Code Readability and Elegance
Shorter, more readable count is easier to maintain, reuse, and edit

Whenever we write a function, we should include a description of what it does
Do this via a “docstring”, which is inside of three double quotes (“””docstring”””) and is the first thing in a function
It lists the meaning and types of each parameter and of what is returned, if something is returned

Format:
“””
:param first_parameter**:** the first parameter for this function
:type first_parameter**:** int, str, float, list, etc.
:return: True - first_parameter is true, False - otherwise
:rtype: bool
“””

Note: from now on, all homework assignments need docstrings
Always put a blank line between sections of code, and two blank lines between functions
Between values, operators, anything, it is standard to include a single space

In-Class Exercise:
Available at http://www.cs.virginia.edu/~up3f/cs1110/inclass/inclass-decisions.html

Problem #1

```
def is_diagonal(wd, tile):
    """
    this function will tell us whether the given tile is along the diagonal
    :param wd: width of a square grid
    :type wd: int
    :param tile: tile to check if it is along the diagonal of the grid
    :type tile: int
    :return: True - a tile is along diagonal, False - otherwise
    :rtype: bool
    """

    #we say (tile - 1) because the tiles start at 1, but the rows and columns start at
zero
    row = (tile - 1) // wd
    col = (tile - 1) % wd

    # if row == col:
    #     return True
    # else:
    #     return False
    #this is one option, but we can shorten it

    return row == col
```

Note: the gray underlines in Pycharm indicate that code is technically correct, but could be more elegant

Problem #2

```
def is_edge(wd, h, tile):
    """
    :param wd: width of a square grid
    :type wd: int
    :param tile: tile to check if it is along the edge
    :type tile: int
    :param h: the height of the grid
    :type h: int
    :return: True - tile is along the edge, False - otherwise
    :rtype: bool
    """

    row = (tile - 1) // wd
    col = (tile - 1) % wd

    # if row == 0 or row == h - 1 will be on edge
    # if col == 0 or col == wd - 1 will be on edge

    return row == 0 or row == h - 1 or col == 0 or col == wd - 1

#   can also be written:
    # if row == 0 or row == h - 1:
    #     return True
    # if col == 0 or col == wd - 1:
    #     return True
    # return False
#   because anytime a function hits "return" it stops executing

print(is_edge(3, 4, 6))
print(is_edge(3, 4, 5))
```

Logical Operators Review

x	y	x and y	x or y	not x	not y
False	False	False	False	True	True
False	True	False	True	True	False
True	False	False	True	False	True
Ture	True	True	True	False	False