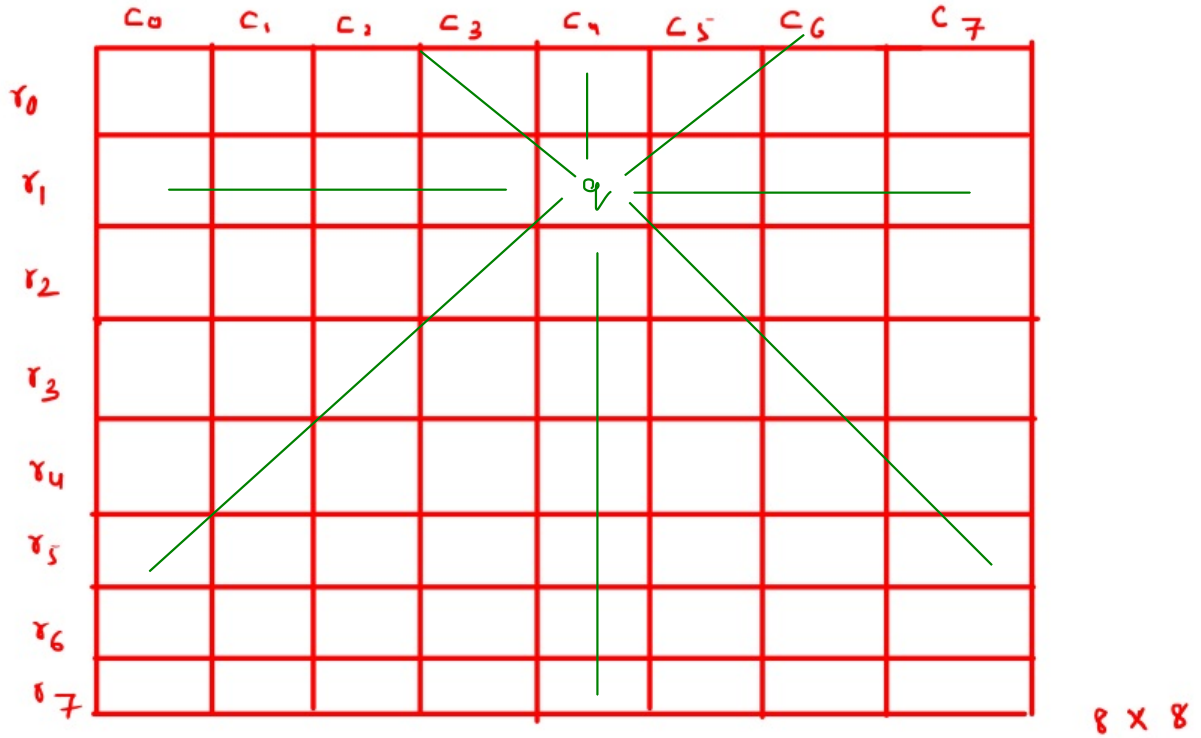
N-Queen Problem :-

n x n

4 x 4



8 x 8

board

	C_0	C_1	C_2	C_3
r_0		1		
r_1				1
r_2	1			
r_3			1	

4x4

sol 1

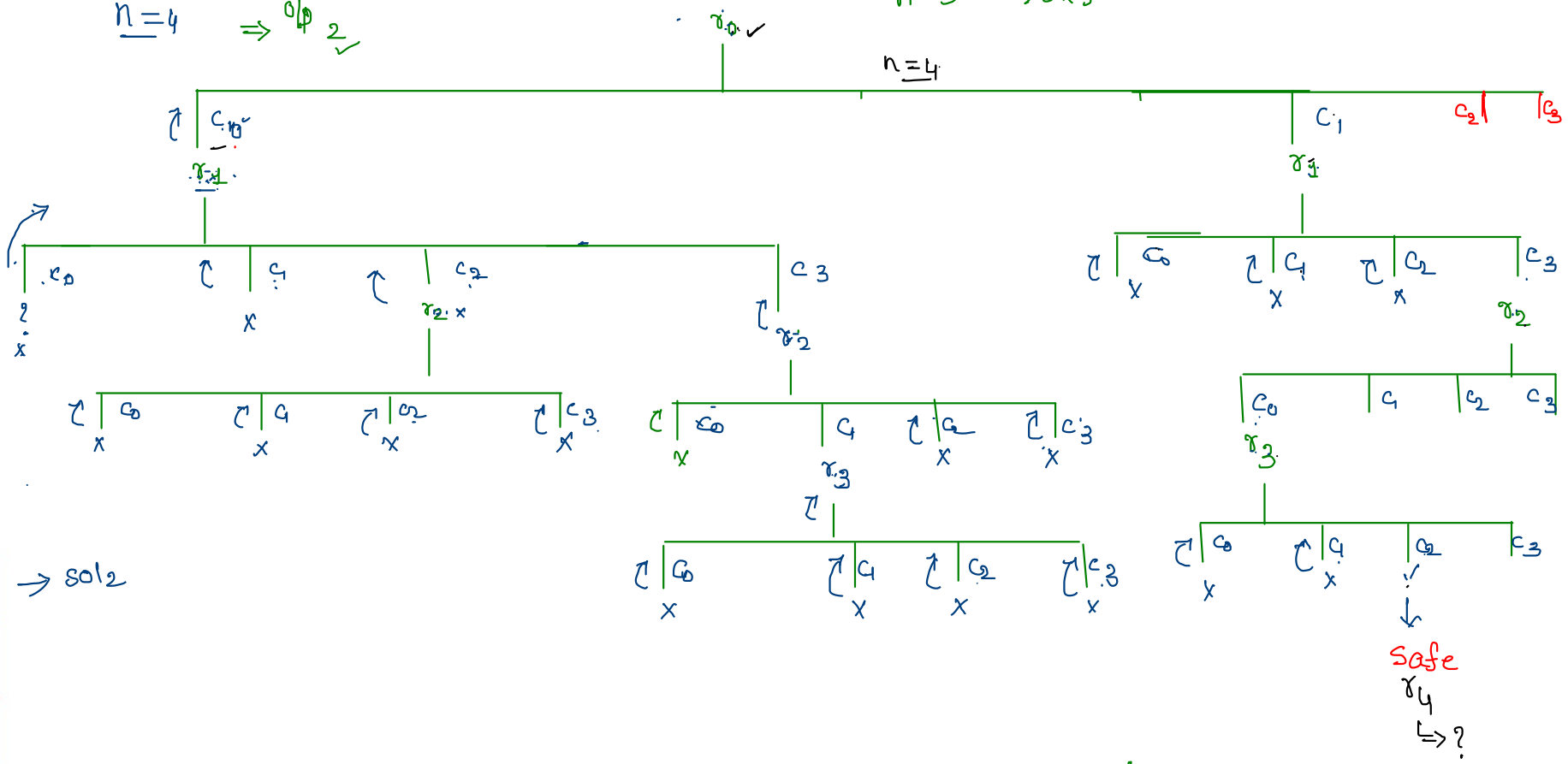
mirror image

	C_0	C_1	C_2	C_3
r_0			1	
r_1	1			
r_2				1
r_3		1		

→ sol 2

$n=4$ ⇒ o/p 2

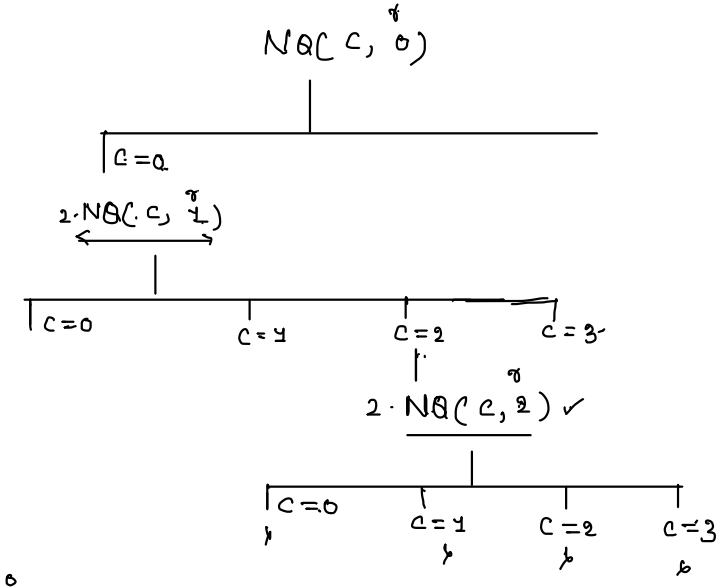
$n=5$ ⇒ 5x5

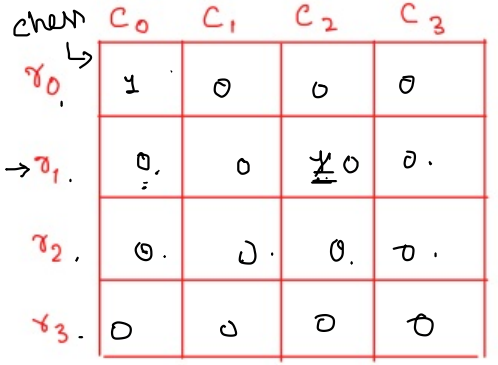


i/p : n ⇒ $n \times n$

o/p → integer

```
count=0
NQueen(chess[][],row)
{
    if(row==chess.length)
    {
        count++
        return count
    }
    for(col=0;col<chess.length;col++)
    {
        if( isQueenSafe(chess,row,col) )
        {
            1. chess[row][col]=1
            2. NQueen(chess,row+1)
            3. chess[row][col]=0
        }
    }

}
```

