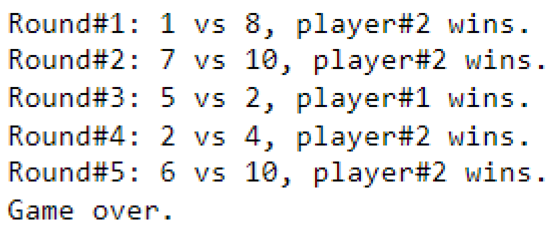
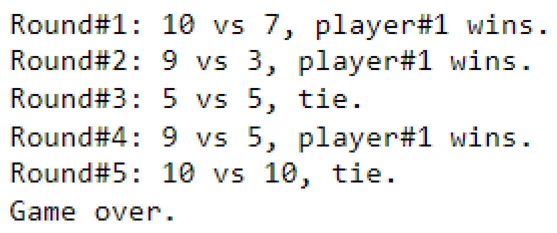
CS128 Homework #4. **Spring 2022**

Task1: 5 rounds of play. Simulate 5 rounds of play for two players. During each round, each player picks a random number between 1 and 10. The player with the higher number wins the round. A tie occurs if the players pick the same number. Display the player's numbers and result for each round. Announce the game is over after 5 rounds of play.

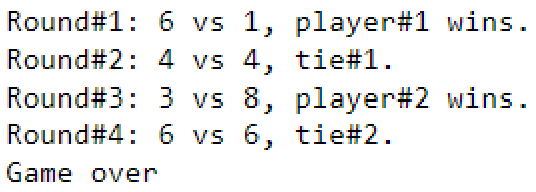


```
Round#1: 1 vs 8, player#2 wins.
Round#2: 7 vs 10, player#2 wins.
Round#3: 5 vs 2, player#1 wins.
Round#4: 2 vs 4, player#2 wins.
Round#5: 6 vs 10, player#2 wins.
Game over.
```

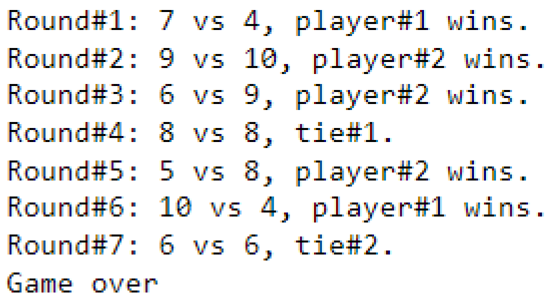


```
Round#1: 10 vs 7, player#1 wins.
Round#2: 9 vs 3, player#1 wins.
Round#3: 5 vs 5, tie.
Round#4: 9 vs 5, player#1 wins.
Round#5: 10 vs 10, tie.
Game over.
```

Task2: Play until second round of ties. Copy the code from task1 and adapt the game so that play continues until two ties have occurred. Announce the game is over after the second round of ties.

