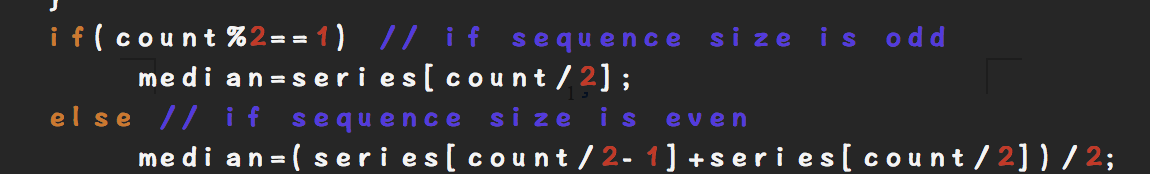
**★★★☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11066: Aragon**

**解題者：周永霖**

**解題日期：2017年4月27日**

**題意：**

在中古歐洲，你得幫助Aragron成功的率領軍隊，越過Misty Mountain。 但在Misty Mountain上常常有cloud籠罩，會有orcs躲在迷霧偷襲。所以聰明的大家要幫助Aragron判斷“哪個Mountain區塊會被cloud籠罩”。  
題目會“先”給一系列“雲”的“點” ，再給一系列“山”的“點”，請見題意範例。

**題意範例：**

**Input：**

2 ->cloud點的個數

0.0 2.0

4.0 2.0

5 ->mountain點的個數

0.0 0.1

1.0 3.0

2.0 1.0

3.0 3.0

4.0 1.0

**output:**

1.0 ->面積

**解法：**

**Step1:透過“頂點”切割區塊（而非“交點”）**

**Step2:判斷是哪種情況**

**Step3:把用兩條線的斜率跟兩點算出焦點的”x”座標**

**F1(x1,y1,x2,y2,r1,r2) return 焦點的x座標;**

**Step4:把mountain的x丟到cloud的func.裡求對應的y**

**F2(x1,y1,r,想找對應的x) return 相對應的y;**

**Step5:這個區塊的三角形面積即為：**

**(Mountain.y - cloud.y) \* (Mountain.x – 焦點.x) \* ½**

**Step6:繼續做下一個區塊（即回step1）**

**解法範例：**

**example1:**

**區塊1:**

因為右邊線Mountain的y>Cloud的y ，所以把焦點的x座標 透過兩個斜率跟兩點求出來(F1),斜率分別是2、0，得出焦點的x座標為1/2。

高：1-(1/2)=1/2

把右邊邊線頂點的x餵給F2,得出cloud的y座標為2

底：3-2=1

Area : (底＊高 ) /2＝(1/2)\*1 / 2 = ¼

依此類推->區塊2、3、4都可求出為1/4

總面積為1/4\*4=1

example2:

**區塊2:**

因為右邊線跟左邊線的Mountain的y都大於Cloud的y ，判斷出為四邊形，所以切割為兩個三角形。

高：左右兩邊的x座標互減（因為共用高）。

把左、右邊邊線頂點的x餵給F2,可得出個別的cloud的y座標

底：即跟區塊1一樣。

Area : 兩塊三角形加起來。

程式：

#include <cstdio>

#include <cmath>

#include <iostream>

#include <algorithm>

#define a 1e-6

using namespace std;

struct Point {

    double x,y,r;

    bool operator <(const Point &q) const { return x < q.x; }

};

double fdx(double,double,double,double,double,double);

double fdy(double,double,double,double);

int num\_c, num\_m, num\_cc;

Point c[1024], m[1024], cc[4096];

int main(){

    while(scanf("%d", &num\_c) == 1 && num\_c>0){

            for(int i=0; i<num\_c ; i++)  cin >> c[i].x >> c[i].y;

            cin >> num\_m;

            for(int i=0 ; i<num\_m ; i++)  cin >> m[i].x >> m[i].y;

            num\_cc = 0;

            for(int i=0 ; i<num\_c-1 ; i++){

                cc[num\_cc].x=c[i].x;

                cc[num\_cc].y=-(c[i+1].y-c[i].y)/(c[i+1].x-c[i].x);

                cc[num\_cc+1].x=c[i+1].x;

                cc[num\_cc+1].y=-cc[num\_cc].y;

                num\_cc += 2;

            }

            for (int i=0; i<num\_m-1 ; i++) {

                cc[num\_cc].x = m[i].x;

                cc[num\_cc].y = (m[i+1].y-m[i].y)/(m[i+1].x-m[i].x);

                cc[num\_cc+1].x = m[i+1].x;

                cc[num\_cc+1].y = -cc[num\_cc].y;

                num\_cc += 2;

            }

            sort(cc,cc+num\_cc);

            double y0=m[0].y-c[0].y,x0=m[0].x, r=0,total=0;

            double s,y1;

            double t,w,yc;

            for (int i=0; i<num\_cc;) {

                if (fabs(x0-cc[i].x)<a){

                    r+=cc[i++].y;

                    continue;

                }

                s=cc[i].x-x0;

                y1=y0+r\*s;

                t=fdx(m[i].x,m[i].y,m[i].r,c[i].x,c[i].y,c[i].r);

                w=fdy(t,m[i-1].x,m[i-1].y,m[i-1].r);

                if(y0>-a&&y1>-a){

                    t=y0/w;

                    w=r\*t;

                    total+=y0\*s+r\*s\*s/2;

                }

                else if(y0>a && y1<-a){

                    t=-y0/r;

                    w=t\*r\*r;

                    total+=y0\*t+r\*t\*t/2;

                }

                else if(y0<-a && y1>a){

                    t=-y0/r;

                    w=s-t;

                    yc=y0+r\*t;

                    total+=yc\*w+r\*w\*w/2;

                }

                y0=y1;

                x0=cc[i].x;

            }

            if (total<a)

                total = 0.0;

            printf("%.2f\n", total);

    }

}

double fdx(double x1,double y1,double r1,double x2,double y2,double r2){

    double fdx,c1,c2;

    c1=y1-r1\*x1;

    c2=y2-r2\*x2;

    fdx=(c2-c1)/(r1-r2);

    return fdx;

}

double fdy(double xxx,double x,double y,double r){

    double fdy,c;

    c=y-r\*x;

    fdy=r\*xxx+c;

    return fdy;

}