**Rightmost different bit**

Show Topic Tags   

Given two numbers **M** and **N**. Write a program to find the position of rightmost different bit in binary representation of numbers.  
  
**Input:**  
First line of input contains a single integer T which denotes the number of test cases. T test cases follows. First line of each test case contains two space separated integers M and N.  
  
**Output:**  
For each test case print the position of rightmost different bit in binary representation of numbers. If both M and N are same then print -1 in this case.  
  
**Constraints:**  
1<=T<=100  
1<=M<=1000  
1<=N<=1000  
  
**Example:**  
**Input:**  
2  
11 9  
52 4  
**Output:**  
2  
5

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

Contributor: Harsh Agarwal

New to competitive programming? Please see [How to Begin?](http://www.geeksforgeeks.org/how-to-begin-with-competitive-programming/)and [How to Pick a Category?](http://practice.geeksforgeeks.org/pickACategory.php)

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

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.math.BigInteger;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.HashMap;

import java.util.HashSet;

import java.util.LinkedHashSet;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication250 {

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

int n = Integer.parseInt(nm[0]);

int m = Integer.parseInt(nm[1]);

String bin\_n = Integer.toBinaryString(n);

String bin\_m = Integer.toBinaryString(m);

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

bin\_n = '0' + bin\_n;

}

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

bin\_m = '0' + bin\_m;

}

// System.out.println(bin\_n);

// System.out.println(bin\_m);

int ans =0;

for(int i = bin\_n.length()-1; i>=0; i--) {

if(bin\_n.charAt(i) != bin\_m.charAt(i)) {

ans = 32 - i;

break;

}

}

System.out.println(ans);

}

/\*

int n = 11,m= 9;

String bin\_n = Integer.toBinaryString(n);

String bin\_m = Integer.toBinaryString(m);

System.out.println(bin\_n);

System.out.println(bin\_m);

\*/

}

}