fullName:_____________________________andrewID:_____________________________ recitationLetter:_____

15-112 F23

Quiz1 version A (25 min)

You **MUST** stop writing and hand in this **entire** quiz when instructed in lecture.

- You may not unstaple any pages.
- Failure to hand in an intact quiz will be considered cheating. Discussing the quiz with anyone in any way, even briefly, is cheating. (You may discuss it only once the quiz has been posted to the course website.)
- You may not use your own scrap paper. If you must use additional scrap paper, raise your hand and we will provide some. You must hand any scrap paper in with your paper quiz, and we will not grade it.
- You may not ask questions during the quiz, except for English-language clarifications. If you are unsure how to interpret a problem, take your best guess.
- You may not use any concepts (including builtin functions) which we have not covered in the notes in week 1 / unit 1.
- You may not use strings, loops, lists, indexing, tuples, dictionaries, sets, or recursion.
- We may test your code using additional test cases. Do not hardcode.
- Assume almostEqual(x, y) and rounded(n) are both supplied for you. You must write all other helper functions you wish to use, unless we specify otherwise.

True or False [3pts ea]

Mark each of the following statements as True or False.

1. ◯ True ◯ False

I understand that it is an Academic Integrity Violation to discuss anything at all about this quiz with anyone other than current 112 TAs and faculty, regardless of which lecture they attend, until the quiz is posted on the course website. Hint: We hope your answer is True

2. ◯ True ◯ False

The following line prints True:

```
print(type(3) == type("3"))
```

3. ◯ True ◯ False

The following line prints True:

```
print(str(3) == "3")
```

4. ◯ True ◯ False

The following code will crash:

```
x = 10
y = 0
print((y>0) and (x/y==0))
```

5. ◯ True ◯ False

The following line prints True:

```
print(isinstance(4.2, float))
```

CT1: Code Tracing [15pts]

Indicate what the following code prints. Place your answers (and nothing else) in the box below.

```
import math
def ct1(n):
    n = n / 10
    print(n)
    m = int(n)
    print(m)
    q = 3 + 2 * m
    print(q)
    q -= math.ceil(n) % 10
    return q

n = 368
print(ct1(n))
print(n)
```

CT2: Code Tracing [15pts]

Indicate what the following code prints. Place your answers (and nothing else) in the box below.

```python
def f(x, y):
    y += 10
    print(y)
    return y // x

def ct2(x, y):
    x = f(x, y) + 2
    print(x)
    y = 2 * f(y, x)
    print(y)

print(ct2(2, 3))
```

Free Response 1: largestInt(x0, x1) [25pts]

Write the function `largestInt(x0, x1)` that takes two values x0 and x1, which can be integers, strings, or floats, and returns the following:

- If both x0 and x1 are integers, `largestInt(x0, x1)` returns the larger integer
- If only one value is an integer, `largestInt(x0, x1)` returns that integer
- If neither x0 nor x1 are integers, `largestInt(x0, x1)` returns None

The function should not crash. Carefully read the test cases below.

```
assert(largestInt(5, 1000) == 1000)
assert(largestInt(1000, 5) == 1000)
assert(largestInt(5, 5) == 5)
assert(largestInt(-15, 10) == 10)
assert(largestInt(-15, "abc") == -15)    #"abc" is a string
assert(largestInt("123", 12) == 12)      #"123" is a string
assert(largestInt(10, 100.0) == 10)      #100.0 is a float
assert(largestInt("abc", 100.5) == None) #Neither are ints
```

Begin your FR1 answer here or on the following page

You may begin or continue your FR1 answer here, if you wish

Free Response 2: isFactorish(n) [30pts]

We will define an integer n to be "factorish" if each of the following are True:

- n has exactly 3 digits
- none of the digits of n are zero
- each digit in n is a factor of the number n itself

Write the function `isFactorish(n)` that takes an integer n (which may be negative) and returns True if n is factorish, and returns False if n is not factorish. Read the test cases carefully.

```
assert(isFactorish(412) == True) # 4, 1, and 2 are all factors of 412
assert(isFactorish(-412) == True) # Must work for negative numbers!
assert(isFactorish(-222) == True)
assert(isFactorish(112) == True)
assert(isFactorish(4128) == False) # 4128 has more than 3 digits
assert(isFactorish(420) == False) # 420 has a 0 (no 0's allowed)
assert(isFactorish(42) == False) # 42 has less than 3 digits
```

Remember, do not use strings, loops, lists, or anything else prohibited on page 1 of this quiz.

Begin your FR2 answer here or on the following page

You may begin or continue your FR2 answer here, if you wish