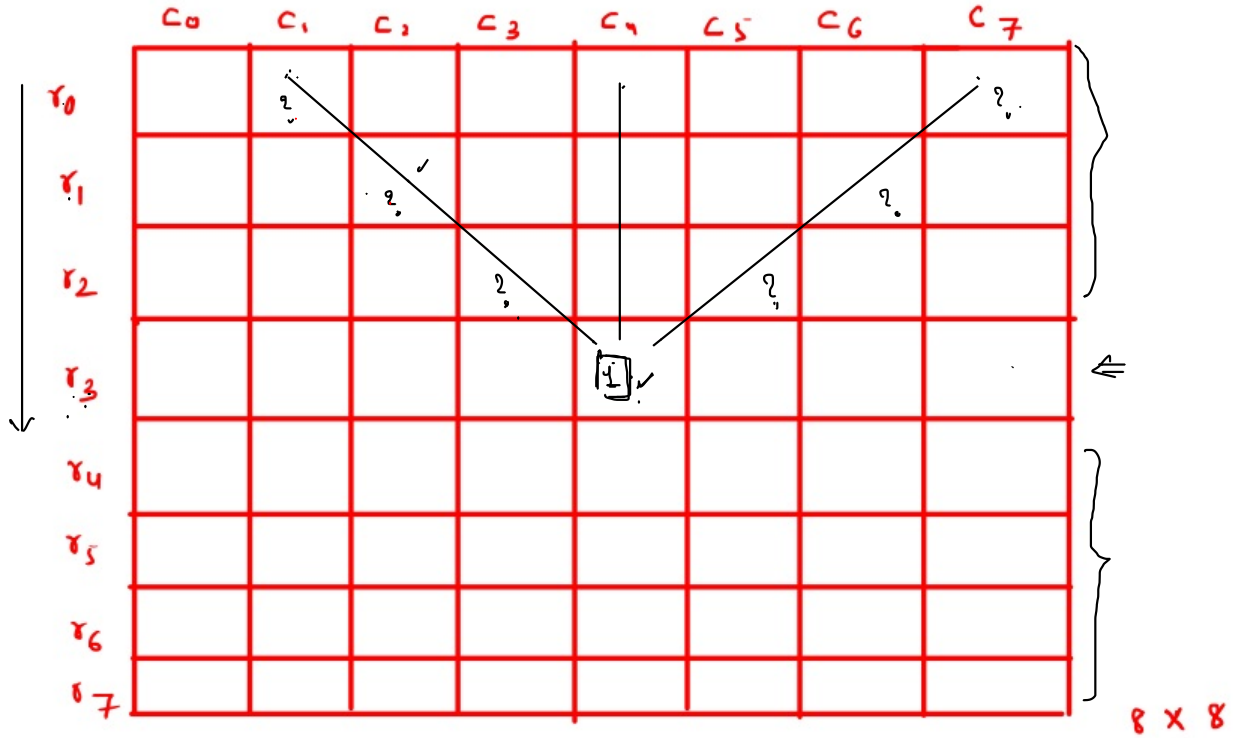




n=4,

4×4

```
main()
{
    1. read : n
    2. 2D-arr : n x n
    3. NQueen(chess, 0)
       I/p : int-value
}
```

