15-112
Fall 2017 Midterm Exam 1
October 19, 2017

Name:

Andrew ID:

Recitation 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 to receive credit.
- You may use the backs of pages as scratch paper. Nothing written on the back of any pages will be graded.
- All code samples run without crashing. Assume any imports are already included as required.
- Do not use these post-midterm 1 topics/constructs: sets, maps/dictionaries, recursion, or classes/OOP.

Don't write anything in the table below.

Question	Points	Score
1	10	
2	10	
3	15	
4	20	
5	20	
6	25	
Total:	100	

1. **Code Tracing**

   Indicate what each will print. Place your answer (and nothing else) in the box below each block of code.

   (a) (5 points) CT1

```
import string
def ct1(s):
    a = ""
    for i in range(len(s)):
        if s[i] in string.digits:
            j = int(s[i])
            print(s[i::j])
        elif s[i] in string.ascii_lowercase:
            s = s.upper()
            a += s[i]
        elif s[i] in string.ascii_uppercase:
            s = s.lower()
            a += s[i]
    print(a)

ct1("2HG5e\td3Ba1")
```

(b) (5 points) CT2

```
import copy
def ct2(a):
    b = a
    c = copy.copy(a)
    d = copy.deepcopy(a)
    c[0] = [4, 6]
    b[0][1] = "orange"
    c[1][0] = "lime"
    c[0][0] = 10
    d[1] = [7, 9]
    c[1].insert(1,3)
    b.append("cat")
    print("a =", a)
    print("b =", b)
    print("c =", c)
    print("d =", d)

z = [ [2, 3], [11, 12] ]
ct2(z)
print("z =", z)
```

2. **Reasoning Over Code**

For each function, find values of the parameters so that the function will return True. Place your answer (and nothing else) in the box below each block of code.

(a) (5 points) ROC1

```
def roc1(n):
    assert(n < 100000)
    (a, b) = (0, 0)
    while n > 0:
        (c, n) = (n % 10, n // 10)
        if c < 5:
            a = 10*a + c
        else:
            b = 10*b + c
    return (a, b) == (30, 795)
```

(b) (5 points) ROC2

```
def roc2(L):
    if not isinstance(L, list):
        return False
    a = len(L)
    s = L.pop()
    for i in range(2):
        s += L.pop(s)
    for i in L:
        s -= (ord(i) - ord('a'))
    return s == 33 and a > 3 and a < 10
```

3. **Short Answer**

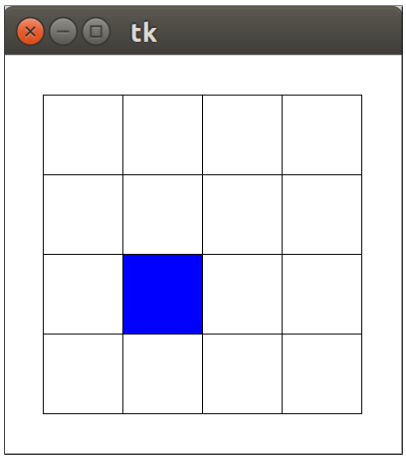
Answer each of the following *very briefly*.

(a) (3 points) Consider the following code that we went over in lecture:

```
def hasBalancedParentheses(s):
    leftParen = 0
    rightParen = 0
    for c in s:
        if c == "(":
            leftParen += 1
        elif c == ")":
            rightParen += 1
    return leftParen == rightParen
```

Describe why this is not a good solution to the problem.

(b) (3 points) Consider the following screenshot of a tkinter game:



The grid you see consists of squares with 80 pixel sides and a margin of 40 pixels all around. The game ends when the user clicks on the colored cell. The variables `data.row` and `data.col` define the row and column, respectively, of which cell is colored. In the example picture, the row is 2 and the column is 1. When the user clicks on the colored cell, `data.gameOver` should be set to true.

Fill in 3-4 lines of python code to complete the `mousePressed` function as described above.

```
def mousePressed(event, data):
```

(c) (3 points) What is the primary difference between a tuple and a list?

