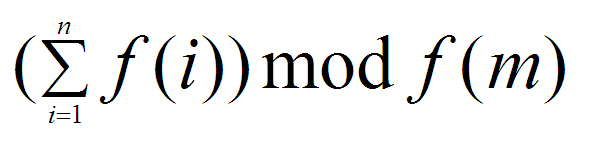
问题描述

　　斐波那契数列大家都非常熟悉。它的定义是：  
  
　　f(x) = 1 .... (x=1,2)  
　　f(x) = f(x-1) + f(x-2) .... (x>2)  
  
　　对于给定的整数 n 和 m，我们希望求出：  
　　f(1) + f(2) + ... + f(n) 的值。但这个值可能非常大，所以我们把它对 f(m) 取模。  
　　公式如下  
  
  
　　但这个数字依然很大，所以需要再对 p 求模。

输入格式

　　输入为一行用空格分开的整数 n m p (0 < n, m, p < 10^18)

输出格式

　　输出为1个整数，表示答案

样例输入

2 3 5

样例输出

0

样例输入

15 11 29

样例输出

25

Thinking:

设f(i)为斐波那契数列第i项,s(i)=f(1)+f(2)+...+f(i),

  i  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15

f(i) 1  1  2  3  5  8 13 21 34  55  89  144 233 377 610

s(i) 1  2  4  7 12 20 33 54 88 143 232  376 609 ... ...

观察发现:

s(i)=f(i+2)-1 (i>0)

又:

s(i)%f(m)=(f(i+2)-1)%f(m)

 =f(i+2)%f(m)+f(m)-1

 =(f(i)+f(i+1))%f(m)+f(m)-1;

得到公式之后就好办多了,但是有一个问题,n,m,p的范围较大,必须用大整数运算才能得满分,我在这里没用大整数,所以,只能过40%的数据,改天再改改~

因为n范围很大，用数组开不下，而最后结果只要fn

所以不用数组存，开两个变量迭代

#include <iostream>

#include <cstdio>

#include<map>

#include<algorithm>

#include<math.h>

#include<string.h>

#include <cmath>

#define ll long long

using namespace std;

int main()

{

ll n,m,p,c,s,fm;

cin>>n>>m>>p;

ll a=1,b=1;

for(int i=3;i<=m;i++)

{

c=a+b;//f[i]=f[i-2]+f[i-1];

a=b;b=c;

}

fm=c;

a=b=1;

for(int i=3;i<=n+2;i++)

{

//f[i]=(f[i-2]+f[i-1])%fm;

c=(a+b)%fm;

a=b;b=c;

}

s=(b%fm+fm-1)%fm;//一定注意这一步，通过最后的公式求出s之后还要再模fm一下！！

s%=p;

cout<<s<<"\n";

return 0;

}