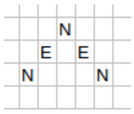
IOITC 2016 Finals, Day 1

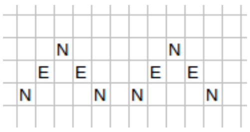
Binary Tree in Grid

Watson has always wanted to draw complete binary trees in grids. In a grid, rows are numbered in increasing order from top to bottom (row-numbers) and columns are numbered in increasing order from left to right (column-numbers). A cell is represented as (r, c), where r is it's row-number and c is it's column-number. Assuming that he has an infinite sized grid, he first draws a complete binary tree T_1 of height 1 in the grid, as shown below:

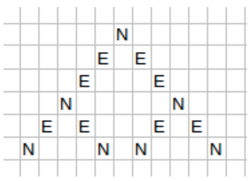


Here character E denotes an edge and character N denotes a node.

To create a complete binary tree of height 2, T_2; Watson will create copy of T_1 and align it with T_1 such that both the roots have the same row-number, and give a separation of one column between rightmost child of T_1 and left most child of copy(T_1). This image depicts result of this process:



Next, he keeps adding edges diagonally from roots of T_1 and copy(T_1) until the root node of T_2 can be added. End result of this is T_2, as shown in the image.



In this way a complete binary tree of any height can be represented in a grid. Watson's algorithm is summarized as follows:

```
//draws complete binary tree of height H in the grid
function drawTree( H ):
        if H == 1:
            Print T1(as defined in statement) in grid
            return

        T = drawTree( H - 1 )
        T' = copy( T )
```

```
Put T' to right of T and align so that the row-numbers of both the roots are the same.
Leave one column between rightmost child of T and leftmost child of T'.

Let (r, c) be coordinates of root of T
Let (r, c') be coordinates of root of T'
// c' will be greater than c.

do:
    r -= 1
    c += 1
    c' -= 1
    put character 'E' at cells (r, c) and (r, c')
while c != c'

//Now, c == c'.
//(r, c) is the coordinates of the root of tree of height H.
put character 'N' at cell (r, c)
```

Now, Sherlock is going to test Watson's algorithm. He gives him an integer H and cell coordinates (r_{root}, c_{root}) and tells Watson to draw a complete binary tree of height H such that root of this tree lies at coordinates (r_{root}, c_{root}). Now, Sherlock will give Q queries of following type:

- r_1, c_1, r_2, c_2: Output one integer, the number of characters N in the rectangular grid defined by top-left and bottom-right corners (r_1, c_1) and (r_2, c_2), respectively.
  Note that the cells (r_1, c_1) and (r_2, c_2) are both included in the query rectangle. That is, the boundaries are included.

Input

First line contains four space separated integers: H, r_{root}, c_{root} and Q, as defined in the statement.
Each of the next Q lines contains four space separated integers, which represent queries of the form r_1, c_1, r_2, c_2 as specified before.

Output

For each query, print the required answer in one line.

Test Data

$0 \leq r_{root}, c_{root}, r_1, c_1, r_2, c_2 \leq 10^{18}$
$r_1 \leq r_2$
$c_1 \leq c_2$

Subtask 1 (20 Points): $1 \leq H \leq 10$ and $1 \leq Q \leq 20$.

Subtask 2 (80 Points): $1 \leq H \leq 50$ and $1 \leq Q \leq 10^5$.

Sample Input1

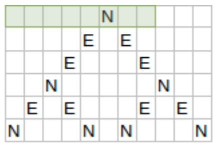
```
2 0 5 3
0 0 0 7
0 5 5 10
3 0 5 4
```

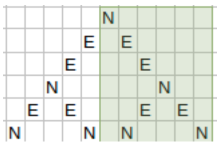
Sample Output1

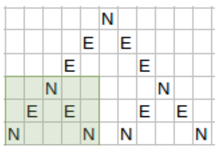
```
1
4
3
```

Explanation

The rectangular query areas are highlighted for all queries in following images:







Limits

Time: 2 seconds

Memory: 256 MB