Name: ____________________________ Section: _____ Andrew Id : ______________________

15-112 Summer-1 2017 Quiz 4
***Up to 50 minutes. No calculator, no notes, no books, no computers. Show your work!**
No recursion!

1. **Code Tracing [10 pts]:** Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
import copy
def ct1(a):
    s = set(a)
    t = copy.copy(s)
    for val in  s:
        if  (val % 3  !=  0): t.remove(val)
    print(sorted(s))
    print(sorted(s.difference(t)))

ct1([(n**2)%10  for n in range(3,8)])
```

2. **Reasoning over code [10 pt]:** Find an argument for the following function that makes it return True. Place your answers (and nothing else) in the boxes below the code.

```
def rc1(L):
    (rows, cols) = (len(L), len(L[0]))
    d = {0 : 0 }
    for row in  range(rows):
        for col in  range(cols):
            key = L[row][col]
            if  (key == 0):
                d[0] += 1
            else:
                d[key] = 10*row + col
    return  (d ==  {0:5, 42:12, 13:3})
```

L =

3. **Big-Oh and better Big-Oh [25 pts]**

Consider the following function:

```
def f(L):
    N = len(L)
    a = []
    for i in range(N):
        a += [i]*L.count(i)
    return a
```

A. Write in just a few words what this function does in general (for a list of N integers, what does it return). Be very brief.

B. Write what the Big-Oh of this function is?

C. Write the python function fasterF(L) that works identically to f(L) but runs in a faster function family. For full credit, your solution must run in O(N), where L has N elements in it, though partial credit will be given for slower solutions.

4. **Free Response - getCourseAverage [25 pts]:**

Consider this dictionary:

```
grades =  {
               "fred":  {"cs1": 75, "bio": 93}
               "wilma": {"physics1": 89}
               "dino":  {"cs1": 92, "physic1": 64}
              }
```

For each student, it shows the courses that each has taken and their final average in that course. So, for example "fred" has taken "cs1" and "bio", and has gotten a 75 in "cs1" and a 93 in "bio".

Given this dictionary, we are now interested at the course averages.

With this in mind, write a function getCourseAverage(grades), which takes a grades dictionary as just described and returns another dictionary mapping each of the courses that appear in the grades dictionary to the course averages.

So, for the example above, the following dictionary would be returned:

```
{"cs1": 85.5, "physics1": 76.5, "bio": 93}
```

(here, for example, the course average for "cs1" was computed by taking the average of fred's and dino's score (75 + 92)/2 = 85.5)

5. **Free Response - movingDot [30 pts]:**

Assuming the run() function is already written, write the functions init, mousePressed, keyPressed, timerFired, and redrawAll so that when run(400,400) is called, the following animation runs:

-The animation starts with a ball bouncing back-and-forth horizontally across the middle of the 400x400 canvas.
-A score of 0 is displayed in top center section of the canvas.
-Each time the user presses the mouse in the moving ball, the score increases by 1 and the ball moves faster.
-If the user misses the ball, the screen displays "Game Over" in the middle, and stays that way without any other movement or responding to any other events, except when the user presses "r", at which point the game resets to a score of 0 and play starts over.