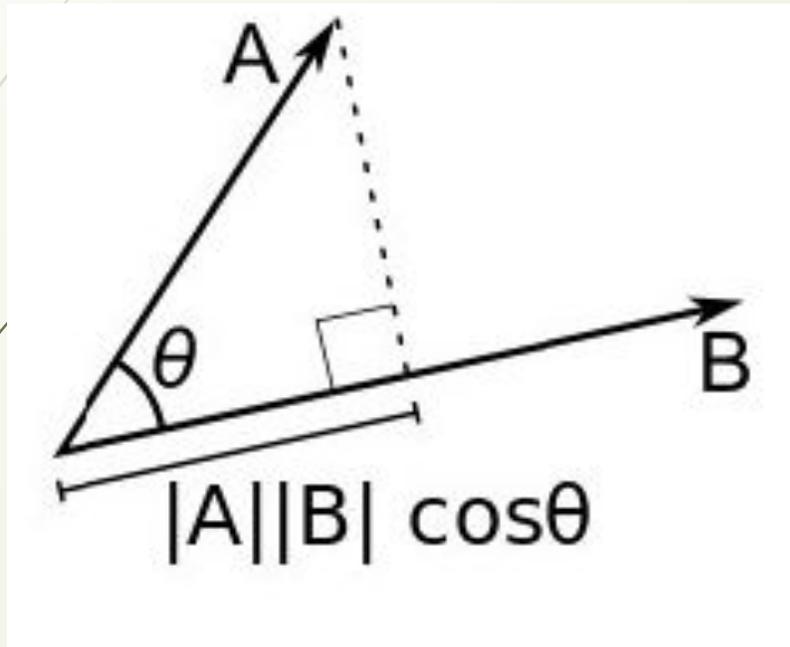




Вычислительная геометрия

Скалярное произведение векторов



$$\square a \cdot b = |a| \cdot |b| \cos \theta$$

$$\square a \cdot b = a_x \cdot b_x + a_y \cdot b_y$$



По введенным трем числам a , b , c определить существует ли треугольник с такими сторонами.

Неравенство треугольника является необходимым и достаточным условием существования треугольника

$a + b > c$

$a + c > b$

$b + c > a$

Определить существует ли треугольник с такими координатами вершин.

Треугольника не существует когда данные три точки лежат на одной прямой.

Проверяется через кросс произведение векторов:

$$[a, b] = x_1y_2 - x_2y_1.$$

Если оно равно нулю, то векторы коллинеарные, то есть все три точки лежат на одной прямой.

**Треугольник задан своими сторонами.
Определить тип треугольника:
тупоугольный, прямоугольный или
остроугольный.**

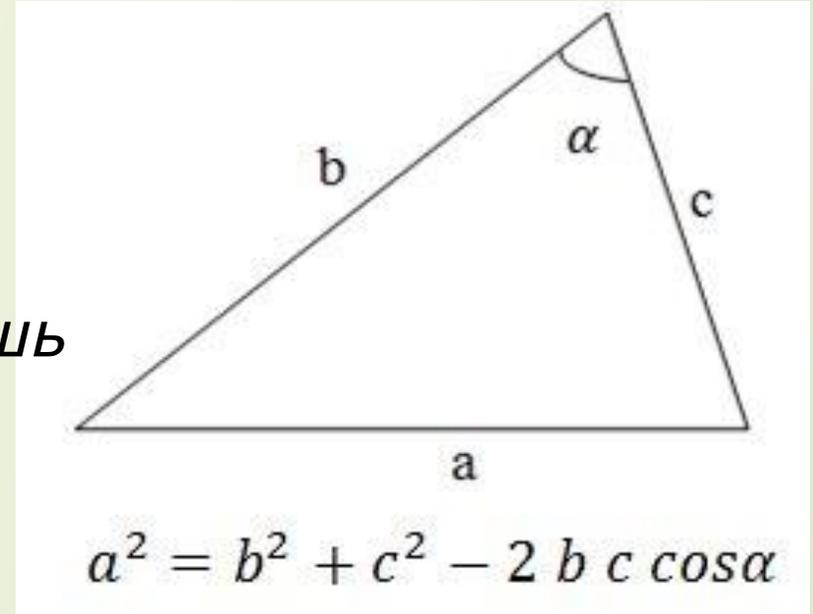
Теорема косинусов:

Вычислять косинус угла не обязательно, необходимо учесть лишь его знак:

