

Задача А. Геометрия

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

Написать класс *Vector* для вектора на плоскости, реализовав основные действия над векторами в виде методов и переопределения операций.

Создать также набор классов — фигур, которые наследуются от абстрактного базового класса *AbstractShape* для работы с двумерными геометрическими примитивами:

- *Point* (точка);
- *Segment* (отрезок);
- *Line* (прямая);
- *Ray* (луч);
- *Polygon* (простой многоугольник — часть плоскости, ограниченная замкнутой ломаной без самопересечений).
- *Circle* (круг)

В базовом классе предусмотреть виртуальные методы:

- сдвига на заданный вектор;
- проверки, содержит ли фигура точку;
- проверки, пересекается ли она с отрезком;
- копирования данного объекта;
- вывода в `cout` данного объекта.

В производных классах — реализовать эти методы.

В этой задаче нужно определить все классы, методы, функции так, чтобы предложенный тестирующий код выводил ожидаемый результат в соответствии с общепринятой семантикой. Исправлять код функций *main* и *CheckFunctions* запрещается.

Тесты:

- 1-6 совпадают с примером
- 7-15 *point*
- 16-46 *segment*
- 47-55 *ray*
- 56-60 *line*
- 61-71 *polygon*
- 72-80 *circle*

Формат входных данных

В первой строчке задается тип геометрического примитива: «point», «segment», «ray», «line», «circle» или «polygon». Далее вводится сам примитив.

После чего вводится две точки *A* и *B*, которые используются в *CheckFunctions*. Все числа целочисленные и не превосходят 10000 по модулю.

Формат выходных данных

Формат вывода можно посмотреть в *main* и в примерах.

Примеры

стандартный ввод	стандартный вывод
point 0 0 -1 -1 1 1	Given shape does not contain point A Given shape crosses segment AB 2 2
segment 0 -1 0 1 0 0 0 1	Given shape contains point A Given shape crosses segment AB 0 0 0 2
ray 0 0 1 1 2 2 3 2	Given shape contains point A Given shape crosses segment AB 1 0 2 1
line 0 0 0 1 1 1 2 1	Given shape does not contain point A Given shape does not cross segment AB 1 0 1 1
polygon 4 0 0 3 0 3 3 0 3 1 1 2 2	Given shape contains point A Given shape does not cross segment AB 4 1 1 4 1 4 4 1 4
circle 0 0 5 0 0 5 5	Given shape contains point A Given shape crosses segment AB 5 5 5

Замечание

Старайтесь избегать использования чисел с плавающей точкой.

Используйте задачи этого конкурса для тестирования некоторых функций.

В данной задаче многоугольник содержит точку, если она лежит в части плоскости, ограниченной замкнутой ломаной. Многоугольник пересекается с отрезком, если отрезок пересекается с замкнутой кривой.

Круг пересекается с отрезком, если есть точка отрезка, лежащая внутри (или на границе) круга

Задача В. Вектор

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

Даны два ненулевых вектора. Требуется вычислить:

- Длину первого и второго вектора (два числа)
- Вектор, образованный сложением данных двух векторов
- Скалярное и векторное произведения данных векторов
- Площадь треугольника, построенного из этих векторов

Формат входных данных

В двух строках входного файла заданы по четыре целых числа, не превосходящих по модулю 10 000, — координаты начала и конца первого вектора, затем второго.

Формат выходных данных

В каждой строке выходного файла — ответ на соответствующий пункт задачи с точностью не менее 10^{-6} .

Пример

стандартный ввод	стандартный вывод
5 1 2 6 1 1 7 8	5.830952 9.219544 3.000000 12.000000 17.000000 -51.000000 25.500000

Задача С. Две прямые

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

Даны коэффициенты A_1, B_1, C_1 уравнения первой прямой и коэффициенты A_2, B_2, C_2 уравнения второй прямой. Требуется:

- Построить направляющие векторы для обеих прямых (вывести координаты)
- Найти точку пересечения двух прямых или вычислить расстояние между ними, если они параллельны

Формат входных данных

В первой строке входного файла находятся три числа — коэффициенты нормального уравнения для первой прямой. Во второй строке — коэффициенты для второй прямой.

Все числа во входном файле по модулю не превосходят 10 000.

