问题描述

　　栋栋正在和同学们玩一个数字游戏。

　　游戏的规则是这样的：栋栋和同学们一共n个人围坐在一圈。栋栋首先说出数字1。接下来，坐在栋栋左手边的同学要说下一个数字2。再下面的一个同学要从上一个同学说的数字往下数两个数说出来，也就是说4。下一个同学要往下数三个数，说7。依次类推。

　　为了使数字不至于太大，栋栋和同学们约定，当在心中数到 k-1 时，下一个数字从0开始数。例如，当k=13时，栋栋和同学们报出的前几个数依次为：

　　1, 2, 4, 7, 11, 3, 9, 3, 11, 7。

　　游戏进行了一会儿，栋栋想知道，到目前为止，他所有说出的数字的总和是多少。

输入格式

　　输入的第一行包含三个整数 n,k,T，其中 n 和 k 的意义如上面所述，T 表示到目前为止栋栋一共说出的数字个数。

输出格式

　　输出一行，包含一个整数，表示栋栋说出所有数的和。

样例输入

3 13 3

样例输出

17

样例说明

　　栋栋说出的数依次为1, 7, 9，和为17。

数据规模和约定

　　1 < n,k,T < 1,000,000；

看到数据范围，肯定不能直接模拟。但是我们知道相邻数字间的增量是一个等差数列，那么就可以很容易的求出下一个数字，进而可以直接计算出下n个数字，对于超过k的数字直接求余即可。即我们知道本次栋栋的报数，可以直接计算出他下一次的报数，这样只需要一层循环就够了。

我们用add表示栋栋每次报数的增量。可以用等差数列求和公式求出来第一次的增量。那么利用高中数学知识可以求出下一个add=add+n\*n。（数列中每一个元素都增加了n）。

注意n虽然是int型的，但n相乘就可能会爆int，要把n转成long long

#include<iostream>

#include<cstdio>

#include<cstring>

#include<algorithm>

#include<set>

#define INF 1e6+10

using namespace std;

typedef long long ll;

int main()

{

//freopen("input.txt","r",stdin);

int n,k,T;

cin>>n>>k>>T;

ll ans,last,add;

ans=1;

last=1;

add=(ll)n\*(n+1)/2;

for(int i=2;i<=T;i++)

{

last=(last+add)%(ll)k;//last是栋栋每次喊到的数字

ans+=last;

add+=(ll)n\*n;

}

cout<<ans<<"\n";

return 0;

}