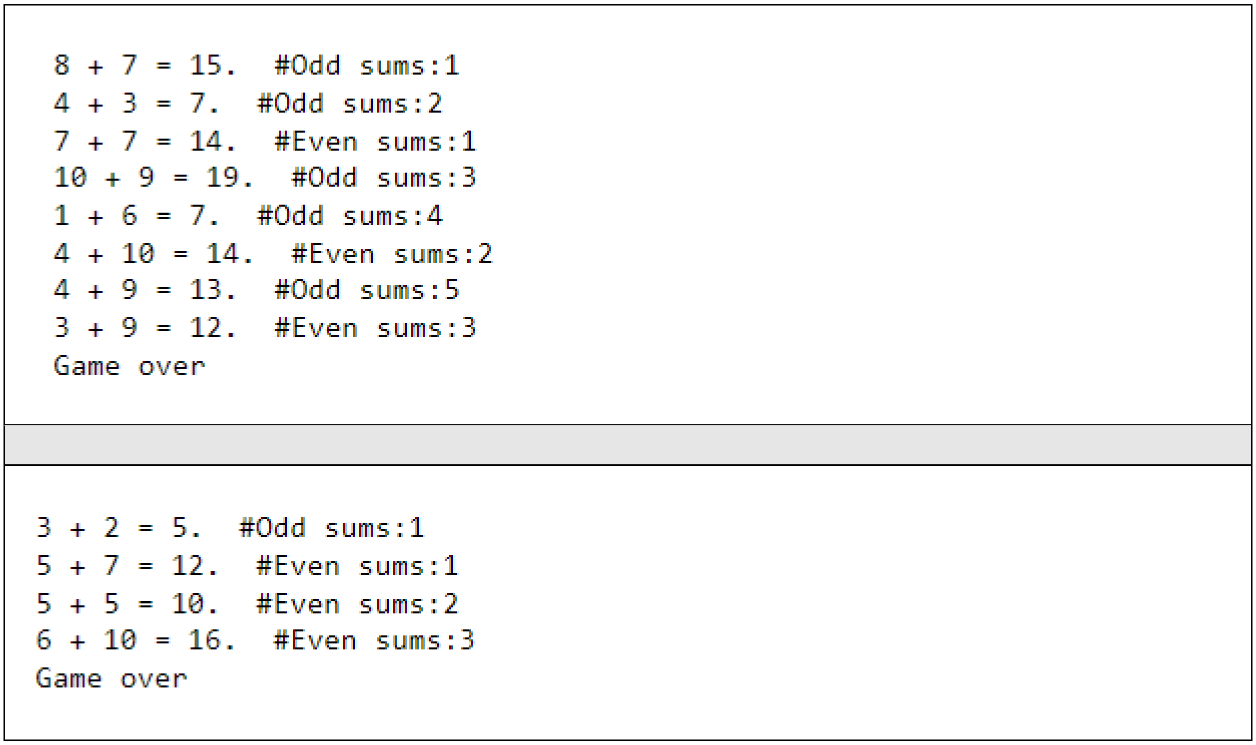


```
Round#1: 6 vs 1, player#1 wins.
Round#2: 4 vs 4, tie#1.
Round#3: 3 vs 8, player#2 wins.
Round#4: 6 vs 6, tie#2.
Game over
```



```
Round#1: 7 vs 4, player#1 wins.
Round#2: 9 vs 10, player#2 wins.
Round#3: 6 vs 9, player#2 wins.
Round#4: 8 vs 8, tie#1.
Round#5: 5 vs 8, player#2 wins.
Round#6: 10 vs 4, player#1 wins.
Round#7: 6 vs 6, tie#2.
Game over
```

Task3: Play until the sum of player numbers is even for 3 rounds. Adapt the game to compute the sum of both player's numbers. Count how many times the sum is even vs odd. Announce the game is over when the count of even sums reaches 3.

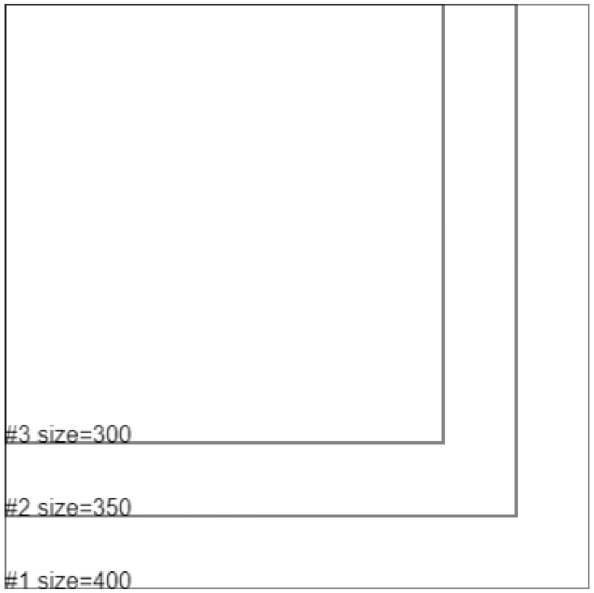


```
8 + 7 = 15.  #Odd sums:1
4 + 3 = 7.  #Odd sums:2
7 + 7 = 14.  #Even sums:1
10 + 9 = 19.  #Odd sums:3
1 + 6 = 7.  #Odd sums:4
4 + 10 = 14.  #Even sums:2
4 + 9 = 13.  #Odd sums:5
3 + 9 = 12.  #Even sums:3
Game over
```

