Basic Data Types

1. **Numbers**
    - Write a Python program to find all the prime numbers between two given numbers.
        - **Example Test Case:**
            - Input: `start = 10`, `end = 30`
            - Expected Output: `[11, 13, 17, 19, 23, 29]`
    - Write a Python program to find the first n numbers that are both prime.
        - **Example Test Case:**
            - Input: `n = 5`
            - Expected Output: `[2, 3, 5, 7, 11]`
2. **Strings**
    - Write a Python function to find the longest word in a given sentence.
        - **Example Test Case:**
            - Input: `"The quick brown fox jumps over the lazy dog"`
            - Expected Output: `"jumps"`
    - Write a Python function to check if two given strings are anagrams of each other.
        - **Example Test Case:**
            - Input: `str1 = "listen"`, `str2 = "silent"`
            - Expected Output: `True`
3. **Booleans**
    - Create a program that evaluates a list of Boolean expressions and returns the count of `True` values.
        - **Example Test Case:**
            - Input: `[True, False, True, True, False]`
            - Expected Output: `3`
    - Write a Python function to evaluate a complex Boolean expression represented as a string (e.g., "True and False or True").
        - **Example Test Case:**
            - Input: `"True and False or True"`
            - Expected Output: `True`
4. **Lists**
    - Write a Python function to find the second largest element in a list.
        - **Example Test Case:**
            - Input: `[10, 20, 4, 45, 99]`
            - Expected Output: `45`
    - Write a Python program to generate all possible permutations of a list of numbers.
        - **Example Test Case:**
            - Input: `[1, 2, 3]`
            - Expected Output: `[[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]`
5. **Tuples**
    - Create a function to merge two sorted tuples into one sorted tuple.

        - **Example Test Case:**
            - Input: `tuple1 = (1, 3, 5), tuple2 = (2, 4, 6)`
            - Expected Output: `(1, 2, 3, 4, 5, 6)`
    - Write a Python program to find all unique combinations of elements in a tuple that sum up to a given number.
        - **Example Test Case:**
            - Input: `tuple = (2, 3, 5, 7), target = 10`
            - Expected Output: `[(3, 7), (2, 3, 5)]`
6. **Dictionaries**
    - Write a Python function to invert a dictionary, swapping keys and values.
        - **Example Test Case:**
            - Input: `{'a': 1, 'b': 2, 'c': 3}`
            - Expected Output: `{1: 'a', 2: 'b', 3: 'c'}`
    - Write a Python program to merge multiple dictionaries and sum the values of common keys.
        - **Example Test Case:**
            - Input: `dict1 = {'a': 1, 'b': 2}, dict2 = {'b': 3, 'c': 4}`
            - Expected Output: `{'a': 1, 'b': 5, 'c': 4}`
7. **Sets**
    - Write a Python program to find the symmetric difference between two sets.
        - **Example Test Case:**
            - Input: `set1 = {1, 2, 3}, set2 = {3, 4, 5}`
            - Expected Output: `{1, 2, 4, 5}`
    - Write a Python function to find all subsets of a given set.
        - **Example Test Case:**
            - Input: `set = {1, 2, 3}`
            - Expected Output: `[{}, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}]`

Control Statements

1. **If Statements**
    - Write a Python program to classify a given year as a leap year or a non-leap year.
        - **Example Test Case:**
            - Input: `year = 2020`
            - Expected Output: `"Leap Year"`
    - Write a Python program to solve a quadratic equation (ax^2 + bx + c = 0) and classify the roots as real, complex, or equal.
        - **Example Test Case:**
            - Input: `a = 1, b = -3, c = 2`
            - Expected Output: `Real roots: 2.0, 1.0`
2. **For Loops**
    - Write a Python function to generate the Fibonacci series up to a given number of terms.
        - **Example Test Case:**

        - Input: `terms = 5`
        - Expected Output: `[0, 1, 1, 2, 3]`
    - Write a Python program to find the longest increasing subsequence in a given list of numbers.
        - **Example Test Case:**
            - Input: `[10, 22, 9, 33, 21, 50, 41, 60, 80]`
            - Expected Output: `[10, 22, 33, 50, 60, 80]`
3. **While Loops**
    - Write a Python program to find the greatest common divisor (GCD) of two numbers using the Euclidean algorithm.
        - **Example Test Case:**
            - Input: `num1 = 48`, `num2 = 18`
            - Expected Output: `6`
    - Write a Python program to implement a basic version of the Collatz conjecture.
        - **Example Test Case:**
            - Input: `number = 13`
            - Expected Output: `[13, 40, 20, 10, 5, 16, 8, 4, 2, 1]`