Формат выходных данных

В каждой строке выходного файла выведите ответ на соответствующий пункт задачи с точностью до 10^{-6} .

Пример

стандартный ввод	стандартный вывод
1 1 -1	1.000000000 -1.000000000
1 -1 0	-1.000000000 -1.000000000
	0.500000000 0.500000000

Задача D. Точка и две точки

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

Определите, принадлежит ли точка C заданной прямой, лучу и отрезку, образованными точками A и B .

Формат входных данных

В первой строке входного файла даны два целых числа — координаты точки C . Во двух следующих строках в таком же формате заданы точки A и B ($A \neq B$).

Все числа во входном файле по модулю не превосходят 10 000.

Формат выходных данных

В первой строке выведите «YES», если точка C принадлежит прямой AB , и «NO» в противном случае. Во второй и третьей строках аналогично выведите ответы для луча AB (A — начало луча) и отрезка AB .

Пример

стандартный ввод	стандартный вывод
1 6	YES
3 7	NO
5 8	NO

Задача Е. Расстояния

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

Даны три точки A, B и C . Необходимо подсчитать расстояния от точки C до прямой, луча и отрезка, образованного точками A и B .

Формат входных данных

В первой строке входного файла даны два целых числа — координаты точки C . Во двух следующих строках в таком же формате заданы точки A и B ($A \neq B$).

Все числа во входном файле по модулю не превосходят 10 000.

Формат выходных данных

В первой строке выходного файла выведите одно вещественное число — расстояние от точки C до прямой. В следующих двух строках выведите соответственно расстояния до луча AB (A — начало луча) и до отрезка AB . Все числа выводить с точностью не менее 10^{-6} . Луч строится по направлению от точки A к точке B .

Пример

стандартный ввод	стандартный вывод
3 0	1.000000000
1 1	1.000000000
2 1	1.414213562

Задача F. Два отрезка

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

Необходимо проверить, пересекаются ли два отрезка.

Формат входных данных

В двух строках входного файла заданы по четыре целых числа, не превосходящих по модулю 10 000, — координаты концов первого отрезка, затем второго.

Формат выходных данных

В первой строке выходного файла выведите «YES», если отрезки имеют общие точки, и «NO» в противном случае.

Пример

стандартный ввод	стандартный вывод
5 1 2 6 1 1 7 8	YES

Задача G. Отрезки на дистанции

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

Найдите расстояние между двумя отрезками.

Формат входных данных

В двух строках входного файла даны по четыре целых числа, не превосходящих по модулю 10 000, — координаты концов сначала первого, затем второго отрезков.

Формат выходных данных

В первой строке выходного файла выведите одно вещественное число — расстояние между отрезками с точностью не менее 10^{-6} .

Пример

стандартный ввод	стандартный вывод
1 1 2 2 2 1 3 0	0.7071067812

Задача Н. (Не)выпуклый

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

Проверьте многоугольник на выпуклость

Формат входных данных

В первой строке одно число N ($3 \leq N \leq 100000$). Далее в N строках по паре целых чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Координаты всех точек целые, по модулю не превосходят 10^7 .

Формат выходных данных

Одна строка «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

Пример

стандартный ввод	стандартный вывод
3 0 0 0 1 1 0	YES

Задача I. Внутри или нет?

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

Проверьте, лежит ли точка внутри многоугольника.

Формат входных данных

В первой строке вводятся три целых числа – N ($3 \leq N \leq 100000$) и координаты точки. Далее в N строках задается по паре целых чисел – координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Формат выходных данных

Выведите одну строку: «YES», если заданная точка содержится в приведённом многоугольнике или на его границе, и «NO» в противном случае.

Пример

стандартный ввод	стандартный вывод
3 2 3 1 1 10 2 2 8	YES

Задача J. Площадь

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

Найдите площадь многоугольника.

Формат входных данных

В первой строке одно число N ($3 \leq N \leq 100\,000$). Далее в N строках по паре чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Координаты целые по модулю не превосходят 10 000.

Формат выходных данных

Одно число — величина площади приведённого многоугольника с абсолютной точностью.

