**威佐夫博弈：**有两堆各若干个物品，两个人轮流从某一堆或同时从两堆中取同样多的物品，规定每次至少取一个，多者不限，最后取光者得胜。

这种情况下是颇为复杂的。我们用（ak，bk）（ak ≤ bk ,k=0，1，2，...,n)表示两堆物品的数量并称其为局势，如果甲面对（0，0），那么甲已经输了，这种局势我们称为奇异局势。前几个奇异局势是：（0，0）、（1，2）、（3，5）、（4，7）、（6，10）、（8，13）、（9，15）、（11，18）、（12，20）。

可以看出,a0=b0=0,ak是未在前面出现过的最小自然数,而 bk= ak + k。

那么任给一个局势（a，b），怎样判断它是不是奇异局势呢？我们有如下公式：

    ak =[k（1+√5）/2]，bk= ak + k  （k=0，1，2，…,n 方括号表示取整函数)

## 奇异局势有如下性质：

1。任何自然数都包含在一个且仅有一个奇异局势中。

由于ak是未在前面出现过的最小自然数，所以有a[k] > a[k-1] ，而 bk= a[k] + k > a[k-1] + k > a[k-1] + k - 1 = b[k-1] > a[k-1] 。所以性质1成立。

2。任意操作都可将奇异局势变为非奇异局势。

事实上，若只改变奇异局势（ak，bk）的某一个分量，那么另一个分量不可能在其他奇异局势中，所以必然是非奇异局势。如果使（ak，bk）的两个分量同时减少，则由于其差不变，且不可能是其他奇异局势的差，因此也是非奇异局势。

3。采用适当的方法，可以将非奇异局势变为奇异局势。

假设面对的局势是（a,b），若 b = a，则同时从两堆中取走 a 个物体，就变为了奇异局势（0，0）；如果a = ak ，b > bk 那么，取走b - bk个物体，即变为奇异局势；如果 a = ak ， b < bk 则同时从两堆中拿走a-a[b-a] 个物体变为奇异局势（ a[b-a], b-a+a[b-a]）；如果a > ak ，b= ak + k 则从第一堆中拿走多余的数量a - ak 即可；如果a < ak ，b= ak + k,分两种情况，第一种，a=aj （j < k）从第二堆里面拿走 b - bj 即可；第二种，a=bj （j < k）从第二堆里面拿走 b - aj 即可。

# 1527 取石子游戏

**Problem Description**

有两堆石子，数量任意，可以不同。游戏开始由两个人轮流取石子。游戏规定，每次有两种不同的取法，一是可以在任意的一堆中取走任意多的石子；二是可以在两堆中同时取走相同数量的石子。最后把石子全部取完者为胜者。现在给出初始的两堆石子的数目，如果轮到你先取，假设双方都采取最好的策略，问最后你是胜者还是败者。

**Input**

输入包含若干行，表示若干种石子的初始情况，其中每一行包含两个非负整数a和b，表示两堆石子的数目，a和b都不大于1,000,000,000。

**Output**

输出对应也有若干行，每行包含一个数字1或0，如果最后你是胜者，则为1，反之，则为0。

**Sample Input**

2 1

8 4

4 7

**Sample Output**

0

1

0

//#include <cstdio>

//#include <cmath>

//#include <algorithm>

//#include <iostream>

//using namespace std;

//int main()

//{

// int ak,bk,tmp,k;

// double t=(sqrt(5)-1.0)/2.0;

// while(scanf("%d%d",&ak,&bk)!=EOF)

// {

// if(ak>bk)

// {

// tmp=ak;

// ak=bk;

// bk=tmp;

// }

// k = ak\*t;

// if(ak != (int)(k\*(1+t)))

// k++;

// if(ak+k!=bk)

// printf("1\n");

// else

// printf("0\n");

// }

// return 0;

//}

#include <cstdio>

#include <cmath>

#include <algorithm>

#include <iostream>

using namespace std;

int main()

{

int ak,bk,tmp,k;

double t=(sqrt(5)+1.0)/2.0;

while(scanf("%d%d",&ak,&bk)!=EOF)

{

if(ak>bk)

{

tmp=ak;

ak=bk;

bk=tmp;

}

k = bk-ak;

if(bk != (int)(k\*t+k))

printf("1\n");

else

printf("0\n");

}

return 0;

}