

Вычислительная геометрия

Вам предлагается разработать классы, реализующие основные геометрические “примитивы”: точку, прямую, луч, отрезок и методы работы с ними.

- *Класс Point (точка)*

В классе `Point` полезно реализовать следующие методы:

- конструктор: создание точки по двум координатам `x` и `y`, каждая по умолчанию равна нулю. Тип координат `int` или `float`;
- метод `__str__`, возвращающий строку вида `(x, y)`, где `x` — абсцисса, `y` — ордината;
- логические операторы сравнения на равенство (`==`) и неравенство (`!=`);
- полярный угол точки в радианах `polar(self)`, принадлежащий $[0, 2\pi)$;
- полярный угол точки в градусах `deg(self)`, принадлежащий $[0, 360)$.

- *Класс Vector (вектор)*

В классе `Vector` полезно реализовать следующие методы:

- конструктор: создание вектора по двум точкам (`Point`), но хранить координаты вектора, а не переданные точки;
- арифметические операторы сложения векторов, вычитания векторов (в т.ч. унарный минус), умножения вектора на число (целое или вещественное): `__add__(self, Vector)`, `__sub__(self, Vector)`, `__neg__(self)`, `__mul__(self, число)`
- метод `length(self)`, возвращающий длину вектора;
- скалярное произведение векторов, `scalar(self, Vector)`;
- косое произведение векторов, `cross(self, Vector)`.

Косым произведением векторов $\vec{a} = (a_x, a_y)$ и $\vec{b} = (b_x, b_y)$ называется выражение

$$\vec{a} \times \vec{b} = a_x \cdot b_y - a_y \cdot b_x$$

Докажите, что косое произведение $\vec{a} \times \vec{b}$ равно площади параллелограмма, натянутого на вектора \vec{a} и \vec{b} , взятое со знаком минус, если поворот от вектора \vec{a} к вектору \vec{b} — по часовой стрелке.

- *Класс Ray (луч)*

В классе `Ray` полезно реализовать следующие атрибуты и методы:

- конструктор: создание луча по точке `Point` и вектору `Vector`;
- метод `__str__`, возвращающий строку вида `[(xA, yA), (xV, yV)]`, где точка `A` — начало луча и вектор `V`, задающий направление луча;
- метод `point_in_ray(self, Point)` — принадлежит ли точка лучу, возвращает `True/False`;
- метод `point_distance(self, Point)` — расстояние от данной точки до луча;
- метод `intersect(self, Ray)` — пересекаются ли два луча, возвращает `True/False`.

- *Класс Segment (отрезок)*

В классе `Segment` полезно реализовать следующие атрибуты и методы:

- конструктор: создание отрезка по двум точкам `Point`;
- метод `length`, возвращающий длину отрезка;
- метод `point_in_segment(self, Point)` — принадлежит ли точка отрезку, возвращает `True/False`;
- метод `point_distance(self, Point)` — расстояние от данной точки до отрезка;
- метод `intersect(self, Segment)` — пересекаются ли два отрезка, возвращает `True/False`.

- *Класс Angle (угол)*

В классе `Angle` полезно реализовать следующие атрибуты и методы:

- конструктор: создание угла по трём точкам `Point` (вершине угла и точкам на сторонах угла);
- метод `value` — величина угла в градусах;
- метод `point_in_angle(self, Point)` — принадлежит ли точка углу, возвращает `True/False`;

- *Класс Line (прямая)*

В классе `Line` полезно реализовать следующие атрибуты и методы:

- конструктор: создание прямой по двум точкам, лежащим на этой прямой или трём коэффициентам a, b, c нормального уравнения прямой $ax + by + c = 0$ (здесь $\vec{n} = (a, b)$ — это вектор, перпендикулярный прямой);
Хранить в самом классе достаточно только точки, а для получения по ним коэффициентов прямой реализовать метод `get_coeff()`.
- метод `point_in_line(self, Point)` — принадлежит ли точка прямой, возвращает `True/False`;
- метод `point_distance(self, Point)` — расстояние от данной точки до прямой;
- метод `norm(self, Point)` — возвращает прямую, перпендикулярную данной и проходящую через заданную точку;

Как используется скалярное произведение векторов:

- $\vec{a} \cdot \vec{b} = 0$: вектора перпендикулярны;
- $\vec{a} \cdot \vec{b} > 0$: угол между векторами острый;
- $\vec{a} \cdot \vec{b} < 0$: угол между векторами тупой.

