

Main Flow Internship

TASK-1

Ankith R

1. The sum of Two Numbers:

<pre>1 # The sum of Two Numbers 2 num1 = int(input("Enter the first number: ")) 3 num2 = int(input("Enter the second number: ")) 4 sum_result = num1 + num2 5 print("The sum is:", sum_result)</pre>	Enter the first number: 56 Enter the second number: 65 The sum is: 121 === Code Execution Successful ===
--	---

2. Odd or Even:

<pre>1 # Odd or Even 2 num = int(input("Enter a number: ")) 3 if num % 2 == 0: 4 print("Even") 5 else: 6 print("Odd")</pre>	Enter a number: 7 Odd === Code Execution Successful ===
---	---

3. Factorial Calculation:

<pre>1 # Factorial Calculation 2 n = int(input("Enter a number: ")) 3 factorial = 1 4 for i in range(1, n + 1): 5 factorial *= i 6 print("Factorial:", factorial)</pre>	Enter a number: 5 Factorial: 120 === Code Execution Successful ===
---	--

4. Fibonacci Sequence:

<pre>1 # Fibonacci Sequence 2 n = int(input("Enter the number of Fibonacci terms: ")) 3 fib_sequence = [0, 1] 4 for i in range(2, n): 5 next_term = fib_sequence[-1] + fib_sequence[-2] 6 fib_sequence.append(next_term) 7 if n == 1: 8 fib_sequence = [0] 9 print("Fibonacci Sequence:", fib_sequence)</pre>	Enter the number of Fibonacci terms: 7 Fibonacci Sequence: [0, 1, 1, 2, 3, 5, 8] === Code Execution Successful ===
---	--

5. Reverse a String:

<pre>1 # Reverse a StringA 2 string = input("Enter a string: ") 3 reversed_string = string[::-1] 4 print("Reversed String:", reversed_string)</pre>	Enter a string: 7 8 5 3 2 Reversed String: 2 3 5 8 7 === Code Execution Successful ===
---	--

6. Palindrome Check:

```
1 # Palindrome Check
2 string = input("Enter a string: ")
3 is_palindrome = string == string[::-1]
4 print(is_palindrome)
5
```

Enter a string: 7 8 8 7
True
=== Code Execution Successful ===

7. Leap Year Check:

```
1 # Leap Year Check
2 year = int(input("Enter a year: "))
3 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
4     print(True)
5 else:
6     print(False)
```

Enter a year: 2028
True
=== Code Execution Successful ===

8. Armstrong Number:

```
1 # Armstrong Number
2 num = int(input("Enter a number: "))
3 num_str = str(num)
4 num_digits = len(num_str)
5 sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)
6 is_armstrong = num == sum_of_powers
7 print(is_armstrong)
```

Enter a number: 153
True
=== Code Execution Successful ===

9. Custom Encryption-Decryption System:

```
1 # Custom Encryption-Decryption System
2 def encrypt(text, key):
3     encrypted_text = ""
4     for char in text:
5         if char.isalpha():
6             shift = key % 26
7             base = ord('A') if char.isupper() else ord('a')
8             encrypted_text += chr((ord(char) - base + shift) % 26 + base)
9         else:
10            encrypted_text += char
11    return encrypted_text
12
13 def decrypt(text, key):
14    return encrypt(text, -key)
15
16
17 message = input("Enter a message: ")
18 key = int(input("Enter a shift key (integer): "))
19 encrypted_message = encrypt(message, key)
20 decrypted_message = decrypt(encrypted_message, key)
21
22
23 print("Encrypted:", encrypted_message)
24 print("Decrypted:", decrypted_message)
25
```

Enter a message: hi
Enter a shift key (integer): 3
Encrypted: kl
Decrypted: hi
=== Code Execution Successful ===