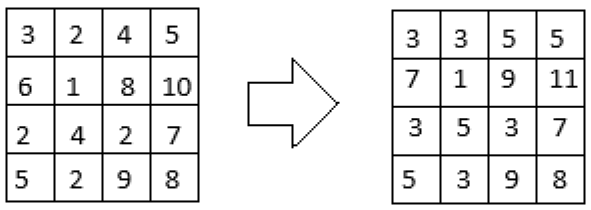
1) (10 pts) DSN (Recursive Coding)

Imagine that a virus is spreading through an area we can model as a 2-dimensional integer array of size ***n*** by ***n***, where each value in the array stores the current level of virus. The virus can be activated at a particular location. If that location currently has a virus level that is an even integer, then the virus level in that square increases by 1, and then the activation recursively activates the locations above, below, left and right of the immediate location. If the virus level in that square was odd, no change occurs and the virus doesn't spread. Luckily, once the virus level in a square increases by 1 due to an activation, it can't increase again. Here is a small example of a before and after picture of activating the virus in row 2, column 0:



3	2	4	5
6	1	8	10
2	4	2	7
5	2	9	8

3	3	5	5
7	1	9	11
3	5	3	7
5	3	9	8

Complete the **recursive** function below so that it takes in a pointer to the array storing the grid, the value of n, and the row and column of where the virus is activated, and updates the virus levels accordingly. Don't forget to make sure you don't go out of bounds of the array!

```
void activate(int** grid, int n, int row, int col) {

    if (row < 0 || row >= n) return;

    if ( col < 0 || col >= n ) return; // col out of bounds

    if ( grid[row][col]%2 == 1 ) return; // odd square

    grid[row][col]++;

    activate(grid, n, row-1, col) ;

    activate(grid, n, row, col-1) ;

    activate(grid, n, row, col+1) ;

    activate(grid, n, row+1, col) ;

}
```

Grading: 2 pts out of bounds check for col
3 pts for odd square check
1 pts for incrementing correct item
1 pt each recursive call, give credit if last two parameters are correct (order doesn't matter)