

15-112
Summer 2019 Midterm Exam
June 7, 2019

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.
- Good luck!

Don't write anything in the table below.

Question	Points	Score
1	10	
2	5	
3	10	
4	10	
5	22	
6	22	
7	21	
Total:	100	

1. (10 points) **Code Tracing**

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

```
import copy

def ct1(L):
    a = copy.copy(L)
    b = copy.deepcopy(L)
    c = [L[0]] + a[0]
    c[0][0] = 2
    a.insert(0, b.pop())
    b[0] = b[1]
    print(c.append(b[1]))
    b[1][1] = 112
    a[2] = c[2]
    a[1][1] = "python"
    print("a = ", a)
    print("b = ", b)
    print("c = ", c)
```

```
L = [[1, 2], [3, 4], [5, 6]]
ct1(L)
print("L = ", L)
```

2. (5 points) **Code Tracing**

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

```
def ct2(s):
    while s != "":
        i = len(s)//2
        print("s =", s)
        if i % 3 == 0:
            print("\t", s[i])
        s = s[:i]+s[i+1:]
print(ct2("aksgyw"))
```

3. (10 points) **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.

```
def roc1(L):  
    res = ""  
    x = 0  
    for i in range(len(L)):  
        assert(len(L[i]) == 4)  
        for j in range(len(L[i])-1, -1, -2):  
            res += str(L[i][j])  
        x += 1  
    return res == "314159" and x == 3
```

4. Short Answer

Answer each of the following *very briefly*.

- (a) (3 points) Recall in class we mentioned that strings are immutable. Write 1-2 lines of Python code that will crash since strings are immutable.

- (b) (2 points) Give an example of a common MVC violation – that is, something that is explicitly disallowed by the rules for Model-View-Controller applications.

- (c) (2 points) In 4 words or less, explain why this code snippet will not crash:

```
return 0 == 0 or 5/0
```

- (d) (3 points) Circle all of the test cases that will fail for the buggy isPrime implementation shown below.

```
def isPrime(n):  
    if (n <= 2): return False  
    for factor in range(2, int(math.sqrt(n))):  
        if (n % factor == 0):  
            return False  
    return True
```

- A. `assert(isPrime(2) == True)`
B. `assert(isPrime(10) == False)`
C. `assert(isPrime(1) == False)`
D. `assert(isPrime(11) == True)`
E. `assert(isPrime(9) == False)`

5. (22 points) **Free Response: nthKamyarNumber**

We will classify a number as a Kamyar number if the number is equal to the sum of each of its digits raised to the power of the number of digits in the number. For example, 153 is a Kamyar number because $153 = 1^3 + 5^3 + 3^3$, where we are raising each digit to the third power because 153 has three digits.

Write the function `nthKamyarNumber` which takes in a non-negative integer `n` and returns the `n`th Kamyar number. **To get full credit, you may not use strings to solve this problem.**

Additional Space 1 for Answer to Question 5

6. (22 points) **Free Response: countPossibleRookMoves**

Write the function `countPossibleRookMoves`, which takes in a board and a potential rook position, and returns a tuple containing the number of possible moves the rook can make, and the number of pieces it can attack.

For this problem, assume that the board is represented as a 2D list where 0 represents an empty space, 1 represents a space occupied by a piece of the same color, and 2 represents a space occupied by a piece of a different color. Furthermore, recall that a rook can move by moving in one direction (up, down, left or right), for any number of squares - it can't jump over any pieces. The tuple `rookPos` will be of the form `(rookRow, rookCol)`, where `rookRow` and `rookCol` represent the row and col the rook will be placed in the 2D list. As an example, consider the following test case:

```
board = [[0, 1, 0, 0]
          [0, 0, 0, 1]
          [2, 1, 2, 1]
          [2, 0, 1, 1]]
```

```
rookPos = (1, 2)
```

In this case, `countPossibleRookMoves(board, rookPos)` would return `(4, 1)`, since there are four possible moves (positions `(1,0)`, `(1,1)`, `(2,2)`, and `(0,2)`), and one of them attacks a piece on the other team (position `(2,2)`). You may always assume `rookPos` refers to a position on the board that is empty.

Hint: Make sure your function will not crash on an invalid move!

Additional Space 1 for Answer to Question 5

Additional Space 2 for Answer to Question 5

7. (21 points) **Free Response: alrightGridGame**

Assuming the `run()` function is already written for you, write `init`, `keyPressed`, `mousePressed`, `redrawAll`, and `timerFired` so that an animation is created with the following properties:

1. A 5x5 grid is drawn that fills the entire screen (you can assume the width and the height are the same)
2. In its initial state, a blue circle appears inside a randomly chosen cell. This circle should have a random diameter greater than 5px and smaller than the size of the cell.

Hint: use `random.randint()`

3. A score should be displayed in the top left corner of the canvas, starting at 0.
4. If a cell that has a circle inside is clicked, increment the score by 1.
5. If a cell that does not have a circle inside is clicked, decrease the score by 1.
6. Every 100ms, a new red circle should appear in a randomly chosen cell. You don't need to detect whether a circle already appears at that cell - simply draw the new circle on top of the old one.
7. When the up arrow is pressed, increase every circles diameter by 10px. If the circle would be greater than the size of the square, this operation should keep the circle the same size.

To save time, you may wish to abbreviate data as `d`, event as `e`, and canvas as `c`.

Additional Space 1 for Answer to Question 7

Additional Space 2 for Answer to Question 7