Name_____

CS 107

Practice Final Exam

Spring 2016

Instructions: closed books, closed notes, open minds,
2 hour time limit.
There are 4 sections.

Part I: Definitions and Theory
Part II: Binary, Logic, and Gates
Part III: Reading and Understanding Code
Part IV: Writing Code
Part V: Reflection

Part I. Definitions and Theory

1. Be familiar with the following terms and ideas from this quarter:

    - What is Computer Science?
    - Who was Charles Babbage? Alan Turing? Ada Lovelace?
    - Algorithm, Program- What are they? How are they related? How are they different?
    - Python, Jython, JES
    - Encoding data- How is text encoded? How are pictures encoded/digitized? How are sounds encoded/digitized?
    - What are Boolean expressions? What is a truth table? What are gates and circuits? How are these all connected?
    - How do you convert between decimal, binary, and hexadecimal numbers?
    - What is an interpreter? What is a compiler?
    - What is a high-level language? What is an assembler language? What is a machine language? How do we get from a high-level language to a machine language? What types of languages do we write programs in? Why do we choose one type over another?
    - Designing programs – top-down vs. bottom-up
    - Complexity of algorithms – tractable, intractable, non-computable
    - Halting Problem

    Note: Should also add Traveling Salesperson problem to this list

Part II. Binary, Logic, and Gates

2. a) Convert the binary number 110101 to a decimal number.

    110101 = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0
    = 32 + 16 + 4 + 1 = 53

    b) Convert the decimal number 75 to a binary number.

    75 = 64 + 11 = 64 + 8 + 3 = 64 + 8 + 2 + 1 = 2^6 + 2^3 + 2^1 + 2^0
    = 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0
    = 1001011

3. a) What is the hexadecimal representation for the string “Hello”?

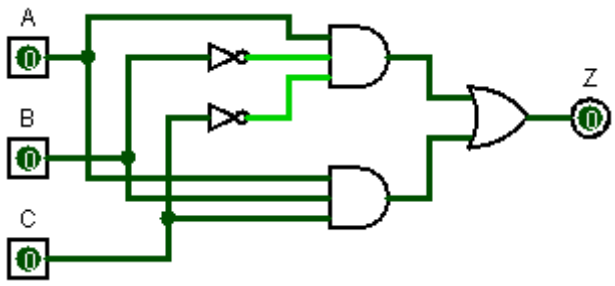
Soln:
Using handout from ascii/binary assignment early in the quarter:

H = 48 in hex
e = 65
l = 6C
l = 6C
o = 6F

b) What is the binary representation for this same string?

48 = 0100 1000
65 = 0110 0101
6C = 0110 1100
6C = 0110 1100
6F = 0110 1111

4. Write down the Boolean expression and truth table that correspond to the following gate diagram:



Soln:

Boolean expression: AB'C' + ABC = Z

Truth table:

A	B	C	AB'C'	ABC	Z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Part III. Reading and Understanding Code

5. The following code snippet will not cause JES to print an error message, but it contains two logic errors so that the beginning of the sound becomes silent. Circle the line(s) with the error(s) and rewrite below.

```
# A function that decreases the volume in a section
# of a sound by half, from the given starting sample
# index to the given ending sample index
def dampenASection(sound, startIndex, endIndex):
  for index in range(0,startIndex):
    value = getSampleValueAt(sound, index)
    setSampleValueAt(sound, index, 0)
```

Soln: The for loop should be:

```
    for index in range(startIndex, endIndex):
```

The line to set the value should be:

```
    setSampleValueAt(sound, index, value/2)
```