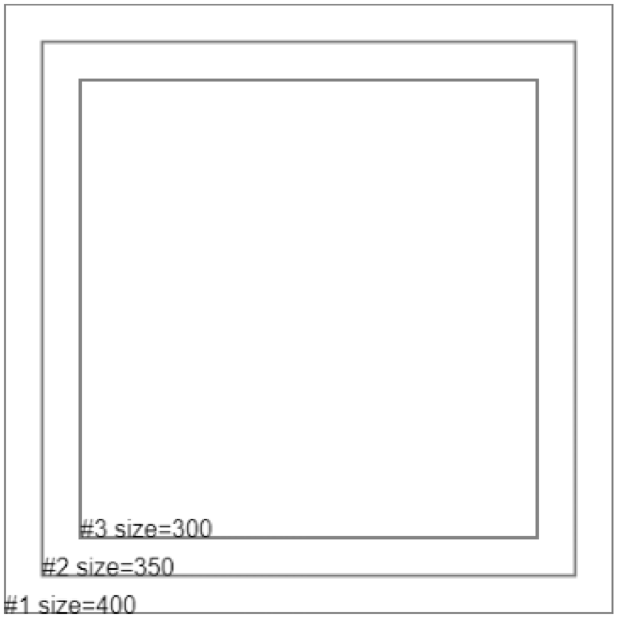
```
3 + 2 = 5.  #Odd sums:1
5 + 7 = 12.  #Even sums:1
5 + 5 = 10.  #Even sums:2
6 + 10 = 16.  #Even sums:3
Game over
```

Task 4a NOTE: You must run the first cell to import the Canvas module prior to running this cell. The existing code creates a 400x400 canvas and draws a square of size 400.

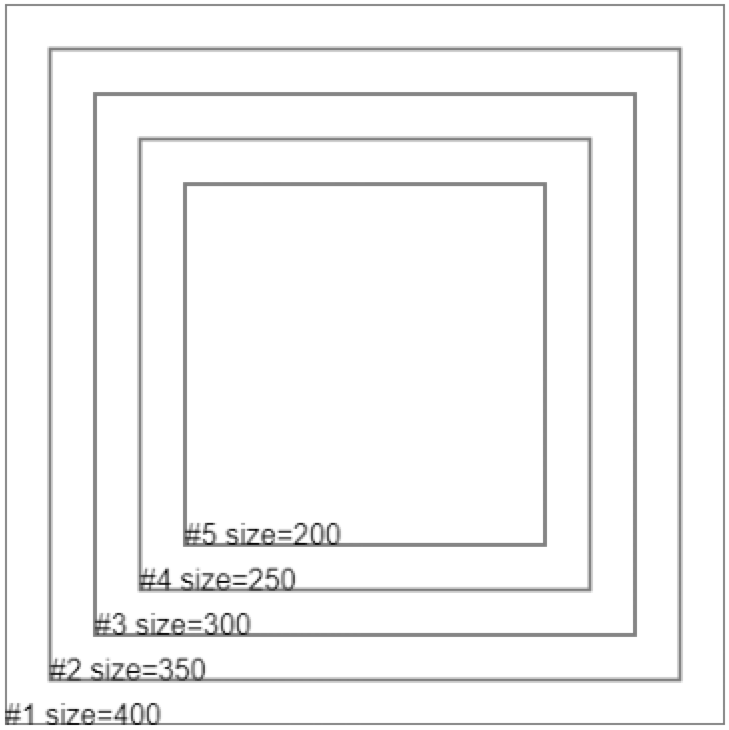
Update the code to draw 3 nested squares, reducing the size of each subsequent square by the shrink amount 50. Draw the largest square first, then the second largest, etc. The squares all have the same upper left corner x,y of 0,0. DO NOT USE A LOOP! This task will help you see the general pattern for drawing nested squares.



