

Name: \_\_\_\_\_ Section: \_\_\_\_\_ AndrewID: \_\_\_\_\_

### 15-112 Fall 2021 Quiz 5b

**\* Up to 25 minutes. \* No calculators, no notes, no books, no computers. \* Show your work!**

**\* No recursion**

**Code Tracing 1 [15pts]:** Indicate what the following code prints. Place your answers (and nothing else) in the box to the right of the code.

# Note: this prints 2 lines!

```
def ct1(M):
    rows, cols = len(M), len(M[0])
    result = []
    for row in range(rows):
        for col in range(1, cols):
            M[row][0] += M[row][col]
        result = [M[row][0]] + result
    return result

L = [ [ 2, 5, 7 ], [ 1, 2, 3 ] ]
print(ct1(L))
print(L)
```

**Code Tracing 2 [20pts]:** Indicate what the following code prints. Place your answers (and nothing else) in the box to the right of the code. Note: ct2 is called twice.

# Note: this prints 3 lines

```
import copy
def ct2(L, A, n):
    A[-1][0] += n
    A[n%2] += [10*n]
    L.append(n)
    print(A)
```

```
L = [[3], [4]]
C = copy.copy(L)
D = copy.deepcopy(L)
ct2(L, C, 1)
ct2(L, D, 2)
print(L)
```

### Free Response 1: insertRowCol [65 pts]

Write the **nondestructive** function `insertRowCol(L, row, col, val)` which takes a rectangular 2D list `L` and three integers and returns a new 2D list where one new row and one new column have been inserted at the locations specified by `row` and `col`. The cells in the new row and new column are all set to `val`, as shown below.

```
L = [[1, 2, 3, 4],  
     [2, 4, 6, 8],  
     [3, 7, 2, 0]]
```

```
assert(insertRowCol(L, 1, 2, 5) == [[1, 2, 5, 3, 4],  
                                   [5, 5, 5, 5, 5],  
                                   [2, 4, 5, 6, 8],  
                                   [3, 7, 5, 2, 0]])
```

```
assert(insertRowCol(L, 3, 0, 9) == [[9, 1, 2, 3, 4],  
                                   [9, 2, 4, 6, 8],  
                                   [9, 3, 7, 2, 0],  
                                   [9, 9, 9, 9, 9]])
```

```
assert(insertRowCol(L, 0, 4, 0) == [[0, 0, 0, 0, 0],  
                                   [1, 2, 3, 4, 0],  
                                   [2, 4, 6, 8, 0],  
                                   [3, 7, 2, 0, 0]])
```

You are guaranteed that `row` and `col` will be non-negative. Also, `row` will be less than or equal to the number of rows of `L`, and `col` is less than or equal to the number of columns of `L`.

**Hint:** How is this similar to the homework problem `nondestructiveRemoveRowCol`?

**(You can write your answer on the next page for more space.)**

(You may answer the FR on this page.)

**Bonus/Optional: Code Tracing [+2pts]**

Indicate what this prints. Place your answer (and nothing else) in the box.

```
import copy
def f(x): return 2*x
def g(x): return copy.copy(x)
def bonusCT(A):
    A[0] = (f, g)
    A.append([A[0][0](A[i]) for i in (1,2)])
    A[0] = A[0][1](A[1:])
    A[-1][-1] *= 10
    return A
print(bonusCT([1, 2, 3]))
```