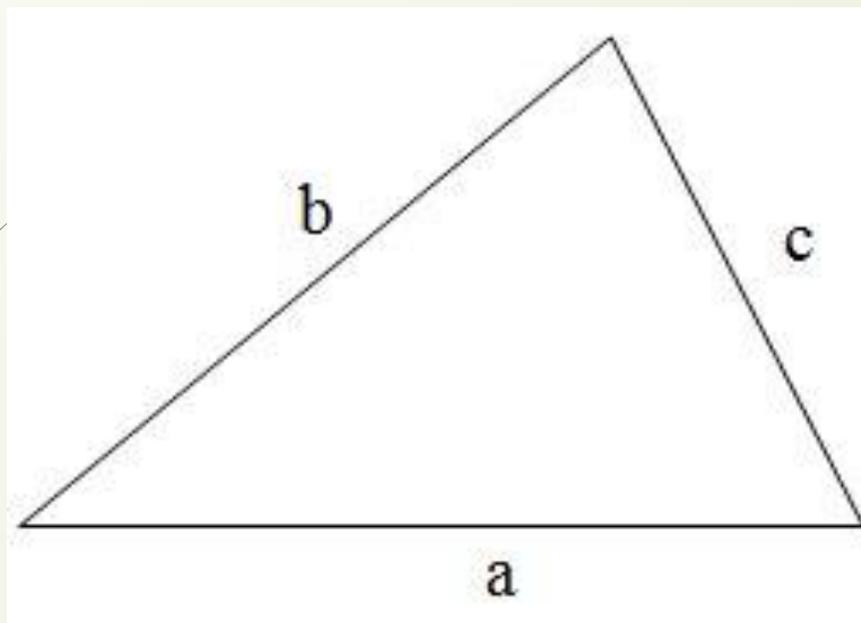
Если $\cos \alpha > 0$, то $a^2 < b^2 + c^2$ –
треугольник остроугольный

• Если $\cos \alpha = 0$, то $a^2 = b^2 + c^2$ –
треугольник прямоугольный

• Если $\cos \alpha < 0$, то $a^2 > b^2 + c^2$ –
треугольник тупоугольный
где a – большая сторона.



**По данным сторонам треугольника
найти его площадь.**



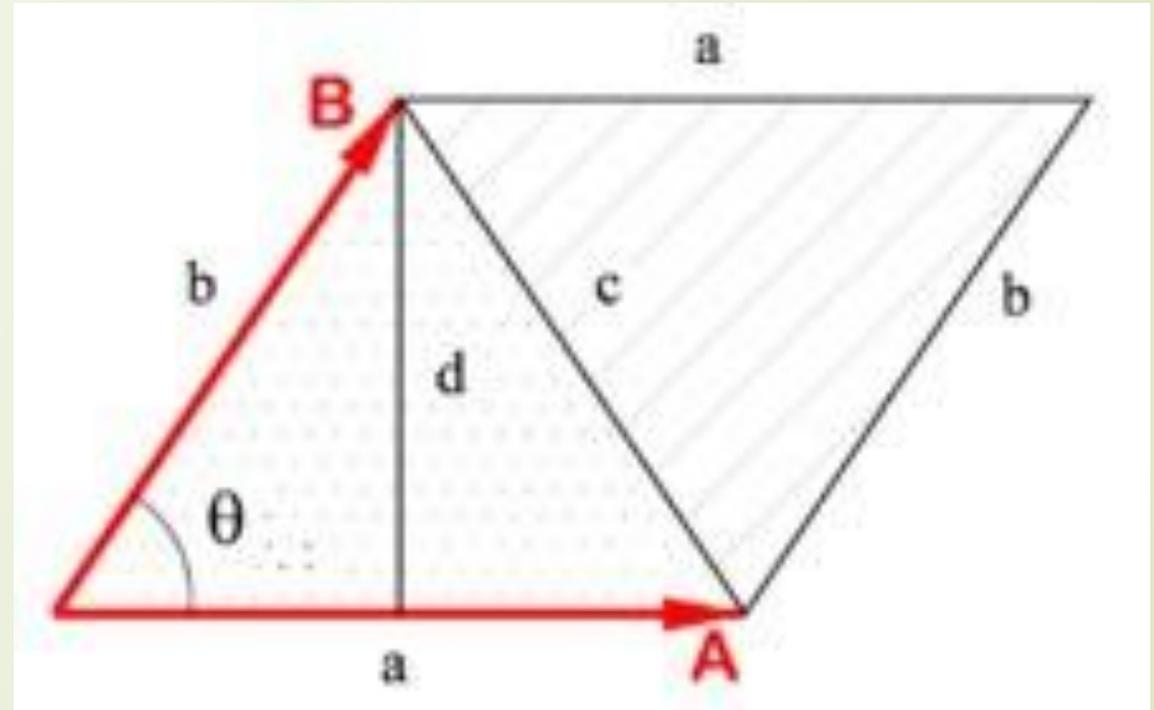
$$p = \frac{a+b+c}{2}$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

Вычислить площадь треугольника заданного координатами своих вершин.

