**Bit Difference**

Show Topic Tags   

[Amazon](http://practice.geeksforgeeks.org/company/Amazon/)[Samsung](http://practice.geeksforgeeks.org/company/Samsung/)

You are given two numbers A and B. Write a program to count number of bits needed to be flipped to convert A to B.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is a and b.  
  
**Output:**

Print the number of bits needed to be flipped.  
  
**Constraints:**

1 ≤ T ≤ 100  
1 ≤ a,b ≤ 1000  
  
**Example:**

Example:  
Input  
1  
10 20

Output  
4

**Explanation:**

A  = 1001001  
B  = 0010101  
No of bits need to flipped = set bit count i.e. 4

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/bit-difference/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication241;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashSet;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication241 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

String[] input = br.readLine().trim().split(" ");

int a = Integer.parseInt(input[0]);

int b = Integer.parseInt(input[1]);

String bin\_a = Integer.toString(a,2); // decimal to binary

String bin\_b = Integer.toString(b,2);

for(int i =0; bin\_a.length() <32; i++) {

bin\_a = '0' + bin\_a;

}

for(int i =0; bin\_b.length() < 32; i++ ) {

bin\_b = '0' + bin\_b;

}

int cont =0;

for(int i =0; i<bin\_a.length(); i++) {

if(bin\_a.charAt(i) != bin\_b.charAt(i)) {

cont++;

}

}

System.out.println(cont);

//System.out.println(bin\_a + "\n" + bin\_b );

}

}

}

---------otra solución usando XOR-------------

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class GFG {

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

String[] input = br.readLine().trim().split(" ");

int a = Integer.parseInt(input[0]);

int b = Integer.parseInt(input[1]);

int xor = a ^b; //como un bit es distinto del otro entonces el xor es igual a uno, hago un xor de los 2 numeros, y después cuento la cantidad de unos en su representación binaria.

//1 xor 0 = 1

//0 xor 1 = 1

Pero:

1 xor 1 = 0

Y

0 xor 0 = 0

int cont =0;

while(xor > 0) {

if(xor%2 ==1){

cont++;

}

xor/=2;

}

System.out.println(cont);

}

}

}