

AcWing-4443：无限区域

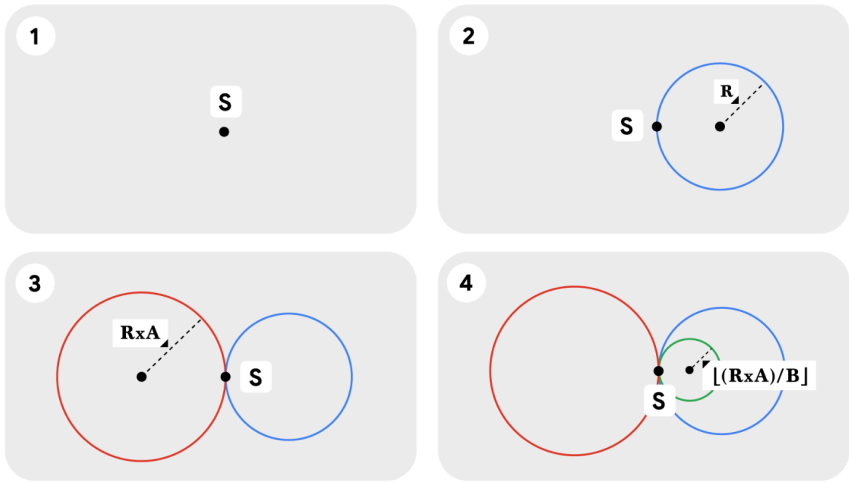
给定一个无限大的二维平面，设点 S 为该平面的中心点。

设经过点 S 的垂直方向的直线为 P ，如果直线 P 是一个圆的切线，且切点恰好为点 S ，那么：

- 如果该圆位于直线 P 的右侧，则称之为右圆。
- 如果该圆位于直线 P 的左侧，则称之为左圆。

现在，给定三个整数 R, A, B ，你需要按照右圆、左圆、右圆、左圆...的顺序不断画圆，具体要求如下：

- 第一个右圆的半径等于 R 。
- 每个左圆的半径等于你画的上一个圆的半径乘以 A 。
- 每个右圆（第一个除外）的半径等于你画的上一个圆的半径除以 B （向下取整）。
- 当你要画的圆的半径等于 0 时，绘画停止。



请你计算，所有画出的圆的面积之和。

保证绘画会在有限数量的步骤后停止。

输入格式

第一行包含整数 T ，表示共有 T 组测试数据。

每组数据占一行，包含三个整数 R, A, B 。

输出格式

每组数据输出一个结果，每个结果占一行。

结果表示为 `Case #x: y`，其中

x 为组别编号（从 1 开始），
 y 为面积和（实数）。

y 在正确答案的 10^{-6} 的绝对或相对误差范围内，则视为正确。

数据范围

$1 \leq T \leq 100$,
 $1 \leq R \leq 10^5$,
 $1 \leq A \leq 500$,
 $2 \times A \leq B \leq 1000$ 。

时/空限制:	1s / 64MB
总通过数:	802
总尝试数:	1673
来源:	Google Kickstart2022 Round B Problem A
算法标签	数学 模拟

输入样例：

```
2
1 3 6
5 2 5
```

输出样例：

```
Case #1: 31.415927
Case #2: 455.530935
```

样例解释

对于 case #1，首先画一个半径为 1 的右圆，然后画一个半径为 $1 \times 3 = 3$ 的左圆，随后停止绘画，因为下一个右圆的半径为 $\lfloor 3/6 \rfloor = 0$ 。

对于 case #2：

1. 第一步画一个半径为 5 的右圆；
2. 第二步画一个半径为 $5 \times 2 = 10$ 的左圆；
3. 第三步画一个半径为 $\lfloor 10/5 \rfloor = 2$ 的右圆；
4. 第四步画一个半径为 $2 \times 2 = 4$ 的左圆；
5. 停止绘画，因为下一个右圆的半径为 $\lfloor 4/5 \rfloor = 0$ 。

简单的模拟题，代码实现上做了一个简单的简化：左右分别判断改为判断一次，在 `r` 处理后即将作为右圆的半径计算时进行判断。

此外注意计算精度，需要记 $\pi = 3.141592654$ 才能达到精度要求。

Code：

```
//
//  main.cpp
//  4443-无限区域
//
//  Created by MacBook Pro on 2023/8/7.
//

#include <iostream>
#include <cmath>
using namespace std;

const double pai = 3.141592654;

int main() {
    int T;
    scanf("%d", &T);
```

```
for(int cases=1;cases<=T;cases++){  
    int r,a,b;  
    scanf("%d%d%d",&r,&a,&b);  
    double ans = .0;  
    while(r!=0){  
        ans += pai*r*r;  
        r = a*r;  
        ans += pai*r*r;  
        r = floor(double(r)/b);  
    }  
    printf("Case #%d: %lf\n",cases,ans);  
}  
}
```

代码提交状态: **Accepted**