

While Loops Challenge 30: Power Ball Simulator App

Description:

You are responsible for writing a program that simulates the Power Ball Lottery. The traditional power ball is played by randomly choosing 5 white balls numbered 1 through 69 then randomly choosing 1 red ball numbered 1 through 26. The traditional power ball has astronomically low odds of winning. Therefore, your program will allow users to adjust the odds by setting how many balls the lottery will choose from. Your program will then calculate the odds the user has of winning. The user will purchase tickets in a set interval, without purchasing repeated losing tickets, until they either win the lottery or choose to give up.

Step By Step Guide:

- Print a welcome message as formatted below.
- The normal Power Ball Lottery chooses 5 random white balls numbered 1 through 69.
- Once a ball is chosen, it cannot be chosen again.
- Using all 69 balls will make winning the lottery nearly impossible. Therefore, we want to give the user the option to set how many white balls they will pick from.
- Get user input for the number of white balls to use.
 - Store this user input in a variable called `white_balls`.
- If the user entered a number less than 5, set the value of `white_balls` to five.
 - This ensures that we have enough numbers to choose from.
- Similarly, the red Power Ball is normally chosen from red balls numbered 1 through 26.
- We want to give the user the option to set how many red balls they will pick from.
- Get user input for the number of red balls to use.
 - Store this input in a variable called `red_balls`.
- If the user entered a number less than 1, set the value of `red_balls` to one.
 - This ensures we have at least one power ball.
- Generate the odds the user will win this specific lottery.
- Create a variable called `odds` and set it equal to 1.
 - You will use this variable to perform repeated multiplication.
- Use a for loop to loop an appropriate number of times to perform the correct multiplication. Your for loop should be a numerical for loop such as `for i in range(5)`:
 - For example, if the user wanted 69 white balls and 26 red balls the odds would be calculated as follows: $(69 \times 68 \times 67 \times 66 \times 65) \times 26 / 120$.
 - For example, if the user wanted 20 white balls and 4 red balls the odds would be calculated as follows: $(20 \times 19 \times 18 \times 17 \times 16) \times 4 / 120$.
 - Try to determine the relationship between the number of white balls used, the value of `i` in the for loop, and the terms being multiplied together to get the odds.
- Determine the odds for any specific lottery given and print the odds to the user.

- We will allow users to purchase tickets in a set interval.
- Get user input for how many tickets they would like to purchase in each interval.
 - Store this input in a variable called `ticket_interval`.
- Next, we need to simulate creating the winning lotto numbers.
- Each lottery ball will be represented by a random integer.
 - Type `import random` as the first line of code in your program.
- Create a blank list called `winning_numbers`.
- Generate the numbers to represent the white balls.
 - While there are less than 5 values in `winning_numbers`:
 - Generate a random integer from 1 up to the users value of `white_balls`.
 - If the integer does not already appear in the list `winning_numbers`:
 - Append the integer to the list `winning_numbers`. We cannot have repeated values.
- Permanently sort the list `winning_numbers`.
- Generate the number to represent the red power ball.
 - Generate a random integer from 1 up to the users value of `red_balls`.
 - Append this to the end of the list `winning_numbers`.
 - Do not sort the list as the red power ball is always at the end of the winning numbers.
- You now have a list of numbers, `winning_numbers` that represent the winning Power Ball numbers; 5 numbers that are ordered and then one final Power Ball number added at the end.
 - For example `[10, 22, 24, 54, 67, 3]`
- Now, simulate the actual Power Ball drawing.
- Print a welcome message as formatted below.
- Display the winning power ball numbers all on one line as formatted below.
 - Use the `end=` argument.
- Prompt the user to press enter to being purchasing tickets.
- Create a variable called `tickets_purchased` and set it equal to 0.
 - This will keep track of how many tickets have been created.
 - Every time your while loop runs you should increment this by 1.
- Create a flag variable called `active` and set it to `True`.
 - This variable will stay `True` as long as the user wants to continue to purchase tickets. After, the number of tickets stored in `ticket_interval` have been created, ask the user if they want to purchase more. If they do not, set this variable to `False`.
- Create a blank list called `tickets_sold`.
 - This will keep track of every ticket that has been sold.

