定理本身不讲了。直接上结论：

著名结论

设自然数a□b和整数n□a与b互素。考察不定方程：

$$ax+by=n$$

其中x和y为自然数。如果方程有解，称n可以被a□b表示。

记A=ab-a-b□由a与b互素□A必然为奇数。则有结论：

对任意的整数n□n与A-n中有且仅有一个可以被表示。

即：可表示的数与不可表示的数在区间[0,A]对称（关于A的一半对称）。0可被表示□A不可被表示；负数不可被表示，大于A的数可被表示。

证明：

由于a□b互素，因此原方程有整数解。设解为：

$$\begin{cases}x=x_0+bt\\y=y_0-at\end{cases}$$

其中t为整数。取适当的t□使得y位于0到a-1之间。这只需在y0上加上或减去若干个a□即可得到这样的t□

第一步：证明大于A的数都可以被表示。当n大于A时：

$$ax=c-by>ab-a-b-by\geqslant ab-a-b-b(a-1)=-a$$

于是x也是非负整数。

第二步：证明A不可被表示，进而n与A-n不可能都被表示。

反证法。若ax+by=ab-a-b有非负整数解x□y□则：

$$ab=a(x+1)+b(y+1)$$

由于a与b互素，所以a整除y+1□b整除x+1□a不超过y+1□b不超过x+1□于是有：

$$ab=a(x+1)+b(y+1)\geqslant ab+ab=2ab$$

矛盾！第二步证完。

第三步：证明如果n不可被表示，则A-n可被表示。

由上可知，若n不可被表示，由于上述方程中已规定y在0到a-1之间，则x为负。所以：

$$ab-a-b-ax-by=a(-x-1)+b(a-1-y)$$

显然-x-1和a-1-y均非负，于是A-n可被表示。

几何意义

重新观察方程ax+by=n，将它看成一条直线。直线与两坐标轴在第一象限围成三角形。

当n小于ab的时候，这个直线在第一象限，至多只能通过一个整点。

根据上述讨论：当n可以被表示的时候，直线恰好经过一个整点；当n不可以被表示的时候，直线不经过整点（在第一象限）。

这结论也可以理解为：作三角形(0,0)(b,0)(0,a)，随着n从0不断增加，直线向右上方平移，整点会一个一个地通过直线，直到最后才撞上两个整点。

因此，小于等于n的能被表示的非负整数的数量，恰好就是直线ax+by=n（含）与两坐标轴（含）在第一象限围成三角形覆盖的整点个数。

（另一种解释）

考虑模b意义下每个剩余系中最小能被表示的值是多少——大于他们的可以通过增加若干个b得到。

观察原方程，a的若干倍数0, a, · · · , (b − 1)a 在mod b 意义下互不相同。这些数恰好是这些最小值。那么当n<ab时，小于等于n的能被表示的非负整数的数量是：

$$ \sum\limits_{i=0}^{\left[\frac{n}{a}\right]}\left[\frac{n - ia}{b}\right]$$

这是一个非常经典的直线下整点问题，恰好是这条直线：

$$y=-\frac{a}{b}x+\frac{n}{b}$$

即ax+by=n。

（解释完）

使用类欧几里得算法可以在O(log max(a,b)) 的时间内求解。因此我们得到了计算小于等于n的能被表示的非负整数的数量的工具。

~~但是直线下整点（类欧几里得）不会。只能等待大佬来完善了。~~

以下是例题。

题目

双城是一个一切都是双胞胎的地方，除了他们的硬币。他们使用两种硬币，其价值相当于A和B克黄金。那里的人很了解欧几里德算法，所以我们保证A和B的最大公约数是1。他们会告诉你为什么硬币系统是有效的：通过求解线性不定方程Ax+by=C，每一个可能的整数C都会有一个结果。

但当它进入现实时，事情变得更加复杂。事实上，改变-换句话说，负的x或y-是很麻烦的，在双土地上的人根本不喜欢它。所以当有必要改变的时候，他们总是多付一点钱。

一个叫伊尼的骗子，住在一个地方，恨恶两个地方的人。他知道当没有合适的非负x，y作为C的价格时，人们会付出更多的代价，于是决定用这样不方便的价格来骗钱。他买了很多货，把它们送到了双人间土地，并以各种价格定价-当然，所有这些价格都是不方便的人在双重土地。

但是艾尼不是很聪明-也许这就是他现在不得不住在一个地方的原因。他发现很难计算出第K个最小的不公平价格。你能帮他吗？他愿意和你分享他从双面人那里骗取的钱！

第一行包含整数T1≤T≤10-测试用例数。

每个测试用例描述只包含一行三个整数ABK1≤AB≤1071≤K≤1018-两种硬币的值，以及所需的K

保证了gcdAB=1且存在第K个最小不公平价格。

对于每个测试用例，在单独的一行上打印一个表示第K个最小不公平价格的整数。

样例

输入

2

2 3 1

314159 233333 123456789

输出

1

123570404

题解

本题利用了上述结论：给定值，判断是第几个。因此最终就是一个二分查找。二分一下就能解决原问题了。时间复杂度O(T log max(A,B) log min(K,AB))可以通过A,B ≤ 1018, T ≤ 1000 的数据，也即本题原本的数据范围。

考虑到直线下整点是一个掏板子工作，因此放过了暴力。暴力可以这么写：假设A < B我们尝试计算[kB, (k + 1)B) 这一段里有多少可以被表达出来的，那其实就是{(iA mod B) + kB|iA < (k + 1)B}的集合大小，枚举iA 计算确定答案在哪一段后，再在段内确定就好了——而段内有多少能被表达的已经被枚举了，打好标记for 一下就行了。时间复杂度O(T max(A,B))

```
#include<stdio.h>

long long solve(long long n, long long a, long long b, long long m)
{
    if(b==0)
    {
        return n*(a/m);
    }
    if(a>=m)
    {
        return n*(a/m)+solve(n,a%m,b,m);
    }
    if(b>=m)
```

```
    {
        return (n-1)*n/2*(b/m)+solve(n,a,b%m,m);
    }
    return solve((a+b*n)/m,(a+b*n)%m,m,b);
}

long long A,B,K;

void work()
{
    scanf("%lld%lld%lld",&A,&B,&K);
    long long l=1,r;
    r=(double)A*B>3e18?2e18+10:((long long)(2e18)+10<A*B-1?(long long)(2e18)+10:A*B-1);// double(A-1)*(B-1)/2>=K
    while(l<r)
    {
        long long mid=(l+r)/2;
        long long n=mid/A+1,m=B,a=mid-(n-1)*A+B,b=A;
        long long tot=solve(n,a,b,m)-1;
        long long cnt=mid-tot;
        if(cnt>=K)
        {
            r=mid;
        }
        else
        {
            l=mid+1;
        }
    }
    printf("%lld\n",l);
}

int main()
{
    int T;
    scanf("%d",&T);
    while(T--)
    {
        work();
    }
}
```