

Name: _____ Section: ____ Andrew Id: _____

15-112 Spring 2016 Quiz 7xa

*** Up to 45 minutes. No calculators, no notes, no books, no computers.**

*** No recursion! * 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(a):
    b = a
    (a, c, d) = (b + [b[0]], copy.copy(b), copy.deepcopy(b))
    d[1] = b[0]
    c[1] = d[0]
    b = b[0:1] + b[1:]
    b[0] = [3,4]
    a[0] += [5]
    c[1][0] = 1 + b[0][1]
    a[0][0] = 6 + d[1][0]
    for (s,L) in (("a",a), ("b",b), ("c",c), ("d",d)):
        print(s,L)
    a = [list(range(2-i)) for i in range(2)]
    print("start:", a)
    ct1(a)
    print("end:", a)
```

2. **Short Answer** [10 pts]

State and very briefly prove the worst-case big-oh runtime of mergesort. A well-labeled picture could suffice as a proof.

3. **Reasoning Over Code** [10 pts]:

Find arguments for the following function that make it return True. You only need one set of arguments for the function, even if there are multiple correct answers.

```
def rc1(n):
    a = [[(r+c) for c in range(n)] for r in range(n)]
    b = [a[i][i] for i in range(n)]
    return (sum(b) == 20)
```

n =

```
def rc2(d):
    (count, key) = (0, d["key"])
    while (key != "key"):
        (count, key) = (1+count, d[key])
    return (count == 3)
```

d =

4. **Big-Oh** [10 pts]

State the worst-case big-oh runtime of each of the following functions. Circle your answers.

```
def bigOh1(L):
    # L is a list of integers all between 0 and 2**32
    N = len(L)
    d = dict()
    a = sorted(L)
    for i in range(len(a)):
        for j in range(i+1, len(a)):
            k = j + d.get(i,5)
            if (k not in a):
                d[a[j]] = i + j + k
    return d
```

```
def bigOh2(n):
    # n is an integer
    N = math.log(n, 2)
    a = [ ]
    m = len(str(n))
    for i in range(2, m):
        count = j = 1
        while (j < m):
            (count, j) = (1+count, i*j)
        a.append((i, count))
    return len(set(a))
```

Name: _____ Section: ____ Andrew Id: _____

5. **Free Response: Random Number Grid** [50 pts]

Assuming the `run()` function is already written for you, write `init`, `keyPressed`, `mousePressed`, and `redrawAll` so that when the animation is first run, a 5x10 grid (with visible gridlines) nearly fills the window, with each cell containing a random digit (between 0 and 9, inclusive) drawn in its center. Each time the user presses a digit key, a yellow circle fills the cell behind each of the matching digits (and only those digits) in the grid. If the user presses the mouse inside any yellow circle, all the yellow circles disappear (until the next time the user presses a digit key). Hint: you might find `random.randint(lo,hi)` helpful here.

[quiz continues on next page]

6. **Bonus/Optional: Code Tracing** [5 pts]

Indicate what this prints. Place your answer (and nothing else) in the box below the code):

```
def bonusCt(L=range(5),k=2):  
    def q(r,k=1): return r*k if (type(r)!=list) else sum(q(s,k) for s in r)  
    return q([q(v) for v in [list(range(i,2*i)) for i in L]],k)  
print(bonusCt())
```