15-112 Fall 2024 Quiz 3

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!

Do not use strings, lists, tuples, dictionaries, sets, try/except, or recursion on this quiz.

1. (5 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box below the code.

   Hint: There are five lines of output.

```python
def helper(a):
    t = 0
    while a % 2 == 0:
        t += 1
        a //= 2
    return t

def ct(x):
    c = 0
    for i in range(x, 10, -1):
        c += i % 2
        if i % 3 != 0:
            continue
        t = helper(i)
        print(i, c, t)
    return c

print(ct(21))
```

```
21 1 0
18 2 1
15 4 0
12 5 2
6
```

2. (8 points) **Free Response**: nthLitPrime

Write a function `nthLitPrime(n)` which takes as an input a number (n) and returns the nth Lit Prime. A Lit Prime is a number where the number is prime, the sum of its digits is prime, and every consecutive pair of digits in the number is prime when considered as a two-digit number.

For example, the number 671131 is a Lit Prime because:

- 671131 is prime.
- $6 + 7 + 1 + 1 + 3 + 1 = 19$ is prime
- The numbers 67, 71, 11, 13, and 31 are all prime.

The first 10 Lit Primes are 11, 23, 29, 41, 43, 47, 61, 67, 83, 89

Here are some other Lit Primes: 719, 137, 6131, 11131, 231131.

Consider the following test cases:

```
assert nthLitPrime(0) == 11
assert nthLitPrime(9) == 89
assert nthLitPrime(75) == 231131
```

Notes/Hints:

- You should probably write a helper function, `isLitPrime(n)` that returns `True` if `n` is a Lit Prime and `False` otherwise.
- You may assume that the helper function `isPrime` has already been written and is available for you to use. (*Do not* write `isPrime`, just pretend it already exists.)

<div align="center">There is additional answers space for this problem on the next page.</div>

Additional space for Question 2

3. (7 points) **Free Response**: Ascending Digits

Write the function `ascendingDigits(n)` which, given an integer n, returns an integer that contains only the digits of n that ascend from right to left.

For example, consider `n = 643512`. Scanning the number from right to left...

- The right-most digit is 2. (64351<u>2</u>)
- The next digit that is larger than 2 is 5. (643<u>5</u>12)
- The next digit that is larger than 5 is 6. (<u>6</u>43512)
- The final answer is 652.

Consider the following additional test cases:

```
assert ascendingDigits(64354512) == 652
assert ascendingDigits(4239758346513) == 98653
assert ascendingDigits(759) == 9
assert ascendingDigits(0) == 0
assert ascendingDigits(-26434512) == -652
```