**解题思路**

好一道数学题，先请出一个公式：

**a + b == a ^ b + (a & b) \* 2**

粗略证明：   
对 a，b 的每一位 ai，bi（1 <= i <= n），将 a + b 分为如下几部分的和：   
1、ai != bi 的部分，和为 a ^ b；   
2、ai == bi && ai == 1 的部分，和为 (a & b) \* 2；   
3、ai == bi && ai == 0 的部分，和为 0。

此题便是给你 a + b 的值 s ，和 a ^ b 的值 x，求满足 s 和 x 的 a、b 对的个数。

由上述公式，得 (s - x) / 2 == a & b 。可知，当 s < x || (s - x)%2 != 0 时直接判无解。

接下来就是玩智商的时候了，因为按位异或是无进位的二进制加法，相加时进位，则亦或后该位为 0。

反之，若亦或后该位为 1，则相加时不进位，有 ai == 1 && bi == 0 || ai == 0 && bi == 1 两种可能；若亦或后该位为 0，仅有 ai == bi 一种可能。

**所以，若 x 的二进制中有 n 位为 1，则 a、b 对的个数为 2n。**

此外，此题还有三个会 WA 的点：   
1、因为 a、b 为正整数，所以当 a&b 为 0 时，需减去 a==0 || b==0 的两种清况；   
2、要用 1ll << one\_bits 表示 long long 类型的 res，而非 1 << one\_bits；   
3、还是由按位异或无进位的性质，可知当 (a&b) & (a^b) != 0 时，也直接判无解。

#include<bits/stdc++.h>

using namespace std;

#define ll long long

int main()

{

ll s,x,xx;

cin>>s>>x;

xx=x;

ll andd=(s-x)>>1;

if(s<x || (s-x)%2!=0 || (x&andd)!=0)

return puts("0"),0;

int bit=0;

while(x)

{

if(x&1)

++bit;

x>>=1;

}

ll ans=1ll<<bit;

if(s==xx)

ans-=2;

cout<<ans<<endl;

return 0;

}