半径为R的圆与多边形面积的交

三角剖分

```
前面还有头文件，pi,eps，sgn，Point结构体+重载，点乘，叉乘
double dis(Point A,Point B){
    return sqrt((A.x-B.x)*(A.x-B.x) + (A.y-B.y)*(A.y-B.y));
}

double Length(Point A){return sqrt(Dot(A,A));}//计算向量的长度

Point rotate(Point a, double b){
    return {a.x * cos(b) + a.y * sin(b), -a.x * sin(b) + a.y * cos(b)};
}

Point norm(Point a){//将向量a标准化
    return a / Length(a);
}

//P点再P1P2上 叉积==0
bool OnSeg(Point P,Point P1,Point P2){
    return sgn(Cross(P1 - P,P2 - P)) == 0 && sgn(Dot(P1 - P,P2 - P)) <= 0;
}

//判断共线
bool Collinear(Point P1,Point P2,Point P3,Point P4){
    double A = Cross(P1 - P2,P3 - P2);
    double B = Cross(P1 - P2,P4 - P2);
    if(!sgn(A) && !sgn(B)) return 1;
    return 0;
}

//两参数方程求交点，(点,向量,点,向量)
Point Getlinenode(Point P,Point v,Point Q,Point w){
    //线1(P1,P2) 线2(P3,P4) (P1,P1-P2,P3,P3-P4)
    Point u = P - Q;
    double t = Cross(w,u) / Cross(v,w);
    return P + v * t;
}

double get_sector(Point a,Point b){//Oa Ob扇形面积
    double angle = acos(Dot(a,b) / Length(a) / Length(b));//aOb夹角
    if(sgn(Cross(a,b)) < 0) angle *= -1.0;//面积是有向的
    return R * R * angle / 2.0;//面积
}

//求圆与直线的交点
double get_C_L_Ist(Point a,Point b,Point &pa,Point &pb){//ab 与 O 相交于 papb
    Point e = Getlinenode(a,b - a,O,rotate(b - a,pi / 2.0));//e是O垂直于ab的垂足
    double mind = dis(O,e);//垂线的距离
    if(!OnSeg(e,a,b)){
        mind = min(dis(O,a),dis(O,b));//O到线段ab的最短距离
    }
    if(sgn(R - mind) <= 0) return mind;//ab均在圆外 返回最短距离 不比再求交点
    double len = sqrt(R * R - dis(O,e)*dis(O,e));//欲加在e上的向量长度 勾股
    pa = e + norm(a - b) * len;
    pb = e + norm(b - a) * len;
    return mind;
}

double get_circle_triangle_area(Point a,Point b){
    double disa = dis(a,O),disb = dis(b,O);
    if(sgn(R - disa) >= 0 && sgn(R - disb) >= 0) return Cross(a,b)/2.0;//a b在圆内 等于S△
    if(Collinear(a,O,b,O)) return 0.0;//O a b三点共线
    Point pa,pb;//ab与圆的两个交点
    double mind = get_C_L_Ist(a,b,pa,pb);//线段ab与O的最近距离
    if(sgn(R - mind) <= 0) return get_sector(a,b);//扇形aOb
    if(sgn(R - disa) >= 0) return get_sector(pb,b)+Cross(a,pb)/2.0;
    if(sgn(R - disb) >= 0) return get_sector(a,pa)+Cross(pa,b)/2.0;
    //扇形+△ 注意有向!
    return get_sector(a,pa)+get_sector(pb,b)+Cross(pa,pb)/2.0;//俩扇形+△
}

double work(){
    double ans = 0.0;
    for(int i = 0;i < n; i++){
        ans += get_circle_triangle_area(p[i],p[(i + 1) % n]);
    }
    return fabs(ans);
}

int main()
{
    O.x = 0,O.y = 0;
    while(~scanf("%lf %d",&R,&n)){
        for(int i = 0;i < n; i++) scanf("%lf %lf",&p[i].x,&p[i].y);
        printf("%.2f\n",work());
    }
}
```