Name: _____ Andrew Id: _____

15-112 Fall 2023 Quiz 3

Up to 20 + 5 minutes (finish within 20 minutes for 1-point proficiency bonus)
No calculators, no notes, no books, no computers. Show your work!
Do not use strings, lists, dictionaries, try/except, or recursion on this quiz.

1. (6 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

```
def ct(n):
    k = 1
    res = 0
    while (k < n):
      if n%10 == k:
        print("debug")
        n = n // 10
      else:
        for i in range(0, k, 2):
          res = res + n %10
          n = n // 10
          print(i, res, k)
      k = 2 * k
print(ct(6531))  # prints 5 lines
```

2. (4 points) **Reasoning Over Code**: Find an argument, `n`, for the following function to cause it to return `True`. Place your answer (and nothing else) in the box below the code.

```
def roc(n):
    if type(n) != int or n > 1000:
        return False
    for i in range(2, 0, -1):
        if n % 10 != i:
            break
        n = n // 10
    return (n == 6)
```

3. (10 points) **Free Response: Count Narrow Numbers**

A *narrow* number (coined term) is an integer whose digit sum is equal to its length (without leading zeros); for example, 1 is a narrow number because its length equals 1, and the sum of its digits also equals 1. 12 is not a narrow number, as the sum of its digits is not equal to 2. 20 is a narrow number because its length and the sum of its digits are both equal to 2.

With that in mind, write the function `countNarrowNumbers(a, b)`, which takes two positive integers `a` and `b` and returns the number of *narrow* numbers that exist between `a` and `b` (inclusive).

Note: You may not use strings in this problem!! A solution that uses strings will receive 0 points.

Here are some examples:

- `countNarrowNumbers(1, 20)` returns 3 because there are 3 narrow numbers between 1 and 20: 1, 11, and 20
- `countNarrowNumbers(1,50)` also returns **3** because 1, 11, and 20 are the only narrow numbers between 1 and 50.
- `countNarrowNumbers(1,300)` returns **9** because 1, 11, 20, 102, 111, 120 201, 210, 300 are the only narrow numbers between 1 and 300.
- `countNarrowNumbers(100, 150)` returns **3** because 102, 111, and 120 are the only narrow numbers between 100 and 150.

Additional Space for Answer to Question 3