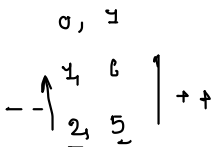


chessCJCJ

row

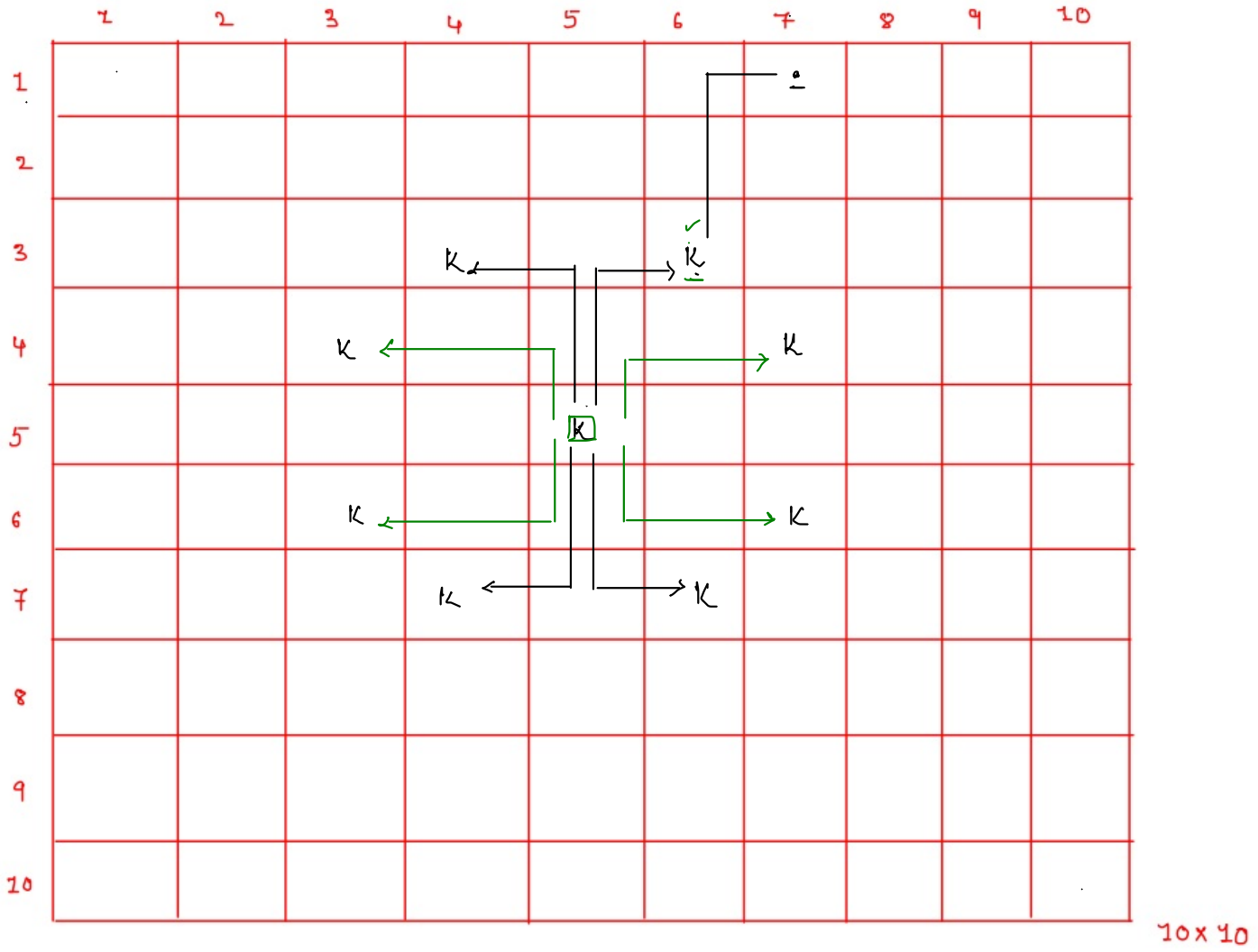
col

Left-Diag:-

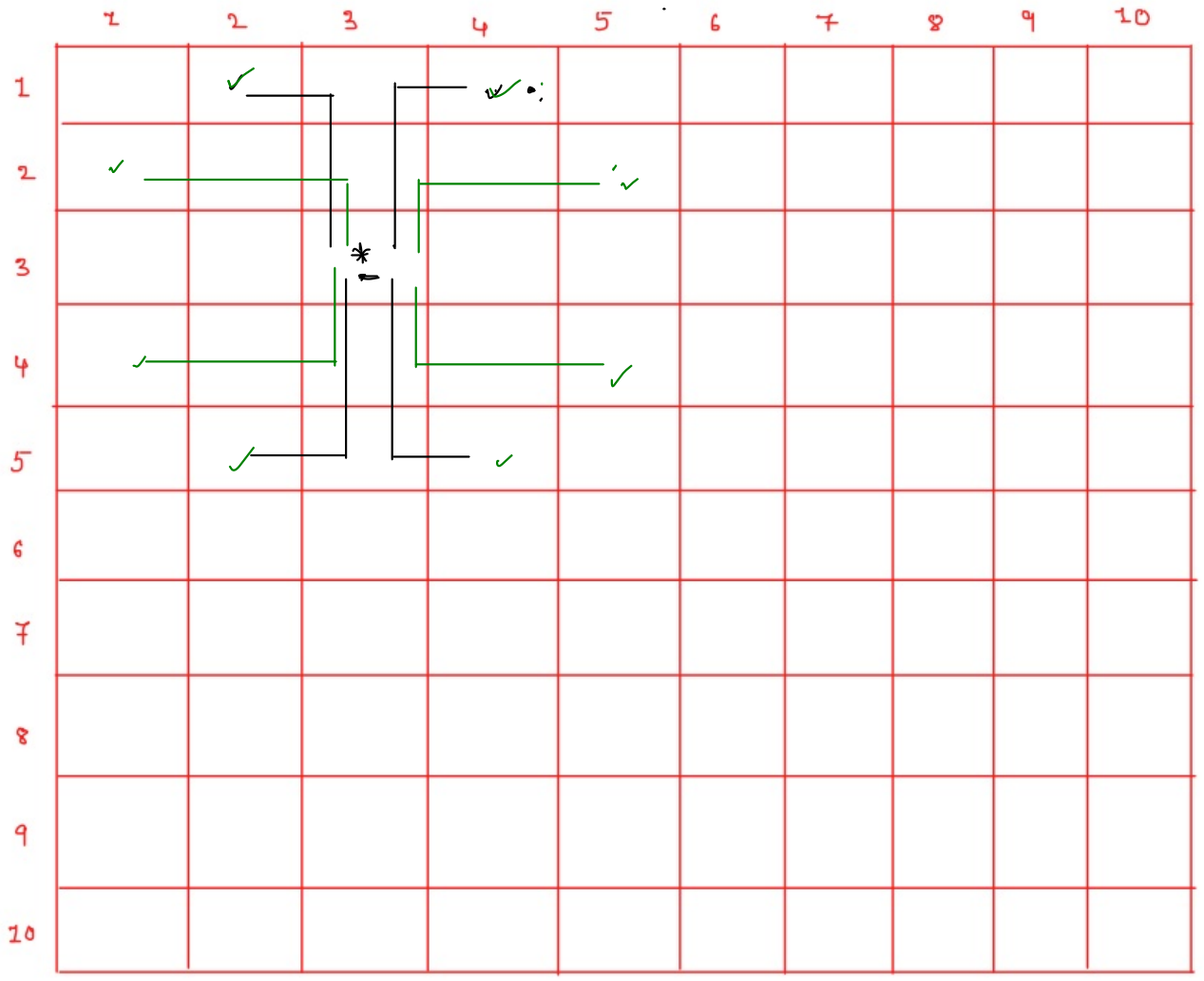


```
function isQueenSafe(chess[][],row,col)
{
	//left-Diag
	for(i=row-1,j=col-1; i>=0 && j>=0 ; i--,j--)
	{
		if(chess[i][j]==1)
			return false
	}

	//vertical
	for(i=row-1; i>=0; i--)
	{
		if(chess[i][col]==1)
			return false
	}

	//right-Diag
	for(i=row-1,j=col+1; i>=0 && j<chess.length; i--,j++)
	{
		if(chess[i][j]==1)
			return false
	}
	return true
}
```

