🧮 Assignment: Function in Python

🎯 **Objective:**
To understand how to define, call, and utilize functions in Python for code reusability, modularity, and problem-solving.

🟢 Level 1 – Easy

✅ Task 1: Create a Simple Function

Define a function `greet()` that prints `"Hello, welcome to Python!"` and call it.

✅ Task 2: Function with Parameters

Create a function `add(a, b)` that takes two numbers and prints their sum. Call it with different values.

✅ Task 3: Function with Return Value

Define a function `square(n)` that returns the square of a number. Call it and print the result.

🟡 Level 2 – Medium

✅ Task 4: Default Parameter

Create a function `greet_user(name="Guest")` that greets the user. If no name is provided, it should greet "Guest".

✅ Task 5: Function with Multiple Return Values

Write a function `calculator(a, b)` that returns the sum, difference, product, and quotient of two numbers.

✅ Task 6: Even or Odd Checker

Define a function `check_even_odd(n)` that returns `"Even"` if the number is even and `"Odd"` if the number is odd.

✅ Task 7: Factorial Function

Write a function `factorial(n)` that returns the factorial of a number using a loop.

🧮 Assignment: Function in Python

🔴 Level 3 – Hard

✅ Task 8: Recursive Function

Write a recursive version of the `factorial(n)` function.

✅ Task 9: Prime Number Checker

Define a function `is_prime(n)` that returns `True` if the number is prime and `False` otherwise.

✅ Task 10: Function with Variable Number of Arguments

Write a function `total_sum(*args)` that accepts any number of arguments and returns their sum.

✅ Task 11: Lambda Function

Use a lambda function to calculate the square of a number and call it with input 5.

✅ Task 12: Function Inside a Function (Nested Function)

Create a function `outer()` which contains a nested function `inner()` that prints `"This is the inner function"`.