Пример

стандартный ввод	стандартный вывод
3 1 0 0 1 1 1	0.5000000000000000

Замечание

Для того, чтобы ваш ответ считался корректным, рекомендую использовать следующую команду `std::cout.precision(15); std::cout << fixed << ...`

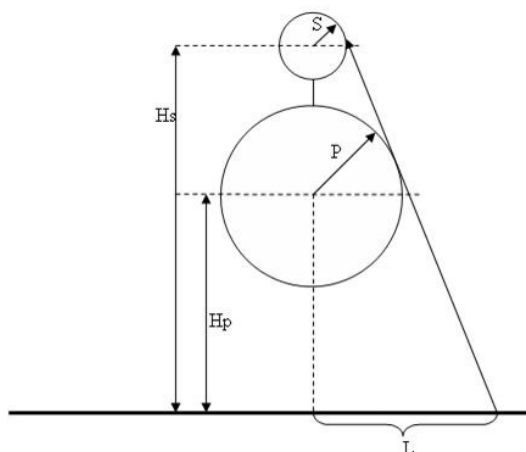
Задача К. Второй орден

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

Второй орден не дает спокойствия повстанцам. Так как в прошлых сериях никакая модель Звезды Смерти не позволяла ордену захватить вселенную, было решено построить *сдвоенную Звезду Смерти*, то есть как одна Звезда Смерти, но их две, при этом их центры находятся на *звездной оси*.

Повстанцы узнали, что в новом оружии их врагов одна из сфер неуязвима, а вторая приводит к разрушению обоих модулей, даже если выстрел пройдет по касательной к ней. Также из расшифрованных переговоров сопротивление выяснило, что новое оружие уже на позиции и готовится к нанесению смертельного удара. Вам необходимо выяснить, на какое минимальное расстояние L от пересечения *звездной оси* с планетой повстанцев надо передвинуть оборонительное оружие, чтобы предотвратить угрозу.

Благодаря шпионам сопротивлению стали известны размеры и положение *сдвоенной Звезды Смерти*. А именно, известно, что неуязвимая сфера имеет радиус P и находится на высоте H_p от планеты, тогда как второй модуль имеет радиус S и находится на высоте H_s . Планету повстанцев считать плоской.



Формат входных данных

В единственной строке входного файла находятся четыре положительных целых числа P, H_p, S, H_s , не превосходящие 10000. Гарантируется, что $H_p < H_s$. Также известно, что модули не могут пересекаться.

Формат выходных данных

Выведите минимальное расстояние L от пересечения *звездной оси* с планетой повстанцев, на которое надо передвинуть оборонительное оружие, чтобы предотвратить угрозу, с точностью не менее 6 знаков после запятой.

Пример

стандартный ввод	стандартный вывод
1 9 10 21	0.0000000

Задача L. Портал для Букли

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

Гарри понадобилось срочно отправить письмо Альбусу. К сожалению, привычном маршрутом Букле не удастся полететь, так как Пожиратели Смерти подготовили ловушку, а именно портал, который поглощает все, что попадет внутрь.

Напишите программу, которая посчитает длину самого короткого безопасного пути для Букли.

Портал Пожирателей Смерти представляет собой идеально круглую дыру. Путь является безопасным, если он не проходит через портал (но может проходить по его границе).

Формат входных данных

Во входном файле записаны сначала координаты Гарри: X_B, Y_B , затем — координаты Альбуса: X_R, Y_R , а затем — координаты центра и радиуса портала: X_T, Y_T, R_T . Все координаты — целые числа из диапазона от -32000 до 32000 . Радиус портала — натуральное число, не превышающее 32000 .

Гарри и Альбус не могут находиться внутри портала, но запросто могут быть на границе.

Формат выходных данных

Выведите в выходной файл одно число — длину самого короткого безопасного пути Букли с точностью не менее 6 знаков после запятой.

Примеры

стандартный ввод	стандартный вывод
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 М. Выпуклая оболочка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
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 K — число вершин выпуклой оболочки. Следующие K 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.

Пример

стандартный ввод	стандартный вывод
5	4
0 0	0 0
2 0	0 2
0 2	2 2
1 1	2 0
2 2	4.0

Замечание

Оболочку нужно вывести в порядке обхода по часовой стрелки, начиная с самой левой (в случае нескольких таких, то с самой левой самой нижней) точки. Иначе ответ считается неверным.

