F 狗头豪的数列求和

出题人:爆哥

考察点:数学推导,矩阵快速幂

本题需要求 $1^k+2^k+...+n^k\ (n\leq 10^{18})$

发现 n 可能会很大不能正常的通过遍历来求和,那么我们来思考每一位

定义 $sum(x,k)$ 为前 x 项 k 次方的和

首先可以得出 $sum(x,k)=sum(x-1,k)+x^k\ (x\geq 2)$

发现 $sum(x,k)$ 是用一个递推关系的，那么思考 $(x-1)^k$ 和 x^k 会有什么关系

可以通过高中学过的二项式展开得到 $(x-1)^k=\sum_{i=0}^{k}C_k^ix^i(-1)^{k-i}$

得出 $x^k=(x-1)^k-\sum_{i=0}^{k-1}x^i(-1)^{k-i}$

那么我们发现这显然是一个线性递推关系的

$k<=10$ 那么我们可以维护一个矩阵的信息 分别处理 $1,x^1,x^2,...,x^k,sum(x,k)$ 的递推联系

便可以通过矩阵快速幂来解决这道题

时间复杂度 $O(k^3log(n))$

当然，如果你知道拉格朗日插值法，那么这个题就变的很裸，不过要注意这个题模数不一定会有逆元，就看你通过什么方式来处理了。

```
#include<bits/stdc++.h>
using namespace std;
const int N=15;
using ll=long long;
int k,mod;
long long n;
struct mat{
    int m;
    ll mp[N][N];
    mat(){
        memset(mp,0,sizeof(mp));
    }
    void re(){
        for(int i=0;i<m;i++){
            mp[i][i]=1;
        }
    }
    mat operator *(const mat &w)const{
        mat res;
        res.m=m;
        for(int k=0;k<m;k++){
            for(int i=0;i<m;i++){
                for(int j=0;j<m;j++){
                    res.mp[i][j]+=mp[i][k]*w.mp[k][j]%mod;
```

```
                    res.mp[i][j]%=mod;
                }
            }
        }
        return res;
    }
};

mat ksm(mat a,long long b){
    mat res;
    res.m=a.m;
    res.re();
    while(b){
        if(b&1) res=res*a;
        a=a*a;
        b>>=1;
    }
    return res;
}

ll c[N][N];

void solve(){
    cin>>n>>k>>mod;
    int m=k+2;
    for(int i=0;i<=m;i++){
        for(int j=0;j<=i;j++){
            if(i==j||j==0){
                c[i][j]=1;
            }
            else{
                c[i][j]=c[i-1][j-1]+c[i-1][j];
                if(c[i][j]>=mod) c[i][j]-=mod;
            }
        }
    }
    mat a;
    a.m=m;
    for(int j=m-1;j>=1;j--){
        a.mp[j][j]=1;
        int k1=k-j+1;
        for(int i=j+1,k2=k1-1,f=1;i<m;i++,k2--,f=mod-f){
            for(int p=i;p<m;p++){
                a.mp[p][j]+=f*c[k1][k2]%mod*a.mp[p][i];
                a.mp[p][j]%=mod;
            }
        }
    }
    a.mp[0][0]=1;
    for(int i=1;i<m;i++){
        a.mp[i][0]=a.mp[i][1];
    }
    mat res;
    res.m=m;
    for(int j=0;j<m;j++){
```

```
        res.mp[0][j]=1;
    }
    res=res*ksm(a,n-1);
    cout<<res.mp[0][0]<<'\n';
}

int main(){
    ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);
    int t=1;
    while(t--){
        solve();
    }
    return 0;
}
```