(d) (3 points) Write a comment of fifteen words or less that describes what the following function does.

```
def puzzle(a):
    assert(isinstance(a, list))
    c = len(a)
    d = 0
    for i in range(c):
        e = c - i - 1
        if i >= e:
            break
        elif a[i] == a[e]:
            d += 1
    return d == c // 2
```

(e) (3 points) Define and briefly explain what the three elements of MVC are.

4. (20 points) **Free Response: Pradish Numbers**

   Note: For full credit, you may not use strings or lists for this problem. For a 10-pt deduction, you may use strings and/or lists.

   A pradish number (coined term) is a number with *all* of the following properties:

   - The number has an even number of digits.
   - The sum of the digits of the number is equal to the number of digits of the number.
   - The sum of the first half of the digits is equal to the sum of the second half of the digits.

   For example, 102003 is pradish. It has 6 digits, which is even. Its digits also sum to 6 ($1 + 0 + 2 + 0 + 0 + 3 = 6$). The sum of the first half of the digits is equal to the sum of the second half of the digits ($1 + 0 + 2 = 0 + 0 + 3$).

   Write the function `isPradish(n)` that takes an integer value n and returns True if the number is pradish and false otherwise. Then, write `nthPradish(n)` that takes an integer value n and returns the nth pradish number. For example, nthPradish(0) returns 11. You may write any additional helper functions you desire.

Additional Space for Answer to Question 4

5. (20 points) **Free Response: Vertical Word Remover**

In this problem you will write the destructive function `verticalWordRemove(L, word)` that takes as arguments a non-ragged, 2-dimensional list `L` of letters and a string `word`. The function goes through L, removing the first instance of the word from the columns of L, shifting the rows to the left to fill in the empty spaces as it goes.

Consider the following example:

```
L = [
 ['t', 'c', 'z', 'd', 'e', 'f', 'g'],
 ['a', 'a', 'd', 'f', 'd', 'c', 'a'],
 ['c', 't', 'q', 'r', 'e', 'a', 'q'],
 ['w', 'e', 'i', 'o', 't', 't', 'w']
]

After calling verticalWordRemove(L, "cat"), then...

L = [
 ['t', 'z', 'd', 'e', 'f', 'g'],
 ['a', 'd', 'f', 'd', 'c', 'a'],
 ['c', 'q', 'r', 'e', 'a', 'q'],
 ['w', 'e', 'i', 'o', 't', 't', 'w']
]
```

The word "cat" was removed from column 1. Notice that even though L has two instances of "cat" in its columns, only the first one is removed.

Notes: When your function is called, you may assume `L` is not ragged. After your function returns, `L` might be ragged. You may assume `L` and `word` will contain only lowercase letters. You should search the columns in left-right order (column 0, then 1, then 2, etc) and within a column search top-down.

Additional Space 1 for Answer to Question 5

Additional Space 2 for Answer to Question 5

6. (25 points) **Free Response: Color Grid Animation**

Assuming the run() function is already written for you, write init, keyPressed, mousePressed, redrawAll, and timerFired so that when the animation is first run:

A. A black grid composed of red cells (each 20 pixels by 20 pixels) is displayed in the window, with no margin. You may assume the window's width and height are each multiples of 20.

B. A blue circle is in the upper-left cell.

Game play proceeds as such:

A. Every half second, the circle moves forward one cell. At the beginning, the circle moves downwards.

B. The circle's color changes automatically every five seconds. The color switches from blue to red and from red to blue.

C. When the circle moves into a new cell, it changes that cell's color to match its own. The cell keeps that color when the circle leaves.

D. If the circle moves off the board, the game is over. This is displayed as a black screen with yellow text in the center which says 'GAME OVER!'.

E. The user can change the circle's movement direction by pressing the arrow keys.

Make reasonable assumptions for anything not specified here. Do not hardcode values for data.width or data.height. You are only allowed to use one timer and one timerFired function. We recommend that, to save time writing, you abbreviate canvas, event, and data: use c, e and d, respectively.

Additional Space 1 for Answer to Question 6

Additional Space 2 for Answer to Question 6

Additional Space 3 for Answer to Question 6