Assignment on Comprehension

1. Find all of the numbers from 1–1000 that are divisible by 8

2. Find all of the numbers from 1–1000 that have a 6 in them

3. Count the number of spaces in a string (*take input from user*)

4. Remove all of the vowels in a string (*take input from user*)

5. Find all of the words in a string that are less than 5 letters (*take input from user*)

6. Use a dictionary comprehension to count the length of each word in a sentence (*take input from user*)

7. Use a nested list comprehension to find all of the numbers from 1–1000 that are divisible by any single digit.

Assignment on Generator

1. We want to generate Fibonacci numbers up to a certain limit. Instead of computing and storing the entire sequence in memory, create generator to yield Fibonacci numbers one by one, conserving memory and allowing for easy iteration.

2. Implement a generator function that yields palindrome numbers. Palindromes are numbers that read the same backward as forward (e.g., 121, 1331). Generate palindromes lazily and infinitely.

3. Write a generator function that mimics the behavior of the built-in range() function. The generator should take start, stop, and step arguments and yield numbers within the specified range.

Assignment of Decorator

1. Develop a memoization decorator that caches the results of function calls and returns the cached result when the same inputs occur again. This can greatly improve the performance of recursive or computationally intensive functions.