Косое произведение двух векторов определяет ориентированную площадь параллелограмма основанного на этих векторах.

$S = (x_1 y_2 - x_2 y_1) / 2$ — ориентированная площадь треугольника
 x_1, y_1 — координаты вектора A



$$S = \left| \frac{(x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)}{2} \right|$$

Где $x_1, y_1, x_2, y_2, x_3, y_3$ — координаты точек.

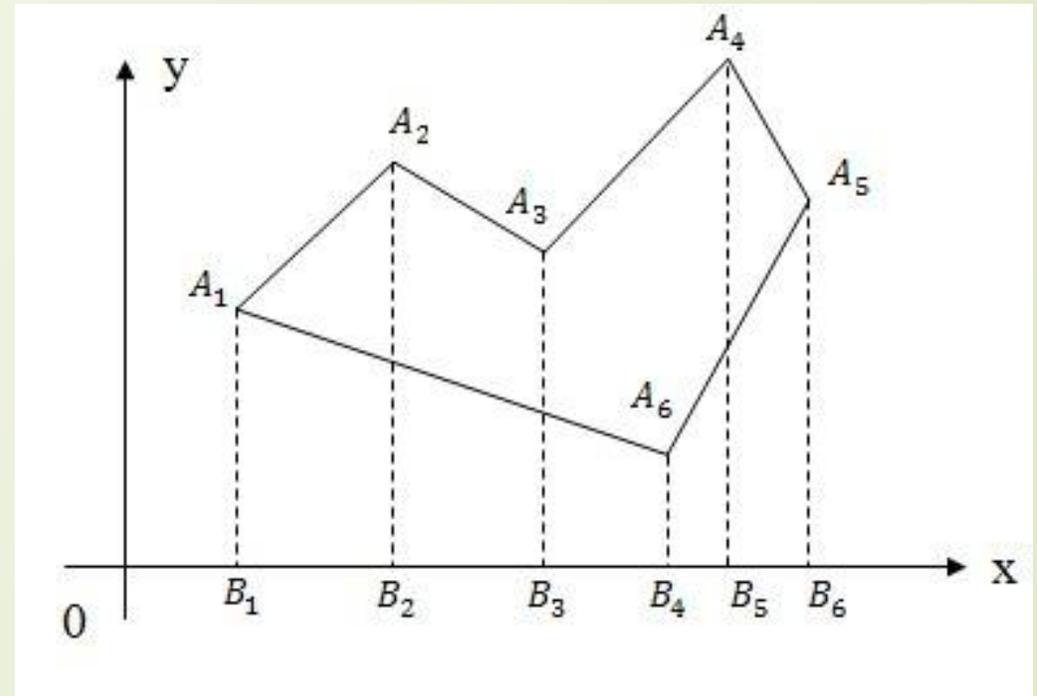
Вычисление площади многоугольника заданного координатами своих вершин.

Метод трапеций

$$S = S_{A_1 A_2 B_2 B_1} + S_{A_2 A_3 B_3 B_2} + S_{A_3 A_4 B_5 B_3} + S_{A_4 A_5 B_6 B_5} - S_{A_5 A_6 B_4 B_6} - S_{A_6 A_1 B_1 B_4}$$

