题意：给你n个硬币排成一圈，编号1-n，只能翻转连续的1~k个的硬币。翻最后一枚硬币者赢。

又是一道博大精深的博弈题目。

已经说了 巴什博弈，威佐夫博弈，尼姆博弈，SG函数，找规律的博弈，这次和找规律有点类似——对称博弈。

一般都是圆啊方啊什么的。

属于环形博弈，我觉得用一句古语来描述很恰当，叫做 分而治之 。

思想就是将剩余的硬币环一分为二，总是不让你一次取完，你只能二取一，最后的硬币由我来取，从而我获胜。

如果先手能一次把硬币拿完，即 k >= n ，那么先手胜

如果每次只能拿一个硬币， 即 k = 1 ，那么如果有奇数个硬币先手胜，如果有偶数个硬币后手胜。

剩下的情况就是先手一次拿不完，而且每次可以拿一个或者拿两个硬币。

剩下的硬币会变成一条链，如果后手能拿完最好，不能拿完的话就拿一个或两个硬币使这条链变成长度相等的两条。

这一定是能做到的，

因为如果这条链有奇数个硬币，那么拿走中间的那个；

如果有偶数个硬币，拿走中间的两个。

变成相等的两条链之后，先手在哪条链拿走多少个硬币，后手就在另外一条链中拿走同样的硬币，直到拿完所有的硬币。

因此后手必胜。

#include<stdio.h>

#include<string.h>

#include<iostream>

#include<set>

#include<algorithm>

#include<map>;

typedef long long ll;

using namespace std;

int main()

{

// freopen("input.txt","r",stdin);

ll n,k;

int T;

cin>>T;

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

{

cin>>n>>k;

printf("Case %d: ",o);

if(k==1)

{

if(n&1)

//注意这里是n不是k

puts("first");

else puts("second");

}

else

{

if(k>=n)

puts("first");

else puts("second");

}

}

return 0;

}