15-110 Midterm – Spring 2019
50 minutes

Name: ________________________________

Andrew ID: ________________@andrew.cmu.edu

Section: _______

- **You may not use any books, notes, or electronic devices during this exam.**
- **You may not ask questions about the exam except for language clarifications.**
- **Show your work on the exam (not scratch paper) to receive credit.**
- **If you use scratch paper, you must submit it with your andrew id on it, and we will ignore it.**
- **All code samples run without crashing. Assume any imports are already included as required.**
- **~~Do not use these post-midterm1 topics: strings, lists, etc~~. You may use strings and lists.**

DO NOT WRITE IN THIS AREA		
Part 1 (CT)	10 points	
Part 2 (Multiple Choice)	20 points	
Part 3 (FR / nonLetterCount)	20 points	
Part 4 (FR / Moving Rectangle)	20 points	
Part 5 (FR / Add/Delete Circles)	30 points	
Part 6/bonus	5 points bonus	
Total	100 points	

1. **[10 pts; 3 pts for ct1 and ct2, 4 pts for ct3] Code Tracing**
Indicate what each will print. Place your answer in the boxes below each block of code. Show your work, outside the box, for partial credit.

```
def ct1(m, n):
    result = 0
    while (n > m):
        n -= 3
        result += n
    return result
print(ct1(5, 14))
```

```
def ct2(n):
    L = [ ]
    for x in range(1, n):
        if (n % x == 0):
             L.append(x)
    for i in range(len(L)):
        L[i] *= 10**i
    return L
print(ct2(10))
```

```
def ct3(s):
    t = ''
    for c in s:
        if ((s.count(c) > 1) and
            (c.isalpha() == True)):
            t += c
    return t
print(ct3('Who knows it? I do!'))
```

2. **[20 pts; 2 pts each] Multiple Choice**
   Circle the letter of the correct answer.

1. Using sign-magnitude, 1101 in binary is ________ in decimal.

A. 13 B. -13 C. 12 D. -12 E. 5 F. -5 G. None of these

2. Each value in an RGB color must be an integer from 0 to 255, inclusive.

From this, we conclude that each value is represented using ______ bits.

A. 256 B. 20 C. 10 D. 5 E. None of these

3. Which of the following can be used to perform subtraction?

A. Lattice B. 10's Complement C. Parity D. Nim E. Coin flips

4. Which of the following can be used to detect errors when transmitting data?

A. Lattice B. 10's Complement C. Parity D. Nim E. Coin flips

5. Circle ALL of the following that are methods we learned to represent strings:

A. Zero Terminated B. Zero Prefixed C. Length Terminated D. Length Prefixed

6. In our example circuit that added two one-bit numbers, the right bit (one's digit)

of the answer was computed using a(n) _______ gate.

A. AND B. OR C. NOT D. XOR E. NAND F. None of these

7. Since our memory circuit in the notes loses its memory when the power is lost,

we say that it is ______.

A. Bad B. Volatile C. Lossy D. Powerful E. None of these

8. When we compute big-oh, we ignore constants and we also ignore __________.

A. lower-order terms B. logs C. higher-order terms D. None of these

9. Since linear search is O(n) and binary search is O(logn), for very large sorted

lists, we would expect linear search to be ________ than binary search.

A. much faster B. a little faster C. about the same D. a little slower E. much slower

10. In the worst case, Selectionsort is O(n**2) and Mergesort is ____________.

A. O(n**2) B. O(nlogn) C. O(n) D. O(logn) E. None of these

3. **[20 pts] Free Response: nonLetterCount(s)**

Write the function nonLetterCount(s) that takes a string s and returns a count of the number of characters in s that are not letters, but only count each such character once even if it occurs multiple times in s. For example, nonLetterCount('This is a test!!') returns 2 (space and exclamation mark).

4. **[20 pts] Free Response: Moving Rectangle**

Note: Do not assume the canvas size. Instead, use data.width and data.height.
Starting from our starter code, write the functions init, timerFired, and drawAll so that:

- A 100x20 red rectangle starts in the right-bottom corner of the canvas.
- The rectangle moves up 10 pixels at a time, and stops moving up when its top reaches the canvas top.
- At that point, the rectangle moves left 10 pixels at a time until its left reaches the canvas left.
- When the rectangle reaches the left-top corner, it remains there from then on.

Note: you can use the next page if you wish.

(This page is blank)

5. **[30 pts] Free Response: Add/Delete Circles**

Note: In the app below, you must store the circles in the list data.circles, and each circle must be an instance of a Circle class that you create.
Hint: your Circle class should look something like this:

```
  class Circle(object):
    def __init__(circle, cx, cy, r):
            ...
```

With that in mind, starting from our starter code, write the functions init, keyPressed, mousePressed, and drawAll so that:

- The canvas is empty at the start with a green background.
- There are two modes -- Add Mode and Delete Mode. We start in Add Mode. Each time any key is pressed, the mode toggles to the other mode.
- The background is green when in Add Mode, and red when in Delete Mode.
- On a mouse press while in Add Mode, we add a circle of radius 20 centered on the mouse press location.
- On a mouse press while in Delete Mode, if there are any circles left, we find the nearest circle to the mouse press and delete it (if two circles are the same distance from the mouse press, you can delete either one).

Note: you can use the next page if you wish.

(This page is blank)

6. Bonus/Optional: **[2.5 pts] What will this print? Clearly circle your answer.**

```
# hint: bin(27) is '0b11011'
def bonusCt1(n):
    i = 0
    for z in range(2**n): i += bin(z).count('1')
    return i
print(bonusCt1(10))
```

Bonus/Optional: **[2.5 pts] What will this print? Clearly circle your answer.**

```
def bonusCt2(L):
    d = L.pop()
    while (str(L).count(str(d)) < 3**d): L.append(L*3)
    return L[-1][-1]
print(bonusCt2([1,2,3,2]))
```