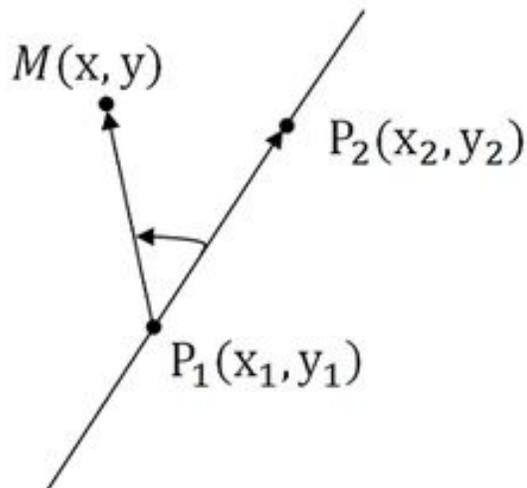
Площади трапеций: полусумма оснований на высоту

$$S_{A_1 A_2 B_2 B_1} = 0.5 * (A_1 B_1 + A_2 B_2) * (B_2 - B_1)$$



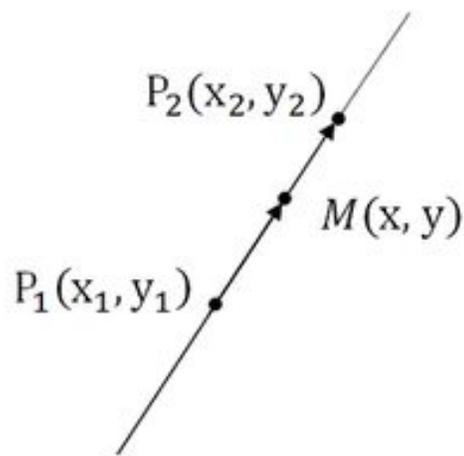
Определить взаимное расположение точки и прямой: лежит выше прямой, на прямой, под прямой.

- Косое произведение двух векторов положительно, если поворот от первого вектора ко второму идет против часовой стрелки, равно нулю, если векторы коллинеарны и отрицательно, если поворот идет по часовой стрелки.



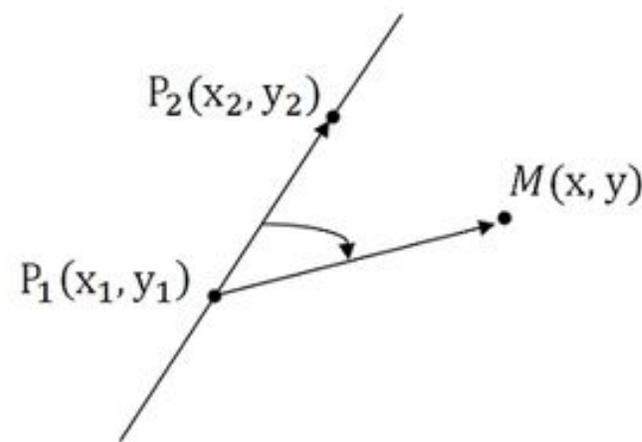
$$[P_1P_2, P_1M] > 0$$

Точка лежит в верхней
полуплоскости



$$[P_1P_2, P_1M] = 0$$

Точка лежит на прямой



$$[P_1P_2, P_1M] < 0$$

Точка лежит в нижней
полуплоскости

Симметрия

(Время: 1 сек. Память: 16 Мб Сложность: 19%)

Многие из вас, вероятно, знакомы с понятием симметрии относительно прямой. Пусть на плоскости расположена прямая L и точка A . Точка B называется симметричной точке A относительно прямой L , если отрезок AB перпендикулярен прямой L и делится пополам точкой пересечения с ней. В частности, если точка A лежит на прямой L , то точка B совпадает с точкой A . Задана прямая L , параллельная одной из осей координат, и точка A . Найдите точку B , симметричную A относительно L .

Входные данные

Первая строка входного файла INPUT.TXT содержит 4 числа: x_1, y_1, x_2, y_2 — координаты двух различных точек, через которые проходит прямая L . Вторая строка входного файла содержит 2 числа x_A и y_A — координаты точки A . Все числа во входном файле целые и не превосходят 10^8 по модулю.

Выходные данные

В выходной файл OUTPUT.TXT выведите числа x_B и y_B — координаты точки B .



№	INPUT.TXT	OUTPUT.TXT
1	0 0 0 1 10 10	-10 10
2	0 0 1 0 10 10	10 -10



```
var x1,y1,x2,y2,ax,ay,bx,by:longint;  
begin  
  assign(input,'input.txt'); reset(input);  
  assign(output,'output.txt'); rewrite(output);  
  readln(x1,y1,x2,y2,ax,ay);  
  if x1=x2 then  
    begin  
      bx:=2*x1-ax; by:=ay;  
    end;  
  if y1=y2 then  
    begin  
      by:=2*y1-ay; bx:=ax;  
    end;  
  writeln(bx,' ',by);  
end.
```

Треугольник и точка

(Время: 1 сек. Память: 16 Мб Сложность: 32%)

В декартовой системе координат на плоскости заданы координаты вершин треугольника и еще одной точки. Требуется написать программу, определяющую, принадлежит ли эта точка треугольнику.

