Rule #1: Determine edge effect in each dimension to create row and column lists using range() for iterating over center cells

Edge effect$_{rows}$ = $\frac{Window\ size_{rows} - 1}{2}$ (the same applies for columns)

E.g., if the window is a 5x5, then the Edge effect$_{rows}$ = $\frac{5-1}{2}$ = 2; for the 3x3 below: Edge effect$_{rows}$ = $\frac{3-1}{2}$ = 1

Rule #2: Create slice indices to subset the window (centered on the current row, col position) from the array

Slice indices -> my_array[current row index - edge effect$_{rows}$: current row index + edge effect$_{rows}$ + 1,

current column index - edge effect$_{cols}$: current column index + edge effect$_{cols}$ + 1]

The 1's here are constant. You add 1 because indexing is exclusive on the high end, i.e., we need to add 1 to the edge effect on the high end.

E.g., For a 5x5 window, the slice indices are: [row - 2 : row + 3, col - 2 : col + 3]; for the 3x3 below: [row - 1 : row + 2, col - 1 : col + 2]

The range function must exclude the edge effect on both ends. Example below for the 3x3 we did in class

Edge effect$_{rows}$

Number of rows in array - Edge effect$_{rows}$

Edge effect$_{cols}$

Number of columns in array - Edge effect$_{cols}$

```
for row in range(1, b3.shape[0] - 1):

    for col in range(1, b3.shape[1] - 1):

        win = b3[row - 1:row + 2, col - 1:col + 2]

        sumArray[row, col] = win.sum()
```

Current row index (row)

Current column index (col)

Slice indices