Practice Problems:

1. Write a program for checking the given number is even or odd.
2. Write a program for displaying reversal of a number.
3. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
4. Implement Python Script to check given number is palindrome or not.
5. Implement Python Script to print sum of N natural numbers.
6. **By considering the terms in the Fibonacci sequence whose values do not exceed 1000, find the sum of the even-valued terms.**
7. Define a function max_of_three() that takes three numbers as arguments and returns the largest of them.
8. **Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.**
9. Write function to compute gcd, lcm of two numbers. Each function shouldn’t exceed one line.
10. Python Program to Make a Simple Calculator using function.

11.Define a function which generates Fibonacci series up to n numbers using RECURSION.

12. Write a Python program to find the power of a number using recursion.

13. Write a Python script to print following patterns:

```
                1
            2   1
        3   2   1
    4   3   2   1
5   4   3   2   1
```

```
            1
        1   2   1
    1   2   3   2   1
1   2   3   4   3   2   1
```

```
            *
        *   *   *
    *   *   *   *   *
        *   *   *
            *
```

14. Find mean, median, mode for the given set of numbers in a list.
15. **Write a function cumulative_product to compute cumulative product of a list of numbers.**
16. **Write a function reverse to reverse a list. Without using the reverse function.**
17. Write a function dups to find all duplicates in the list.
18. Implement Python Script to check given string is palindrome or not.
19. Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it.
20. Finding the sum and average of given numbers using lists.
21. **With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.**
22. **Write a program to count the numbers of characters in the string and store them in a dictionary data structure.**
23. **Remove spaces from a string using recursion.**
24. **24.**Write a program combine_lists that combines two lists into a dictionary.

25. Write a function dups to find all duplicates in the list.

26. Write a function unique to find all the unique elements of a list.

File Handling:

27.Write a program to compute the number of characters, words and lines in a file.

28. Write a Python program to read a file line by line and store it into a list.

29. Write a python program to find the longest words.

30. Write a Python program to write a list to a file.

31. Write a Python program to combine each line from first file with the corresponding line in second file

Classes:

1. Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case.
2. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.
3. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.
4. Write a Python class to reverse a string word by word :

   Input string : 'hello Python'

   Expected Output : 'Python hello'
5. Write a Program to create a class by name Students, and initialize attributes like name, age, and grade while creating an object.
6. Write a program, to create a child class Teacher that will inherit the properties of Parent class Staff.

7. Create a class Teacher with name, age, and salary attributes, where salary must be a private attribute that cannot be accessed outside the class.
8. Write a Python program that checks if one class is a subclass of another?
9. Write a Program in Python to implement a Stack Data Structure using Class and Objects, with push, pop, and traversal method.
10. Define a class called Songs, it will show the lyrics of a song. Its __init__() method should have two arguments:selfanf lyrics.lyricsis a list. Inside your class create a method called sing_me_a_songthat prints each element of lyricson his own line.
11. Bank Account class:
    1. Create a Python class called BankAccount which represents a bank account, having as attributes: accountNumber (numeric type), name (name of the account owner as string type), balance.
    2. Create a constructor with parameters: accountNumber, name, balance.
    3. Create a Deposit() method which manages the deposit actions.
    4. Create a Withdrawal() method which manages withdrawals actions.
    5. Create an bankFees() method to apply the bank fees with a percentage of 5% of the balance account.
    6. Create a display() method to display account details.
    7. Give the complete code for the BankAccount class.
12. Computation class

    1 - Create a Coputation class with a default constructor (without parameters) allowing to perform various calculations on integers numbers.

    2 - Create a method called Factorial() which allows to calculate the

factorial of an integer. Test the method by instantiating the class.

3 - Create a method called Sum() allowing to calculate the sum of the first n integers 1 + 2 + 3 + .. + n. Test this method.

4 - Create a method called testPrim() in the Calculation class to test the primality of a given integer. Test this method.

4 - Create a method called testPrims() allowing to test if two numbers are prime between them.

5 - Create a tableMult() method which creates and displays the multiplication table of a given integer. Then create an allTablesMult() method to display all the integer multiplication tables 1, 2, 3, ..., 9.

6 - Create a static listDiv() method that gets all the divisors of a given integer on new list called Ldiv. Create another listDivPrim() method that gets all the prime divisors of a given integer

Function Decorator:

Write a Python program to make a chain of function decorators (bold, italic, underline etc.).

Operator Overloading:

1. Overload arithmetic operator * (multiplication). The magic method should take two parameters, first a string and an integer. It should print the string number of times equal to the integer passed. e.g. for (BVICAM, 4), it should return BVICAM BVICAM BVICAM BVICAM.
2. Overload comparison operator 'Less Than' (<) to compare two string. It should return true / false.
3.