Name: _____ Recitation: _____ Andrew Id: _____

15-112 Spring 2019 Quiz 1

Up to 15 minutes. No calculators, no notes, no books, no computers, **no extra paper.** Show your work!
Do not use string indexing, loops, lists, dictionaries, try/except, or recursion on this quiz.
You may import the math library and use its functions.

1. (30 points) **Free Response:** You are in charge of buying pizzas for a group of hungry TAs. Write the function `numPizzas(tas, slicesEach)` which returns the number of pizzas you should buy, where `tas` is the number of TAs you must feed, and `slicesEach` is the number of slices each TA will eat. Assume all pizzas have 8 slices. **You cannot buy fractional pizzas, so make sure you return an integer value!** For example, `numPizzas(7,3)` should return 3, because 3 pizzas give the needed 21 slices plus a few extra.

2. (25 points) **Free Response:** Write the function `rectPeg(r,w,h)`, which returns True if a rectangular peg with width w and height h can pass through a round hole with radius r, and False otherwise. (This function is identical to `rectPegRoundHole(r,w,h)` from lab1!)

   Assume this is a 2D problem (i.e. assume the pegs and holes are infinitely long, and don't check if 3D rotations can cram a short peg through a thin slot, etc.).

   For example, `rectPeg(1,5,6)` should return `False` since a rectangle with side lengths of 5 and 6 cannot fit inside a circle with radius 1, and `rectPeg(10,2,2)` should return `True` since a rectangle with side lengths of 2 can easily fit inside a circle with radius 10.

3. (10 points) **Free Response:** This function will crash when called with certain integer inputs, like `buggy(6,-3)` or `buggy(-14,7)`. Modify the **one** line inside the function so that it will return `False` when it would otherwise crash. **You may not add any additional lines.**

   The function should not crash on any integer inputs. You are guaranteed that the inputs will be integers. **The function's behavior should remain the same for non-crashing inputs** (i.e. `buggy(-1,1)` should return `False` and `buggy(5,2)` should return `True`).

```
def buggy(x,y):
    return (100/(x+2*y)) < 50
```

4. (20 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box to the right of the code.

```
def foo(x):
    x + 1
    return x - 1

def bar(y):
    a = y % 3
    print("a:",a)
    a += 1
    return a

def ct(x):
    print(x + 3)
    y=foo(bar(x) + foo(12))
    print("y:",y)
    print(2*bar(y))
    return y

print("Go!")
ct(5)
```

5. (15 points) **Reasoning Over Code**: Find an input value for `x` that makes `roc(x)` return `True`. Place your answer (and nothing else) in the box to the right of the code.

```
def almostEqual(x, y):
    return abs(x - y) < 10**-9

def roc(x):
    y = 2 * x
    if not x // 5 == 5:
        y = 0
    elif x % 2 == 0:
        if x % 3 == 2:
            y += x
        x = x / y
    else:
        return False
    if almostEqual(x, 1/3):
        return True
```