BMIS2542: Data Programming Essentials with Python

Assignment-1: Python Basics (Challenge option)

1. Create a Python program that prompts a user to enter a number, determines and prints the number of odd, even, and zero digits read from the user. Here is an example:

Enter a number: 123450

Odd Digits: 3

Even Digits: 2

Zero Digits: 1

1. A “[hailstone sequence](https://mathworld.wolfram.com/HailstoneNumber.html)” of numbers can be produced through the following steps:

Take an initial integer, and if the number is even, divide it by 2. If the number is odd, multiply it by 3 and add 1. Repeat the process for the resulting number.

For example, see the sequence for the following numbers:

n=3; 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, …

n=4; 2, 1, 4, 2, 1, …

n=5; 16, 8, 4, 2, 1, 4, 2, 1, …

n=6; 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, …

n=7; 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, …

You can see that every initial seed number yields a sequence that eventually hits 1 and then repeats: 4,2,1,4,2,1, …

Write a program that prompts the user to enter an initial seed number, and then runs the hailstone sequence for the number until it reaches 4, 2, 1, 4, 2, 1 for the number. Print out all the numbers in the hailstone sequence for the seed number like the examples above.

1. A “necklace” sequence of numbers is created with two single “seed” numbers. The next number is obtained by adding the first two numbers together and saving only the ones digit. This process is repeated until the "necklace" closes by returning the original two numbers.

For example, if you start with 1 and 8 as the “seed” numbers, the necklace sequence would be:

1, 8, 9, 7, 6, 3, 9, 2, 1, 3, 4, 7, 1, 8

Create a Python program that prompts the user to enter the first two “seed” numbers, and then derives the necklace sequence for the seed numbers and displays all the numbers in the sequence.

1. Develop a Python program that lets the user enter a name, prints the name, prints the name in the reverse order (e.g., narayan and nayaran), and lists the number of vowels in the name.
2. Write a Python program that lets the user enter the total precipitation in inches for each of the 12 months of a year. The program should calculate and display the total precipitation for the year, the average monthly precipitation, and the months with the highest and lowest amounts of precipitation.
3. Create a Python program to simulate the [Rock-Paper-Scissors game](https://en.wikipedia.org/wiki/Rock_paper_scissors) played between a user and the computer.