**Copy Set Bits in Range :-**

Easy Accuracy: 39.22% Submissions: 23K+ Points: 2

Given two numbers **X** and **Y**, and a range [**L**, **R**] where 1 <= L <= R <= 31. You have to copy the set bits of **'Y'** in the range L to R in**'X'.** Return this modified X.

Note: Range count will be from Right to Left & start from 1.

**Example 1:**

**Input:**

X = 44, Y = 3

L = 1, R = 5

**Output:**   
47

**Explaination:**   
Binary represenation of 44 and 3 is 101100 and 0000**11**. So in the range 1 to 5 there are two set bits of 3 (1st & 2nd position). If those are set in 44 it will become 1011**11** which is 47.

**Example 2:**

**Input:**

X = 16, Y = 2

L = 1, R = 3

**Output:** 18

**Explaination:** Binary represenation of 16 and 2 is 10000 and **10**. If the mentioned conditions are applied then 16 will become 100**10** which is 18.

**Your Task:**  
You do not need to read input or print anything. Your task is to complete the function **setSetBit()** which takes the numbers X, Y, L and R as input parameters and returns the modified value of X.

**Expected Time Complexity:** O(R - L)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 ≤ X, Y ≤ 109  
1 ≤ L ≤ R ≤ 32

**Code :-**

//{ Driver Code Starts

// Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

// User function Template for C++

class Solution{

public:

int setSetBit(int x, int y, int l, int r){

int temp=1<<(r-1);

for(int i=r-2; i>=(l-1); i--){

temp = temp | (1 << i);

}

temp = temp & y;

x = x | temp;

return x;

}

};

//{ Driver Code Starts.

int main(){

int t;

cin>>t;

while(t--){

int x, y, l, r;

cin>>x>>y>>l>>r;

Solution ob;

cout<<ob.setSetBit(x, y, l, r)<<"\n";

}

return 0;

}

// } Driver Code Ends

**T.C :- O(R-L)**

**S.C :- O(1)**