2) Follow the Knight



K
8



10 x 10

fun(x, y, n)

3 3 4

1 2

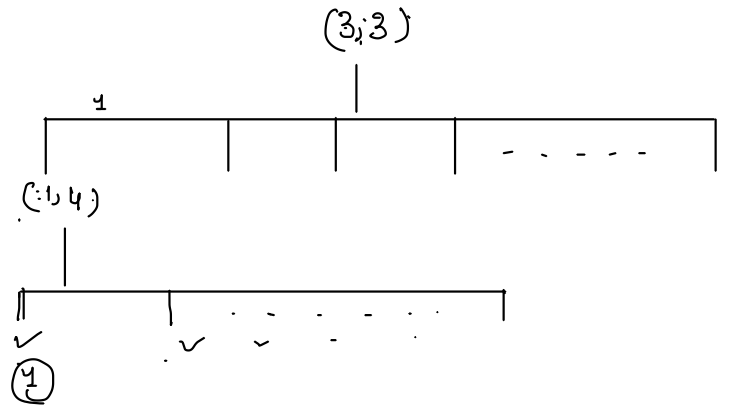
1

0

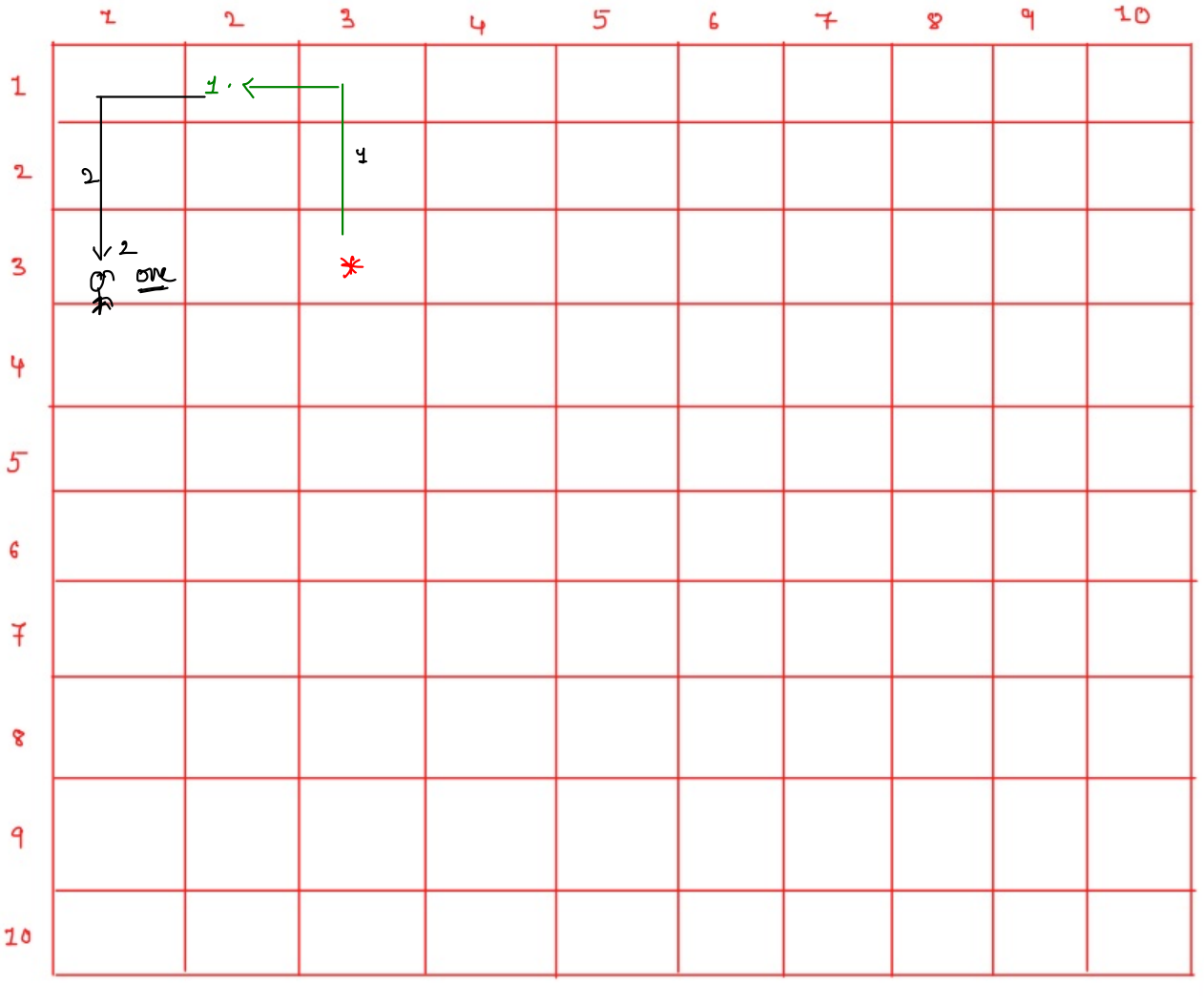
3, 3, N = 4.

