D - Marking (atcoder.jp)

简介

给定 D　和　 N。初始 x = 0;标记0已经经过

不断进行下列的操作:

1. x - > (x + D) % N;
2. while(mark[D]) x++;
3. mark x;

查询第k个被标记的数字。

solve

Editorial - Toyota Programming Contest 2023 Spring Qual B（AtCoder Beginner Contest 290）

首先观察到（或者记录总结一个结论。）

定义 A ， B 为正整数。 g = gcd (A ， B);

$A = ag, B = bg$

$\{0, B\%A, 2B\%A....(a-1)\%A\} = \{0, g, 2g, 3g....(a-1)g\}$

举例子

$$\begin{aligned} A = 12 = ag, B = 9 = bg \\ a = 4, b = 3, g = 3 \end{aligned} \tag{1}$$

对于第一个运算数列如下:

0 ， 9 ， 6 ， 3

对于第二个运算数列如下:

0 ， 3　6　9

推论

当gcd = 1时。0　....　.....　（N - 1）D逐渐会将$[0, N)$每一个数字填满。

当gcd > 1时。

定义$f(i) = i * D\%A$

1. 当$0 <= i < a$ ， $f(i)$两两不同。
2. 当$i = a$　$a * A = a * g * b = Ab$. $f(i) = 0$.该点已经被标记 x++

   x = 1 ;
3. 当$a <= i < a + g$

   显然相当于上一次地集合往前面挪一遍。
4. 将a次分为一轮。每一次等距的一些数字往后不断的挪一步。直到所有都填满。

综上

$f(k) = (k-1)/a + (k-1)D\%N$

生长思考

1. 总结别然总结规律的角度:

k - 1将数字投到 0 .... a-1 这样的划分中。是一个技巧。当计数从1开始， 同时涉及 mod运算，将1 ....k块化分成若干连续len块， 并且要求对于整一块中，它们的mod 值 连续上升。此时考虑投射到 0 ..... k的值域。

2. 第二部分， 不用取mod。因为余的部分，自然取模。结果为0.

code

```cpp
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;

const int oo = 0x0fffffff;
const int N = 1E6 + 10;

ll gcd(ll x , ll y) {
    return (x == 0) ? y : gcd(y % x , x);
}

void work(int testNo)
{
    ll n , d , k;
    cin >> n >> d >> k;
    ll  g = gcd(n , d);
    ll  a = n / g;
    k --;
    cout << k / a + k*d % n << '\n';
}


int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);

    int t; cin >> t;
    for (int i = 1; i <= t; i++)work(i);
}

/* stuff you should look for
* int overflow, array bounds
* special cases (n=1?)
* do smth instead of nothing and stay organized
* WRITE STUFF DOWN
* DON'T GET STUCK ON ONE APPROACH
*/

```