CS 1400 – Spring 2019
Assignment #8

Introduction
You will complete two programs to give you experience with:

- While loop
- For loop
- Range()
- Break
- Continue
-

Task 1
Programming Exercise 5.35

Complete on the Task 1 tab. Make sure to read the rubric to see the additional requirements.

Rubric
3 pts: Software Development Lifecycle Plan
10 pts: Find the four perfect numbers
5 pts: Print the total running time, in seconds, of the program
10 pts: At the end of the program print the total number of iterations of the inner loop that were completed
4 pts: Fewer than 48,500,000 iterations of the inner loop are completed
4 pts: Fewer than 24,900,000 iterations of the inner loop are completed
4 pts: Fewer than 16,000,000 iterations of the inner loop are completed
5 pts: Follow proper coding conventions

NOTE: Your code must discover all four perfect numbers (This means that you cannot use the information about 6 and 28 that the book mentions) using a nested loop structure. You cannot use any information about properties of perfect numbers other than what is described. Your program may know that there will be exactly four perfect numbers with values of 10,000 or less. You may use basic math principles to help you reduce the number of iterations. If you're reading through wikipedia or doing google searches about perfect numbers or trying to discover prime numbers then you are over-thinking it (The #1 problem for students who had trouble in the past). Think through in your planning how you can reduce the number of iterations without affecting the results...and think simply.

Task 2
In 1990, Marilyn vos Savant answered a letter in her *Ask Marilyn* column in the *Parade* magazine. Her response caused a flood of letters to come in, telling her that she was wrong. In the end, she proved that she was correct. You are going to write a program to prove she was right.

Here's the problem:
Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?

In short, it is to your advantage to switch. You can read the details here: https://en.wikipedia.org/wiki/Monty_Hall_problem

Your task is to write a program to simulate the game. This means your program will automatically play the game 100,000 times. The only input the user will give is if they want to stay with their original pick, or switch picks after the first goat is revealed. Their pick will be the decision for all 100,000 simulations.

Use random numbers to decide which door hides the car, and for the user's initial selection for each iteration of the game. You will save a ton of time if you do the Software Development Lifecycle Plan first. Think through the process. It is not hard if you plan it out. Do this on the Task 2 tab.

Rubric
3 pts: Software Development Lifecycle Plan
5 pts: User is prompted for input (stay or switch)
10 pts: Display the number of times won as well as percentage won
10 pts: Winning percentage is what it should be (Ask Marilyn if you have a question)
5 pts: Use random numbers properly
10 pts: Code logic is accurate. This means that it actually simulates all parts of the game. It does not determine results using statistical analysis.
5 pts: Follow proper coding conventions
7 pts: Ask user if they want to play again
Start over if they enter "Y" or "y"
End game with a message if they answer anything else

Helpers
Remember that you can find solutions to the even programming exercises online. Check Canvas for a link. These are suggestions for you to do. They are not part of the assignment, and you do not have to turn them in.

Exercises: 5.20, 5.24, 5.28