

Name: _____ Section: ____ Andrew Id: _____

15-112 Fall 2016 Quiz 1a

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

* No strings, lists, loops, or recursion

1. **Code Tracing** [25 pts]: Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
def ct1(x, y, z):
    print(x ** x - y / x + y * x - z)
    y += max(x,y,z) // min(x,y,z)
    z %= x
    return 100*x + 10*y + z
print(ct1(2, 3, 5)) # hint: prints 2 values
```

```
def f(x): return 2*x+1
def g(x): return f(x//2)
def h(x):
    if (x%2 == 0): return f(x+g(x))
    else: return f(x)-g(x)
def ct2(x):
    print(h(x))
    print(h(x+1))
print(ct2(5)) # hint: prints 3 values
```

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(n):
    if ((not isinstance(n, int)) or (n < 100) or (n > 999)): return False
    a = n%10
    b = n//10
    return (almostEqual(b**0.5, a) and (a + b == 42))
```

n =

3. **Short Answers** [10 pts]

- a. Give a value of x such that `roundHalfEven(x) == roundHalfUp(x)` and a value of y such that `roundHalfEven(y) != roundHalfUp(y)`.

$x =$ _____ $y =$ _____

- b. Assuming x and y are both positive ints, for any given value of y , what is the largest possible value of $(x \% y)$?

- c. If a function would produce a syntax error, and a runtime error, and a logical error, Python will only report one of them. Which one, and very briefly, why?

- d. Which Python arithmetic operator associates right-to-left?

4. **Debugging** [5 pts]

Recall that the `eggCartons` function from `check1` takes a non-negative integer number of eggs, and returns the smallest integer number of cartons required to hold that many eggs, where a carton may hold up to 12 eggs. The following code is a nearly-correct implementation of this function, but it contains one bug -- one operator is incorrect. First, circle the incorrect operator, then write and circle the correct operator that belongs in its place.

```
def eggCartons(eggs):
```

```
    cartons = eggs / 12
```

```
    if (eggs % 12 == 0): return cartons
```

```
    else: return cartons + 1
```

Name: _____ Section: ____ Andrew Id: _____

5. **Free Response 1: setKthDigit(n, k, d)** [25 pts]

[quoted from lab1:]

Write the function `setKthDigit(n, k, d)` that takes three integers -- `n`, `k`, and `d` -- where `n` is a possibly-negative int, `k` is a non-negative int, and `d` is a non-negative single digit (between 0 and 9 inclusive), and returns the number `n` but with the `k`th digit replaced with `d`. Counting starts at 0 and goes right-to-left, so the 0th digit is the rightmost digit.

For example:

`setKthDigit(468, 0, 1)` returns 461

`setKthDigit(468, 1, 1)` returns 418

`setKthDigit(468, 2, 1)` returns 168

`setKthDigit(468, 3, 1)` returns 1468

Quiz continues on next page!

6. **Free Response 2: isAlmostSquare(n)** [25 pts]

Write the function `isAlmostSquare(n)` that takes any Python value and returns `True` if it is an int within 2 (inclusive) of a perfect square, and `False` otherwise. For example, since 25 is a perfect square (5^2), the function returns `False` if `n` is 22, `True` if `n` is 23, 24, 25, 26 or 27, and `False` if `n` is 28. It also returns `False` (without crashing) if `n` is 25.0 or "25", as those are not ints.

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

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

```
def bonusCt1(n):
    def f(n): return 1 - 1/n
    return max(n, f(n), f(f(n)), f(f(f(n))), f(f(f(f(n)))),
              f(f(f(f(f(n))))), f(f(f(f(f(f(n)))))))
print(0.5 + bonusCt1(0.5))
```

```
def bonusCt2(x, y):
    def f(x): return 1 if (x < 2) else f(x-1)+f(x-2)
    return x if (f(x)>y) else bonusCt2(x+1, y)
print(bonusCt2(0, 40))
```