Published

Week 6: Loops 11 of 11

6 Week 6 Assignment

Computer SCIENCE

IMPORTANT: Please review the statement on Academic Integrity. Submitting code that is not your own may result in an automatic F for the class.
Statement on Academic Integrity

REMINDER: There is extra credit available. It must be done before you start this assignment: **PyPhanon Extra Credit**

You will complete two programming tasks in this assignment based on the new things you learned in this unit. You may complete the tasks as you complete certain sections or finish them all at once at the end of the unit. Make sure to use proper naming conventions for your project and name the files for each task as specified below. As a reminder, the naming convention for the project should be <LastName>-<FirstName>-Assn6. When complete, you will create a zip file from the project which will have the same name with a *.zip* at the end.

Objectives

1. While loop
2. For loop
3. Range()
4. Break
5. Continue

Task 1 - Fluky Numbers

Save your program in a file called task1.py and your software development plan in a file called plan1.txt. You will write a program that will find Fluky Numbers. A Fluky Number is a number that is equal to a random number, where the random number is the first random number from generated after seeding a random number generator with the sum of all factors of a number, not including itself. That's quite a mouthful, so here it is again in simpler terms. All of the following requirements must be met for a number to be a Fluky Number. [NOTE: It might be useful to think

of steps 1-5 backwards, meaning the first thing you need is a factor before you can do anything else]

1. A Fluky Number is a number, called The Number, that is equal to a randomly generated number.
2. The random number is generated with a random number function. Use randint() from the random module.
3. The randomly generated number is in the range of 0 to the The Number, inclusive. (Ex. If The Number is 10, the random number can be 0 through 10, including 0 and 10)
4. The random number is the first number generated after setting a specific seed value (Use the random.seed() function).
5. The seed value is the sum of all the factors of The Number, not including itself. (Ex. The factors of the number 10 that will be used for the seed are 1, 2, and 5)
6. A number must have a seed value (meaning the sum of factors) greater than 0. [Really this means that 1 cannot be a Fluky number because its only factor is 1, but the sum may not include The Number itself. Your code must determine this in code, you cannot just start at 2, or have a conditional to skip the iteration if The Number == 1]

Example

The number 25 is a Fluky Number. The factors of 25, not including itself, are 1 and 5. The sum of 1 and 5 is 6. The first random number from 0 to 25, after using 6 as a seed, is 25. Thus, 25 is a Fluky Number. Here is some code to help you: **seeddemo.py** ⬇ **(https://usu.instructure.com/courses/681553/files/82266323/download?download_frd=1)**

Requirements

You must write a program to find all Fluky Numbers from 1 to 10000.

Your program may know that there are 7 Fluky numbers between 1 and 10000.

Your program must print the total running time in seconds to two decimal places. It must print the total number of inner loops executed. Here is an example output, without actual numbers showing: (Note: An 'X' does not indicate the number of digits or any other meaning. It's just a filler)

```
Fluky Number: XX
Fluky Number: XX
Fluky Number: XX
Fluky Number: XX
Fluky Number: XX
Fluky Number: XX
Fluky Number: XX
Total Time:  XX.XX seconds
Total Loops:  XXXXXXXXXX
```

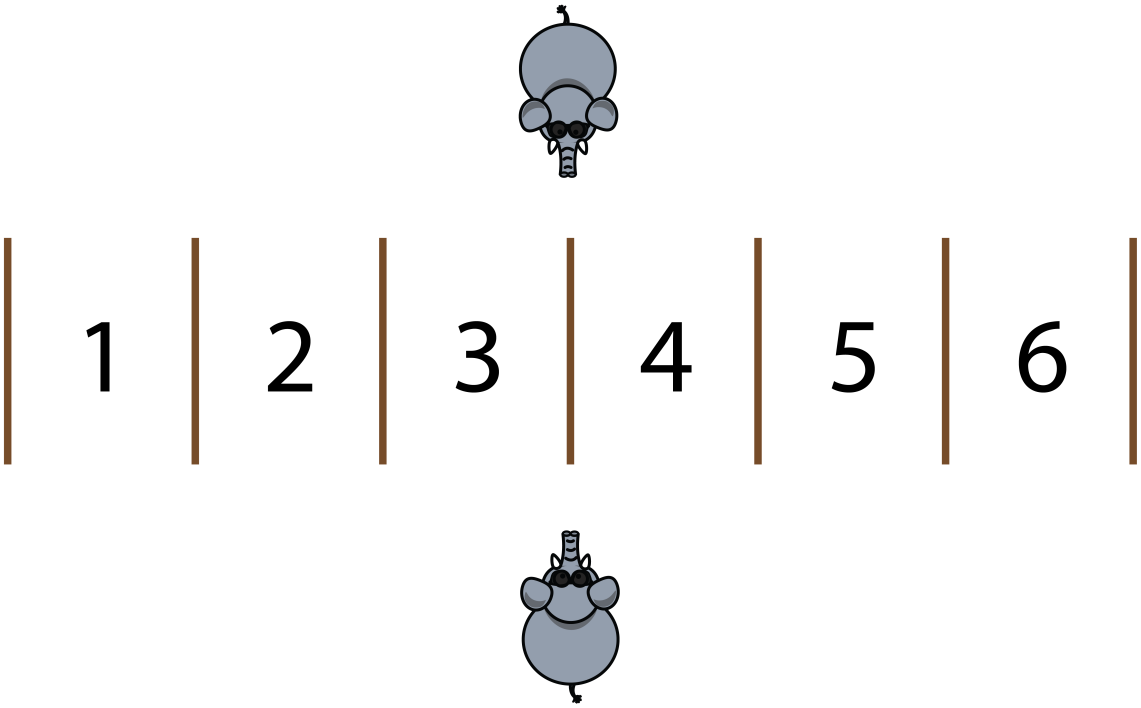
See Rubric for all requirements. Make sure to catch the points for reducing the number of required loops.

