fullName:____________________________andrewID:____________________________ recitationLetter:_____

15-112 N23

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 use any concepts (including builtin functions) which we have not covered in the notes in week 1.
- 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 rounded(n) are both supplied for you. You must write all other helper functions you wish to use, unless we specify otherwise.

Multiple Choice [2pts ea]

MC1. What is the output of the following code? Select the best answer (fill in one circle).

```
a = 3
b = (a!=3)
print(b)
```

- ○ True
- ○ False
- ○ 3
- ○ It will crash with a syntax error
- ○ It will crash with a runtime error

MC2. What is the output of the following code? Select the best answer (fill in one circle).

```
a = None
b = a and True
print(b)
```

- ○ True
- ○ False
- ○ None
- ○ It will crash with a syntax error
- ○ It will crash with a runtime error

MC3. What is the output of the following code? Select the best answer (fill in one circle).

```
a = 0
b = a or a/a
print(b)
```

◯ True

◯ False

◯ 0

◯ It will crash with a syntax error

◯ It will crash with a runtime error

MC4. What is the output of the following code? Select the best answer (fill in one circle).

```
a = False
b = a and a/a
print(b)
```

◯ True

◯ False

◯ 0

◯ It will crash with a syntax error

◯ It will crash with a runtime error

MC5. Is the following an infinite loop? Select the best answer (fill in one circle).

```
a=1
b=0
while a=1:
    b+=a
print(b)
```

◯ Yes

◯ No, it will print 1

◯ No, it will print None

◯ No, it will crash with a syntax error

◯ No, it will crash with a runtime error

CT1: Code Tracing [10pts]

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

```
def ct1(n):
    while n < 5:
        n += 1
        print(n)
    return n == 6
    print(n)

n=2
print(ct1(n))
print(n)
```

CT2: Code Tracing [10pts]

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

```
def ct2(n):
    x, y = 0, 0
    for i in range(n):
        if i % 2 == 1:
            x += 1
        elif i == 4:
            break
        else:
            y += 2
            continue
        print(x, y)
    print(x + y)
print(ct2(6))
```

Free Response 1: isPalindrome(n) [30 points]

Write the function isPalindrome(n) that takes a non-negative integer and returns True if n is a palindrome and False otherwise.

A palindrome is any number which is the same when its digits are reversed, such as: 5, 18481, 323, and 88

Here are some test cases:

```
assert(isPalindrome(0) == True)
assert(isPalindrome(1) == True)
assert(isPalindrome(7) == True)
assert(isPalindrome(11) == True)
assert(isPalindrome(121) == True)
assert(isPalindrome(333) == True)
assert(isPalindrome(2147447412) == True)
assert(isPalindrome(2147483647) == False)
assert(isPalindrome(2142) == False)
assert(isPalindrome(1234567890) == False)
```

Begin your FR1 answer here or on the following page

Begin or continue your FR1 answer here

Free Response 2: nthPalindromicComposite(n) [40 points]

Write the function nthPalindromicComposite(n) that takes a non-negative integer and returns the nth palindromic composite.

A composite number is a positive integer greater than 1 that can be formed by multiplying two smaller integers. In other words, it is a number that has at least one divisor other than 1 and itself. If a number is greater than 1 and not prime, then it is composite. Some composite numbers include 4, 6, 9, 20, and 102.

The 0th palindromic composite is 4. The beginning of the sequence of palindromic composites is as follows: 4, 6, 8, 9, 22, 33, 44, 55, 66, 77, 88, 99, 111, 121, 141...

You must write isComposite(n) as a helper function, which takes an integer n and returns True if n is composite and False otherwise. (You may write other helper functions if you wish, but it is not required.)

You should assume isPalindromic works correctly from the previous question and you can call that function here.

Here are some test cases for isComposite(n):

```
assert(isComposite(-7) == False)
assert(isComposite(0) == False)
assert(isComposite(1) == False)
assert(isComposite(4) == True)
assert(isComposite(14) == True)
assert(isComposite(101) == False)
assert(isComposite(303) == True)
```

...and here are some test cases for nthPalindromicComposite(n):

```
assert(nthPalindromicComposite(0) == 4)
assert(nthPalindromicComposite(1) == 6)
assert(nthPalindromicComposite(4) == 22)
assert(nthPalindromicComposite(5) == 33)
assert(nthPalindromicComposite(15) == 161)
assert(nthPalindromicComposite(100) == 2332)
```

Begin your FR2 answer on the following page

Begin your FR2 answer here

You may continue your FR2 answer here

You may continue your FR2 answer here