



PYTHON DEVELOPER

TASK-1

1. The sum of Two Numbers

Objective: Write a program to calculate the sum of two numbers.

Input: Two integers.

Output: Single integer, their sum.

Hint: Use the simple addition (+) operator.

2. Odd or Even

Objective: Determine whether a number is odd or even.

Input: A single integer.

Output: "Odd" or "Even".

Hint: Check the remainder when divided by 2 (`number % 2 == 0` for even).

3. Factorial Calculation

Objective: Compute the factorial of a given number n (i.e., $n! = n \times (n-1) \times \dots \times 1$).

Input: A single integer n .

Output: Integer value of $n!$.

Hint: Use a loop or `math.factorial` library.

4. Fibonacci Sequence

Objective: Generate the first n numbers in the Fibonacci sequence (e.g., 0, 1, 1, 2, 3, 5, ...).

Input: Integer n .

Output: List of n Fibonacci numbers.

Hint: Use a loop where $F(n) = F(n-1) + F(n-2)$.

5. Reverse a String

Objective: Reverse the characters in a string.

Input: A string.



Output: The reversed string.

Hint: Use slicing (`[::-1]`) or loop through characters.

6. Palindrome Check

Objective: Check if a string reads the same backward as forward.

Input: A string.

Output: `True` or `False`.

Hint: Compare the string with its reversed version.

7. Leap Year Check

Objective: Determine whether a year is a leap year.

Input: An integer year (e.g., 2024).

Output: `True` if leap year, otherwise `False`.

Hint: A year is a leap year if divisible by 4 but not by 100 unless divisible by 400.

8. Armstrong Number

Objective: Check if a number equals the sum of its digits raised to the power of the number of digits.

Input: An integer nnn.

Output: `True` or `False`.

Example: 153 is an Armstrong number because $1^3 + 5^3 + 3^3 = 153$.
 $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$.

Hint: Convert to a string to calculate the length and power.

1. Custom Encryption-Decryption System

- **Description:** Develop a program that encrypts and decrypts messages using custom encryption algorithms like substitution ciphers or matrix transformations.
- **Challenges:**
 - Create a robust algorithm for encryption and decryption.
 - Handle edge cases (e.g., special characters, spaces).
 - Optionally, implement multi-layer encryption.
- **Skills:** Algorithm design, string manipulation, and logical reasoning.

Main Flow Services and Technologies Pvt. Ltd.

Contact Us. +91 9389641586, +91 97736 99074

Email-Add. contact.mainflow@gmail.com

www.mainflow.in



-
- **Restriction:** No use of built-in encryption libraries like `cryptography` or `hashlib`.
 - **Reason:** The goal of this project is for students to **build an understanding of how encryption works** at a low level. By manually implementing encryption and decryption algorithms (such as a Caesar cipher, Vigenère cipher, or even a simple substitution cipher), students gain deeper insight into data security principles, such as the importance of keys and encryption algorithms. Using pre-built libraries would only teach them how to apply the algorithm, not how it works.
 - **Learning Outcome:** Students will learn about **algorithm design**, **data transformation**, and **security principles** without relying on ready-made solutions.
-

Deadline Compliance

- **Restriction:** Submit the project within **7 days** from the start date.
 - **Reason:** Meeting deadlines is crucial in the real-world software development environment. This restriction helps students practice **time management** and **task prioritization**. In professional settings, tight deadlines are often the norm, and learning to meet them without compromising quality is an essential skill.
 - **Learning Outcome:** Students will learn to manage their time effectively, complete projects under pressure, and **deliver results on time**, which are all important skills in the workplace.
-

General Guidelines for All Projects

- **Restriction:** Submit with a **report/documentation** explaining the approach, key challenges, and solutions.
- **Reason:** Documentation forces students to **articulate their thought process**, reinforcing their learning. It also simulates real-world expectations, where developers must often explain their code to stakeholders, team members, or clients. This is an essential skill that enhances understanding and communication.
- **Learning Outcome:** Students will develop the ability to **communicate complex technical ideas** and **reflect on their problem-solving approaches**, which is a key skill for professional development.

Main Flow Services and Technologies Pvt. Ltd.

Contact Us. +91 9389641586, +91 97736 99074

Email-Add. contact.mainflow@gmail.com

www.mainflow.in



Main Flow Services and Technologies Pvt. Ltd.
Contact Us. +91 9389641586, +91 97736 99074
Email-Add. contact.mainflow@gmail.com
www.mainflow.in