DATA STRUCTURES AND ALGORITHMS IN PYTHON

Assignment #2

TOPICS: LOOPS AND RANDOM NUMBERS

Indian Creek School

DUE : *11 September 2023*

For problems that ask you to write a program, place your code in a .py file and write a comment at the beginning of the file that includes your name, date, and description of the contents of the file. Suggested file names appear in bold at the beginning of problems that ask you to write a program.

EXAMPLE:

```
# Name: Doug McNally
# Date: 2023-06-10
# Description: Lucky number checker

num = int( input("Pick a number: ") )
if num == 42:
     print("You entered the right number!")
else:
     print("You lose!")
```

PROBLEMS THAT REQUIRE YOU TO WRITE A PYTHON FILE: 2 – 4, 8 – 10, 13 – 15
PROBLEMS THAT REQUIRE ANSWERS IN A PDF DOCUMENT: 5 – 7, 11 – 12

1. Read section 1.4 in *Data Structures and Algorithms in Python*.

2. **NameRepeat.py** Write a Python program that will use a `for` loop to print your name 10 times, and then the word "Done" at the end.

3. **Countdown.py** Write a program that uses a `while` loop to count down from 10 then prints the words "Blast off!" Remember to use a `while` loop and not a `for` loop. Example output:

   10
   9
   8
   ⋮
   1
   0
   Blast off!

4. **D20.py** Dice with 20 numbers on them are very common in many tabletop role-playing games. Write a program that "rolls" a 20 sided die (commonly referred to as a "d20" by gamers) and prints out the result in a user-friendly format such as "The d20 landed on 5."

5. Consider the following program:

```
count = 0
total = 0
num = int(input("Enter a value (-1 to quit): "))
while num != -1:
    total = total + num
    count = count + 1
    num = int(input("Enter a value (-1 to quit): "))

if count != 0:
    print("The average is", (total / count) )
else:
    print("No values entered.")
```

   a) Explain what this program does. b) Why is the `if` statement after the loop necessary? c) What would the output of the program be if the user entered the values 10, 15, 30, 42, -1?

6. There are at least three things wrong with this program, explain what they are.

```
print("This program takes three numbers and returns the sum.")
total = 0

for i in range(3):
    x = input("Enter a number: ")
    total = total + i
print("The total is:", x)
```

7. Explain what the following program outputs and why:

```
num = 1
for i in range(12):
    print(num)
    num = num * 2
```

8. **NumberStuff.py** Write a program using loop(s) that will:
   - Ask the user to enter 6 different numbers
   - Print the sum of the 6 numbers entered
   - Print the number of negative values entered

   For example, if the user enters 12, -7, 24, 42, 8, -42 as their six numbers, your program should report that the sum is 37 and that 2 negative values were entered.

9. **PowersOf5.py** Write a program that calculates and prints out the values of the first 20 powers of 5, i.e. $5^0, 5^1, 5^2, \ldots, 5^{19}$. Remember that 5^8 means "multiply 5 by itself 8 times."

10. **EvenNumbers.py** Create a loop that prints even numbers until it reaches your year of age or, if your age is an odd number, prints out odd numbers until it reaches your age. For example, it might print out something like this:

```
2
4
6
8
10
12
```

    Hint: Odd numbers can be generated using the formula $2n - 1$, where plugging in $n = 1$ will give you the first odd number (1), $n = 2$ will give you the second odd number (3), and so on. Even numbers can be generated the same way using the formula $2n$.

11. The print command doesn't always have to start a new line after it prints something out. By default it inserts what is called a new line character `"\n"` at the end of what it prints (this is the character that gets "typed" when you press Enter on your keyboard). You can change this by specifying an `end` value in the print statement. For example, the following code will print the numbers 0 through 9 all on the same line, separated by spaces:

```
for i in range(10):
    print(i, end=" ")
```

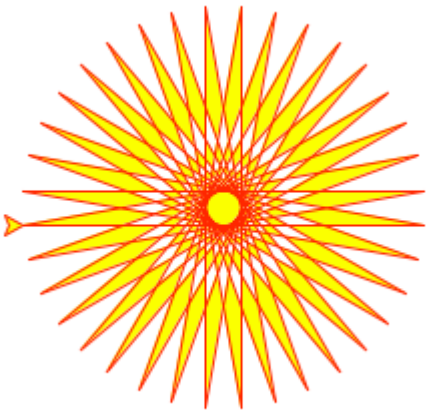
Try running the following code to see what it does:.

```
a = 1
b = 1
for c in range(1, 10):
    print(a, end=" ")
    n = a + b
    a = b
    b = n
```

Do you recognize the sequence of numbers printed out? If so, where have you seen it before? Explain why this code prints out the numbers it does. Go back and modify your solution to the previous problem to make it print out the numbers separated by hyphens by setting the `end` value to `"-"` in your print statements.

12. Explain the purpose of whitespace in Python code.

13. **TurtleArt.py** In your terminal, type the following command `python3 -m turtledemo`. A window should pop up that contains many examples of programs using the `turtle` module. The simplest ones are `yingyang` and `peace`. For this problem, write a Python program that uses the turtle module to create a piece of art. You can use inspiration from the demos, or come up with your own ideas. Remember that you can look up "Python turtle graphics reference" online to see all of the turtle commands that are available. The only requirements for your artwork is that it should use multiple colors and at least 20 lines of Python code.



14. **mystery.py - Bonus problem** *This problem is optional.* Write a program to calculate the number you get by adding up the following series for 1000 terms by using a `for` loop.

$$\frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \ldots$$

Do you recognize the number that it adds up to? What happens if you do 20000 terms instead?

15. **RockPaperScissors.py** Remember the childhood game "Rock, Paper, Scissors"? It is a two-player game in which each person simultaneously chooses either rock, paper, or scissors. Rock beats scissors but loses to paper, paper beats rock but loses to scissors, and scissors beats paper but loses to rock. Write a program that takes input from the user for each of the options, for instance 1: Rock, 2: Paper, 3: Scissors.

    The computer should generate a random move as well. Use `if` statements to check whether your move in Rock, Paper, Scissors beats the computer's move. You may want to use nested if statements. A nested if statement is just one if inside of another. For example the following code first checks if the player picked Rock and then considers each possible outcome for the computer's move.

```
if playerMove == 1: # player picked Rock
    if compMove == 1: # check if the computer also picked Rock
        print("Tie game!")
    elif compMove == 2: # check if the computer picked Paper
        print("Computer wins!")
    else: # computer picked Scissors
        print("Player wins!")
elif playerMove == 2: # player picked Paper
     # your code goes here
```

    Here are two example runs of my program. This is just to give you an idea of how it works, your program doesn't have to be identical to mine.

```
---- Welcome to Rock, Paper, Scissors ----

What is your move? Enter 1 for Rock, 2 for Paper, 3 for Scissors: 2

Rock...
Paper...
Scissors...
Shoot!

Scissors cuts paper, I win. MUAHAHAHA!

---- Welcome to Rock, Paper, Scissors ----

What is your move? Enter 1 for Rock, 2 for Paper, 3 for Scissors: 2

Rock...
Paper...
Scissors...
Shoot!

Paper covers rock, you win. Bummer!
```