

TASK 1

1. **The Sum of Two Numbers:** Calculate and display the sum of two numbers.

Steps:

1. Input two integers using input().
2. Convert them to integers.
3. Use the + operator to calculate the sum.
4. Print the result.

```
a = int(input("Enter the first number: "))  
b = int(input("Enter the second number: "))  
print("Sum:", a + b)
```

2. **Odd or Even:** Determine whether a number is odd or even.

Steps:

1. Input an integer.
2. Use modulo operation % 2.
3. Print "Even" if divisible by 2, else "Odd".

```
num = int(input("Enter a number: "))  
print("Even" if num % 2 == 0 else "Odd")
```

3. **Factorial Calculation:** Compute the factorial of a number.

Steps:

1. Input an integer.
2. Use a loop or recursion to calculate factorial.

- Alternatively, use `math.factorial()`.

```
import math

n = int(input("Enter a number: "))

print("Factorial:", math.factorial(n))
```

4. **Fibonacci Sequence:** Generate the first n Fibonacci numbers.

Steps:

1. Input an integer n.
2. Use a loop to generate Fibonacci numbers.

```
n = int(input("Enter the number of Fibonacci terms: "))

fib = [0, 1]

for i in range(2, n):

    fib.append(fib[-1] + fib[-2])

print("Fibonacci sequence:", fib[:n])
```

5. **Reverse a String:** Reverse the characters in a string.

Steps:

1. Input a string.
2. Use slicing `[::-1]` to reverse the string.

```
s = input("Enter a string: ")

print("Reversed string:", s[::-1])
```

6. **Palindrome Check:** Check if a string is a palindrome.

Steps:

1. Input a string.

2. Compare the string with its reversed version.

```
s = input("Enter a string: ")  
print("Palindrome:", s == s[::-1])
```

7. **Leap Year Check:** Determine if a year is a leap year.

Steps:

1. Input an integer.
2. Check divisibility rules:
 - Divisible by 4 but not by 100, or divisible by 400.

```
year = int(input("Enter a year: "))  
is_leap = (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)  
print("Leap year:", is_leap)
```

8. **Armstrong Number:** Check if a number is an Armstrong number.

Steps:

1. Input a number.
2. Convert to string to calculate the length.
3. Calculate the sum of its digits raised to the power of the length.

```
num = int(input("Enter a number: "))  
digits = str(num)  
armstrong = sum(int(d)**len(digits) for d in digits)  
print("Armstrong number:", armstrong == num)
```

Custom Encryption-Decryption System

Encrypt and decrypt messages using a substitution cipher.

Approach

1. Define functions for encryption and decryption.
2. Use a substitution cipher such as the Caesar cipher.
3. Handle special characters and spaces.

Code:

```
# Encryption
```

```
def encrypt(text, shift):
```

```
    encrypted = ""
```

```
    for char in text:
```

```
        if char.isalpha():
```

```
            shift_base = ord('A') if char.isupper() else ord('a')
```

```
            encrypted += chr((ord(char) - shift_base + shift) % 26 + shift_base)
```

```
        else:
```

```
            encrypted += char
```

```
    return encrypted
```

```
#Decryption
```

```
def decrypt(text, shift):
```

```
    return encrypt(text, -shift)
```

```
message = input("Enter a message: ")
```

```
key = int(input("Enter shift value: "))
```

```
encrypted = encrypt(message, key)
```

```
print("Encrypted:", encrypted)
```

```
print("Decrypted:", decrypt(encrypted, key))
```