XOR sum hackerrank

An array, , is defined as follows:

* for , where  is the symbol for [XOR](https://en.wikipedia.org/wiki/Exclusive_or)

You will be given a left and right index . You must determine the XOR sum of the segment of  as .

For example, . The segment from  to  sums to .

Print the answer to each question.

**Function Description**

Complete the *xorSequence* function in the editor below. It should return the integer value calculated.

xorSequence has the following parameter(s):

* *l*: the lower index of the range to sum
* *r*: the higher index of the range to sum

**Input Format**

The first line contains an integer , the number of questions.  
Each of the next  lines contains two space-separated integers,  and , the inclusive left and right indexes of the segment to query.

**Constraints**

**Output Format**

On a new line for each test case, print the *XOR-Sum* of 's elements in the inclusive range between indices  and .

**Sample Input 0**

3

2 4

2 8

5 9

**Sample Output 0**

7

9

15

**Explanation 0**

The beginning of our array looks like this:

*Test Case 0:*

*Test Case 1:*

*Test Case 2*:

**Sample Input 1**

3

3 5

4 6

15 20

**Sample Output 1**

5

2

22

**Explanation 1**

. Perform the xor sum on each interval:  
  
  
#include<bits/stdc++.h>

#define pp pop\_back

#define pb push\_back

#define int long long int

#define INF 1e18

#define vec vector<int>

#define pii pair<int,int>

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int xr(int x)

{

   if(x%4==0)

   return x;

   if(x%4==1)

   return 1;

   if(x%4==2)

   return x+1;

   else

   return 0;

}

int cale(int x)

{

   return xr(x/2)\*2;

}

int calo(int x)

{

  if(((x-1)/2)%2==0)

  return xr((x-1)/2)\*2+1;

  else

  return xr((x-1)/2)\*2;

}

int B(int x)

{

  if(x%2)

  return calo(x);

  else

  return cale(x);

}

int32\_t main()

{

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

  cin.tie(NULL);

  cout.tie(NULL);

  int t=1;

  cin>>t;

  while(t--)

  {

    int l,r;

    cin>>l>>r;

    cout<<(B(r)^B(l-1))<<"\n";

  }

}