Name: _____ Recitation: _____ Andrew Id: _____

15-112 Fall 2018 Quiz 10

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!

1. (10 points) **Short Answer:** Briefly, write at least one significant reason why you might want to write a wrapper function for a recursive problem.

2. (25 points) **Free Response:** Write the function `listFiles(path)` which takes an input string and returns a list of all files in that path or its subdirectories. For example, `listFiles('myDir')` might return:

   `['myDir/todo.txt', 'myDir/pets/cats.jpg', 'myDir/pets/dogs.jpg', 'myDir/CS112/hw10.py']`

   You can assume `os` has been imported.

3. (30 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box to the right of the code.

```python
def sortCT(L, depth=0):

    print(depth,":",L)

    if (len(L) < 2):
        result = L
    else:
        first = L[0]
        lo = []
        hi = []
        for x in L[1:]:
            if x<first:
                lo+=[x]
            else:
                hi+=[x]
        loNums=sortCT(lo, depth+1)
        hiNums=sortCT(hi, depth+1)
        result = loNums + [first] + hiNums

    return result

print("Result:", sortCT([2,7,6,9,3,0]))
```
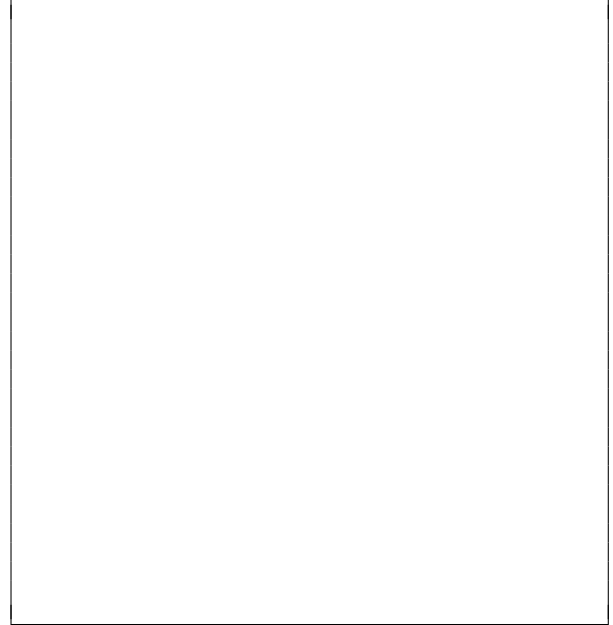
Answer box:

```
0 : [2, 7, 6, 9, 3, 0]
1 : [0]
1 : [7, 6, 9, 3]
2 : [6, 3]
3 : [3]
3 : []
2 : [9]
Result: [0, 2, 3, 6, 7, 9]
```

4. (35 points) **Free Response:** Write the function `getValidList(n)` which takes an integer n and returns a list of length n that meets the following requirements. If no valid list can be constructed, it returns None. A valid list has the following properties:

1. A list of length n contains every integer from 1 to n exactly once

2. Integers must alternate between even and odd. (i.e. no even numbers are next to each other, and no odd numbers are next to each other.)

3. The difference between two adjacent numbers must be greater than 1

According to these rules, the following results are valid:

- `getValidList(1)==[1]`
- `getValidList(4)==None`
- `getValidList(7)==[3, 6, 1, 4, 7, 2, 5]`
- `getValidList(8)==[1, 4, 7, 2, 5, 8, 3, 6]`

And these lists violate one or more rules:

- `[1, 4, 1, 4]` violates rule 1
- `[1, 3, 4, 2]` violates rule 2
- `[1, 2, 3, 4]` violates rule 3

This function must be written using recursion in a meaningful way!

Hint: Consider backtracking!

YOU MAY CONTINUE WRITING CODE ON THIS PAGE.