fullName:____________________________andrewID:____________________________
recitationLetter:_____

15-112 F22

Quiz2 version B

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) we have not covered in the notes this semester.
- You may not use strings, lists, indexing, tuples, dictionaries, sets, or recursion.
- We may test your code using additional test cases. Do not hardcode.
- Assume almostEqual(x, y) and roundHalfUp(n) are both supplied for you. You must write all other helper functions you wish to use.

CT1: Code Tracing [12pts]

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

```
def ct1(n):
    z = 3
    for x in range(2, n, 2):
        z = 10*z + x
        for y in range(n, 2*x):
            z = 10*z + y
    return z

print(ct1(6))
```

CT2: Code Tracing [12pts]

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

```
def ct2(a, b, c):
    while c > a:
        c -= b
        if c > a + b :
            print(f'X {c}')
        elif c == 15:
            print(f'Y {b}')
            continue
        b += 1
    return c
print(f'Z {ct2(10, 7, 30)}')
```

Free Response 1: isRarePrime(n) [46 points]

We will say an integer is a "rare prime" (a coined term) if it is a prime number and none of its **non-zero** digits appear more than once.

Here are some assorted examples of rare primes:

```
2, 5, 47, 89, 103, 761, 2897, 3001
```

And these values are not rare primes (because they are not prime):

```
0, 1, -761, 187, 4
```

And these values are not rare primes (because a non-zero digit appears more than once):

```
11, 199, 797, 1303
```

With this in mind, and without using strings, lists, or other prohibited concepts or functions which are not in the week 1 or week 2 notes, write the function isRarePrime(n) that takes any integer and returns True if n is a rare prime, and False otherwise.
Note: It's a good idea to write helper functions for this problem!

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

Free Response 2: nthRarePrime(n) [30 points]

Using the definition of "rare primes" from FR1, write the function nthRarePrime(n) that takes a non-negative integer and returns the nth rare prime. For this problem, assume isRarePrime(n) is written properly, given the definition in FR1 (i.e. you do not need to rewrite isRarePrime(n) for this question, and you may simply call it and assume it is correct.)

The 0th rare prime is 2. The beginning of the sequence of rare primes is as follows:

```
2, 3, 5, 7, 13, 17, 19, 23, 29, 31, 37...
```

Do not use strings, lists, or other prohibited concepts or functions which are not in the week 1 or week 2 notes.

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

bonusCT: Code Tracing [2pts bonus]

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

```
def bonusCt1(n):
    (a,b,c) = (0, 1000, 100)
    while c < 1000:
        for x in range(a, b, c):
            (a,b,c) = (a+1, b-1, c+50)
    return a - n
print(bonusCt1(2))
```