6. What will we see and/or hear if we call this function with the following parameters:
   `mysteryFunction1(sound1, sound2)`
   where `sound1` is from a file "`go.wav`", and `sound2` is from a file "`stop.wav`".

```
def mysteryFunction1(sound1, sound2):
  canvas = makeEmptyPicture(300, 300)
  show(canvas)
  play(sound1)
  for x in range(0, getWidth(canvas)-10, 10):
       addRectFilled(canvas, x, 1, 10, 10, magenta)
       repaint(canvas)
       addRectFilled(canvas, x, 1, 10, 10, white)
  for y in range(0, getHeight(canvas)-10, 10):
       addRectFilled(canvas, getWidth(canvas)-10-y, y, 10, 10, blue)
       repaint(canvas)
       addRectFilled(canvas, getWidth(canvas)-10-y, y, 10, 10, white)
  for x in range(0, getWidth(canvas)-10, 10):
       addRectFilled(canvas, x, getHeight(canvas)-10, 10, 10, yellow)
       repaint(canvas)
       addRectFilled(canvas, x, getHeight(canvas)-10, 10, 10, white)
  play(sound2)
```

Soln: We will first see an empty white canvas and then hear "go". Then we will see a magenta-colored square move across the top of the white canvas. We will then see a blue square move diagonally from the top right corner to the bottom left corner of the white canvas, and then finally, a yellow square move across the bottom. The boxes never move off the canvas. We then hear "stop".

Part III. Writing Code

7. Write a function called `minEltInList` that finds the minimum value in a list of integer elements. This function should take a list as a parameter and should return the value of the minimum element in the list. You may assume that the list contains unique integers and is not empty. You may not use the built-in Python function `min`. (Note that `len(myList)` will return the number of elements in the list called `myList`.)

Soln:

```
def minEltInList(someList):
     minValue = someList[0]
     for index in range(1, len(someList)):
          if (someList[index] < minValue):
               minValue = someList[index]
     return minValue
```

8. Write a function that takes a sound as a parameter and creates a new sound the same length as the original sound, according to the following rules. For every sample in the original sound: if the absolute value of the sample value is greater than 16,000, set the corresponding sample value in the new sound to be 0. Otherwise double it. Since this function creates a new sound, the new sound should be returned.

Soln:

```
def modifySound(sound):
     newSound = duplicateSound(sound)
     for index in range(0, getLength(newSound)):
          value = getSampleValueAt(newSound, index)
          if (abs(value) > 16000):
               setSampleValueAt(newSound, index, 0)
          else:
               setSampleValueAt(newSound, index, value*2)
     return newSound
```

9. Write a function that creates a square yellow canvas, moves a blue box along the top-left-to-bottom-right diagonal of the canvas, then along the top-right-to-bottom-left diagonal of the canvas, and then finishes up in the middle of the canvas. This function should take the width to make the canvas, and the width of the box as parameters, and does not need to return anything.

Soln:

```
def moveBoxes(canvasWidth, boxWidth):
    canvas = makeEmptyPicture(canvasWidth, canvasWidth, yellow)
    show(canvas)
    for x in range(0, canvasWidth-boxWidth, 10):
      addRectFilled(canvas, x, x, boxWidth, boxWidth, blue)
      repaint(canvas)
      addRectFilled(canvas, x, x, boxWidth, boxWidth, yellow)
    for y in range(0, canvasWidth-boxWidth, 10):
      addRectFilled(canvas, canvasWidth-boxWidth-y, y, boxWidth, boxWidth, blue)
      repaint(canvas)
      addRectFilled(canvas, canvasWidth-boxWidth-y, y, boxWidth, boxWidth, yellow)
    addRectFilled(canvas, (canvasWidth-boxWidth)/2, (canvasWidth-boxWidth)/2, boxWidth, boxWidth,
      blue)
    repaint(canvas)
```

Bonus: Modify the function you just wrote so that the box does not go off the edge of the screen in either direction.

Soln: The function above has been written this way already.

Part IV. Reflection

In your final programming projects, you had an opportunity to put together many of the aspects of multimedia programming that we learned this quarter. Reflecting back on what you did, answer the following questions. There are no right or wrong answers here, it is expected that you will give this some thought and an honest effort.

a) In terms of design, some of you wrote your programs all in one main function, while others used functions for each part of the animation. Describe the advantages and disadvantages that might be involved in using **each** of these designs.

b) When creating your own animation, what limitations did you run into? Were there things you wanted to do, but didn't, either because you didn't know how to program them, or because you thought they would take too long to run?

.

Soln: Answers will vary.