Another Long Sequence Inversion

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given three non-negative integers L, R and X in binary. Find the inversion number of the integer sequence $(L \oplus X, (L+1) \oplus X, \ldots, R \oplus X)$ of length $R - L + 1$, and output it in binary.

Here, $\oplus$ is the bitwise XOR operation.

You have T test cases; solve each of them.

Definition of the inversion number

The inversion number of a sequence $B = (B_1, B_2, \ldots, B_M)$ of length M, is the number of pairs of integers $(i, j)(1 \leq i < j \leq M)$ such that $B_i > B_j$.

Input

The input is given from Standard Input in the following format:

T
case_1
case_2
$\vdots$
case_T

Each test case is given in the following format:

L R X

- $1 \leq T \leq 2 \times 10^5$
- $0 \leq L \leq R < 2^{2 \times 10^5}$
- $0 \leq X < 2^{2 \times 10^5}$
- L, R and X are given in binary representation without leading zeros (except that 0 is considered a one-digit integer).
- The total number of digits in the binary representation of R over all test cases does not exceed 2×10^5.
- The total number of digits in the binary representation of X over all test cases does not exceed 2×10^5.
- All input values are integers.

Output

Output T lines. On the i-th line, print the answer for the i-th test case in binary.

Example

standard input	standard output
3 101 1000 10 1101 10111010 0 1000110 1110011011101 100011110010	10 0 11100100001101111010100

Note

In the first test case, L, R and X in decimal are $L = 5, R = 8$ and $X = 2$, respectively. Since $(5 \oplus 2, 6 \oplus 2, 7 \oplus 2, 8 \oplus 2) = (7, 4, 5, 10)$, the inversion number is 2.