

解题思路

奇怪的离线算法

在你回答正确的前提, 当解密出的 a, b, c, d 均为 0 时一定是最后一个问题, 根据加密式子可以推出上一个问题的答案为 $-a$, 代入原式可以依次推出所有答案

设第 i 次读入的四个数为 a_i, b_i, c_i, d_i , 倒推知第 i 个问题的答案为 ans_i , 设 $j = ans_i$, 带入到式子中稍做整理:

$$tmp = a_i * (x_j \oplus j)^2 + b_i * (x_j \oplus j) * x_j + c_i * x_j^2 + d_i$$

$$tmp + lastAns * ((x_j \oplus j)^2 + (x_j \oplus j) * x_j + x_j^2 + 1) = 0$$

其中 $lastAns$ 即为 ans_{i-1}

AC代码

```
/*
Author: 赵淞铭
Result: AC Submission_id: 4590545
Created at: Fri Sep 09 2022 08:56:08 GMT+0800 (China Standard Time)
Problem_id: 5927 Time: 1023 Memory: 15864
*/

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

const int N = 1e5 + 7;

int n, t; LL x[N], a[N*4], b[N*4], c[N*4], d[N*4], ans[N*4];

int main() {
    cin >> n;
    for (int i = 1; i <= n; i++) cin >> x[i];
    while (cin >> a[t+1] >> b[t+1] >> c[t+1] >> d[t+1]) t++;
    ans[t-1] = -a[t];
    for (int i = t - 1; i > 1; i--) {
        LL j = ans[i];
        LL tmp = a[i] * (x[j] ^ j) * (x[j] ^ j) + b[i] * (x[j] ^ j) * x[j] + c[i]
* x[j] * x[j] + d[i];
        ans[i-1] = -tmp / ((x[j] ^ j) * (x[j] ^ j) + (x[j] ^ j) * x[j] + x[j] *
x[j] + 1);
    }
    for (int i = 1; i < t; i++)
        cout << ans[i] << '\n';
    return 0;
}
```