Name: ____________________ Recitation: ______ Andrew Id: ____________

15-112 Fall 2017 Quiz 10

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!

1. (20 points) **Short Answer:** Briefly answer the following questions.

   (a) Can memoization be used to speed up the listFiles() function from the notes? Why or why not?

   (b) Describe the main difference between `*args` and `**kwargs`.

   (c) In words (not code) describe the functionality of the base case of `drawSierpinskyTriangle()` from the notes.

   (d) Describe the difference between a powerset and a set of permutations.

2. (25 points) **Code Tracing:** Indicate what the following program prints. Place your answers (and nothing else) in the box below the code.

```
def ct1(n, depth=0):
    print("Depth " + str(depth) + " input: " + str(n))
    result = None
    if n <= 2:
        result = 1
    elif n % 2 != 0:
        result = 1 + ct1(n - 1, depth+1)
    else:
        result = 2 + ct1(n // 2, depth+1)
    print("Depth " + str(depth) + " output: " + str(result))
    return result
ct1(6)
```

3. (15 points) **Code Tracing:** Indicate what the following program prints. Place your answers (and nothing else) in the box below the code.

Hint: Take careful note of the fact that `L` is an optional argument that initializes a list, which as mentioned in the notes is generally a bad idea.

```
def f(x, L=[]):
    for i in range(x):
        L.append(2*i)
    return L

def ct2(n):
    a = f(n)
    print(a)
    b = f(n//2)
    print(b)
    c = f(n//4,a)
    print(c)
    return a

print(ct2(4))
```

4. (40 points) **Free Response:** Write a recursive, backtracking function `pairsSum(a, sum)` that, given an list of positive integers `a`, splits the list into pairs of integers such that each pair sums to `sum`. The function should return a list of tuples containing the pairs, or `None` if there is no solution.

Example 1

```
a = [2, 3, 1, 4, 5, 3]
L = pairsSum(a, 6)
print(L)
```

Outputs: `[(2, 4), (1, 5), (3, 3)]`

Example 2

```
a = [2, 3, 1, 4, 5, 4]
L = pairsSum(a, 6)
print(L)
```

Outputs: `None`

A few notes:

- You may assume that `a` will always contain an even number of elements.
- If there is more than one way to split the list into pairs, you only need to return one of the correct possibilities.
- In order to receive points, you must use recursion and backtracking to solve this problem.
- You may write any helper or wrapper functions that you need.

Additional Space for Answer to Free Response Question