Name:_____ Section:___ Andrew Id: _____ a

15-112 Fall 2019 Quiz 2a

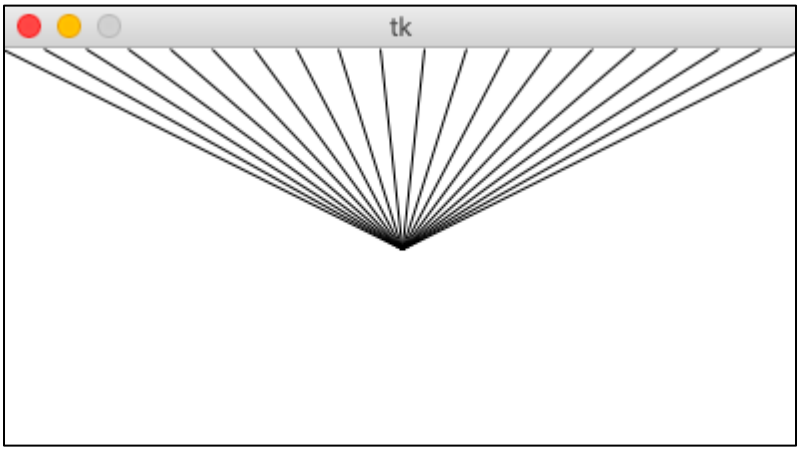
*** Up to 25 minutes. No calculators, no notes, no books, no computers. * Show your work!**

*** No strings, lists, string or list indexing, or recursion**

1. **Free Response: nthTenlyPrime(n)** [40 pts]
   We will say that a number is "tenly" (a made-up term) if the digits of the number add up to 10. So 1153 is tenly, but 153 is not. With this in mind, write the function nthTenlyPrime(n) that takes a non-negative int n and returns the nth number that is both tenly and prime. You should also write *all* the required helper functions.

   The first several tenly primes are: 19, 37, 73, 109, 127…

2. **Free Response: drawLinesToCenter(lines, canvas, width, height)** [25 pts]:
Write the function drawLinesToCenter(lines, canvas, width, height) that uses only lines to draw a this picture like this (in this case, where lines is 20):



- The first parameter (lines) is the number of lines to draw
- The tops of the lines are equally spaced along the top of the canvas, with the first line starting at the left-top corner and the last line starting at the right-top corner.
- All the lines end at the middle of the canvas
- You do not need to write runDrawing(width, height). Assume it is provided, and calls drawLinesToCenter(lines, canvas, width, height)

3. **Code Tracing** [25 pts]:
Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
def ct1(n):
    z = 0
    for x in range(2, n, 2):
        z = 10*z + x
        for y in range(n, 2*x):
            z = 10*z + y
    while z < 123456:
        z = 10*z + 9
    return z
print(ct1(6))
```

```
def ct2(a, b, c):
    while (c > a):
        c -= b
        if (c > a+b):
            print('X', c)
        elif (abs(c-14) == 1):
            print('Y', c)
            continue
        b += 1
    return c
print('Z', ct2(10, 7, 30))
```

4. **Reasoning Over Code** [10 pts]:
Find arguments for the following functions that makes them return True. Place your answers (and nothing else) in the box beside the code:

```
def f(n):
    t = 1
    while (n != 0):
        t *= n%10
        n //= 10
    return t

def rc1(z):
    r = 0
    for i in range(100):
        if (f(i) == 32):
            r = i
    return (r == z)
```

z =

5. **Bonus/Optional: Code Tracing** [2.5 pts]
Indicate what this prints. Very clearly circle your answer (and nothing else):

```
def bonusCt1(r):
    def f(x, d):
        z = 0
        while x: z,x = z or not (x%10-d),x//10
        return z
    for i in range(11):
        d = (i**2)%10
        if (not f(r, d)): r = 10*r + d
    return r
print(bonusCt1(4))
```

6. **Bonus/Optional: Code Tracing** [2.5 pts]
Sketch what this draws in a 300x300 canvas. Very clearly circle your answer, including a box to represent the 300x300 canvas.

```
def bonusCt2Drawing(canvas, width, height):
    # Sketch what this draws
    x, y, z = width/2, height/2, min(width, height)//2
    for x in range(z):
        for y in range(z):
            dx = x - z
            dy = y - z
            if (((dx >= dy/2) or (dy >= dx/2)) and
                abs(dx**2 + dy**2) <= z**2):
                canvas.create_line(x, y, z, z)
```