首先你要会快速幂

```
inline LL ksm(LL x,LL p){LL s=1;for(;p;(p&1)&&(s=s*x%mod),x=x*x%mod,p>>=1);return s;}
```

这样暴力就可以 30 $\texttt{pts}$ 了。

注意到这个和式的结果为（等比数列求和）$\dfrac{a^{k+1}-1}{a-1}$，于是求逆元快速幂乘一下即可，拿到 30 $\texttt{pts}$，结合上面有 60 $\texttt{pts}$。

正解递归。记 $f(n)=1+a+a^2+\cdots+a^n$，我们注意到：

1. 当 $n=2k-1$ 时，$f(n)=1+a+a^2+\cdots+a^{k-1}+a^k+a^{k+1}+a^{k+2}+\cdots+a^{2k-1}=(a^k+1)(1+a+a^2+\cdots+a^{k-1})=(a^k+1)f(k-1)$
2. 当 $n=2k$ 时，$f(n)=f(n-1)+a^n=(a^k+1)f(k-1)+a^n$。

于是 $f(n)$ 就转换为求 $f(\left\lfloor\dfrac{n-1}{2}\right\rfloor)$，递归求解即可，边界条件为 $f(0)=1$。100 $\texttt{pts}$。

code：

```
#include<bits/stdc++.h>
#define LL long long
#define fr(x) freopen(#x".in","r",stdin);freopen(#x".out","w",stdout);
using namespace std;
LL a,k,mod;
inline LL ksm(LL x,LL p){LL s=1;for(;p;(p&1)&&(s=s*x%mod),x=x*x%mod,p>>=1);return s;}
LL sol(LL x)
{
	if(!x) return 1;
	LL t=sol((x-1)>>1)*(ksm(a,(x+1)>>1)+1)%mod;
	if(x&1^1) t=(t+ksm(a,x))%mod;return t;
}
int main()
{
	cin>>a>>k>>mod;
	cout<<sol(k);
	return 0;
}
```