


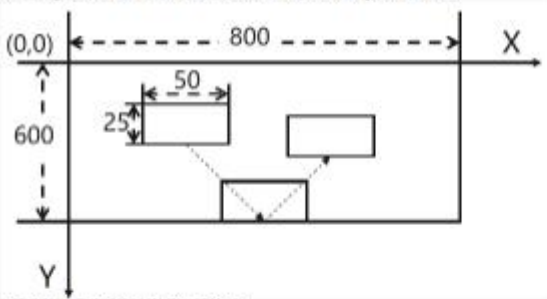
经典屏保

题目描述: DVD机在视频输出时,为了保护电视显像管,在待机状态会显示“屏保动画”,如下图所示。DVD Logo在屏幕内来回运动,碰到边缘会反弹:



请根据如下要求,实现屏保Logo坐标的计算算法。

1. 屏幕是一个800*600像素的矩形,规定屏幕的左上角点坐标原点,沿横边向右方向为X轴,沿竖边向下方向为Y轴;
2. Logo是一个50*25像素的矩形,初始状态下,左上角点坐标记做 (x, y) ,它在X和Y方向上均以1像素/秒的速度开始运动;
3. 遇到屏幕四个边缘后,会发生镜面反弹,即以45°碰撞边缘,再改变方向以45°弹出;
4. 当Logo和四个角碰撞时,两个边缘同时反弹的效果是Logo会原路返回。



请编程实现, t 秒后Logo左上角点的坐标。

输入描述: 输入3个数字,以空格分隔:
 $x\ y\ t$
第一个数字表示Logo左上角点的初始X坐标;
第二个数字表示Logo左上角点的初始Y坐标;
第三个数字表示时间 t ,题目要求即求 t 秒后Logo左上角点的位置。

输出描述: 输出2个数字,以空格分隔:
 $x\ y$
第一个数字表示 t 秒后,Logo左上角点的X坐标
第二个数字表示 t 秒后,Logo左上角点的Y坐标

补充说明: 所有用例均保证:

1. 输入的 x 和 y 坐标会保证整个Logo都在屏幕范围内,Logo不会出画;
2. 所有输入数据都是合法的数值,且不会出现负数;
3. t 的最大值为100000。

示例 1

输入:

0 0 10

输出:

10 10

说明:

输入样例表示 Logo 初始位置在屏幕的左上角点,10s 后,Logo 在 X 和 Y 方向都移动了 10 像素,因此输出 10 10。

示例 2

输入：

500 570 10

输出：

510 570

说明：

输入样例表示初始状态下，Logo 的下边缘再有 5 像素就碰到屏幕下边缘了，5s 后，会与屏幕碰撞，碰撞后，斜向 45° 弹出，又经过 5s 后，Logo 与起始位置相比，水平移动了 10 像素，垂直方向回到了原来的高度。

```
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int x1 = sc.nextInt();
        int y1 = sc.nextInt();
        int m = sc.nextInt();

        int x = x1 + m;
        int y = y1 + m;

        while(x < 0 || x + 50 > 800 || y < 0 || y+25 > 600){
            if(x<0){
                x = -x;
            }
            if(y<0){
                y = -y;
            }
            if(x+50 > 800){
                x = 1500-x;
            }
            if(y +25 > 600){
                y = 1150 -y;
            }
        }
        System.out.println(x+" "+y);
    }
}
```