Как используется косое произведение векторов:

- модуль косого произведения это площадь параллелограмма, натянутого на вектора;
- косое произведение $\vec{a} \times \vec{b} = 0$: вектора коллинеарны; площадь натянутого параллелограмма равна 0;
- косое произведение $\vec{AB} \times \vec{BC} > 0$: тройка точек A, B, C «поворачивает» против часовой стрелки;
- косое произведение $\vec{AB} \times \vec{BC} < 0$: тройка точек A, B, C «поворачивает» по часовой стрелке;
- какая из точек P и Q находится ближе к прямой, проходящей через точки A и B : сравнение модулей косых произведений $\vec{AB} \times \vec{AP}$ и $\vec{AB} \times \vec{AQ}$;

Задачи

Использование скалярного и крестного произведения векторов

Внимание! При решении задач А-М использование вещественной арифметики допускается только при решении задач А, В, С, D, I, J.

A. Расстояние между двумя точками

Даны координаты двух точек.

Найдите расстояние между ними.

Input	Output
0 0 1 1	1.4142135623730951

B. Полярный угол точки

Даны два числа: координаты точки, не совпадающей с началом координат.

Выведите её полярный угол (величину от 0 до 2π).

Input	Output
2 3	0.98279

C. Угол между векторами

Даны четыре числа: координаты двух невырожденных векторов.

Выведите величину неориентированного угла между ними.

Input	Output
2 1 3 5	0.56673

D. Площадь треугольника

Даны шесть чисел: координаты трех вершин треугольника.

Выведите значение площади треугольника.

Input	Output
1 0 2 4 5 2	7.0

E. Классификация векторов

Даны четыре числа: координаты двух ненулевых векторов.

Если эти вектора коллинеарны, выведите 1. Если эти вектора перпендикулярны, выведите 2. Иначе выведите 0.

В этой задаче нельзя использовать обратные тригонометрические функции.

Input	Output
1 1 2 2	1
0 1 1 0	2
1 2 2 1	0

F. Три точки

Программа получает на вход шесть чисел: координаты трех точек.

Программа должна вывести YES, если эти точки лежат на одной прямой, или NO в противном случае.

Input	Output
0 0 1 1 2 2	YES

Г. *Принадлежность точки лучу*

Программа получает на вход шесть чисел: координаты точки и координаты начала и конца вектора. Проверьте, принадлежит ли данная точка лучу, задаваемому данным вектором (вершина луча совпадает с точкой — началом вектора).

Программа должна вывести YES, если точка принадлежит лучу, или NO в противном случае.

Input	Output
1 6 3 7 5 8	NO

Н. *Принадлежность точки отрезку*

Программа получает на вход шесть чисел: координаты точки и координаты концов отрезка. Проверьте, принадлежит ли данная точка данному отрезку.

Программа должна вывести YES, если точка принадлежит отрезку, или NO в противном случае.

Input	Output
3 3 1 2 5 4	YES

И. *Расстояние от точки до луча*

Дано шесть чисел: координаты точки, координаты начала и конца вектора.

Программа должна вывести единственное число: расстояние от точки до луча, заданного вектором.

Input	Output
2 1 1 1 0 2	1.0

Ж. *Расстояние от точки до отрезка*

Дано шесть чисел: координаты точки и координаты двух концов отрезка.

Программа должна вывести единственное число: расстояние от данной точки до данного отрезка.

Input	Output
0 4 2 3 2 5	2.0

К. *Принадлежит ли точка углу*

Дан угол $\angle AOB$ (O — вершина угла, A и B — точки на сторонах) и точка P . Определите, принадлежит ли точка P углу $\angle AOB$ (включая его стороны: лучи OA и OB).

