B - Flip Bits

You are given two numbers A and B. Write a program to count the number of bits to be flipped to change the number A to the number B. Flipping a bit of a number means changing a bit from 1 to 0 or vice versa.

**Input Format**

First line of input contains T - number of test cases. Each of the next T lines contains 2 integer A and B, separated by space.

**Constraints**

1 <= T <= 100000  
0 <= N <= 109

**Output Format**

For each test case, print the number of bit flips required to convert A to B, separated by new line.

**Sample Input 0**

4

20 10

16 8

1 153

549 24

**Sample Output 0**

4

2

3

6

**Explanation 0**

Self Explanatory

#include <iostream>

using namespace *std*;

bool isKthBitSetV1(long long n, int k) {

return (n & (1LL << k)) != 0; // 'true' if set.

}

int contSetBits(long long n)

{

int count = 0;

while (n)

{

if (isKthBitSetV1(n, 0))

count++;

n = n >> 1;

}

return count;

}

int main(void) {

*ios\_base*::*sync\_with\_stdio*(false);

*cout*.*tie*(nullptr);

*cin*.*tie*(nullptr);

int t; *cin* >> t;

while (t--)

{

long long a, b; *cin* >> a>>b;

long long num = a ^ b;

*cout* << contSetBits(num) << "\n";

}

return 0;

}