Входные данные

В четырех строках входного файла INPUT.TXT находятся пары целых чисел - координаты точек. Числа в первых трех строках - это координаты вершин треугольника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , в четвертой строке - координаты тестируемой точки (x_4, y_4) . Все координаты не превышают 10000 по абсолютной величине.

Выходные данные

В выходной файл OUTPUT.TXT необходимо вывести слово «In», если точка находится внутри треугольника и «Out» в противном случае.



No	INPUT.TXT	OUTPUT.TXT
1	0 0 100 0 0 100 100 100	Out
2	0 0 100 0 0 100 10 10	In
3	0 0 100 0 0 100 50 50	In
4	0 0 100 0 0 100 0 0	In

➤ Φορμουλα

$$S = \left| \frac{(x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)}{2} \right|$$

- var x1,y1,x2,y2,x3,y3,x4,y4:integer;
- s,s1,s2,s3:real;
- begin
- assign(input, 'input.txt'); reset(input);
- assign(output, 'output.txt'); rewrite(output);
- readln(x1,y1,x2,y2,x3,y3,x4,y4);
- s:=abs(((x2-x1)*(y3-y1)-(x3-x1)*(y2-y1))/2);
- s1:=abs(((x4-x1)*(y3-y1)-(x3-x1)*(y4-y1))/2);
- s2:=abs(((x4-x2)*(y2-y1)-(x2-x1)*(y4-y2))/2);
- s3:=abs(((x4-x2)*(y3-y2)-(x3-x2)*(y4-y2))/2);
- if (s1+s2+s3)<=s then writeln('In') else writeln('Out');
- end.

Фонарики

(Время: 1 сек. Память: 16 Мб Сложность: 31%)

«Одна голова хорошо, а две лучше. Одна лампочка хорошо, а две лучше!» - подумал Миша, и решил собрать фонарик с двумя лампочками. Теперь он хочет узнать, насколько фонарик с двумя лампочками лучше, чем фонарик с одной. Для этого Миша посветил фонариком на стену, и каждая из лампочек осветила на ней круг.

Эффективность фонарика Миша хочет оценить через площадь освещенной части стены. Миша догадался измерить координаты центров освещенных кругов и их радиусы (которые оказались одинаковыми). Причем, площадь, освещаемая фонариком с одной лампочкой известна, т.к. описана в документации, прилагаемой к фонарику. Но что делать дальше он не знает. Напишите программу, которая поможет Мише.

Входные данные

В первых двух строчках входного файла INPUT.TXT содержатся координаты (x_1, y_1) и (x_2, y_2) - центры кругов от лампочек собранного Мишей фонарика. В третьей строке задан радиус r описанных выше кругов, а четвертая строка содержит площадь освещения s фонариком из одной лампочки. Все числа целые и удовлетворяют следующим ограничениям: $1 \leq x_1, y_1, x_2, y_2, r \leq 100$, $1 \leq s \leq 10^5$. Так же заметим, что площади, освещаемые разными фонариками, отличаются друг от друга более чем на 10^{-3} .

Выходные данные

В выходной файл OUTPUT.TXT выведите «YES», если Мишин фонарик лучше старого (т.е. освещает большую площадь) и «NO» в противном случае.



№	INPUT.TXT	OUTPUT.TXT
1	1 2 3 4 2 22	YES
2	1 1 100 100 1 7	NO

□ **Фонарики**

```
□ uses math;
□ var x1,y1,x2,y2,r,s:longint;
□ d,h,st,a,al,ss,sp:double;
□ begin
□ assign(input, 'input.txt'); reset(input);
□ assign(output, 'output.txt'); rewrite(output);
□ readln(x1,y1,x2,y2,r,s);
□ d:=sqrt(sqr(x1-x2)+sqr(y1-y2));
□ if (x1=x2) and (y1=y2) then
□ begin
□ if pi*r*r>s then writeln('YES') else writeln('NO');
□ exit
□ end;
□ if d>=2*r then sp:=0
□ else begin
□ h:=d/2;
□ a:=sqrt(sqr(r)-sqr(h));
□ st:=a*h;
□ al:=arccos((r*r-2*a*a)/r*r);
□ ss:=pi*r*r*al/360;
□ sp:=2*(ss-st);
□ end;
□ if (2*pi*r*r-sp)>s then writeln('YES') else writeln('NO');
□ end.
```