**UVA - 11582 Colossal Fibonacci Numbers!**

|  |
| --- |
| /\*  题意：  输入两个非负整数a、b和正整数n（0<=a,b<2^64，1<=n<=1000）,你的任务是计算f(a^b)除以n的余数，其中f(0)=f(1)=1;  且对于所有非负整数i，f(i+2)=f(i+1)+f(i)。  思路：  假设n=3，序列F[i]的前10项为1,1,2,0,2,2,1,0,1,1  猜想这是有规律的，根据递推公式，第十一项会等于第3项，第12项会等于第4项  因为余数最多n种，所以最多n^2项就会重复出现。设周期为M，则只需计算出f[0]~f[n^2]求的周期M，然后用快速幂算出a^b，，，计算F[a^b]等于那一项即可  \*/  #include<iostream>  using namespace std;  typedef unsigned long long ULL;  const int maxn = 1000+10;  ULL F[maxn\*maxn];  ULL quick\_pow\_mod(ULL a,ULL n,ULL m){  if (n == 0)return 1;  ULL res = 1;  while (n > 0){  if (n & 1)  res = res\*a%m;  a = (a%m)\*(a%m)%m;  n >>= 1;  }  return res;  }  int main(){  ULL a, b, n,t,T,i;  cin >> t;  while (t--){  cin >> a >> b >> n;  F[0] = 0; F[1] = 1;  if (n == 1){ printf("0\n"); continue; }  for (i = 2;i<n\*n; i++){  F[i] = (F[i - 1] + F[i - 2]) % n;  if (F[i] == 1 && F[i - 1] == 0)break; //找到周期  }  T = i - 1; //周期  ULL id = quick\_pow\_mod(a, b, T);  cout << F[id] << endl;  }  return 0;  } |