

15-112
Summer 2018 Midterm Exam
June 8, 2018

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.
- You may complete the problems in any order you'd like; you may wish to start with the last three problems, which are worth most of the credit.
- All code samples run without crashing. Assume any imports are already included as required.
- Do not use these post-midterm topics/constructs: sets, maps/dictionaries, recursion, or classes/OOP.

Don't write anything in the table below.

Question	Points	Score
1	15	
2	5	
3	10	
4	22	
5	23	
6	25	
Total:	100	

1. (15 points) **Code Tracing**

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

(a) (10 points) CT1

```
import copy
def ct1(L):
    L[2] = L[1]
    a = copy.copy(L)
    b = copy.deepcopy(L)
    c = L
    a[0][0] = 113
    a[2] = 15
    b[0][0] = 112
    c[1] = 119
    print("a:", a)
    print("b:", b)
    print("c:", c)
    print("L1: ", L)
L = [[2,0],[1,3], [4]]
ct1(L)
print("L2:", L)
```

(b) (5 points) CT2

```
def ct2(s):
    result = ""
    i = 0
    for c in s:
        if c.isdigit():
            result += s[int(c)]
        elif c == c.upper():
            i += 3
        else:
            result += s[i % 2]
            i += 1
    return result
s = "xY6z78Abc"
print(ct2(s))
```



2. (5 points) **Reasoning Over Code**

Find an argument (the value of `s`) for `roc1` that makes it return `True`. Place your answer (and nothing else) in the box below the code.

```
def roc1(s):
    assert(len(s) == 10)
    result, t = "", "qakcu"
    n = len(s)
    for i in range(n // 2):
        if int(s[n - i - 1]) < int(s[i]):
            result = "%s%s" % (result,t[0])
            t = t[1:]
        elif int(s[n - i - 1]) > int(s[i]):
            result = "%s%s" % (result,t[-1])
            t = t[:-1]
    return result == "quack"
```

3. (10 points) **Short Answer**

Answer each of the following *very briefly*.

- (a) (2 points) Write one line of code that would be an MVC violation if it was placed in `redrawAll`.

- (b) (2 points) The code below violates multiple style rules in the 112 style guide. List 2 rules that are violated.

Note: Answers such as "lacks clarity" or "inefficient" will not be accepted.

Be brief but precise!

```
def digit_count(b):  
    count = 0  
    while b > 0:  
        count += 1  
        b //= 10  
    return count  
    print("done!")
```

- (c) (2 points) Why do we need to use `almostEquals` when comparing floats?

- (d) (2 points) Give an example of a situation where you would need to use a while loop instead of a for loop.

- (e) (2 points) You're given a function `foo(L)` that operates on a list of integers. Write a few lines of code to test whether that function is destructive.

4. (22 points) **Free Response: isReversiblePrime(n) and nthReversiblePrime(n)**

A reversible prime is a number n such that n is prime, and the number formed by reversing the digits of n is also prime.

For example, 13 is a reversible prime because 13 is prime and so is 31.

Write the function `isReversiblePrime(n)` that takes a value n and returns `True` if it is a reversible prime and `False` otherwise. Then, write `nthReversiblePrime(n)` that takes an integer value n and returns the n th reversible prime number. For example, `nthReversiblePrime(0)` returns 2.

Here are the first couple reversible prime numbers: 2,3,5,7,11,13,17,31,37,71,73,79.

You may NOT use strings in this problem. However, you may assume that `isPrime(n)` is already written for you.

Additional Space for Answer to Question 4

5. (23 points) **Free Response: isLowerTriangular**

For this problem assume you're given a 2D list of integers `L`. Also assume that `L` is `NxN` where `N` is at least 2.

We define the diagonal of `L` to be elements `L[row][col]` such that `row == col`.

In this problem you will write the function `isLowerTriangular(M)` that takes a 2D list and returns `True` if the list is lower triangular. A lower triangular 2D list is a list where every element above the diagonal is 0.

For example, the following 2D list is lower triangular:

```
[[1, 0, 0, 0],  
 [5, 2, 0, 0],  
 [5, 5, 3, 0],  
 [5, 5, 5, 4]]
```

The elements on the diagonal in this example are 1,2,3, and 4. The elements above the diagonal are all 0s (which is what makes this lower triangular).

The following list is not lower triangular (because of the 7 that is above the diagonal):

```
[[1, 0, 0, 0],  
 [5, 2, 7, 0],  
 [5, 5, 3, 0],  
 [5, 5, 5, 4]]
```

Additional Space 1 for Answer to Question 5

6. (25 points) **Free Response: clickMeIfYouCan**

The following animation is a 2 player game where one player uses the arrow keys to move an oval between columns as the oval falls to the bottom of the screen, and the other player attempts to "catch" the oval by clicking on it.

Using our animation framework and assuming `run()` is already written, write the `init(data)`, `keyPressed(event, data)`, `mousePressed(event, data)`, `timerFired(data)`, and `redrawAll(canvas, data)` functions for an animation which has the following elements:

1. A grid with 10 rows and 3 cols is displayed. The grid should fill the entire screen for any width and height specified.
2. A blue oval starts in the top middle of the screen (`row=0, col=1`).
3. The blue oval is moved left and right using the arrow keys. Pressing an arrow key should move the oval into an entirely new cell. The oval should always be centered in its current cell, and the oval's size should not be larger than the cell.
4. Every 100ms the oval's row should move down by one. When the oval reaches the last row, its location should reset to the top row. Hint: 100ms is the default for `data.timerDelay`.
5. If the user clicks in the grid that contains the oval, the game is over. Once the game is over, only the text "Game Over" should be displayed in the center of the screen.
6. You do not need to deal with bounds checking for movement, and you do not need to worry about ignoring `keyPressed/mousePressed` when the game is over.

Make reasonable assumptions for anything not specified here. We recommend that, to save time writing, you abbreviate `canvas`, `event`, and `data`: use `c`, `e` and `d`, respectively.

Note: While we generally don't grade for style on exams, MVC violations are an exception. You will lose points if there are MVC violations in your code.

Additional Space 1 for Answer to Question 6

Additional Space 2 for Answer to Question 6