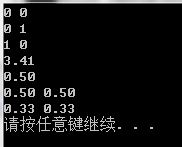
**12-1三角形**

问题描述

　　为二维空间中的点设计一个结构体，在此基础上为三角形设计一个结构体。分别设计独立的函数计算三角形的周长、面积、中心和重心。输入三个点，输出这三个点构成的三角形的周长、面积、外心和重心。结果保留小数点后2位数字。

样例输出

与上面的样例输入对应的输出。  
例：  


数据规模和约定

　　输入数据中每一个数的范围。  
　　例：doule型表示数据。

**C++：**

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

struct Spot

{

double x;

double y;

};

class Triangle

{

private :

double la;

double lb;

double lc;

public :

Triangle(struct Spot,struct Spot,struct Spot);

double S();

struct Spot G(struct Spot a,struct Spot b,struct Spot c);

struct Spot Q(struct Spot a,struct Spot b,struct Spot c);

double p;

};

Triangle::Triangle(struct Spot a,struct Spot b,struct Spot c)

{

la=sqrt(pow(a.x-b.x,2)+pow(a.y-b.y,2));

lb=sqrt(pow(a.x-c.x,2)+pow(a.y-c.y,2));

lc=sqrt(pow(b.x-c.x,2)+pow(b.y-c.y,2));

p=(la+lb+lc)/2;

}

double Triangle::S()

{

return sqrt(p\*(p-la)\*(p-lb)\*(p-lc));

}

struct Spot Triangle::G(struct Spot a,struct Spot b,struct Spot c)

{

struct Spot g;

g.x=(a.x+b.x+c.x)/3;

g.y=(a.y+b.y+c.y)/3;

return g;

}

struct Spot Triangle::Q(struct Spot a,struct Spot b,struct Spot c)

{

double A1,B1,C1;

double A2,B2,C2;

struct Spot q;

A1=2\*(b.x-a.x);

B1=2\*(b.y-a.y);

C1=pow(b.x,2)+pow(b.y,2)-pow(a.x,2)-pow(a.y,2);

A2=2\*(c.x-b.x);

B2=2\*(c.y-b.y);

C2=pow(c.x,2)+pow(c.y,2)-pow(b.x,2)-pow(b.y,2);

q.x=((C1\*B2)-(C2\*B1))/((A1\*B2)-(A2\*B1));

q.y=((A1\*C2)-(A2\*C1))/((A1\*B2)-(A2\*B1));

return q;

}

int main()

{

struct Spot a,b,c,g,q;

cin>>a.x;

cin>>a.y;

cin>>b.x;

cin>>b.y;

cin>>c.x;

cin>>c.y;

Triangle T(a,b,c);

g=T.G(a,b,c);

q=T.Q(a,b,c);

cout<<setprecision(2)<<std::fixed<<T.p\*2<<endl;

cout <<setprecision(2) <<std::fixed <<T.S()<<endl;

cout<<setprecision(2)<<std::fixed<<q.x<<" "<<q.y<<endl;

cout<<setprecision(2)<<std::fixed<<g.x<<" "<<g.y<<endl;

return 0;

}

**C:**

#include<stdio.h>

#include<math.h>

typedef struct node

{

double x;

double y;

}V;

typedef struct Node

{

V A;

V B;

V C;

}Three;

int main()

{

Three Q;

double a,b,c,p,C,S;

V O,G;

scanf("%lf%lf",&Q.A.x,&Q.A.y);

scanf("%lf%lf",&Q.B.x,&Q.B.y);

scanf("%lf%lf",&Q.C.x,&Q.C.y);

a=sqrt(pow((Q.B.x-Q.C.x),2)+pow((Q.B.y-Q.C.y),2));

b=sqrt(pow((Q.A.x-Q.C.x),2)+pow((Q.A.y-Q.C.y),2));

c=sqrt(pow((Q.A.x-Q.B.x),2)+pow((Q.A.y-Q.B.y),2));

C=a+b+c;

printf("%.2f\n",C);

p=C/2;

S=sqrt(p\*(p-a)\*(p-b)\*(p-c));

printf("%.2lf\n",S);

O.x=(pow(Q.A.x,2)\*(Q.B.y-Q.C.y)+pow(Q.B.x,2)\*(Q.C.y-Q.A.y)+pow(Q.C.x,2)\*(Q.A.y-Q.B.y)-(Q.A.y-Q.B.y)\*(Q.B.y-Q.C.y)\*(Q.C.y-Q.A.y))/(2\*(Q.A.x\*(Q.B.y-Q.C.y)+Q.B.x\*(Q.C.y-Q.A.y)+Q.C.x\*(Q.A.y-Q.B.y)));

O.y=(pow(Q.A.y,2)\*(Q.B.x-Q.C.x)+pow(Q.B.y,2)\*(Q.C.x-Q.A.x)+pow(Q.C.y,2)\*(Q.A.x-Q.B.x)-(Q.A.x-Q.B.x)\*(Q.B.x-Q.C.x)\*(Q.C.x-Q.A.x))/(2\*(Q.A.y\*(Q.B.x-Q.C.x)+Q.B.y\*(Q.C.x-Q.A.x)+Q.C.y\*(Q.A.x-Q.B.x)));

printf("%.2lf %.2lf\n",O.x,O.y);

G.x=(Q.A.x+Q.B.x+Q.C.x)/3;

G.y=(Q.A.y+Q.B.y+Q.C.y)/3;

printf("%.2lf %.2lf",G.x,G.y);

return 0;

}

**Java：**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

double x1 = in.nextDouble();

double x2 = in.nextDouble();

double y1 = in.nextDouble();

double y2 = in.nextDouble();

double z1 = in.nextDouble();

double z2 = in.nextDouble();

double a, b, c;

a = Math.sqrt((z1 - y1) \* (z1 - y1) + (z2 - y2) \* (z2 - y2));

b = Math.sqrt((x1 - z1) \* (x1 - z1) + (x2 - z2) \* (x2 - z2));

c = Math.sqrt((x1 - y1) \* (x1 - y1) + (x2 - y2) \* (x2 - y2));

// System.out.println("a=" + a + " b=" + b + " c=" + c);

double perimeter, area, center1, center2, core1, core2;

perimeter = a + b + c;

area = Math.abs(((x1 - z1) \* (y2 - z2) - (y1 - z1) \* (x2 - z2)) / 2);

core1 = (x1 + y1 + z1) / 3;

core2 = (x2 + y2 + z2) / 3;

center1 = ((x1 \* x1 - y1 \* y1 + x2 \* x2 - y2 \* y2) \* (x2 - z2) - (x1

\* x1 - z1 \* z1 + x2 \* x2 - z2 \* z2)

\* (x2 - y2))

/ (2 \* (x2 - z2) \* (x1 - y1) - 2 \* (x2 - y2) \* (x1 - z1));

center2 = ((x1 \* x1 - y1 \* y1 + x2 \* x2 - y2 \* y2) \* (x1 - z1) - (x1

\* x1 - z1 \* z1 + x2 \* x2 - z2 \* z2)

\* (x1 - y1))

/ (2 \* (x2 - y2) \* (x1 - z1) - 2 \* (x2 - z2) \* (x1 - y1));

System.out.printf("%.2f\n", perimeter);

System.out.printf("%.2f\n", area);

System.out.printf("%.2f %.2f\n", center1, center2);

System.out.printf("%.2f %.2f\n", core1, core2);

}

}

// (a·x1/(a+b+c)+b·x2(a+b+c)+c·x3(a+b+c)，a·y1/(a+b+c)+b·y2/(a+b+c)+c·y3(a+b+c))