Name: ______________________________ Recitation: ____ Andrew Id: ________________

15-112 Fall 2017 Quiz 5

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!
No recursion.

1. (20 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

```
import copy
def ct1(a):
    b = a
    c = copy.copy(a)
    d = copy.deepcopy(a)
    c[0] = [5, 6]
    b[0][1] = "two"
    c[1][0] = "twenty"
    c[0][0] = 9
    d[1] = [7, 8]
    c[1].insert(0, c[1].pop())
    print("a =", a)
    print("b =", b)
    print("c =", c)
    print("d =", d)

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

2. (30 points) **Free Response**: Write the function `transposeList(L)` that non-destructively transposes the rows and columns of 2D list L. It should return the transposed list. For example...

```
[ [1, 2],           [ [1, 4, 7],
  [4, 5],    ---->    [2, 5, 8] ]
  [7, 8] ]
```

You may assume L is not ragged and will not be empty.

QUIZ IS ON BOTH SIDES OF THE PAGE

3. (40 points) **Free Response**: `getClosestEnemy(board)`

   Background: This problem involves a simple game where a hero moves on a 2D board and is attacked by enemies. The board is represented as a 2D list of integers where 0 is empty, 1 is the hero, and other numbers are enemies. There is always exactly one hero on the board, but there can be multiple enemies. As the game progresses, the characters can move, but each time they move they can only step either up, down, left, or right. (They can't step diagonally in one move.) The distance from the hero to an enemy is measured as the number of steps it would take one to the move to the other.

   With this in mind, write the function `getClosestEnemy(board)` that takes a non-empty board as described above and determines the identity and distance (in steps) of the closest enemy. You should return a tuple of the form `(enemy number, distance)`. In the event of a tie, return the enemy with the larger enemy number.

   Consider this example:

```
[ [ 0, 10, 0, 0, 0, 0, 0  ]
  [ 0, 0,  0, 1, 0, 0, 0  ]
  [ 0, 0,  0, 0, 0, 0, 11 ] ]
```

   The hero is at (1,3), enemy 10 is at (0,1), and enemy 11 is at (2, 6). The closest enemy is enemy 10, with a distance of 3. So, the return should be (10, 3).

   Hint: You will probably want to write two helper functions. One to find the hero, and the other to calculate the distance between two different cells.