8

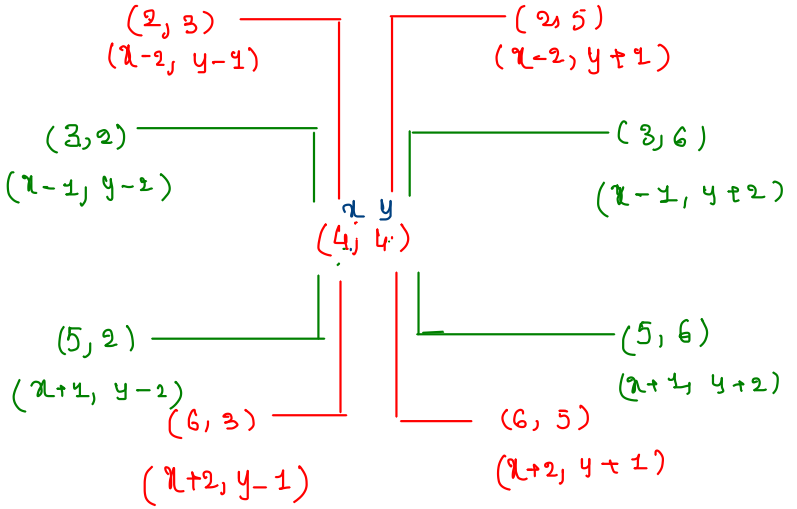
Let N = 2



100

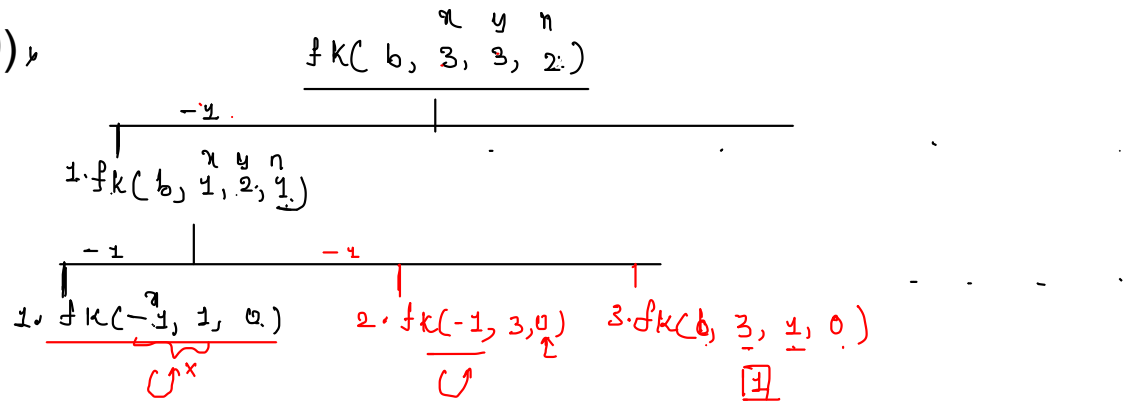

1
2
3
4
5
6
7
8
9
10
1
2
4
2
or one
*
10 x 10

y
x(i)
x
y(j)

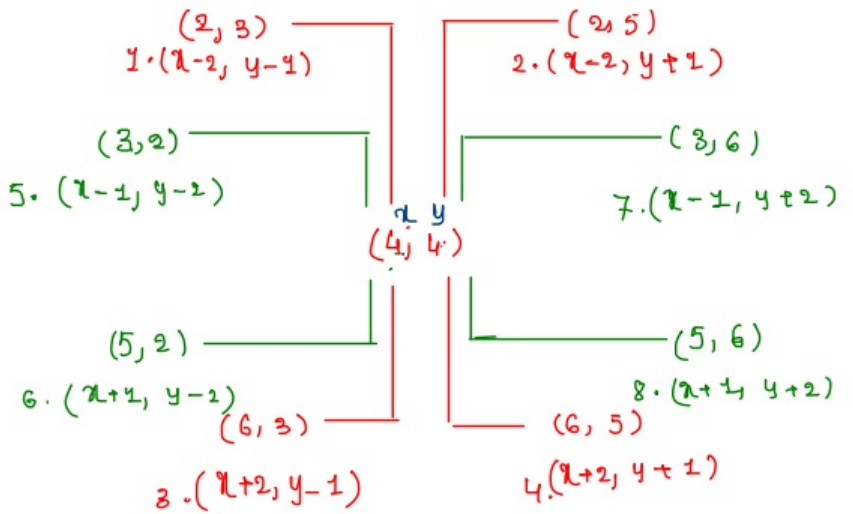

(2, 3)
(x-2, y-1)
(2, 5)
(x-2, y+1)
(3, 2)
(x-1, y-2)
(3, 6)
(x-1, y+2)
x y
(4, 4)
(5, 2)
(x+1, y-2)
(5, 6)
(x+1, y+2)
(6, 3)
(x+2, y-1)
(6, 5)
(x+2, y+1)

N
↳ N-1
↳ N-2 . . . 0

```
function followKnight(board[][],x,y,n) // fill all values of board with 0's , its size is 10*10
{
    if(x<0 || x>9 || y<0 || y>9)
            return;
     if(n==0)
      {
            board[x][y]==1
            return
      }
    followKnight(board,x-2,y-1,n-1)
    followKnight(board,x-2,y+1,n-1)
    followKnight(board,x+2,y-1,n-1)
    followKnight(board,x+2,y+1,n-1)
    followKnight(board,x-1,y-2,n-1)
    followKnight(board,x+1,y-2,n-1)
    followKnight(board,x-1,y+2,n-1)
    followKnight(board,x+1,y+2,n-1)
}
```




```
count=0;
for(i=0;i<10;++)
{
        for(j=0;j<10;j++)
        {
                if(board[i][j]==1)
                        count++;
        }
}
console.log(count);
```


(2, 3)
1. (x-2, y-1)
(2, 5)
2. (x-2, y+1)
(3, 2)
5. (x-1, y-2)
(3, 6)
7. (x-1, y+2)
x, y
(4, 4)
(5, 2)
6. (x+1, y-2)
(5, 6)
8. (x+1, y+2)
(6, 3)
3. (x+2, y-1)
(6, 5)
4. (x+2, y+1)