A XOR Problem

Let us suppose we have a “N” bit integer X. We call an integer X’ as neighbor-X, if X’ can be obtained by shuffling the bits of integer X.

For example If X = 6 and N = 5, then X will be represented as X = $(00110)_2$, So all the 5 bit integers that have exactly two 1s in their binary representation are called neighbor-X. For example $(00011)_2$, $(01001)_2$, $(10100)_2$, ... all of these are neighbor-X.

Now you are given two N bit integers “X” and “Y”, then you have to find the maximum possible value of (X’ xor Y’) where X’ is neighbor-X.

xor operator takes two bit strings and performs XOR operation between them. For example:

$(10010)_2$ xor $(01011)_2$ = $(11001)_2$

Input

The first line of the input will contain an integer “T”, the number of test cases. Each of the next “T” lines will contain three integers “N”, “X” and “Y”, N is the number of bits in integers X and Y.

Output

For each test case you have to print the maximum possible value of (X’ xor Y’) in a separate line.

Constraints

1 <= T <= 50

1 <= N <= 30

0 <= X, Y < 2^N

Example

Input:
3
3 5 4
5 0 1
4 3 7

Output:
7
16
14

Explanation

In the first case X = 5 = $(101)_2$ and Y = 4 = $(100)_2$, So we can have X' = $(101)_2$ and Y' = $(010)_2$, X' xor Y' = $(111)_2$ = 7

In the second case X = 0 = $(00000)_2$ and Y = 1 = $(00001)_2$, So we can have X' = $(00000)_2$ and Y' = $(10000)_2$, X' xor Y' = $(10000)_2$ = 16

In the third case X = 3 = $(0011)_2$ and Y = 7 = $(0111)_2$, So we can have X' = $(0011)_2$ and Y' = $(1101)_2$, X' xor Y' = $(1110)_2$ = 14