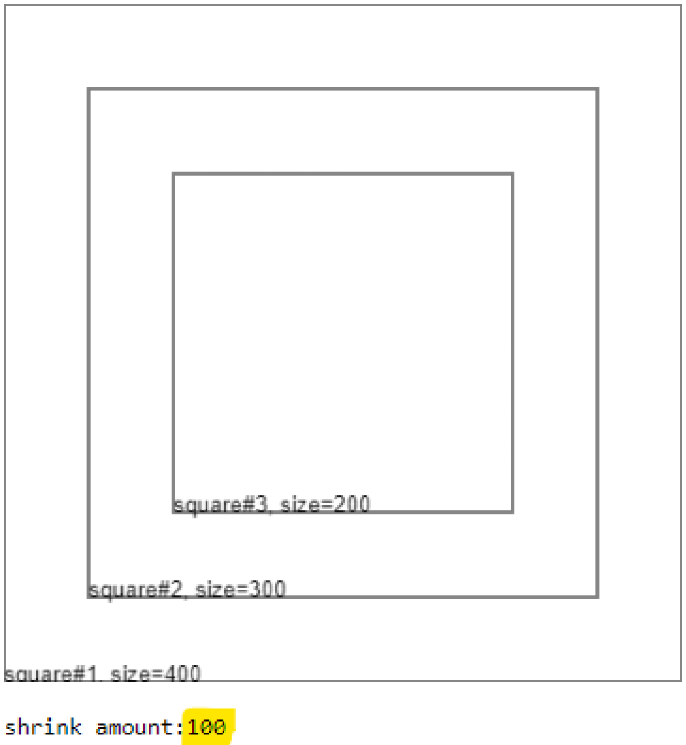
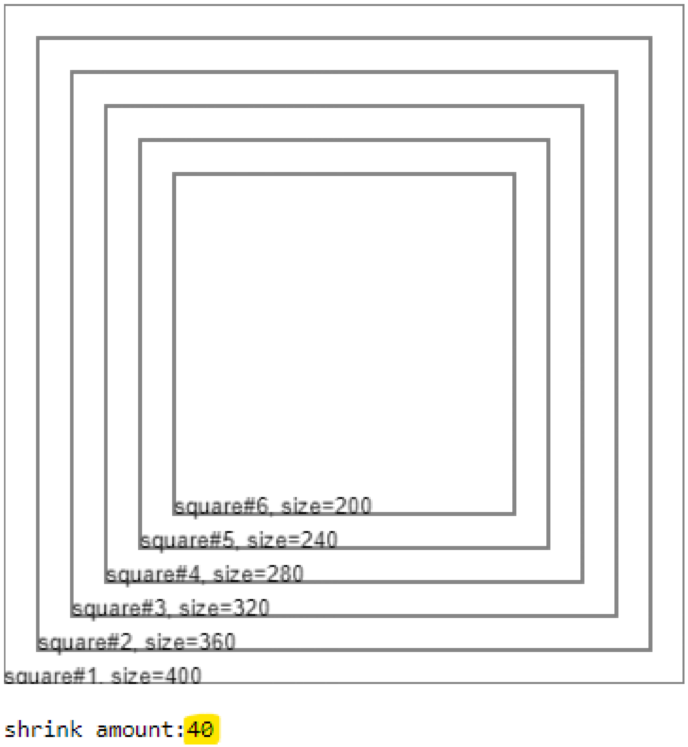
Task 4b Copy the code from task4a into task4b. Update the code to adjust the x,y of each square to appear centered in the canvas as shown.



Task 4c Copy the code from task4b into task4c. Update the code to use a while loop to produce 5 nested squares.



Task 4d Copy the code from task4c into task4d. Prompt the user to enter the shrink amount. Use a while loop to produce nested squares as long as the size is at least 200.

Sample drawing when user enters shrink amount of 100. User input →→	 square#3, size=200 square#2, size=300 square#1, size=400 shrink amount:100
Sample drawing when user enters shrink amount of 40. User input →→	 square#6, size=200 square#5, size=240 square#4, size=280 square#3, size=320 square#2, size=360 square#1, size=400 shrink amount:40

Save your notebook before exporting. Submit hw4.ipynb and hw4.pdf to Canvas.