

Name: \_\_\_\_\_ Andrew Id: \_\_\_\_\_

### 15-112 Spring 2024 Quiz 7

Up to 25 minutes. 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 answers (and nothing else) in the box next to the code. (Hint: There is blank space at the bottom of the page so that you can draw a picture as you work.)

```
def ct1(a):
    b = a
    c = copy.copy(b)
    d = copy.deepcopy(a)

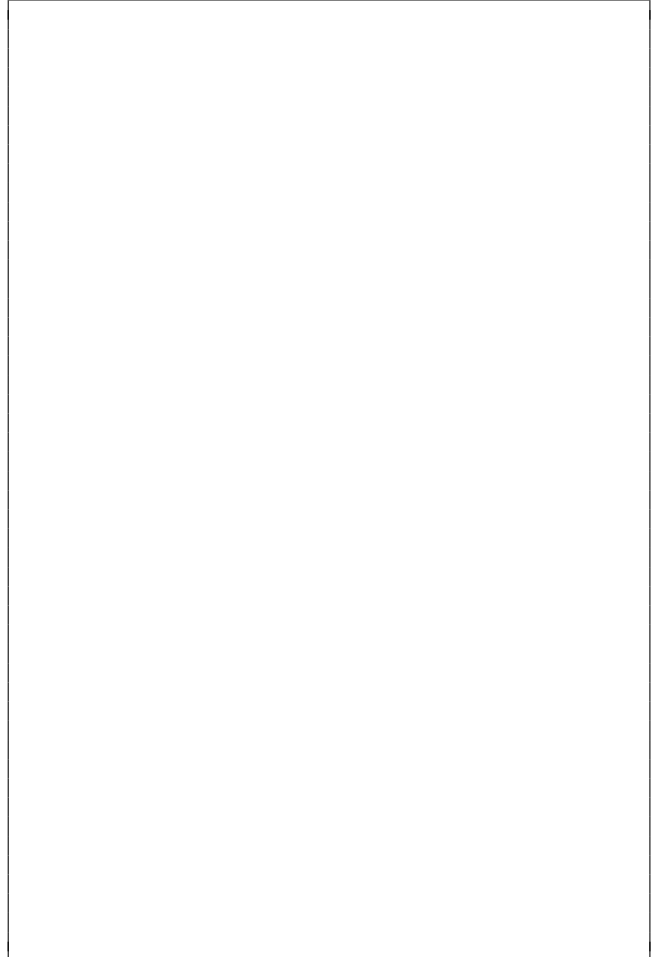
    a[1][1] = 15
    a[0].append(20)
    b[0].append(25)
    c[1].append(30)
    d[1].append(35)

    print("a = ", a)
    print("b = ", b)
    print("c = ", c)
    print("d = ", d)

    c[1] = c[1][:]
    d[1] = a[1]
    a[0][0] = 50
    b[0][1] = 60
    c[1][0] = 70
    d[1][1] = 80

    print("a = ", a)
    print("b = ", b)
    print("c = ", c)
    print("d = ", d)

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



2. (4 points) **Code Tracing:** Indicate what the following program prints. Place your answers (and nothing else) in the box below the code.

```
def ct2(s):
    m = []
    for i in range(s + 1):
        m.append([0] * s)

    cnt = 0
    for r in range(len(m)):
        for c in range(len(m[r])):
            m[r][c] = cnt
            cnt += 1

    for c in range(len(m[0])):
        for r in range(len(m)):
            print(f"{m[r][c]} ", end="")
        print()
```

ct2(2)

3. (8 points) **Free Response:** Color Sequences

Imagine a game that uses a 2D board of colored squares, where each square is either red, green, or blue. Whenever there are three or more squares of the same color in a vertical line, that line of colored squares is eliminated from the board.

Write *destructive* function `blankVerticalSequences(board)` which, given a board, finds any vertical sequences of length three or more and replaces all of the color codes with a space. The function is destructive, so it does not return anything.

Consider the following example:

```
board = [  
  ["B", "R", "R", "G"],  
  ["G", "R", "G", "G"],  
  ["G", "B", "R", "G"],  
  ["G", "B", "G", "R"],  
  ["R", "G", "R", "G"],  
  ["G", "R", "B", "B"],  
  ["R", "G", "B", "R"],  
  ["R", "B", "B", "G"],  
  ["R", "B", "B", "B"],  
  ["R", "R", "B", "G"],  
]  
blankVerticalSequences(board)
```

After this code runs, board contains:

```
[  
  ["B", "R", "R", " "],  
  [" ", "R", "G", " "],  
  [" ", "B", "R", " "],  
  [" ", "B", "G", "R"],  
  ["R", "G", "R", "G"],  
  ["G", "R", " ", "B"],  
  [" ", "G", " ", "R"],  
  [" ", "B", " ", "G"],  
  [" ", "B", " ", "B"],  
  [" ", "R", " ", "G"],  
]
```