- We will use this list to make sure that we don't purchase a losing ticket more than once.
- Use a while loop that will continue to run until a winning ticket is picked or user wants to stop wasting their money. This loop needs to check two conditions: The winning ticket has not been sold and the user still wants to play.
 - Create a blank list called `lottery_numbers` which will hold the current lottery numbers of a ticket that is being purchased.
 - Once the ticket is created and determined to be a loser, you can use this same variable for the next ticket.
 - Simulate picking a lottery ticket.
 - You should do this in a similar manner to picking the actual power ball numbers. Pick 5 numbers, sort them, and then add a 6th.
 - If this current ticket is not in your list of tickets sold:
 - Increase the number of purchased tickets by 1.
 - Add it to the list of tickets sold,
 - Print the ticket as a list.
 - Else:
 - Print a message that a losing ticket was generated that has already been purchased and that the ticket is being disregarded.
 - After picking the number of tickets in `ticket_interval`, which can be checked using modulo division:
 - Print the total number of tickets purchased.
 - Get user input for if they would like to continue purchasing tickets.
 - If the user does not want to continue:
 - Set your flag variable to False
- The while loop will end either by purchasing the winning ticket or by giving up.
- If the current lottery ticket is equal to the winning ticket:
 - You won the lottery!
 - Print the winning ticket numbers.
 - Print the total number of tickets purchased.
- Else:
 - You didn't win the lottery.
 - Print how many tickets were purchased.
- Use at least 2 comments to describe sections of your code.
- "Chunk" your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output 1:

```
-----Power-Ball Simulator-----
```

How many white-balls to draw from for the 5 winning numbers (Normally 69): 7
How many red-balls to draw from for the Power-Ball (Normally 26): 1
You have a 1 in 21.0 chance of winning this lottery.
Purchase tickets in what interval: 5

-----Welcome to the Power-Ball-----

Tonight's winning numbers are: 1 2 3 5 7 1
Press 'Enter' to begin purchasing tickets!!!
[2, 3, 5, 6, 7, 1]
[1, 3, 4, 5, 6, 1]
[1, 2, 4, 5, 6, 1]
[1, 3, 4, 5, 7, 1]
Losing ticket generated; disregard...
[1, 2, 5, 6, 7, 1]
5 tickets purchased so far with no winners...

Keep purchasing tickets (y/n): y
[1, 4, 5, 6, 7, 1]
[1, 3, 5, 6, 7, 1]
[1, 2, 3, 5, 7, 1]

Winning ticket numbers: 1 2 3 5 7 1
Purchased a total of 8 tickets.

Example Output 2:

-----Power-Ball Simulator-----

How many white-balls to draw from for the 5 winning numbers (Normally 69): 69
How many red-balls to draw from for the Power-Ball (Normally 26): 26
You have a 1 in 292201338.0 chance of winning this lottery.
Purchase tickets in what interval: 20

-----Welcome to the Power-Ball-----

Tonight's winning numbers are: 31 37 55 57 61 12
Press 'Enter' to begin purchasing tickets!!!
[11, 36, 40, 50, 52, 10]
[8, 20, 45, 49, 51, 14]
[35, 42, 57, 60, 67, 3]
[19, 21, 28, 53, 67, 12]
[5, 17, 22, 59, 67, 18]

[1, 3, 12, 15, 50, 17]
[18, 44, 45, 54, 66, 24]
[6, 33, 47, 55, 61, 16]
[28, 35, 36, 47, 52, 5]
[7, 9, 37, 42, 57, 24]
[12, 14, 21, 56, 60, 18]
[20, 52, 55, 62, 69, 16]
[11, 38, 41, 61, 63, 17]
[5, 24, 25, 33, 37, 7]
[36, 38, 44, 51, 62, 15]
[20, 23, 59, 60, 66, 20]
[4, 9, 23, 29, 33, 21]
[11, 14, 35, 36, 60, 3]
[7, 19, 48, 50, 68, 26]
[6, 24, 27, 37, 58, 18]

20 tickets purchased so far with no winners...

Keep purchasing tickets (y/n): y

[4, 8, 14, 22, 68, 16]
[3, 24, 34, 41, 57, 12]
[30, 36, 39, 40, 59, 16]
[6, 23, 33, 34, 50, 26]
[1, 19, 54, 60, 61, 23]
[1, 35, 43, 52, 54, 13]
[3, 6, 7, 37, 50, 13]
[2, 6, 22, 28, 47, 25]
[6, 19, 46, 55, 60, 5]
[8, 44, 50, 52, 61, 2]
[39, 44, 52, 54, 64, 17]
[16, 19, 24, 27, 49, 13]
[18, 24, 27, 34, 68, 12]
[15, 19, 27, 44, 57, 23]
[40, 42, 47, 55, 65, 10]

[13, 30, 36, 45, 59, 17]
[4, 16, 22, 28, 57, 2]
[17, 27, 51, 65, 66, 14]
[7, 15, 22, 43, 53, 4]
[11, 18, 20, 30, 53, 21]

40 tickets purchased so far with no winners...

Keep purchasing tickets (y/n): n

You bought 40 tickets and still lost!
Better luck next time!