Name: _____ Andrew Id: _____

15-112 Fall 2023 Quiz 7

Up to 20 + 5 minutes (finish within 20 minutes for 1-point proficiency bonus)
No calculators, no notes, no books, no computers. Show your work!
Do not use dictionaries, sets, try/except, or recursion on this quiz.

1. (8 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code. Ensure strings are enclosed with quotes and the uppercase and lowercase are distinguishable. Ensure lists are enclosed with brackets.

```
import copy
def ct1(a):
    b = a
    c = copy.copy(b)
    d = copy.deepcopy(c)
    a[0][0] = 3
    print(a[0].append(7))
    a[1] = c[1]
    c[1] = d[1]
    c[0] = [4]
    print(b is a)
    b = b[1:]
    print(b)
    if 3 in b[0]:
        b[0].remove(3)
    b[0][0] += 1
    d[1][1] = "yey"
    print("a = ", a)
    print("b = ", b)
    print("c = ", c)
    print("d = ", d)

L = [[1,2], [3,4]]
ct1(L)
print("L = ", L)
```

2. (12 points) **Free Response**: Numerical Tic-Tac-Toe

Numerical Tic-Tac-Toe is a variation of the classic game invented by the mathematician Ronald Graham. The two players take turns marking the spaces in a three-by-three grid with a number. The player who succeeds in making one horizontal, vertical, or diagonal line sum to 15 is the winner. The line may contain both even and odd numbers. In this task, you will create a simplified version in which the numbers they play in each turn are predefined and players can only choose where to play.

Here are the specifications you need to implement:

- Game Setup:
  - The game should display a 3x3 grid with numbers.
  - Initially, all grid cells contain the number 0.
  - The grid should be white on a black background.
  - A countdown timer is displayed at the bottom of the screen.
  - Player One starts, and the countdown timer starts at 20 seconds.
- Gameplay Rules:
  - Player One plays with odd numbers (1, 3, 5, 7, 9), and Player Two plays with even numbers (2, 4, 6, 8).
  - The sequence of plays is always fixed: Player One places 1, Player Two places 2, Player One places 3, and so on.
  - On each player's turn, they must select an empty cell on the grid.
  - If the selected cell contains a 0 (empty), the player can make a move (place their predefined number), and the countdown timer resets to 20 seconds.
- Winning Condition:
  - The player who succeeds in making one horizontal, vertical, or diagonal line of numbers on the grid sum to 15 is the winner. The winning line may contain both even and odd numbers and 0s.
  - If the countdown timer reaches 0 before the current player makes a move, the other player wins.
  - When a player wins, a message appears at the top of the window indicating "Player One Won!" or "Player Two Won!" accordingly.
- Draw Condition:
  - If the game ends with a draw (all cells are filled without a winning line), the message "It's a Draw" appears at the top of the window.

Your task is to write the Python code using `cmu_graphics`.

Here are some helper functions you may use in your solution.

```
def resetBoard(app):
    app.board = [ [0,0,0], [0,0,0], [0,0,0] ]

def drawGrid(app):
    cellSize = (app.width)//3, (app.height)//3
    for row in range(3):
        for col in range(3):
            # compute coordinates
            x1 = cellSize[0] * col
            y1 = cellSize[1] * row
            # draw the cell and its content
            drawRect(x1, y1, cellSize[0], cellSize[1],
                     border='white', borderWidth=5)
            drawLabel(app.board[row][col],
                      x1 + cellSize[0]//2, y1 + cellSize[1]//2,
                      size=60, fill='white')
```

Make reasonable assumptions for anything not specified here.

You may start your work on the back of this page...

Answer to Question 2

3. (2 points) **Reasoning Over Code (BONUS)**: Find an argument, `s`, for the following function to cause it to return `True`. Place your answer (and nothing else) in the box below the code.

```
import string
def roc(s):
    assert(type(s) == str and len(s) == 8)
    L = ['quiz', 6, 'cmu', '112', 42]
    while s != "":
        a = s[0]
        b = s[-1]
        s = s[1:-1]
        if a.isdigit() and b.isdigit():
            L.remove(int(a) * int(b))
            break
        elif a == b:
            L.pop(0)
        else:
            L.insert(len(L)//2, '15')
    return L == ['cmu', '15', '112'] and s == ""
```