Name: ______________________________ Recitation: ____ Andrew Id: ________________

15-112 Spring 2019 Practice Quiz 9

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use backtracking on this quiz.

1. (25 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box to the right of the code.

```
def ct(a, b):
    print(a, b)
    if b == 1:
        print("done!")
        return a
    elif b % 2 == 0:
        print("split", end=" ")
        return ct(a, b//2) * ct(a, b//2)
    else:
        print("boring")
        return a * ct(a, b-1)

print(ct(2, 3))
```

2. (25 points) **Reasoning Over Code**: Find an argument for the following program (function roc) that makes it return True. Place your answer (and nothing else) in the box under the code.

```
def roc1Helper(L,z):
    if L == []:
        z[0] += 1
        return 0
    elif isinstance(L[0], list):
        return roc1Helper(L[0],z) + roc1Helper(L[1:], z)
    elif isintance(L[0], int):
        z[1] += 1
        return L[0] + roc1Helper(L[1:], z)

def roc(L):
    z = [0,0]
    n = roc1Helper(L,z)
    return n == 12 and z[0] == 3 and z[1] == 6
```

3. (50 points) **Free Response:** You are going on a trip and need to figure out what to bring with you. However, you have a lot of items you want to take but only one bag to put them all into. Your bag can only hold a certain amount of things before it breaks. Each item you wish to bring with you has an integer value (that represents how important this item is) and an integer weight that specifies how heavy the item is. Furthermore, your bag has a total weight - a maximum weight it can hold before it breaks. Your goal is to find the set of items you should pack such that their total value is maximized, but their total weight remains less than the max weight that the bag can support.

   With this in mind write the function packItems(items, maxWeight) that takes in a list of items and an integer max weight that your bag can support and returns a set of the names of the items that you should pack according to the description above. The list of items contains tuples with 3 elements - the name of the item, the value of the item and the weight of the item in that order.

   Here is an example:

```
items = [
("Food", 1, 8),
("Guitar", 100, 1),
("Toothbrush", 10, 4),
("Gum", 5, 3),
("Clothes", 8, 4)
]
maxWeight = 9
```

   With the above parameters your function should return {"Guitar", "Toothbrush", "Gum"} as these 3 items have a total weight of 8 which is less than 9, and they have the highest cumulative value.

   You may assume that a solution will always exist.

   You must use recursion to solve this problem to receive any credit

   Hint: Consider trying to include and exclude each item in your bag, and keeping track of the best bag you have found so far