Задача N. Take me back to London by Ed Sheeran

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

«*London is the capital of Great Britain*» ©Учитель английского

Как известно, Лондон делится на несколько зон. Между ними разная стоимость проезда, в них разные цены на жилье и в принципе уровень жизни в них различный. Лондонское метро *London underground* старейшее в мире и одно из крупнейших на нашей планете. По большей части зоны современного Лондона определялись именно расположением станций.

Вам дан набор из N достопримечательностей с их координатами (x, y) , при этом множество этих мест таково, что если построить их выпуклую оболочку, то все, что попадет внутрь, будет нулевой зоной. Если же выбросить из набора точки, образующие нулевую оболочку (или попавшую на ее контур), то все, что лежит внутри выпуклой оболочки оставшихся точек является уже первой зоной. По индукции можно построить разбиение города на «концентрированные» выпуклые оболочки. Номер зоны, в которой лежит заданная точка, определяется максимальным номером выпуклой оболочки, в которой она *еще* лежит внутри.

Историк Боб хочет узнать номер зоны, в которой расположена каждая из N достопримечательностей списка выше.

Формат входных данных

В первой строке вам дается число N ($3 \leq N \leq 8 \cdot 10^3$) — количество достопримечательностей для разбиения города на зоны.

Далее идут N строк по два целых числа через пробел (x_i, y_i) ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты достопримечательностей. Гарантируется, что одинаковых точек нет.

Формат выходных данных

Выведите N строк из одного целого числа, где на i -й строке номер зоны i -й достопримечательности на вводе.

Пример

стандартный ввод	стандартный вывод
8	0
0 0	1
1 1	0
5 5	1
3 4	0
5 0	1
4 1	1
2 2	0
0 5	

Замечание

Рекомендую послушать песню из названия:)

Задача О. Инфекция

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	8 мегабайт

Неизвестная инфекция начала распространяться по всей столице. Известно, что она распространяется почти одинаково во все стороны, поэтому было решено разбить город на зоны, чтобы лучше мобилизовать ресурсы в случае внештатной ситуации.

Студент Артур долго работал над такой моделью и выяснил, что лучше всего разбить город следующим образом. Задано множество точек размера N — социально значимых объектов в столице. По этому множеству точек разбиение строится так:

1. строится выпуклая оболочка на этом множестве, потом все точки, что в нее попали (или на ее границу) выбрасываются.
2. повторяют первый шаг, пока не останется два или меньше социально значимых объектов.

То есть по индукции построили множество выпуклых оболочек. Точка лежит в зоне с номером i , если она лежит внутри i -й оболочки и не лежит внутри $(i + 1)$ -й оболочки. Точка лежит в оболочке, если попадает на ее границу или лежит внутри. Зоны нумеруются с нуля, так как там еще нет инфекции.

Ученые заинтересовались его исследованием и решили, что это действительно хорошая модель разбиения, так как точки подобраны согласно последним данным о распространении инфекции. Теперь им стало интересно, в какой зоне будут лежать научно важные объекты, например, университеты или лаборатории. Всего таких объектов K , а так как таких объектов очень много, ученые попросили автоматизировать процесс. Справитесь ли вы с этим поручением?

Формат входных данных

В первой строке вам дается число N ($3 \leq N \leq 10^3$) — количество социально значимых объектов для разбиения города на зоны.

Далее идут N строк по два целых числа через пробел (x_i, y_i) ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты очередного социально значимого объекта. Гарантируется, что одинаковых точек нет. Считайте, что точки, лежащие снаружи нулевой оболочки лежат в нулевой зоне.

На следующей строке идет число K ($1 \leq K \leq 10^5$) — количество научно значимых объектов, про которые ученым надо знать их зону.

Далее идут K строк по два целых числа через пробел (x_i, y_i) ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты университетов, лабораторий ...

Формат выходных данных

Выведите K строк из одного целого числа, где на i -й строке номер зоны i -го научно значимого объекта.

Пример

стандартный ввод	стандартный вывод
11	0
0 0	0
6 0	1
1 1	0
5 1	1
2 2	2
4 2	
5 4	
4 4	
2 5	
6 6	
0 6	
6	
6 6	
5 5	
5 4	
4 5	
2 4	
3 3	

Замечание

В данной задаче предполагается, что все оболочки невырождены. То есть, если все точки лежат на одном отрезке на какой-то момент построения, то дальше оболочки строить не нужно.