Программа получает на вход координаты точек A, O, B, P . Все координаты — целые, не превосходят 100 по модулю. Точки A, O, B не лежат на одной прямой.

Программа должна вывести слово YES или NO.

Input	Output
0 1 0 0 1 0 1 1	YES
1 0 0 0 0 1 -1 -1	NO

L. *Пересекаются ли два луча*

Даны два луча: AB и CD (A и C — вершины лучей, B и D лежат на лучах). Проверьте, пересекаются ли они. Будем говорить, что лучи пересекаются, если у них есть хотя бы одна общая точка.

Программа получает на вход координаты точек A, B, C, D . Все координаты — целые, не превосходят 100 по модулю.

Программа должна вывести слово YES или NO.

Input	Output
0 1 1 2 1 -1 1 0	YES
0 0 1 0 0 1 1 2	NO

Указание: при решении этой задачи можно обойтись только определением принадлежности точки отрезку и косым произведением векторов.

Косое произведение, по-первых, позволяет определить, лежит ли пара точек C и D по одну сторону от прямой, которая задана точками A и B . Во-вторых, косое произведение позволяет определить, какая из двух точек C и D лежит дальше от прямой, содержащей точки A и B .

M. *Пересекаются ли два отрезка*

Даны два отрезка AB и CD . Проверьте, пересекаются ли они.

Программа получает на вход координаты точек A, B, C, D . Все координаты — целые, не превосходят 100 по модулю.

Программа должна вывести слово YES или NO.

Input	Output
0 1 1 2 1 -1 1 3	YES
0 0 1 0 0 1 1 2	NO

Указание: при решении этой задачи можно обойтись только определением принадлежности точки отрезку и косым произведением векторов. Косое произведение позволяет определить, лежит ли пара точек C и D по одну сторону от прямой, которая задана точками A и B .

Уравнение прямой

Во всех задачах можно считать, что коэффициенты A и B нормального уравнения прямой $A \cdot x + B \cdot y + C = 0$ не равны одновременно нулю.

N. Уравнение прямой

Прямая задана двумя точками.

Выведите коэффициенты A , B , C нормального уравнения прямой ($Ax + By + C = 0$).

Input	Output
1 2 3 1	-1 -2 5

O. Перпендикулярная прямая

Дано уравнение прямой и координаты точки.

Выведите коэффициенты уравнения прямой, перпендикулярной данной прямой и проходящей через данную точку.

Input	Output
0 1 -1 0 0	1 0 0

P. Принадлежность точки прямой

Даны координаты точки и уравнение прямой.

Определите, принадлежит ли точка прямой, выведите YES или NO.

Input	Output
3 7 -2 1 -1	YES

Q. Взаимное расположение двух точек

Даны две точки и уравнение прямой, точки не лежат на прямой.

Выведите YES, если точки лежат по одну сторону от прямой и NO в противном случае.

Input	Output
0 0 2 4 2 -1 -1	YES

R. Классификация двух прямых

Программа получает на вход шесть чисел: коэффициенты уравнений двух прямых.

Программа должна вывести 1, если эти прямые совпадают, 2 — если параллельны, 3 — если перпендикулярны и 0 во всех остальных случаях.

Input	Output
1 2 3 -1 -2 -3	1

S. Расстояние от точки до прямой

Даны пять чисел: координаты точки и коэффициенты нормального уравнения прямой.

Программа должна вывести одно число: расстояние от данной точки до данной прямой.

Input	Output
1 1 1 1 -1	0.70711

T. Параллельная прямая

Даны четыре числа: коэффициенты нормального уравнения прямой и величина d .

Программа должна вывести три числа: коэффициенты нормального уравнения любой из прямых, параллельных данной, и лежащих от нее на расстоянии d .

Input	Output
0 -1 1 1	0 -1 2

U. Основание перпендикуляра

Дано пять чисел: координаты точки и коэффициенты нормального уравнения прямой.

Программа должна вывести два числа: координаты основания перпендикуляра, опущенного из данной точки на данную прямую.

