Name:___________________________________ Section:___ Andrew Id: _____________________

15-112 Spring 2017 Quiz10a
*** Up to 35 minutes. No calculators, no notes, no books, no computers. * Show your work!**

1. **Free Response: A class** [35 pts]
Write the class A so that the following test function passes, and using OOP properly (so, for example, you may not hardcode any test cases).

```
a1 = A('fred')
assert((a1.name == 'fred')  and (str(a1) == '<name is fred>'))
a2 = A()
assert(str(a2) == '<name is wilma>')
a3 = a1.either(a2)
assert((a3.name == 'fred-or-wilma') and (isinstance(a3, A)))
assert((a1 == A('fred')) and (a1 != A('wilma')) and (a1 != 42)) # don't crash!
s = set([A('fred')])
assert(A('fred') in s)
```

2. **Free Response: onlyEvenDigits(L)** [35 pts]
   Without using iteration and without using strings, write the recursive function onlyEvenDigits(L), that you can abbreviate as oed(L), that takes a list L of non-negative integers (you may assume that), and returns a new list of the same numbers only without their odd digits (if that leaves no digits, then replace the number with 0). So:
   onlyEvenDigits([43, 23265, 17, 58344]) returns [4, 226, 0, 844]
   Also the function returns the empty list if the original list is empty. Remember to not use strings.

Name:___________________________________ Section:___ Andrew Id: _____________________

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

```
def ct1(L):
    if (L == [ ]):
        return L
    elif (L[-1] % 2 == 0):
        R = L[::-1]
        return [ 7, max(L[:2]) ] + ct1(R[:-2])
    else:
        return ct1(L[:-2]) + [ 8, sum(L[:2]) ]
print(ct1([1,3,4,6,2,5])) # prints a list with 6 ints
```

```
def ct2(n, m=0, depth=0):
    indent = '    '*depth
    print(indent, '(%d,%d)' % (n, m))
    if (n <= 0):
        result = int(str(n)+str(m))
    else:
        result = ct2(n//2, m+1, depth+1) + ct2(n-4, m+2, depth+1)
    print(indent, '-->', result)
    return result
ct2(3) # Note: you must correctly indicate newlines and indents
       # Hint: prints 10 lines
```

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

```
def rc(n):
    def g(n):
        if (n < 10): return True
        elif (n//10%10 != 1 + n%10): return False
        else: return g(n//10)
    def f(n, c=0, d=-1):
        if (n < 10): return ((c == 3) and (d == 6))
        else: return f(n//10, c+1, n%10)
    return f(n) and g(n)
```

n =

5. **Bonus/Optional: Code Tracing** [5 pts; 2.5 pts each] What will these print? Place your answer in the boxes.

```
def bonusCt1(n):
    def f(n): return 0 if (n <= 0) else 2*n-1+f(n-1)
    def g(n, m): return 0 if (m == 0) else f(n)+g(n,m-1)
    return (g(n, n) - f(n)) // f(n)
print(bonusCt1(42))
```

```
def bonusCt2(L):
    def R(f, L, r): return r if (L == []) else R(f, L[1:], f(r,L[0]))
    def M(f, A, B):
        if (A == [ ]): return [ ]
        else: return [f(A[0], B[0])] + M(f, A[1:], B[1:])
    f = lambda L: R(lambda x,y: x+[len(y)], L, [ ])
    g = lambda L: R(lambda x,y: x+[len(str(y))], L, [ ])
    h = lambda L: M(lambda x,y: x*10**y, g(L), f(L))
    return h(L)
print(bonusCt2([[42], "two"]))
```