NOTES

1. Your code must discover all seven Fluky Numbers using a nested loop structure.
2. You cannot use any information about properties of Fluky Numbers other than what is described. Your program may know that there will be exactly seven Fluky Numbers numbers with values of 10,000 or less.
3. You cannot start checking at 25, just because the instructions mentioned that 25 is a Fluky Number. It may or not be the first fluky number, actually. Your program must start checking at number 1.
4. You may use basic math/logic principles to help you reduce the number of iterations. If you're reading through wikipedia or doing google searches then you are over-thinking it. Think through in your planning how you can reduce the number of iterations without affecting the results...and think simply.
5. Big Hint: One approach to minimizing the number of iterations is to stop when you have found 7 Fluky Numbers. The other approaches are similar to this level of analysis.

Task 2 - The Zookeeper Statistician

Save your program in a file called task2.py and your software development plan in a file called plan2.txt. Here is a quick story about two elephants who live in the Potowatomi Zoo. Every evening the elephants each randomly select one of six sleeping pens to spend the night in. After deciding on which pen, they simultaneously enter the pen from opposite sides. Being slightly vision impaired, they cannot see if the other elephant entered the same pen until it is too late to turn around and go to another pen to spend the night. Thus, sometimes they must each spend an uncomfortable night sharing a small sleeping area with another elephant.



After the elephants are asleep the zookeeper randomly picks one of the pens to check. During a break, the zookeeper and the nighttime custodian were talking. The zookeeper told the custodian that 1/3rd of the time when he checks a pen, there's at least one elephant in there, and of the times that there's at least one elephant in the pen, 1/6th of the time the other elephant is in there as well. The custodian said that those percentages cannot be correct, but the zookeeper insisted they are.

Requirements

Your job is to write a program to simulate this nightly activity 100,000 times to determine if the zookeeper or the custodian is correct. As a simulation, you will program all steps of this process. You are not writing a statistical analysis program to calculate probabilities.

You will report the following statistics.

1. What percentage of the time there is at least one elephant in the pen the zookeeper checks?
2. When there is an elephant in the pen the zookeeper checks, what percentage of time are both elephants in the pen?
3. Based on the results of your simulation, is the zookeeper or the custodian correct? (Use 2% points as the margin of error)

Your results should be displayed as percentages with two decimal places.

After your results are displayed, you will prompt the user to choose to end the program or run the simulation again. That way, whomever the results indicate is wrong, can easily try another simulation to see if the results were a fluke. You can use a simple message such as:

```
Run the simulation again? (yes or no):
```

Read the rubric for all requirements.

TIPS

1. Your software design is very important. Make sure you've planned for every requirement and thought of a way to code each requirement individually and step-by-step. Then you can increase your confidence that your simulation is correct.
2. Double check the description for what you're looking for when both elephants are in the same pen.

Points 100

Submitting a file upload

Due	For	Available from	Until
Oct 18, 2021	Everyone	-	-

Assignment 6

Criteria	Ratings		Pts
Task 1: Software Development Plan Test cases not required	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Find the 7 Fluky Numbers Properly No extra information assumed other than what is in the description.	**20 pts** **Full Marks**	**0 pts** **No Marks**	20 pts
Task 1: Print total running time message print in seconds with two decimal places	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Print total number of iterations of inner loop	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Fewer than 48,500,000 loops	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Fewer than 1,800,000 loops	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Fewer than Fewer than 900,000 loops	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Fewer than 55,000 loops	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Proper course coding conventions	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 2: Software Development Plan Test cases not required	**3 pts** **Full Marks**	**0 pts** **No Marks**	3 pts
Task 2: Results are accurate You must rely on the accuracy of your simulation. See TIP in assignment description	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts

Criteria	Ratings		Pts
Task 2: Comment properly indicate code that determines presence of elephant and presence of both elephants	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 2: Use random numbers properly	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 2: Code logic is accurate This means that it actually simulates all parts of the problem. It does not determine results using statistical analysis/formulas.	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 2: Proper course coding conventions	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 2: Ask user if they want to play again Start over if they enter non-case-sensitive "yes". This means they can enter YES, Yes, yes, YeS, or any combination of capitalization. End game with a message if they answer anything else	**7 pts Full Marks**	**0 pts No Marks**	7 pts
			Total Points: 100