**Toggle bits given range**

Show Topic Tags   

Given a non-negative number N and two values L and R. The problem is to toggle the bits in the range L to R in the binary representation of N, i.e, to toggle bits from the rightmost Lth bit to the rightmost Rth bit. A toggle operation flips a bit 0 to 1 and a bit 1 to 0.

**Input:**  
First line of input contains a single integer T which denotes the number of test cases. Then T test cases follows. First line of each test case contains three space separated integers N, L and R.

**Output:**  
For each test case , print the number obtained by toggling bits from the rightmost Lth bit to the rightmost Rth bit in binary representation of N.

**Constraints:**  
1<=T<=100  
1<=N<=1000  
1<=L<=R  
L<=R<= Number of bits(N)

**Example:**  
**Input:**  
2  
17 2 3  
50 2 5  
**Output:**  
23  
44

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

Contributor: Harsh Agarwal

<http://practice.geeksforgeeks.org/problems/toggle-bits-given-range/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.\*;

import java.math.\*;

import java.util.\*;

/\*\*

\*

\* @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[] input = br.readLine().trim().split(" ");

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

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

int r = Integer.parseInt(input[2]);

char[] bin = Integer.toBinaryString(n).toCharArray();

int cont =1;

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

if(cont >= l && cont <= r) {

if(bin[i] == '0') {

bin[i] ='1';

}

else if(bin[i] == '1') {

bin[i] = '0';

}

}

cont++;

}

System.out.println(Integer.parseInt(new String(bin),2));

}

}

}