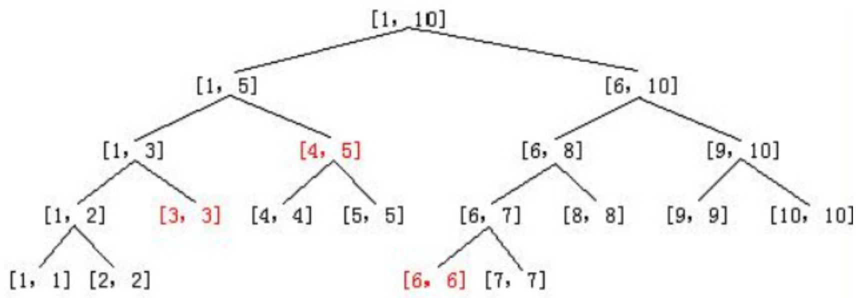
Cover

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Segment tree is a kind of binary tree, and each of its node can cover an interval. A node in the segment tree is a leaf node if and only if it simply covers one index, which means its interval only includes an integer. To a non-leaf node covering interval $[a, b]$, its left child's interval is $[a, \lfloor\frac{a+b}{2}\rfloor]$, and its right child's interval is $[\lfloor\frac{a+b}{2}\rfloor + 1, b]$. So we can determine the structure of one segment tree uniquely when we know the coverage interval of its root node $[1, n]$.

It can be proved that to a given interval $[l, r]$, there is only one **gorgeous-cover** method, we note it as $w[l, r]$, which represents the minimal number of intervals needed to cover $[l, r]$.



For example, in this $[1, 10]$ segment tree, $w[3, 6] = 3$ .The selected intervals have been painted red ($[3, 3], [4, 5], [6, 6]$).

Given n, l, r, k, you need to find the shortest interval $[l', r']$ in the $[1, n]$ segment tree which satisfies $l' \le l \le r \le r'$ and $w[l', r'] \le k$. It's guaranteed that $k \le w[l, r]$.

Input

The first line contains an integer $T (T \le 10^5)$, the number of the test cases.

Each of the next T lines contains 4 integers $n, l, r, k (1 \le l \le r \le n \le 10^{18}, 1 \le k \le w[l, r])$.

Output

To each of the test cases, output one integer in a line, the length of the shortest interval. **Note that the length of interval $[l, r]$ is $r - l$.**

Example

standard input	standard output
2 10 3 6 1 10 2 6 3	9 5