Input	Output
1 1 1 1 -1	0.5 0.5

V. *Пересечение двух прямых — 1*

Дано шесть чисел: коэффициенты нормальных уравнений двух непараллельных прямых.

Программа должна вывести два числа: координаты точки пересечения данных прямых.

Input	Output
1 1 -1 1 -1 0	0.5 0.5

W. *Пересечение двух прямых — 2*

На плоскости даны две прямые. Каждая прямая задается парой точек, через которые она проходит. Требуется установить, пересекаются ли эти прямые, и найти координаты точки пересечения.

Вводятся сначала координаты двух различных точек, через которые проходит первая прямая, а затем — координаты еще двух различных (но, быть может, совпадающих с первыми двумя) точек, через которые проходит вторая прямая. Координаты каждой точки — целые числа, по модулю не превышающие 1000.

Если прямые не пересекаются, выведите одно число 0. Если прямые совпадают, выведите 2. Если прямые пересекаются ровно в одной точке, то выведите сначала число 1, а затем два вещественных числа — координаты точки пересечения.

Input	Output
1 1 2 2 1 10 2 11	0
0 0 1 1 1 0 -1 2	1 0.5 0.5

Треугольники и окружности

Термины для геометрических объектов:

- median — медиана;
- angle bisector — биссектриса;
- perpendicular bisector — серединный перпендикуляр;
- incenter — центр вписанной окружности;
- circumcenter — центр описанной окружности;
- excenters — центры внеписанных окружностей;
- incircle — вписанная окружность;
- circumcircle — описанная окружность;
- centroid — центр тяжести, точка пересечения медиан;
- orthocenter — точка пересечения высот;

X. Биссектриса угла

Даны координаты трех точек O , A , B .

Постройте уравнение прямой, являющейся биссектрисой угла $\angle AOB$. Выведите коэффициенты A , B , C нормального уравнения этой прямой.

Input	Output
1 1 1 0 0 1	-1.0 1.0 0.0

Y. Точка пересечения медиан

Дан треугольник. Найдите точку пересечения его медиан.

Input	Output
0 0 1 0 0 1	0.333333 0.333333

Z. Точка пересечения биссектрис

Дан треугольник. Найдите точку пересечения его биссектрис.

Input	Output
0 0 1 0 0 1	0.292893 0.292893

ZA. Точка пересечения высот

Дан треугольник. Найдите точку пересечения его высот.

Input	Output
0 0 1 0 0 1	0.0 0.0

ZB. Точка пересечения серединных перпендикуляров

Дан треугольник. Найдите точку пересечения его серединных перпендикуляров.

Input	Output
0 0 1 0 0 1	0.5 0.5

ZC. Вписанная окружность

Выведите радиус окружности, вписанной в данный треугольник.

Input	Output
0 0 1 0 0 1	0.2928932188134525

ZD. Описанная окружность

Выведите радиус окружности, описанной вокруг данного треугольника.

Input	Output
0 0 1 0 0 1	0.7071067811865476

ZЕ. *Принадлежность точки треугольнику*

Дан треугольник и точка.

Выведите слово YES, если точка принадлежит треугольнику (она может лежать на границе треугольника) и слово NO в противном случае.

Input	Output
0 0 10 0 0 5 1 2	YES

ZF. *Минимальная окружность*

Дан треугольник. Найдите минимальную окружность, содержащую внутри себя треугольник.

Выведите три числа: координаты центра и радиус данной окружности.

Input	Output
0 0 1 0 0 1	0.5 0.5 0.7071067811865476

ZG. *Расстояние от прямой до окружности*

Даны шесть чисел: координаты центра окружности, ее радиус (в первой строке), коэффициенты нормального уравнения прямой (во второй строке).

Выведите единственное число: расстояние от данной окружности до данной прямой.

Input	Output
0 0 1 1 0 -2	1.0

ZH. *Пересечение прямой и окружности*

Даны шесть чисел: координаты центра окружности, ее радиус (в первой строке), коэффициенты нормального уравнения прямой (во второй строке).

