15-112 Spring 2019
Practice Midterm 1
80 minutes

Name:

Andrew ID:

Recitation Section:

- **This practice midterm is written by TAs and may not reflect the true difficulty or length of the actual midterm.**
- 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 complete the problems in any order you'd like; you may wish to start with the free response problems, which are worth most of the credit.
- All code samples run without crashing unless we state otherwise. Assume any imports are already included as required.
- Do not use these concepts: generators, sets, dictionaries, OOP or recursion

Don't write anything in the table below.

Question	Points	Score
1	15	
2	15	
3	5	
4	15	
5	15	
6	15	
7	20	
Total:	100	

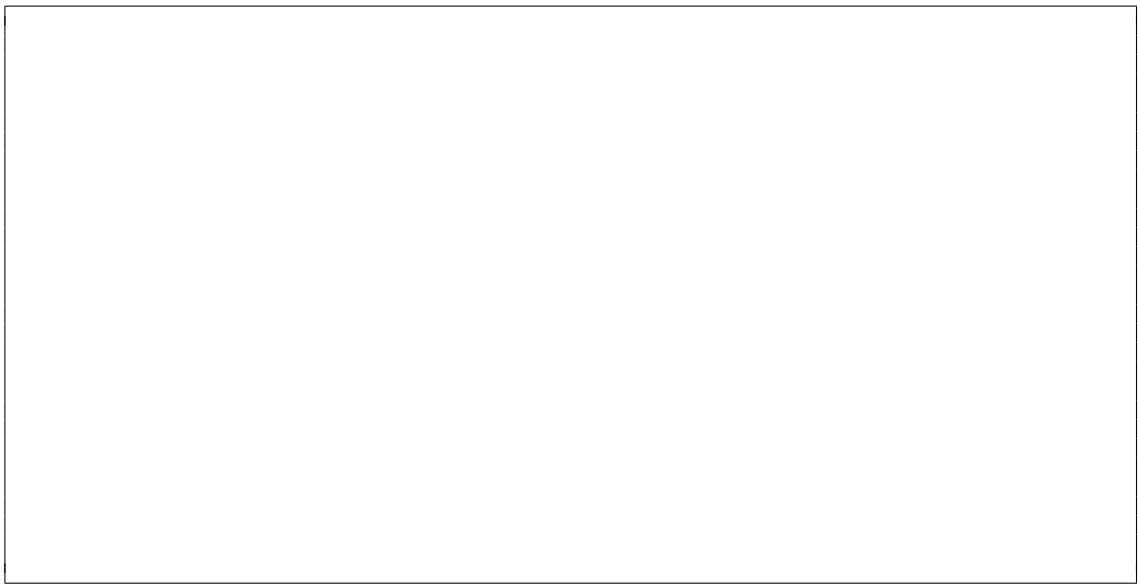
1. **Code Tracing**

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

   (a) (5 points) CT1

```
def ct1(a):
    L, M, n = [ ], [ ], len(a)//2
    for val in a[n-1:0:-1]:
        L.append(val % 10)
        M += [val // 10 % 10]
        if (sum(M) > 3):
            L.append(M.pop(len(M)-1))
    return L + M

print(ct1(list(range(17, 25))))
```



(b) (5 points) CT2

```
import copy

def ct2(a):
    b = a
    c = b[:]
    d = copy.deepcopy(c)
    c[0] = [4, 6]
    b[0][1] = "had"
    c[1][0] = "ley"
    c[0][0] -= d[0][0]
    d[1] = [7, 9]
    c[1].insert(1,3)
    print(b.append(c.pop()))
    for L in [a, b, c, d]:
        print(L)

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

(c) (5 points) CT3

```
def ct3(s):
    t = ""
    r = 0
    for c in s:
        if c.isdigit() and int(c) % 2 == 0:
            t += c
        elif c.isalpha():
            r += 5
        elif c.isalnum():
            r += 1
        elif c.isspace():
            t = "%s%d" % (t,r)
            r = 0
        else:
            print("0", end="")
    return t

print(ct3("a12?\\n89!e\t"))
```

2. **Reasoning Over Code**

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

(a) (7 points) ROC1

```
def roc1(n):
    count = 0
    temp = n
    while(temp > 0):
        count += 1
        temp //= 10
    assert(count % 2 != 0 and count**2 > 25)

    for i in range(count//2):
        assert((n//(10**i) % 10) == (n//10**(count-i-1) %10))
    return True
```

(b) (8 points) ROC2

```
def roc2(s):
    i = 0
    assert(len(s) > 3)
    temp = ""
    while(i < len(s)):
        assert(s[i].islower() if i % 2 == 0 else s[i].isupper())
        assert(s[i] not in temp)
        temp += s[i]
        if(i > 1 and ord(s[i]) < ord(s[i-2])):
            return False
        if(i == len(s)//2):
            assert(s[i] == "z")
        i += 1
    return True
```

3. **Short Answer**

Answer each of the following *very briefly*.

(a) (1 point) In no more than two sentences, briefly explain what top-down design is.

(b) (1 point) What is the length of the following string:

```
"\\\t112rox\n"
```

(c) (1 point) What type of error (syntax, runtime, or logical) will occur in this code below and why?

