

CPSC 1301, Computer Science I Programming Exercise

Week 13, Roulette.py

Playing Roulette is a very simple game. A roulette wheel has 38 bins. Thirty-six bins are numbered from 1 to 36. The last two bins contain 0 and 00. The two zero bins are colored green. The others are colored randomly red and black, 18 of each color. As the wheel spins, a ball is dropped into the wheel. When the wheel stops spinning, the ball comes to rest in one bin. See Figure 1.

Betting Betting is much more complicated. Bets are pictured in Figure 2. The following bets can win:

1. Straight: the number of the bin
2. Dozens: row thirds, 1 – 12, 13 – 24, 25 – 36
3. HighLow: low (1 – 18) or high (19 – 38) numbers.
4. Street: rows, e.g., 1/2/3 or 22/23/24
5. Columns: first, second, or third columns
6. OddsEvens: even or odd numbers
7. Colors: red or black colored numbers
8. Split: at the edge of any two contiguous numbers, e.g., 1/2, 11/14, and 35/36
9. DoubleStreet: double rows, e.g., 1/2/3/4/5/6 or 22/23/24/25/26/26
10. Squares: at the intersection of any four contiguous numbers, e.g., 1/2/4/5, or 23/24/26/27

Figure 1: Roulette Wheel



Figure 2: Roulette Bets

00	3	6	9	12	15	18	21	24	27	30	33	36	2 to 1
	2	5	8	11	14	17	20	23	26	29	32	35	2 to 1
	1	4	7	10	13	16	19	22	25	28	31	34	2 to 1
0	1st 12				2nd 12				3rd 12				
	1 to 18		EVEN		RED		BLACK		ODD		19 to 36		

Assignment You are to write a program that models a roulette wheel. The ball can fall randomly into one of 38 different bins. You are to calculate all the winning bets according to the bin the ball falls into.

Use arrays to model this game. You should use two arrays to model the wheel (numbers and colors). You can use arrays as appropriate to determine the winning bets.

As output, your program should print the winning bin to standard output, together with all the winning bets. For example, if the ball comes to rest in square 26, the winning Split bets could be 23/26, 26/27, 26/29, and 25/26, while the winning Corner bets could be 22/23/25/26, 23/24/26/27, 25/26/28/29, and 26/27/29/30.

Grading There are ten bets in this exercise. Each bet carries ten points. You will receive 10 points for every bet implemented, for a total of 100 points.

1 Exercise requirements

1.1 Part 1, Setup

This module sets up the game. I have written it as follows:

make_wheel() Declares and initializes a 38 element array representing the Roulette wheel, from '0' to '00'.

make_rb() Declares and initializes a 38 element array with two Green slots, and 18 each of Red and Black slots. Greens are at '0' and '00', while Reds and Blacks are randomized.

print_wheel() Prints the Roulette wheel, both slots and colors.

Wins This implements the winning values for ten kinds of bets:

- Straight()
- Split()
- Street()
- Square()
- DoubleStreet()
- Columns()
- Dozens()
- LowsHighs()
- EvensOdds()
- RedsBlacks()

1.2 Part 2, Play

This module contains the routines that play Roulette. I have written it as follows:

init() Initializes the game.

spinning() “Spins” the wheel. This generates a random slot.

play() This function allows users to play Roulette().

betting() This function allows users to make bets.

Outcomes This implements the winning outcomes for the bets placed by th player.

- Straight()
- Split()
- Street()
- Square()
- DoubleStreet()
- Columns()
- Dozens()
- LowsHighs()
- EvensOdds()
- RedsBlacks()

1.3 Part 3, Main

In your “Main” program (name it `Roulette_lastname.py`), you should import the modules you have written, other necessary modules, and call the appropriate routines to play Roulette.

2 Starter template

There is no starter template. Write this exercise as you have written the others.

3 Sample output

See the demonstration in class.