Name:_____ Section:___ Andrew Id: _____

15-110 Fall 2018 Quiz 5

*** 25 minutes**
*** No calculators, no notes, no books, no computers.**
*** Show your work when possible!**

1. **Code Tracing [10 pts]** Indicate what the following program prints. Place your answer in the box.

```
def ct1(L, i):
    L[i] += i
    L = L + [ ]
    L[i] += i
    return L

a = [5, 6]
print(ct1(a, 1))
print(a)
```

2. **Code Tracing [10 pts]** Indicate what the following program prints. Place your answer in the box.

```
def ct2(L):
    L = L + [ ]
    while (len(L) > 1):
        a = L.pop()
        b = L.pop()
        L.append(a+b+1)
    return L[0]

a = [2, 5, 3, 4]
print(ct2(a))
print(a)
```

3. **Code Tracing [10 pts]** Indicate what the following program prints. Place your answer in the box.

```
def ct3(s):
    M = s.split(' ')
    s = M[0] + M[2] # s is a string!
    M = s.split('a')
    return M[0] + M[2]

s = 'zab cdae fa2 bma3!'
print(ct3(s))
```

4. **Fill in the blanks [20 pts, 4 pts each]**

```
# monte_carlo_with_coins.py [from the course notes]

# Confirms that if you flip a coin 4 times, the odds
# of getting at least 2 heads is 11/16.

import random

def flipCoin():
    # Hint: this returns 'H' or 'T' with equal probability

    return ____________________________________________________


def flipCoins(times):
    result = [ ]
    for i in range(times):


        ____________________________________________________________
    return result


def trialSucceeds():
    flips = flipCoins(4)

    return ____________________________________________________


def oddsOfatLeastTwoHeadsInFourFlips(trials=100):
    successes = 0
    for trial in range(trials):

        if ____________________________________________________
            successes += 1

    return ____________________________________________________
```

5. **Free Response: goUp(steps) [25 points]**

Write the function goUp(steps) that takes a list of steps that are like those in upDownLeftRight from hw5, and only handles the steps that go up (ignoring left, right, and down). The function returns the resulting y value. Assume all letters are in lowercase, and assume all directions and integers are properly formatted (so you should ignore all illegal format issues). For example:

```
goUp(['left 5', 'up', 'right', 'up 7'])
```

The up steps are 'up' (which is the same as 'up 1') and 'up 7', so the function returns 8 here.

6. **Free Response: hasConsecutiveValues(L) [25 points]**
Write the function hasConsecutiveValues(L) that takes a list L and returns True if two consecutive values in L are equal, and False otherwise. Here are some test cases:

```
assert(hasConsecutiveValues([ ]) == False)
assert(hasConsecutiveValues([ 1, 2, 1 ]) == False)
assert(hasConsecutiveValues([ 1, 2, 2, 1 ]) == True)
assert(hasConsecutiveValues([ 1, 'b', 'b', True ]) == True)
```

7. **Bonus/Optional: Code Tracing [2.5 pts each]:**
Indicate what each of the following programs prints. Clearly circle your answers.

```
def bonusCt1():
    for i in range(1000):
        L = list(range(1,i+1))
        s = sum(L * (len(L) ** len(L))) // 1000
        if (s > 0): return (i, s)
print(bonusCt1())

def bonusCt2(L):
    try:
        L.append(L)
        L[0] += L[-1][1]
        L.pop().append(L[2][1])
        L[0] += L[-1][1]
    except: return [sum(L)]
a = [1,2,3]
print(bonusCt2(a) + a)
```