题意，有1~n个塔 开始只有两座塔a，b，两人轮流建塔，只能建现有的任意两座塔x+y号或x-y号，谁先不能建谁输

简单博弈

首先最优的肯定是能建哪个建哪个，因为能建早晚会被建，先手建了就可以让对手再选择，其次，规律，能建的最小号的塔肯定是gcd(a,b);

解题思路

d = gcd(a,b), k\*d的位置都是可以选的，判断试下k的个数是奇数还是偶数即可。

思路：本来还以为是博弈，但是后来把数字的加减都模拟一遍之后发现，无论怎样，最后得到的这个集合里面的数列，其实是一个等差数列，所以这就简单了，由一开得到的 a 和 b 来相减，然后不断地取最小的两个数相减，然后等到等差数列的差，这个差同时也是等差数列一开始的那个数字，然后拿 n 除以这个差就是在 n 的范围内可以得到的数字的个数了，然后因为分先手和后手，所以最后只要判断一下个数的奇偶数就可以得到答案了。

【思路】(n/gcd(a,b)&1)即可，为什么这样说呢？思考一下：a + b或a - b或b -a这样的数次操作之后，无论怎样，最后得到的这个集合里面的数列，其实是一个等差数列，想到这这就简单了，由最开始的 a 和 b 来相减，不断地取最小的两个数相减，然后等到等差数列的差，这个差同时也是等差数列的首项（即为gcd(a,b)），然后拿 n 除以这个差就是在 n 的范围内可以得到的数字的个数了，然后因为分先手和后手，所以最后只要判断一下个数的奇偶数就可以得到答案了。

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

int main()

{

int T,n,a,b,gcd;

scanf("%d",&T);

for(int cas=1;cas<=T;cas++)

{

scanf("%d%d%d",&n,&a,&b);

gcd=\_\_gcd(a,b);

int ans=n/gcd;

if(ans&1)

printf("Case #%d: Yuwgna\n",cas);

else printf("Case #%d: Iaka\n",cas);

}

return 0;

}