```
def findAndRemoveRepeats(lst, n):
    count = 0
    for i in range(len(lst)):
        if lst[i] == n:
            count += 1
            if count > 1:
                lst.pop(i)
    return lst

findAndRemoveRepeats([1, 1, 3, 1, 1, 8, 9], 1)
```

(d) (1 point) What errors will occur if you try to draw things in controller functions (like keyPressed or mousePressed)?

(e) (1 point) How do you set the color of a tkinter shape to be transparent?

4. (15 points) **Free Response 1: vigenereCipher(msg, key)**

   The vigenere cipher is a method of encrypting text by using a series of interwoven Caesar ciphers, based on the letters of a keyword. This cipher resisted all efforts to break it for three centuries, which earned it the description le chiffre indechiffrable (french for the indecipherable cipher). The vigenere cipher takes in two inputs: a message to be scrambled and a keyword. The keyword is then repeated to match the length of the message, before applying the shift alphabetically.

   For instance, take the following: msg = ATTACKATDAWN and key = LEMON.

   We get:

   Message: ATTACKATDAWN

   Key: LEMONLEMONLE

   Ciphertext: LXFOPVEFRNHR

   The shift can be described algebraically where the letters A-Z are taken to be the numbers 0-25, and addition is performed modulo 26. So for the first letter, we get A+L=0+11=11, which corresponds back to the letter L. Similarly for the second letter, we get T+E=19+4=23, which corresponds back to the letter X. With this in mind, write the function vigenere(msg, key) that takes as input the msg and key, and returns the associated ciphertext. You are guaranteed that the msg and the key will have length at least 1, and that the msg and key will only contain capital letters. You are not guaranteed that the msg is longer than the key, and neither are you gauranteed that the key will be as long as the msg. (in which case you should repeat the key until it becomes as long as the msg)

Additional Space for Answer to Question 4

5. (15 points) **Free Response 2: nthTrimorphicNum(n)**

   A trimorphic number is a number whose cube ends in the number itself. For example, some trimorphic numbers are 4 (as 4^3 =**64**), 24 (as 24^3 = 138**24**), and 249 (as 249^3 = 15438**249**). The first several trimorphic numbers are 0, 1, 4, 5, 6, 9, 24, 25, 49, 51, 75, 76, 99, 125...

   Write a function nthTrimorphic(n) that takes in a non-negative integer n and returns the nth trimorphic number. nthTrimorphic(0) should return 0.

6. (15 points) **Free Response 3: longestIncreasingWeave(L)**

   Background: We will define a weave on a 2D list as the traversal of the list down the first column, up the second column, down the third column and so on. For example, the weave of the following list:

   [[1,2,3],

   [4,5,6],

   [7,8,9]]

   would be [1,4,7,8,5,2,3,6,9]

   With that in mind, write the function longestIncreasingWeave(L) which takes a rectangular, non-empty 2D list and returns the longest weave where the numbers are strictly increasing. If there is a tie, return the weave that occurs first in the list. In the above example, the function should return [1,4,7,8]

Additional Space for Answer to Question 6

7. (20 points) **Free Response 4: Tic-Tac-Toe**

In this question we will write a simplified version of the popular game Tic-Tac-Toe. Assuming the run function is written for you already, please write the **init**, **keyPressed**, **timerFired** and **redrawAll** functions so that the game can be played according to the following specifications. (please read the following very carefully)

- The board should have 20 rows and 25 columns.
- The game should keep track of the current cell the user is in. The currently selected cell should be highlighted a different color and should start of as row 0 column 0 at the beginning.
- You should allow the user to move the current cell using the left, up, right and down arrow keys. If you move too far off the board the cell should wraparound correctly.
- You should also allow the user to change the currently selected cell by clicking in the cell.
- If the highlighted cell is empty, you should allow the user to place an "x" or an "o" in that cell. These characters will be entered using their respective keys on the keyboard.
- In the top left of the canvas you should display a timer that is the number of seconds that have elapsed since the game started. To achieve this you **may not** change the value of data.timerDelay directly. (the run function sets data.timerDelay = 100 by default)
- You should check if the game has been won after every move. The game finishes with a "win" if there are 3 adjacent "o"'s or 3 adjacent "x"'s on the board (horizontally, vertically or diagonally) For this you can assume that **wordSearch(board, word)** has already been written for you. wordSearch(board, word) returns True if the word 'word' exists on 'board' and False otherwise. After a win, all further key presses and mouse presses must be ignored. You do not need to worry about what happens if the board fills up with no win.

Additional Space for Answer to Question 7

Additional Space for Answer to Question 7