15-112 Summer 2019 Quiz 2

Up to 50 minutes. No calculators, no notes, no books, no computers. Style will not be graded on quizzes. Show your work!

1. (20 points) **Code Tracing:** Indicate what the following program prints. Place your answers (and nothing else) in the box under the code.

```python
import copy

def ct1(a):
    b = a
    c = copy.copy(b)
    d = copy.deepcopy(c)
    a[0][0] = 3
    print(a[0].append(7))

    a[1] = c[1]
    c[1] = d[1]
    c[0] = [4]
    b = b + [5]
    b[0].insert(0, 17)
    c[1][0] = 5
    b[1][0] = 6
    d[1][1] = "raptors in 7"

    print("a = ", a)
    print("b = ", b)
    print("c = ", c)
    print("d = ", d)

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

```python
def rc1(L):
    result = []
    while (L != []):
        result.append(L.pop())
        result.append(L.pop(0))
        L = L[2:]
    return (result == [4, 2, 0, -2, -4, -6])
```

[2, 0, 0, -2, 0, 0, -6, -4, 0, 4]

3. (30 points) **Free Response:** Write the function drawLaotianFlag(canvas, x0, y0, flagWidth, flagHeight), which takes in canvas, x0, y0, the width and height of the flag and draws the Laotian flag with x0,y0 as the top left corner. For reference, the Laotian flag is shown below. You may assume that the entire flag is of size flagWidth by flagHeight, and that each red band is exactly 1/4 the height of the entire flag.

Furthermore, the diameter of the inner circle should be 4/5th of the height of the blue band. You may assume that runDrawing is already written for you.
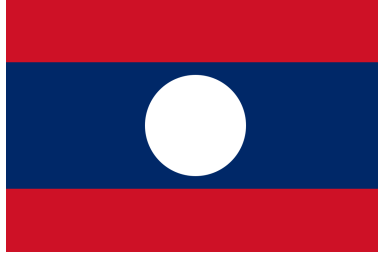
4. (40 points) **Free Response:** Write the function isIncreasingTwistyPath which takes a 2D list of integers as input and returns whether or not the integers are strictly increasing by following the specified path shown below. The path should traverse the columns of the list by going up the first column, then down the second, then up the third, etc., until it reaches the last column. You are guaranteed that the provided list will be rectangular (not ragged). An example is shown below:
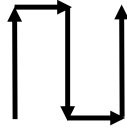

Figure 1: The specified path for the problem

```
lst = [[3, 4, 11],
       [2, 5, 10],
       [1, 6, 7]]
```

In this case, the function would return true, as you would start with 1, traverse the column until you get to 3, and then start with 4 in the next column, go all the way down to 6, and then finally start at 7 in the last column and loop until 11. Hint: you might want to write a helper function to get a specified column in a list.