Name:_____ Section:___ Andrew Id: _____

15-112 Fall 2015 Quiz 9x (a)
*** Up to 40 minutes. No calculators, no notes, no books, no computers.**
*** To receive credit (in Code Tracing), show your work.**

1. **Code Tracing** [20 pts]:Indicate what this prints. Place your answer (and nothing else) in the box below the code.

```
def ct1(L, n=4):
    if (L == [ ]):
        return L
    else:
        return [L[0]**n] + ct1(L[1:], n-1)
print(ct1([2,3,4]))
```


```
def ct2(n):
    print(n, " ", end="") # don't miss this!
    if (n < 1):
        return 2+abs(n)
    else:
        return n + ct2(n-2) + ct2(n//2)
print(ct2(3))
```


2. **Reasoning Over Code [10 pts]:** For the following function, find the value of s (not m, which you may not set!) that will make the function return True. You only need one set of arguments, even if there are multiple correct answers. Show your work. Place your answer (and nothing else) in the box below the code.

```
def rc1(s, m=1):
    t = str(m**2)
    if (s == ""):
        return (m == 6)
    elif (not s.endswith(t)):
        return False
    else:
        return rc1(s[:len(s)-len(t)], m+1)
```

s =

3. **Free Response**: **Oopy Animation** [40 pts]

Note: another Free Response follows, on the last page, before the bonus!

Assuming our run() function is already written, write an animation where each time the user clicks in the canvas, even if that click is inside an existing square, an additional square is added, centered on the click, with a label set to the next available letter. So the first square is labeled "a", the next "b", and so on. After 26 squares are created, future clicks are simply ignored.

Also, when created, each square is assigned a randomly-chosen size, as small as 5x5 up to as large as 20x20.

Once created, squares sweep across the screen from left-to-right, and when they go completely offscreen on the right, they reappear on the left side and continue sweeping left-to-right.

Finally, if the user presses the letter corresponding to a square's label, that square increases its velocity by one, so it sweeps faster to the right, unless its velocity reaches 5x normal, in which case it resets to 1x normal.

To do this, you must create a Square class, where each square is an instance of the Square class. Your Square class must include at least a constructor, a moveRight method, and a draw method, all of which must be used properly by your event handlers and redrawAll function. You may add additional methods as you wish.

This page left blank for Oopy Animation.

Note: another Free Response follows, on the next page, before the bonus!

4. **Free Response**: **nthWithTwo3s(n)** [30 pts]
Without using iteration or strings, write the recursive function nthWithTwo3s(n) that takes a non-negative int n and returns the nth integer that contains exactly two 3's, so:
nthWithTwo3s(0) returns 33
nthWithTwo3s(1) returns 133
nthWithTwo3s(2) returns 233
nthWithTwo3s(3) returns 303
Note that you can add optional parameters if you wish.

5. **Bonus/Optional**: **Code Tracing** [2.5 pts each]: Indicate what these print. Circle your answer. Show your work!

```
def bonus1(f, n): return 1 if (n == 0) else f(n) + bonus1(g(f), n-1)
def g(f): return (lambda n: f(f(n)))
def f(n): return n+5
print(bonus1(f, 2))

def bonus2(x, y=0, z=None):
    (x, z) = (int(x), z or int(x))
    if (x+y == 0): return 0
    elif (y<0): return bonus2(x-1, z, z)
    else: return 1 + bonus2(x, y-1, z)
print(bonus2(math.pi))
```