В первой строке программа должна вывести одно число K , равное количеству точек пересечения прямой с окружностью. Далее в K строках координаты самих точек.

Input	Output
1 1 1 1 -1 0	2 1.70711 1.70711 0.29289 0.29289

ZI. *Угол обзора*

Даны пять чисел: координаты центра окружности, ее радиус (в первой строке), координаты точки, лежащей вне окружности (во второй строке).

Выведите единственное число: угол (в радианах), под которым видна данная окружность из данной точки.

Input	Output
1 1 1 2 2	1.5707963267948966

ZJ. *Поворот точки*

Даны три числа: координаты точки и угол (в радианах, задан в виде действительного числа).

Выведите координаты точки, полученной поворотом данной точки вокруг начала координат на данный угол в положительном направлении.

Input	Output
1 0 1.5707963267948966	0.0 1.0

ZK. *Поворот прямой*

Даны четыре числа: коэффициенты нормального уравнения прямой и угол (в радианах, задан в виде действительного числа).

Выведите три числа: коэффициенты нормального уравнения прямой, полученной поворотом данной прямой вокруг начала координат на данный угол в положительном направлении.

Input	Output
1 1 -1 1.5707963267948966	1.0 -1.0 1.0

ZL. *Точки касания*

Даны пять чисел: координаты центра окружности, ее радиус (в первой строке), координаты точки (во второй строке).

Выведите одно число K, равное количеству точек пересечения всевозможных касательных к окружности, проходящих через данную точку. Далее в K строках координаты самих точек пересечения касательных с окружностью.

Input	Output
1 1 1	2
2 2	1.0 2.0
	2.0 1.0

ZM. *Длина дуги*

Даны семь чисел – координаты центра и радиус окружности (в первой строке) и действительные координаты двух точек на ней (во второй и третьей строке), с точностью до пятого знака после запятой.

Выведите одно число: длину меньшей из дуг окружности, заключенной между указанными точками.

Input	Output
1 1 1	1.5707963267948966
2.0 1.0	
1.0 2.0	

ZN. *Пересечение окружностей*

Даны шесть чисел – координаты центров и радиусы двух окружностей окружности.

В случае если количество общих точек окружностей конечно, в первой строке вывести одно число K, равное этому количеству, далее в K строках координаты самих точек.

Если указанных точек бесконечно много, вывести единственное число "3".

Input	Output
2 3 1	2
3 2 1	3.0 3.0
	2.0 2.0

Многоугольники

ZO. Площадь многоугольника

Задан многоугольник: сначала задано количество вершин многоугольника N , не превосходящее 10^5 , затем N вершин многоугольника в порядке обхода.

Выведите единственное число: площадь многоугольника.

Input	Output
3 1 0 0 1 1 1	0.5

ZP. Выпуклость многоугольника

Задан многоугольник: сначала задано количество вершин многоугольника N , не превосходящее 10^5 , затем N его вершин, как в предыдущей задаче.

Выведите YES, если многоугольник является выпуклым и NO в противном случае.

Input	Output
3 1 0 0 1 1 1	YES

ZQ. Принадлежность точки многоугольнику

Первая строка входных данных содержит число N ($3 \leq N \leq 10^5$). Далее идет N точек — координаты вершин многоугольника. Затем идут координаты проверяемой точки.

Программа должна вывести YES, если точка лежит внутри многоугольника или на его границе и NO в противном случае.

Координаты всех точек могут быть дробными числами.

Input	Output
3 3 0 0 3 0 0 1 1	YES

ZR. Принадлежность точки выпуклому многоугольнику

Первая строка входных данных содержит число N , ($3 \leq N \leq 10^5$).

Далее, в N строках записаны координаты вершин выпуклого многоугольника.

Затем записано число K , ($3 \leq N \leq 10^5$) — количество проверяемых точек. Затем в K строчках записаны координаты K проверяемых точек.

Для каждой проверяемой точки выведите YES, если точка лежит внутри многоугольника или на его границе и NO в противном случае.

Координаты всех точек могут быть дробными числами.

Input	Output
3 0 0 2 0 0 2 2 1 1 2 2	YES NO