

数字加减游戏

知识点 [广搜](#) 

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

小明在玩一个数字加减游戏，只使用加法或者减法，将一个数字 s 变成数字 t 。
每个回合，小明可以用当前的数字加上或减去一个数字。
现在有两种数字可以用来加减，分别为 $a, b (a \neq b)$ ，其中 b 没有使用次数限制。
请问小明最少可以用多少次 a ，才能将数字 s 变成数字 t 。
题目保证数字 s 一定能变成数字 t 。

输入描述：

输入的唯一一行包含四个正整数 $s, t, a, b (1 \leq s, t, a, b \leq 105)$ ，并且 $a \neq b$ 。

输出描述：

输出的唯一一行包含一个整数，表示最少需要使用多少次 a 才能将数字 s 变成数字 t 。

示例1

输入：

1 10 5 2

输出：

1

说明：

初始值1加一次a变成6，然后加两次b变为10，因此a的使用次数为1次。

示例2

输入：

11 33 4 10

输出：

2

说明：

11减两次a变成3，然后加三次b变为33，因此a的使用次数为2次。

解题思路：

这道题主要的就是需要换个思路：

题目说可以加也能减，如果+a和-a同时存在，则正负相抵，啥都没做但已经使用了2次，所以可得-a和+a只能有一个存在，则存在下面三种情况：

$$a、s - a*i + b*j = t \rightarrow t - (s - a*i) = b*j \rightarrow (t - (s - a*i)) \% b = 0$$

$$b、s + a*i + b*j = t \rightarrow t - (s + a*i) = b*j \rightarrow (t - (s + a*i)) \% b = 0$$

$$c、s + a*i - b*j = t \rightarrow t - (s + a*i) = -b*j \rightarrow (t - (s + a*i)) \% b = 0$$

这样一来就很清楚了。

t 减去 s对a的递增或者递减 整除 b，递增或者递减的次数则是a的使用次数。递增递减同时进行，先整除b的则是最小使用次数。

```

public class Main{

//    public static int t,a,b;
//    public static int min = Integer.MAX_VALUE;

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int s = sc.nextInt();
        int t = sc.nextInt();
        int a = sc.nextInt();
        int b = sc.nextInt();

//        int min = Integer.MAX_VALUE;
//
//        for(int i=0;i<100000/a; i++){
//            if((t - (s + a*i))%b == 0){
//                min = Math.min( min, i);
//                break;
//            }
//        }
//
//        for(int i=0;i<100000/a; i++){
//            if((t - (s - a*i))%b == 0){
//                min = Math.min( min, i);
//                break;
//            }
//        }

        int res = 0;
        int add = s;
        int sub = s;
        while (true){
            if((t - add)%b == 0){
                break;
            }
            if((t - sub)%b == 0){
                break;
            }
            add += a;
            sub -= a;
            res++;
        }
    }
}

```

```

        //handle( s, 0 , "");

        System.out.println(res);
    }

    //    public static void handle(int s, int n, String operation){
    //
    //        if((t - s) % b == 0){
    //            min = Math.min( min, n);
    //        }else {
    //            if(!operation.equals("add")){
    //                handle(s - a , n+1, "sub");
    //            }
    //            if(!operation.equals("sub")){
    //                handle(s + a , n+1, "add");
    //            }
    //        }
    //    }
    // }

}

```