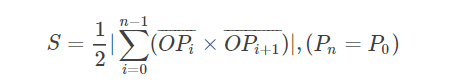
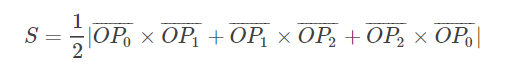
多边形面积的计算一般是将其剖分为三角形，利用海伦公式计算每个三角形的面积，然后将所有三角形的面积加起来。但是，这种方法程序实现起来比较复杂，因为无法预知多边形的形状，需要判断多边形的“凸凹”，从而避免重复计算。本文采用向量的叉乘的方法，不管坐标原点怎样选取，只要顺序输入多边形每个顶点的坐标，按同样的顺序（顺时针或反时针）两两叉乘。这些叉乘的和的绝对值的一半就是该多边形面积。这就是所谓的“鞋带公式”(Shoelace formula)。

计算公式

设O为原点，多边形由点P0,P1,...,Pn−1P0,P1,...,Pn−1连线围成，则面积S为：



举一个最简单的例子，n=3。也就是三角形的面积。



正多边形面积：设正n边形的半径为R，边长为an，中心角为αn，边心距为rn，则αn=360°÷n，an=2Rsin(180°÷n)，rn=Rcos(180°÷n)，R^2=r n^2+(an÷2)^2， 周长pn=n×an，面积Sn=pn×rn÷2。

并查集

#include<iostream>

#include<cstdio>

#include<cstring>

#include<cmath>

#include<algorithm>

using namespace std;

int pre[1010]; //里面全是掌门

int unionsearch(int root)

{

int son, tmp;

son = root;

while (root != pre[root]) //寻找掌门ing……

root = pre[root];

while (son != root) //路径压缩

{

tmp = pre[son];

pre[son] = root;

son = tmp;

}

return root; //掌门驾到~

}

int main()

{

int num, road, total, i, start, end, root1, root2;

while (scanf("%d%d", &num, &road) && num)

{

total = num - 1; //共num-1个门派

for (i = 1; i <= num; ++i) //每条路都是掌门

pre[i] = i;

while (road--)

{

scanf("%d%d", &start, &end); //他俩要结拜

root1 = unionsearch(start);

root2 = unionsearch(end);

if (root1 != root2) //掌门不同？踢馆！~

{

pre[root1] = root2;

total--; //门派少一个，敌人（要建的路）就少一个

}

}

printf("%d\n", total);//天下局势：还剩几个门派

}

return 0;

}