**Report: Python Developer - Task 1**

**Objective**

To develop Python programs addressing basic and intermediate-level tasks, focusing on computational problems, algorithm design, and logical reasoning.

**Completed Tasks:**

1. **Sum of Two Numbers**
   * Objective: Calculate the sum of two integers.
   * Logic: Use the + operator to add two integers.
   * Input: Two integers.
   * Output: Their sum.
2. **Odd or Even Check**
   * Objective: Determine if a number is odd or even.
   * Logic: Use the modulus operator % to check divisibility by 2.
   * Input: A single integer.
   * Output: "Odd" or "Even".
3. **Factorial Calculation**
   * Objective: Compute the factorial of a given number.
   * Logic: Use a loop or the math.factorial function.
   * Input: A single integer.
   * Output: The factorial value.
4. **Fibonacci Sequence**
   * Objective: Generate the first n numbers in the Fibonacci sequence.
   * Logic: Use a loop and the Fibonacci formula: F(n)=F(n−1)+F(n−2)F(n) = F(n-1) + F(n-2)F(n)=F(n−1)+F(n−2).
   * Input: A single integer n.
   * Output: A list of Fibonacci numbers.
5. **Reverse a String**
   * Objective: Reverse a given string.
   * Logic: Use string slicing or a loop to reverse characters.
   * Input: A string.
   * Output: The reversed string.
6. **Palindrome Check**
   * Objective: Check if a string is a palindrome.
   * Logic: Compare the string with its reverse.
   * Input: A string.
   * Output: True or False.
7. **Leap Year Check**
   * Objective: Determine if a year is a leap year.
   * Logic: A year is a leap year if divisible by 4, not by 100, unless divisible by 400.
   * Input: A year (integer).
   * Output: True or False.
8. **Armstrong Number Check**
   * Objective: Check if a number is an Armstrong number.
   * Logic: Compute the sum of each digit raised to the power of the number of digits.
   * Input: An integer.
   * Output: True or False.
9. **Custom Encryption-Decryption System**
   * Objective: Develop a program to encrypt and decrypt messages.
   * Logic: Implement a substitution cipher or a matrix transformation algorithm.
   * Input: A plaintext string for encryption; an encrypted string for decryption.
   * Output: Encrypted or decrypted string.

**Challenges Faced:**

* Designing robust algorithms for encryption and decryption.
* Handling edge cases for string manipulation tasks.