Name: ______________________________ Andrew Id: ____________________

15-112 Spring 2025 Quiz 5

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use lists, dictionaries, try/except, or recursion on this quiz.

1. (8 points) **Code Tracing**:

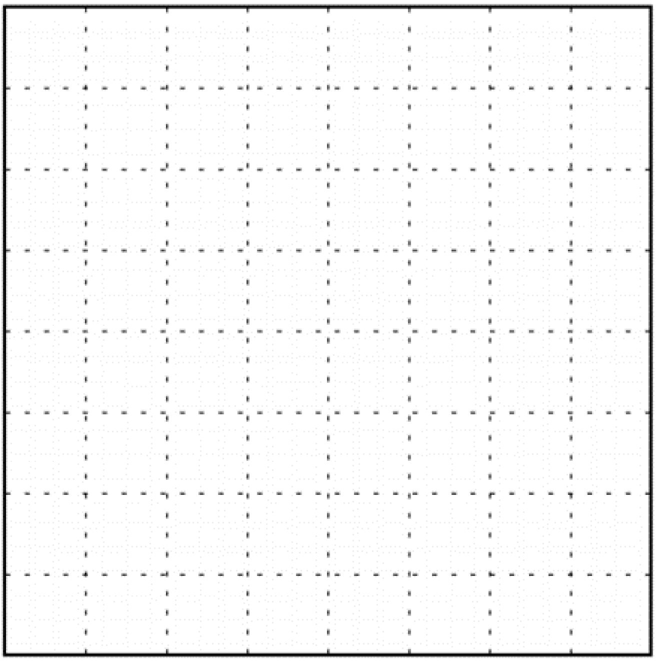
```
def drawCT(app):
    msg = 'abcde'
    cnt = 0
    for i in range(4):
        for j in range(4):
            if (i+j)%2 != 0:
                drawLabel(msg[cnt%len(msg)], i*100+50, j*100+50, align='center', size=40)
                cnt += 1
            else:
                drawRect(i*100, j*100, 100, 100, fill='black')

def redrawAll(app):
    drawCT(app)

runApp(400, 400)
```

Given that the box below is your canvas, with a width and height of 400 each, draw what the above code would display.

Hint: Each of the small boxes on the canvas is 50x50 pixels.



2. (12 points) **Free Response**: Blocky Hourglass

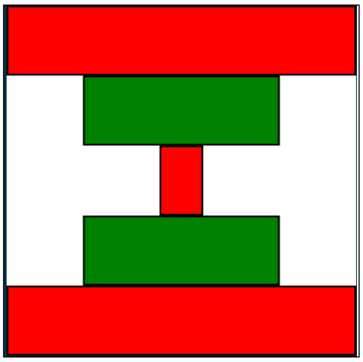
Write the function `drawBlockyHourGlass(app, n)`, which takes an integer $n > 0$ that specifies the number of blocks that form a blocky hourglass shape. The blocks are stacked on top of each other in a symmetric pattern, with their widths gradually decreasing toward the center before increasing symmetrically back to their original size.

Here are the specifications:

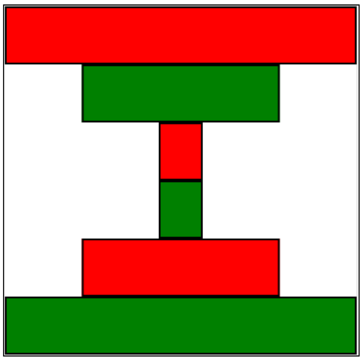
- There are exactly n blocks.
- The blocks have the same height.
- The width of the blocks decreases proportionally towards the center and then increases back symmetrically.
- The smallest block(s) always have a width of 50 pixels.
- The largest blocks span the entire window width.
- The colors of the blocks alternate between red and green, with the topmost block always being red.

Examples:

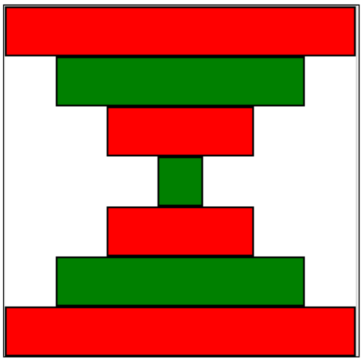
`drawBlockyHourGlass(app, 5)` produces the graphics below:



`drawBlockyHourGlass(app, 6)` produces the graphics below:



`drawBlockyHourGlass(app, 7)` produces the graphics below:



You may continue your work on the next page...

Additional Space for Answer to Question 2