12.三角形四心计算

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define LD long double  const LD eps=1e-9;  int cmp(LD x)  {  if(fabs(x)<eps)  return 0;  if(x>0)  return 1;  else  return -1;  }  const LD pi=acos(-1.0);  inline LD sqr(LD x)  {  return x\*x;  }  struct point  {  LD x,y;  point () {}  point (LD a,LD b):x(a),y(b) {}  void input()  {  cin>>x>>y;  }  friend point operator +(const point &a,const point &b)  {  return point(a.x+b.x,a.y+b.y);  }  friend point operator -(const point &a,const point &b)  {  return point(a.x-b.x,a.y-b.y);  }  friend bool operator ==(const point &a,const point &b)  {  return cmp(a.x-b.x)==0&&cmp(a.y-b.y)==0;  }  friend point operator \*(const point &a,const LD &b)  {  return point(a.x\*b,a.y\*b);  }  friend point operator\*(const LD &a,const point &b)  {  return point(a\*b.x,a\*b.y);  }  friend point operator /(const point &a,const LD &b)  {  return point(a.x/b,a.y/b);  }  LD norm()  {  return sqrt(sqr(x)+sqr(y));  }  };  point Triangle\_Mass\_Center(point a,point b,point c)  {  return(a+b+c)/3.0;  }  point CircumCenter(point p0,point p1,point p2)  {  point cp;  LD a1=p1.x-p0.x,b1=p1.y-p0.y,c1=(a1\*a1+b1\*b1)/2.0;  LD a2=p2.x-p0.x,b2=p2.y-p0.y,c2=(a2\*a2+b2\*b2)/2.0;  LD d=a1\*b2-a2\*b1;  cp.x=p0.x+(c1\*b2-c2\*b1)/d;  cp.y=p0.y+(a1\*c2-a2\*c1)/d;  return cp;  }  point Orthocenter(point a,point b,point c)  {  return Triangle\_Mass\_Center(a,b,c)\*3.0-CircumCenter(a,b,c)\*2.0;  }  point Innercenter(point a,point b,point c)  {  point cp;  LD la,lb,lc;  la=(b-c).norm();  lb=(c-a).norm();  lc=(a-b).norm();  cp.x=(la\*a.x+lb\*b.x+lc\*c.x)/(la+lb+lc);  cp.y=(la\*a.y+lb\*b.y+lc\*c.y)/(la+lb+lc);  return cp;  }  int main()  {  point a,b,c;  a.input();  b.input();  c.input();  point d;  cin>>d.x>>d.y;  point ans1=Orthocenter(a,b,c);  point ans2=Innercenter(a,b,c);  point ans3=CircumCenter(a,b,c);  point ans4=Triangle\_Mass\_Center(a,b,c);  if((d==ans1)||(d==ans2)||(d==ans3)||(d==ans4))  printf("Yes\n");  else  printf("No\n");  return 0;  } |