Name:_____ Section:___ Andrew Id: _____

15-112 Spring 2017 Quiz 3a

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

*** No lists, or recursion**

1. **Code Tracing** [20 pts]:Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
def ct1(s):
  print(chr(ord('G') + ord(s[1]) - ord(s[0])), end='')
  t, count = '', 0
  for c in s:
    if (not c.isalnum()): t += c
    if (c.isdigit()): print(c, end='')
    elif (c.isupper()): print(c.lower(), end='')
    else: count += 1
  return ('\tt=%s\t%d' % (t, count))
print(ct1('ae1#B2cD!'))
```

```
def ct2(s):
  r = t = ''
  for i in range(len(s)):
    if (s[i] in s[i+1:]): r += str(i)
    else: t += s[i]
  return r + t[::-1]
print(ct2('aebacab'))
```

2. **Reasoning Over Code** [10 pts]:
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(s):
  if (not isinstance(s, str)): return False
  t = string.ascii_uppercase
  return ((s[3::3] == t[0:2]) and
          (s[-1::-2] == t[:5]) and
          (s.count(s[0]) > 3))
```

```
s =
```

3. **Short Answers** [10 pts]:
Unlike the rest of this quiz, the questions in this section (and just this section) cover check4 (Graphics and 1d Lists). Answer each of the following in **just a few brief words** or a line or two of code, as appropriate.

   a. In just a few words, where on the canvas is the origin (0,0) in Tkinter graphics?

   b. Assuming canvas already exists, write one line of code that draws a circle in the canvas, centered at (50, 100), with radius 20.

   c. Assuming L is a list, what does L[:] evaluate to?

   d. In just a few words, given a list L, what is the difference between L.sort() and sorted(L).

   e. In just a few words, what is the difference between tuples and lists?

Name:_____ Section:___ Andrew Id: _____

4. **Free Response: encode(s, pwd)** [50 pts]

Here we consider a simple way to encode an all-lowercase string s, using a positive integer pwd as the password. Let's start with an example: if s is 'abyzc' and pwd is 234, we repeat the digits of the password as necessary so each character in s has a digit beneath it, like so:

a	b	y	z	c
2	3	4	2	3

Next, to find the encoded string, we offset each character by the integer below it, wrapping around as necessary. That is:

a	b	y	z	c
2	3	4	2	3
c	e	c	b	f

Thus, we see that encode('abyzc', 234) returns 'cecbf'.

With this in mind, write the function encode(s, pwd) so that it works as described

5. **Bonus/Optional: Code Tracing** [5 pts] Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
def bonusCt1(s, t=''):
  while len(s)>0: (s, t) = (s[2:-2:2][::-1], t+s[-1])
  return t
print(bonusCt1(string.ascii_lowercase))
```

```
def bonusCt2(s):
  s = (s[1::3]*3)[::-1][1::5]
  return ''.join([chr(ord('G')+ord(c)-ord('b')) for c in s])
print(bonusCt2("Carpe diem!"))
```