

Name: _____ Section: _____ AndrewID: _____

15-112 Fall 2021 Quiz 1a

*** Up to 20 minutes. No calculators, no notes, no books, no computers. * Show your work!**

*** No strings, lists, string or list indexing, loops, or recursion**

Code Tracing 1 [20pts]: Indicate what the following code prints. Place your answers (and nothing else) in the box to the right of the code.

```
import math
def ct1(x):
    if (type(x) != int):
        x = 2 * x
        return math.ceil(2 * float(str(x)))
    x = abs(x)
    if (x < 0):
        return '0'
    elif (x > 10):
        return x/10
print(ct1(-45))
print(ct1(6))
print(ct1(7.8))
print(ct1('4'))
```

Code Tracing 2 [20pts]: Indicate what the following code prints. Place your answers (and nothing else) in the box to the right of the code.

```
def f(x):
    print(f'f({x})') # Don't miss this line
    return 10*x if (x < 25) else 10+x
def ct2(x):
    if (x % 5 < x % 10):
        x += f(x + f(x))
    return f(x)
print(ct2(15) + ct2(20))
# Hint: this prints 5 lines
```

Free Response 1: isSmallFair(n) [60 pts]

We will say that a value n is "fair" if it is an integer and it has the same number of even digits as odd digits (ignoring leading 0's). A "small fair" number is a fair number with exactly 4 digits.

For example, 1083, 1081, and -1092 are each small fair numbers because each have two odds and two evens.

With this in mind, and without using strings or loops, write the function `isSmallFair(n)` that takes a value n , that may or may not be an integer, and returns `True` if n is a small fair number, and `False` otherwise. Do not crash if n is not an integer! **Again, do not use strings or loops here.**

```
def testIsSmallFair():
    assert(isSmallFair(1001) == True)
    assert(isSmallFair(-1083) == True)
    assert(isSmallFair(1112) == False)
    assert(isSmallFair(83) == False)
    assert(isSmallFair('axolotl') == False)
```

Bonus/Optional: Code Tracing [+2.5pts]

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

```
def f(g, h, x): return g(h(x)) + h(g(x))
def g(h): return h**2 + h
def h(g): return g**3 - g + 1
def bonusCt():
    print(f(g,h,1) + f(h,g,2))
bonusCt()
```