

```

# Sean MacBride
# Program: smacbrideP5.py
# Description: A program that simulates a european roulette table at a casino, where you can bet
in 5$ increments.
# Input: Your starting bankroll, the amount you are willing to bet for bet 1, where you would like
to bet for bet 1 (Must be a number 0-36 for numbers, R or B for Colors, or X Y or Z for rows),
the amount you would like to bet for bet 2 and where you would like to bet it, if applicable. The
amount you are betting will repeat until you input 0$ for your first bet.
# Output: The result of the roulette spin, with the appropriate effect on your bankroll, before
prompting you to bet again. Will repeat until the input of bet 1 is 0. At that point, will print out
the final bankroll value

```

```

#Importing random
import random as rand

```

```

#A function that calculates the winnings of a particular bet.
#Takes the bankroll, the location of the bet, and the amount bet at that location as parameters.
#Returns the updated bankroll
#This function is only called if a bet is a winner
def winnings(bankroll, betspot,betval):
    if betspot<=36 or betspot==42: #Winnings calculation for a number bet
        winnings=36*betval
    elif betspot<=38: #Winnings calculation for a color bet
        winnings=2*betval
    elif betspot<=41: #Winnings calculation for a row bet
        winnings=3*betval
    bankroll+=winnings
    return bankroll #Returning the bankroll

```

```

#A function to check if a spin is a win based on the bet
#Takes the spin number, the spin color, the spin row, and location of the bet as parameters
#Returns True or False if it is a win or not a win
def checkWin(spinnumber,spincolor,spinrow,betspot):
    if betspot==spinnumber or betspot==spincolor or betspot==spinrow:
        return True
    else:
        return False

```

```

#A function to Spin the wheel
#Takes no parameters
#Returns a numerical value for the row, color, and number
def getSpin():
    numberval=rand.randrange(0,37,1) #The random number generator
    redlist=[1,3,5,7,9,12,14,16,18,19,21,23,25,27,30,32,34,36] #List of all red numbers
    blacklist=[2,4,6,8,10,11,13,15,17,20,22,24,26,28,29,31,33,35] #List of all black numbers
    if numberval in redlist: #If statements to determine the color of the number

```

```

    color=38 #using spec sheet numeric representation
elif numberval in blacklist:
    color=37 #using spec sheet numeric representation
else:
    color=42 #letting the number 42 as a color value equal green
    if color!=42: #Making sure that the color is not green. If it isn't green, The row will return 42.
Just a way to save not going through the loop
    if numberval%3==0: #If statements to determine the row of the number
        row=41
    elif numberval%3==1:
        row=39
    elif numberval%3==2:
        row=40
else:
    row=42 #Assigning the row value=42 for a green slot
    #I used the number values you gave in the spec sheet in my code, and added row and color
    values of 42 to be attributed to green
    #Returning the number, color, and row values
    return numberval, color, row

#The controller function that asks for the bankroll
#Takes no parameters
#Calls the wager function, which does most of the work
def controller():
    #Asking for the first bankroll
    print()
    bankroll=int(input("Enter Your Starting Bankroll! $"))
    wager(bankroll)

#The converter function that helps convert bet placement inputs
#Takes the location of the bet as an input
#Returns a numerical value of the betspot
#I used the numerical values given in the spec sheet, with the exception of green, which has
number 0, row 42, and color 42
def converter(betspot):
    if betspot=="B":
        return 37
    elif betspot=="R":
        return 38
    elif betspot=="X":
        return 39
    elif betspot=="Y":
        return 40
    elif betspot=="Z":
        return 41
    else:

```

```
    return eval(betspot)
```

```
#A function that returns the finished string of the roulette spin
```

```
#Takes the number and color of the spin as parameters
```

```
#returns the finished string of the result of the roulette spin
```

```
def stringer(number,color):
```

```
    if color==37:
```

```
        return str(number)+" Black"
```

```
    elif color==38:
```

```
        return str(number)+" Red"
```

```
    else:
```

```
        return "0 Green"
```

```
#The wager function, which does most of the work with print statements and calling other functions
```

```
#Takes the bankroll for parameter
```

```
#Outputs the bet amounts, bet locations, results of the bets, and repeats until you enter 0 as you first bet amount
```

```
def wager(bankroll):
```

```
    print() #A print statement for nice formatting
```

```
    bet1amount=int(input("First bet amount : $")) #Asking for the first bet amount
```

```
    while bet1amount!=0: #Running a loop that will repeat until you enter 0 in bet1amount (at the end of the loop)
```

```
        bet1point=input("Name your bet location : ") #Asking for the location of bet1
```

```
        bankroll=bankroll-bet1amount #initially updating the bankroll
```

```
        bet1num=converter(bet1point) #Calling the converter function that converts the location of bet1 to a numeric value, makes it easier to deal with
```

```
        bet2amount=int(input("Second bet amount : $")) #Asking for a second bet
```

```
        if bet2amount!=0: #Similar to the first loop, but this time will check to see if bet2amount is not 0. If it is 0, there's no need to ask for the location, and convert it to a numeric value, or update the bankroll
```

```
            bet2point=input("Name your bet location : ") #Asking for the location of bet2
```

```
            bankroll=bankroll-bet2amount #Updating the bankroll from bet2
```

```
            bet2num=converter(bet2point) #Calling the converter function that converts the location of bet2 to a numeric value, makes it easier to deal with
```

```
            spinnumber, spincolor, spinrow = getSpin() #Spinning the wheel with the getSpin function and getting the values of the wheel
```

```
            spinstring=stringer(spinnumber,spincolor) #Calling the stringer function and returning it to get the final string value
```

```
            result1=checkWin(spinnumber,spincolor,spinrow,bet1num) #The result of the first bet
```

```
            if bet2amount!=0: #As long as bet2 is not 0, will check to see the result of the second bet
```

```
                result2=checkWin(spinnumber,spincolor,spinrow,bet2num) #The result of the second bet
```

```
            else:
```

```
                result2=False #Letting result2=false for a loop later in the code, as to not create any "referenced before assignment" errors
```

```
                print() #a print statement for nice formatting
```

```

    if result1==True or result2==True: #Printing the results of the bet if it won
        print("RESULT - ", spistring, " - WINNER", sep="") #The winning print statment
        if result1==True: #Calling the winnings function to update the bankroll if result1 was a
winner
            bankroll=winnings(bankroll,bet1num,bet1amount)
        if result2==True: #Calling the winnings function to update the bankroll if result2 was a
winner
            bankroll=winnings(bankroll,bet2num,bet2amount)
    else: #A print statement for a spin where you did not win on either bet
        print("RESULT - ", spistring, " - NO WIN", sep="")
    print() #a print statement for nice formatting
    print("Bankroll: $",bankroll,sep="") #The updated bankroll from your bet
    print() #a print statement for nice formatting
    bet1amount=int(input("First bet amount      : $")) #Asking for the first bet again
    print() #a print statement for nice formatting
    print("Final Bankroll: $",bankroll,". Thanks for playing!", sep="") #A print statement of the
final bankroll

```

```

#Main, which calls controller
def main():
    controller()
main()

```

#I have abided by the Wheaton Honor Code in this work