**UVA - 12169 Disgruntled Judge**

|  |
| --- |
| 题意：  找3个整数x1,a,b；然后按照递推公式xi=(axi−1+b)mod10001xi=(axi−1+b)mod10001计算出一个长度为2T的序列，其中T为测试数据的组数。然后把奇数项当做输入，偶数项当做输出。  /\*UVa12169 - Disgruntled Judge  --a,b都在10000以内，这样可以枚举a的值，然后根据x1和x3计算出b的值，再遍历数组X判断他们是否合法  --很容易得到x3=(a\*a\*x1+(a+1)\*b)%10001;变形一下可以得到：(a+1)\*b+10001\*k=x3-a\*a\*x1;直接利用扩展欧几里得算法  求解b在[0,10000]的唯一解。  \*/  #define \_CRT\_SECURE\_NO\_DEPRECATE  #include<bits/stdc++.h>  using namespace std;  const int maxn =100 + 10;  const int mod = 10001;  typedef long long LL;  LL F[maxn \* 2];  //扩展欧几里德算法，求解ax+by=gcd(a,b)系数  void extendEuclid(LL a, LL b, LL &d, LL &x, LL&y){  if (!b){ d = a; x = 1; y = 0; }  else{ extendEuclid(b, a%b, d, y, x); y -= x\*(a / b); }  }  //求ax+by=n的一个解(x0,y0)  //所有解为(x0+t\*b,y0-t\*a),t是任意整数  LL solveLiner(LL a, LL b, LL n, LL&x, LL&y){  LL d;  extendEuclid(a, b, d, x, y);  if (n%d)return 0; //无解  x = (n / d)\*x; y = (n / d)\*y;  return 1;  }  inline LL answer(LL a, LL b, LL x){ return (a\*x + b) % mod; }  int main(){  int T,i;  LL a, b,k;  while (~scanf("%d", &T)){  for (i = 1; i <= 2 \* T - 1; i += 2)  scanf("%d", &F[i]);  for (a = 0; a < mod; a++){  solveLiner(a + 1, mod, F[3] - a\*a\*F[1], b, k);  b = (b%mod + mod) % mod; //计算出b在[0,10000]的解  for (i = 2; i <= 2 \* T; i++){  LL temp = answer(a, b, F[i - 1]);  if (i & 1 && temp != F[i])break;  F[i] = temp;  }  if (i == 2 \* T + 1)break;  }  for (i = 2; i <= 2 \* T; i += 2)  printf("%lld\n", F[i]);  }  return 0;  } |