Name:___________________________________ Section:___ Andrew Id: ____________________

15-112 Fall 2015 Quiz 4x (a)
*** Up to 25 minutes. No calculators, no notes, no books, no computers.**
*** No lists, or recursion! * To receive credit (in Code Tracing), show your work.**

1. **Code Tracing** [30 pts]: Indicate what this prints. Place your answer (and nothing else) in the box below the code.

```
def ct1(s, n):
    result = ""
    d = 0
    while (n > 0) and (len(s) > 0):
        if (s[-1].isdigit()):
            result += str((n%10)%int(s[-1]))
        else:
            result += chr(ord('D') + d)
        n //= 10
        s = s[1:-1]
        d += 1
    return result
print(ct1("abc3c3", 2468))
```

```
x = 1
def f(x):
    x += 2
    return x
def g(y):
    global x
    x *= 2
    return x + y
def ct2(n):
    result = ""
    for i in range(n):
        result += str(g(f(i))) + "."
    return result
print(ct2(3))
```

2. **Reasoning Over Code [20 pts]:** For each of the following functions, find values of the arguments that will make the function return True. You only need one set of arguments for each function, even if there are multiple correct answers. Show your work. Place your answer (and nothing else) in the box below the code.

```
def rc1(n):
    assert(type(n) == int)
    s = str(n)
    return ((2000 > n > 1000) and (n == int(s[0] * len(s))))
```

```
n =
```

```
import string
def rc2(s):
    if ((s.count("c") != 1) or (s.count("e") != 1)): return False
    result = ""
    while (len(s) > 0):
        c = s[0]
        for d in s:
            c = min(c, d) if (len(result)%2 == 0) else max(c,d)
        result += c
        s = s[1:]
    return (result == "cecc")
```

```
s =
```

3. **Free Response**: **nthNearlySquarish(n)** [50 pts]
We will say that a positive integer is squarish (a coined term) if its non-zero digits form a perfect square. For example, 100201 is squarish because 121 is 11**2. Further, we will say that an integer is nearly-squarish if it is within 10, inclusive, of a squarish number. So every integer from 399 to 419 inclusive is nearlySquarish because all are within 10 of 409, which is squarish because 49 is 7**2. With this in mind, and without using strings, write the function nthNearlySquarish(n) that takes a non-negative int n and returns the nth nearly-squarish number.

Note: you may not use strings on this problem.

4. **Bonus/Optional:** [5 pts]
Code Tracing: Indicate what this prints. Place your answer (and nothing else) in the box below the code.

```
import string
def bonusCT(s):
    for i in range(len(s)-1, -1, -1):
        if (s[i] < s[i-1]): return(s[i-1]+s[i])
print(bonusCT(string.ascii_letters))
```

5. **Bonus/Optional:** [5 pts]
Reasoning Over Code : For the following function, find values of the argument that will make the function return True. You only need one set of arguments,, even if there are multiple correct answers. Show your work. Place your answer (and nothing else) in the box below the code.

```
def bonusRC(n):
    c = d = 0
    for a in range(n):
        for b in range(n**2): c += n
    for a in range(c): d += c
    return (d == 4**4)
```

n =