

#B、控枪

题目描述

在枪战类游戏里每一把枪在开枪时都会有后坐力，这一后坐力会使玩家的枪不由自主地随机移动，我们称之为轨迹偏移。

这类游戏里为了模拟真实情况，一把枪有一个弹匣，弹匣中装有 n 发子弹；每射出 n 发子弹，玩家会换上下一个弹匣。

在游戏中，为了程序实现方便，同一把枪的后坐力导致的轨迹偏移往往是固定的，并且这一轨迹偏移序列的周期为弹匣的容量。因此，我们可以记住这把枪的轨迹偏移，然后用一个相反的鼠标移动抵消这一偏移，这样就实现了「控枪」。

于是，要达到枪枪必中，只需要令每一次移动鼠标追踪目标时的位移向量等于敌人的位移向量加上上一枪的轨迹偏移向量的相反向量。

输入格式

首先是一行两个正整数 n, m ，分别表示枪的轨迹偏移周期和需要开的枪数。

之后是 $n - 1$ 行，每行两个整数 a_i, b_i ，分别表示第 i 发子弹射出后的轨迹偏移。由于第 n 次射击后要换弹匣，所以不考虑第 n 次射击产生的轨迹偏移，因此显然每个弹匣的第 1 枪没有轨迹偏移。

之后是 m 行，每行两个整数 x_i, y_i ，分别表示每一次开枪时敌人所在的位置。

输出格式

输出 m 行，每行两个整数 $ansx_i, ansy_i$ ，表示开第 i 枪所需的鼠标位移向量。我们认为开第一枪之前鼠标位于 $(0, 0)$ 。

样例输入1

```
3 3
3 3
-3 -3
1 1
0 0
1 1
```

样例输出1

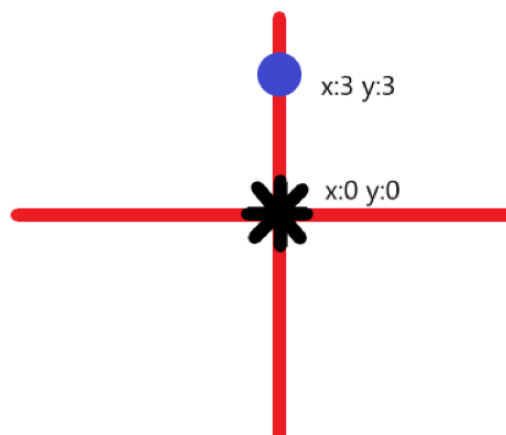
```
1 1
-4 -4
4 4
```

数据范围

对于全部的数据 $2 \leq n \leq 10^5, 2 \leq m \leq 10^5, -10^8 \leq a_i, b_i, x_i, y_i \leq 10^8$ 。

这道题其实很简单，相信很多同学不会做是因为题目没有看懂。这道题的本意就是你需要将子弹打到某一个点上，但是子弹会有一点偏移，所以你需要将子弹往相反的方向偏移，抵消子弹偏移这个效果，最后打到某个点上。

图内敌人x轴应是0

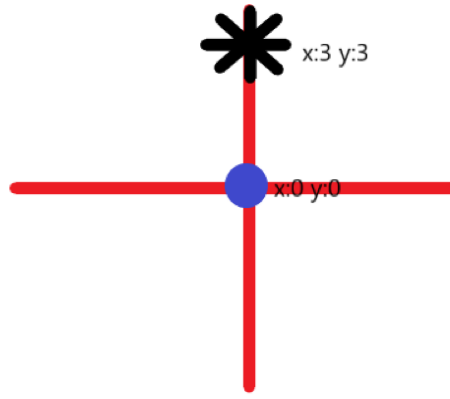


就比如上图，这里黑色的是你的枪口，蓝色的是敌人，红色的是坐标线，而此时是第一枪，子弹没有偏移，所以我们只需要对准敌人发射就可以了。

枪数: 2

偏移程度: x:3 y:3

图内枪口x轴应是0



再看图二，刚刚我们的枪口为了打到第一个点，于是对准了x:0 y:3的位置，而此时第二个点却来到了x:0 y:0之地，这是，子弹还会往x轴偏3，与y轴偏3，那么我们想要打到这个点，就必须要把枪口瞄准x:0 y:0这个点，但是子弹还会偏一点，所以枪口应该加上偏移的相反值，也就是x: -3 y: -3。

当我们了解这两个例子之后，其实这道题就变得很简单了。我们只需用一个循环，输入坐标后存在一个数组里，然后一个敌人来就遍历一次，用一个变量存起来就行了。

```
for(int i=0;i<m;i++,xx++){
    cin>>x>>y;
    cout<<x-ax<<' '<<y-ay<<endl;
    ax=x+ox[xx%n],ay=y+oy[xx%n];
}
```