题意：给你一个求和的循环，让你计算最后的结果是多少？

思路：首先，我们得明确在这个循环里面，a[0]……a[n-1]出现的次数是一样多的，

看一个样例：

它有2个数字，3层循环

1、a[0] a[0] a[0]

2、a[0] a[0] a[1]

3、a[0] a[1] a[0]

4、a[0] a[1] a[1]

5、a[1] a[0] a[0]

6、a[1] a[0] a[1]

7、a[1] a[1] a[0]

8、a[1] a[1] a[1]

可以推出每个数的累加次数为(n^k)\*k/n=(n^(k-1))\*k,

所以结果就等于(sum(a[i])\*(n^(k-1))\*k)%mod

n^(k - 1)要用快数幂来求解

因为溢出导致一直WA…….

要知道k和sum都可能是很大的数

所以不能写成这样sum=sum\*k\*pow\_mod(n,k-1,mod)%mod;

要分开

ll ans=pow\_mod(n,k-1,mod)%mod;

ans=(ans\*k)%mod;

ans=(ans\*sum)%mod;

#include<stdio.h>

#include<math.h>

#include<string.h>

#include<stack>

#include<queue>

#include<vector>

#include<iostream>

#include<algorithm>

typedef long long ll;

using namespace std;

long long pow\_mod(long long a,long long n,long long m)//a^n mod m

{

long long res=1;

while(n>0)

{

if(n&1==1)

res=res\*a%m;

a=a\*a%m;

n>>=1;

}

return res;

}

//记得在最后输出结果的时候再模m一次

int main()

{

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

int T;

cin>>T;

for(int cas=1;cas<=T;cas++)

{

ll sum(0);

int n,mod;

ll k,x;

cin>>n>>k>>mod;

for(int i=0;i<n;i++)

{

cin>>x;

sum+=x;

}

ll ans=pow\_mod(n,k-1,mod)%mod;

ans=(ans\*k)%mod;

ans=(ans\*sum)%mod;

// sum=sum\*k\*pow\_mod(n,k-1,mod)%mod;

printf("Case %d: %lld\n",cas,ans);

}

return 0;

}