15-112 Fall 2022 Lecture 3
Quiz 6B
15 minutes

Name: ____________________________________

Andrew ID: ___________________@andrew.cmu.edu

Section: _______

- **You may not use any books, notes, or electronic devices during this quiz.**
- **You may not ask questions about the quiz except for language clarifications.**
- **Show your work on the quiz (not scratch paper) to receive credit.**
- **If you use scratch paper, you must submit it with your andrew id on it, and we will ignore it.**
- **All code samples run without crashing unless we state otherwise. Assume any imports are already included as required.**
- **Do not use these topics: sets/dictionaries and recursion.**
- **You may use almostEqual() and rounded() without writing them. You must write everything else.**

Scoring: this quiz contains 10 short questions. Here is how we convert raw scores to quiz6 scores (without bonus, which is up to 5 points, and is added after the conversion). Fractional scores are interpolated.

Raw Score	**Quiz6 Score (before bonus)**
10/10	**100**
9/10	**94**
8/10	**88**
7/10	**80**
6/10	**72**
5/10	**62**
4/10	**52**
3/10	**40**
2/10	**28**
1/10	**14**
0/10	**0**

Part A: Fill-in-the-blank (code completion)

Each of the following functions (or portions of functions) are copied from the course notes on Animations with 2d Lists. We then removed some part of the code. For each of the following, fill in the blank with the missing code. First, as a reminder, here are the values in our model:

app.rows	The number of rows in our board.
app.cols	The number of columns (or "cols") in our board.
app.boardLeft	The x value of the left edge of the board on the canvas.
app.boardTop	The y value of the top edge of the board on the canvas.
app.boardWidth	The width of the board in pixels.
app.boardHeight	The height of the board in pixels.
app.cellBorderWidth	The width of each cell's border (typically 1 or 2).

1.

```
# returns the (left, top) pixel of the cell at (row, col) on the board:
def getCellLeftTop(app, row, col):
    cellWidth, cellHeight = getCellSize(app)

    cellLeft = ___________________________________________________________
    ...
```

2.

```
# draws the black border around the entire board.  called at the end
# of redrawAll, after drawBoard already called drawCell on every cell.
# Hint: app.cellBorderWidth may be useful here.
def drawBoardBorder(app):
  drawRect(app.boardLeft, app.boardTop, app.boardWidth, app.boardHeight,
           fill=None, border='black',

           borderWidth=___________________________________________________)
```

3.

```
# returns the (width, height) of each cell on the board:
def getCellSize(app):
    cellWidth =___________________________________________________________
    ...
```

4.

```
# returns the (row, col) of the cell that (x, y) is in, or None if it is
# not on the board.
def getCell(app, x, y):
    dx = x - app.boardLeft
    dy = y - app.boardTop
    cellWidth, cellHeight = getCellSize(app)

    row = ________________________________________________________________
    ...
```

5.

```
# Note: this is not from the notes.  Rather, it is taken from our sample
# solution to rotate2dListClockwise, which was written as a step in Tetris:
def rotate2dListClockwise(L):
    oldRows, oldCols = len(L), len(L[0])
    newRows, newCols = oldCols, oldRows
    M =  [([None] * newCols) for row in range(newRows)]
    for oldRow in range(oldRows):
        for oldCol in range(oldCols):
            newRow = oldCol

            newCol = ______________________________________________________
            M[newRow][newCol] = L[oldRow][oldCol]
    return M
```

6.

```
# Note: this is taken from the writeup for "Moving the Piece" in Tetris.
# This drops the current piece as far as it can legally go.
def hardDropPiece(app):
    while movePiece(app, +1, 0):

        ___________________________________________________________________
```

Part B: Very Short Answers

Answer each of the following **in 10 words or less**. Answers that include more than 10 words will not receive any credit. Write clearly!

7. The Tic-Tac-Toe writeup says: "The approach we will take here will be a bit more general than strictly necessary. In particular, we will not hardcode the board to be 3x3 when searching for 3-in-a-row, but instead our solution will work for any size board." What is the reason it gave for this being especially helpful?

8. The Snake writeup says that "the starter code includes the function setAutogradeMode(app), which helps the autograder. First, it makes the board smaller...." What is the second change that happens in autograde mode, which makes it much easier to autograde your code?

9. In "Immortal Snake with Wraparound", how did you make the snake wraparound?

10. In the "Removing Full Rows" step in Tetris, what was the purpose of the function loadTestBoard(app, key) that we provided for you?

11. Bonus CT [Up to 5 pts]
Indicate what this prints. Place your answer (and nothing else) in the box next to the block of code.

```
def bonusCt(e):
    a = eval(e)
    b = eval(e.replace(':2', e[-1]).replace('3:', e[0]))
    for c in b:
        d = list(c)[0]
        if d in a: c.add(a[d]); a[d]=c
        else: a[d] = c
    return a
print(bonusCt('{4:2, 3:1}'))
```