

考虑 $\text{mod } B = 0 \cdots B - 1$ 的最小能被表示的值是多少——大于他们的可以通过 $+kB$ 得到。假设这样的最小值分别为 m_0, m_1, \dots, m_{B-1} 那么小于等于 K 的能被表示的非负整数的数量是：

$$\sum_{i=0}^{B-1} \lfloor \frac{\max(K - mi + B, 0)}{B} \rfloor$$

考虑如何求 m_i 由反证法可知 $0, A, \dots, (B - 1)A$ 在 $\text{mod } B$ 意义下必然互不相同，而这些数就枚举了 m_i 因此，当 $K < AB$ 时，上式化为：

$$\sum_{i=0}^{\lfloor \frac{K}{A} \rfloor} \lfloor \frac{K - iA}{B} \rfloor$$

这是一个非常经典的直线下整点问题，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 K 的能被表示的非负整数的数量的工具，再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 $A, B \leq 1018, T \leq 1000$ 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 $\{(iA \text{ mod } B) + kB | iA < (k + 1)B\}$ 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();
    }
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team
Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E8%A3%B4%E8%9C%80%E5%AE%9A%E7%90%86%E4%B8%8E%E4%B8%80%E6%AC%A1%E4%B8%8D%E5%AE%9A%E6%96%B9%E7%A8%BB&rev=1589203090>
Last update: 2020/05/11 21:18

