

Quiz 6 FitB, CT, ROC (35 pts)

Time Limit: 12 mins

Name: _____ AndrewID: _____

This part does not contain any free responses. Do not run any code when completing this part of Quiz 6.

Time limit: 12 mins

Part 1 — Multiple Choice (4 pts total)

1. (2 pts) What is L after running the following code?

```
L = [[1]*2]*3  
L[0][0] = 5
```

- ☐ [[5, 1], [1, 1], [1, 1]]
- ☐ [[5, 1], [5, 1], [5, 1]]
- ☐ [[5, 1, 1], [1, 1, 1]]
- ☐ [[5, 1, 1], [5, 1, 1]]

2. (2 pts) What is L after running the following code?

```
L = [[1]*2 for _ in range(3)]  
L[0][0] = 5
```

- ☐ [[5, 1], [1, 1], [1, 1]]
- ☐ [[5, 1], [5, 1], [5, 1]]
- ☐ [[5, 1, 1], [1, 1, 1]]
- ☐ [[5, 1, 1], [5, 1, 1]]

Part 2 — Code Trace / Reasoning Over Code (31 pts total)

Code Trace 1 (6 pts)

What does the following code print?

```
# THIS IS A FILL IN THE BLANK QUESTION, NOT A CT
# Fill in the blanks in the wordSearchFromCellInDir function we wrote in lecture.

# Write your answer (and nothing else) in the box below in the format:
# blank1
# blank2
# blank3

'''
# Returns True if word can be found in board starting at
# (startRow, startCol), looking in direction (drow, dcol)
# Returns False otherwise
def wordSearchFromCellInDir(board, word, startRow, startCol, drow, dcol):
    rows, cols = len(board), len(board[0])
    currRow, currCol = startRow, startCol
    for c in word:
        if ((currRow < 0) or (currRow >= rows) or
            (currCol < 0) or (currCol >= cols) or
            (_____)): # blank 1
            return False
        _____ # blank 2
        _____ # blank 3
    return True

'''
```

Output:

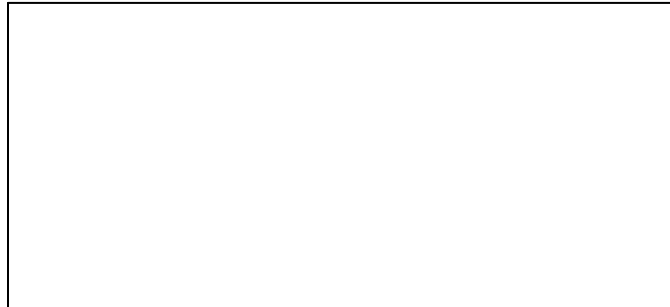
Code Trace 2 (15 pts)

What does the following code print?

```
import copy
def ct(L):
    M = copy.copy(L)
    N = copy.deepcopy(L)
    A = [L[0], N[0][1]]
    M[0].pop()
    L += [[9]]
    L = L * 2
    L[1][-1] += 10
    A.extend(N.pop())
    print(L)
    print(M)
    print(N)
    print(A)

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

Output:



Reasoning Over Code 1 (10 pts)

Find an argument for the function `roc(L)` that makes it return `True`. Write your answer (and nothing else) in the box below.

```
def roc(L):
    if len(L) != 3:
        return False
    m = 0
    for i in range(len(L)):
        A = [ L[i][j] for j in range(len(L[i])) ]
        if len(A) != i+1:
            return False
        if min(A) < m:
            return False
        m = max(A)
    return True
```

This page intentionally left blank for scratch work

This page intentionally left blank for scratch work

This page intentionally left blank for scratch work