import random

from collections import Counter

# numbers = [3,3,3, 1, 5]

# print(three\_of\_a\_kind(numbers))

# quit()

dice\_rolls = []

score = 0

for i in range(5):

die = random.randint(1, 6)

dice\_rolls.append(die)

print(dice\_rolls)

for i in range(3):

indices = input("Choose a dice index between 0 and 4: ")

if indices == "":

break

else:

indices = indices.replace(",", " ")

indices = indices.split(" ")

print("Rerolling die at indexes:", indices)

for index in indices:

# if they type in a number, convert the number to an integer

index = int(index)

die = dice\_rolls[index]

die = random.randint(1, 6)

dice\_rolls[index] = die

print(dice\_rolls)

print(dice\_rolls)

def check\_if\_duplicates\_1(dice\_rolls):

if len(dice\_rolls) == len(set(dice\_rolls)):

return False

else:

return True

result = check\_if\_duplicates\_1

if result:

print("Yes list contains duplicates")

else:

print("No duplicates found")

c = Counter(dice\_rolls)

print(c)

class Roll:

def \_\_init\_\_(self):

pass

def check\_one\_pair(self, dice\_rolls):

pass

def check\_two\_pairs(self, dice\_rolls):

pass

def check\_three\_kind(self, dice\_rolls):

pass

def check\_four\_kind(self, dice\_rolls):

pass

def check\_low\_straight(self, dice\_rolls):

pass

def check\_high\_straight(self, dice\_rolls):

dice\_list.sort()

dice\_list[0] == 2

if len(set(dice\_list)) == 5 and dice\_list[4] == 6 and

return True

else:

return False

def check\_full\_house(self,dice\_rolls):

dice\_rolls.sort

if (len(set(dice\_rolls))) != 2:

return False

elif dice\_rolls[0] != dice\_rolls[3] or dice\_rolls[1] != dice\_rolls[4]:

return True

def add\_chance(self, dice\_rolls):

pass

def check\_yatzy(self, dice\_rolls):

if len(set(dice\_rolls)) == 1:

return True

return False

dices = Roll()

assert dices.check\_yatzy([1,1,1,1,1]) == True

assert dices.check\_yatzy([1,1,1,2,1]) == False

print(dices)