1. **Code Tracing** [10 pts]: Indicate what these print or (for graphics) draw. Place your answers (and nothing else) in the boxes below the code.

```python
def ct1(L):
    a = L
    b = copy.copy(L)
    c = copy.deepcopy(L)
    b[0] = a[1] * a[1][0]
    a[0][0] += a.pop()[0]
    b[1] = c[0]
    return b
```
L = [[1],[2],[3]]
print(ct1(L))
print(L)

2. **Reasoning Over Code** [10 pts]:
Find an argument for the following function that makes it return True. Place your answers (and nothing else) in the boxes below the code:

```python
def rc1(n):
    assert((isinstance(n, int)) and (100 <= n <= 999))
    (n, r, c ) = (n//100, n//10%10, n%10)
    L = [ ([0] * c) for row in range(r) ]
    # note col is the outer loop
    for col in range(c):
        for row in range(r):
            L[row][col] = n
            n += 1
    return ((L[0][2] - L[0][0] == 10) and
            (sum([len(R) for R in L]) == 20) and
            (sum(L[0]) == 42))
```

```python
n =
```
3. **Fill in the Blank** [10 pts]:
Fill in the 3 blanks with the missing code from the case study in the notes.

```python
def wordSearchFromCellInDirection(board, word, startRow, startCol, drow, dcol):
    (rows, cols) = (len(board), len(board[0]))
    dirNames = ["up-left", "up", "up-right"],
                ["left", "", "right" ],
                ["down-left", "down", "down-right" ]
    for i in range(len(word)):
        row = ______________________________________________________
        col = ______________________________________________________
        if ((row < 0) or (row >= rows) or
             (col < 0) or (col >= cols) or
             __________________________________________________________):
            return None
    return (word, (startRow, startCol), dirNames[drow+1][dcol+1])
```

4. **Free Response #1: zeroRectCount(L) [35 pts]**

Background: given a 2d list of integers L, we will say that a rectangular region of L is a "zeroRect" (a coined term) if the sum of the values in that region equals 0. For example, consider this list:

```
L = [ [ 1, 2, -3, 5, 1 ],
     [ 3, -6, 4, 0, 1 ] ]
```

Here are the rectangular regions of L that sum to 0:

```
R1 = [ [ 1, 2, -3 ] ] # 1x3 in top-left of L
R2 = [ [ 1, 2 ], [ 3, -6 ] ] # 2x2 in top-left of L
R3 = [ [ 0 ] ] # 1x1 near bottom-right of L
```

With this in mind, write the function zeroRectCount(L) that takes a rectangular 2d list of integers L, and returns the total number of zeroRects in L. For example, with L as above, zeroRectCount(L) returns 3.

Hint: while you may solve this any way you wish, our sample solution used a large number of nested 'for' loops (so don't be discouraged if your solution does so as well).
This page is intentionally blank for your zeroRectCount solution.

Quiz continues on next page!
5. **Free Response #2: biggerDarkerDot** [35 pts]
Using our animation framework and assuming run() is already written, write init(data), keyPressed(event, data), and redrawAll(canvas, data) so the app works as such:

a. At first, a bright red dot of radius 20 is centered in the window (which may be any dimensions, as specified in the call to run).

b. Each time the user presses 'r', the radius increases by 5.

c. Each time the user presses 'd', the dot gets a bit darker red, unless the dot is already black, in which case it becomes bright red again. Note: you may assume rgbString(red, green, blue) from the graphics notes is also already written.


```python
def bonusCt1(n, b):
    while (b[-1]**0.5 < 1+2+3):
        n += 1; b = [sum([list(range(k)) for k in range(n)][i][:i-1]) for i in range(n)]
    return b[-2]
print(bonusCt1(10, [1]))

def bonusCt2(k, result=0):
    for m in range(2**k):
        while (m > 0): (m, result) = (m//2, 1+result+m%2)
    return result
print(bonusCt2(5,5))
```