I

Strange Summation

Input: Standard Input
Output: Standard Output

Mr Foolizm is learning binary numbers. He has a software that converts any decimal integer to binary and vice versa. So, if he is asked to add some decimal numbers, he first converts them to binary. Then he adds them in binary. But he doesn't understand the carry principle yet and he even doesn't know the idea of positioning the numbers. Instead of LSD (least significant digit), he starts summing from MSD (most significant digit). So, if he is asked to add four integers from 1 to 4, he does the following:

1. He first takes 1 and 2 and convert them to binary using his software, and then he adds them like the following:

   ```
   1

   10

   -----

   00
   ```

2. Then he takes 3, converts it to binary and add it with the previous result

   ```
   00

   11

   -----

   11
   ```

3. He then takes 4, converts it to binary and add it with the previous result

   ```
   11

   100

   -----

   010
   ```

4. And finally he converts the result back to decimal, so, the result is 2.

Now you are given two integers **p** and **q**, your task is to find the result if Mr Foolizm adds all integers from **p** to **q** (inclusive) in his procedure.

Input

Input starts with an integer **T (≤ 10000)**, denoting the number of test cases. Each case starts with a line containing two integers: **p** and **q** **($0 < p < q < 2^{63}$)**.

Output

For each case, print the case number and the result.

Sample Input

```
4
1 4
2 5
1 2147483647
7 12
```

Output for Sample Input

```
Case 1: 2
Case 2: 3
Case 3: 1610612736
Case 4: 2
```