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

所谓威佐夫博弈，是ACM题中常见的组合游戏中的一种，大致上是这样的：  
有两堆石子，不妨先认为一堆有 10，另一堆有 15 个，双方轮流取走一些石子，合法的取法有如下两种：  
1、在一堆石子中取走任意多颗；  
2、在两堆石子中取走相同多的任意颗；  
约定取走最后一颗石子的人为赢家，求必胜策略。

两堆石头地位是一样的，我们用余下的石子数(a,b)来表示状态，并画在平面直角坐标系上。

和前面类似，(0,0)肯定是 P 态，又叫必败态。(0,k),(k,0),(k,k)系列的节点肯定不是 P 态，而是必胜态，你面对这样的局面一定会胜，只要按照规则取一次就可以了。再看 y = x 上方未被划去的格点，(1,2)是 P 态。k > 2 时，(1,k)不是 P 态，比如你要是面对(1,3)的局面，你是有可能赢的。同理，(k,2)，(1 + k, 2 + k)也不是 P 态，划去这些点以及它们的对称点，然后再找出 y = x 上方剩余的点，你会发现(3,5)是一个 P 态，如此下去，如果我们只找出 a ≤ b 的 P 态，则它们是(0,0)，(1,2)，(3,5)，(4,7)，(6,10)……它们有什么规律吗？

忽略(0,0)，很快会发现对于第 i 个 P 态的 a，a = i \* (sqrt(5) + 1)/2 然后取整；而 b = a + i。居然和黄金分割点扯上了关系。  
前几个必败点如下：(0,0)，(1,2)，(3,5)，(4,7)，(6,10)，(8,13)……可以发现，对于第k个必败点(m(k),n(k))来说，m(k)是前面没有出现过的最小自然数，n(k)=m(k)+k。  
判断一个点是不是必败点的公式与黄金分割有关（我无法给出严格的数学证明，谁能给出严格的数学证明记得告诉我），为：  
m(k) = k \* (1 + sqrt(5))/2  
n(k) = m(k) + k；

#include <stdio.h>

#include<algorithm>

#include<iostream>

#include<stdlib.h>

#include<cmath>

typedef long long ll;

using namespace std;

void swap(int &a,int &b)

{

a^=b;

b^=a;

a^=b;

}

int main()

{

ll a,b;

while(cin>>a>>b)

{

if(a>b)swap(a,b);

ll k=b-a;

ll m=k\*(1+sqrt(5))/2;

if(m==a)//这是必败点

puts("0");

else

puts("1");

}

return 0;

}