, что в Большом Ежовом лесу n деревьев, и ежи распределены равномерно по всей территории леса и вокруг него, требуется выбрать такие три дерева, чтобы площадь образованного ими треугольника, а, следовательно, и ёжеёмкость охраняемой территории, были максимально возможными.

Формат входного файла

В первой строке задано число n ($3 \leq n \leq 7\,000$), далее n строк с координатами деревьев — целыми числами x_i y_i , не превосходящими 10 000 по модулю. Гарантируется, что в лесу найдутся три дерева, не лежащие на одной прямой.

Формат выходного файла

В первой строке выведите площадь максимального треугольника с двумя знаками после запятой. Во второй строке — три различных числа, номера деревьев, образующих треугольник, на которые Штирлиц должен установить датчики. Если оптимальных решений несколько, разрешается вывести любое из них.

Пример

<code>triangle.in</code>	<code>triangle.out</code>
5	2.00
0 0	1 4 2
2 2	
1 1	
0 2	
2 0	