Internship Task-1 Report

Name: suryakant kendre

Internship Role: Python Developer

Company: Main Flow Services and Technologies Pvt. Ltd.

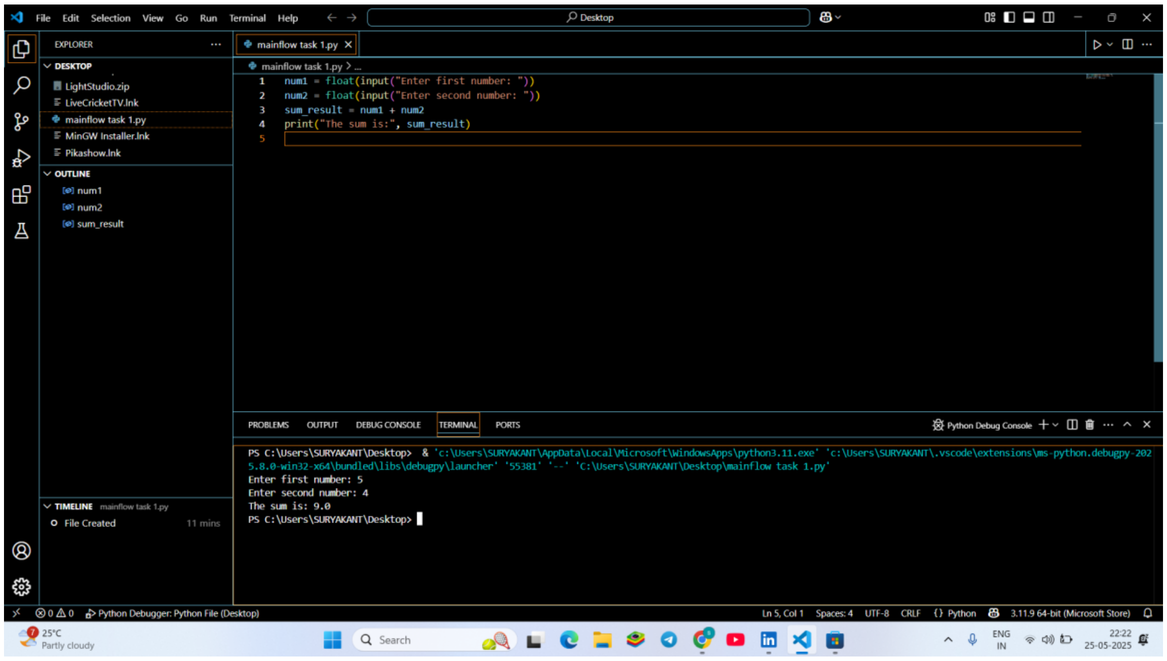
Task: 1 - Python Programs

1. Sum of Two Numbers

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

Code:

```
a = int(input("Enter first number: ")) b

= int(input("Enter second number: "))

print("Sum:", a + b)
```



```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
sum_result = num1 + num2
print("The sum is:", sum_result)
```

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-202
5.8.0-win32-x64\bundled\libs\debugpy\launcher' '55381' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter first number: 5
Enter second number: 4
The sum is: 9.0
PS C:\Users\SURYAKANT\Desktop>
```

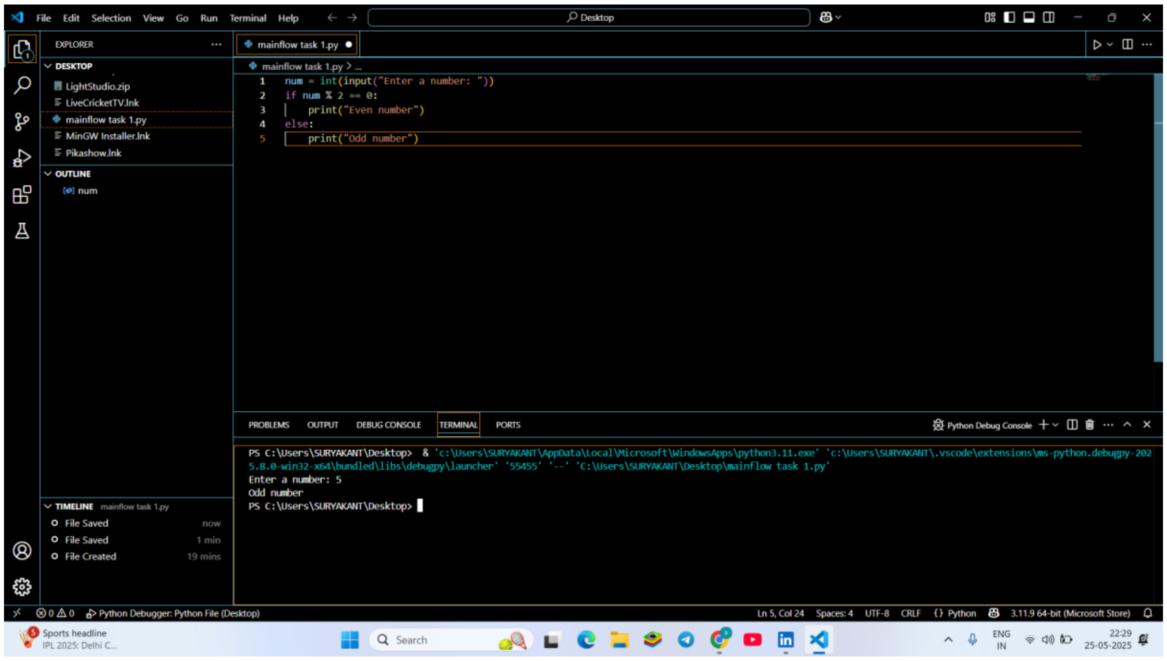
2. Odd or Even

Objective: Determine whether a number is odd or even.

Code:

num = int(input("Enter a number: ")) print("Even"

if num % 2 == 0 else "Odd")



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

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-2025.8.0-win32-x64\bundled\libs\debugpy\launcher' '55455' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter a number: 5
Odd number
PS C:\Users\SURYAKANT\Desktop>
```

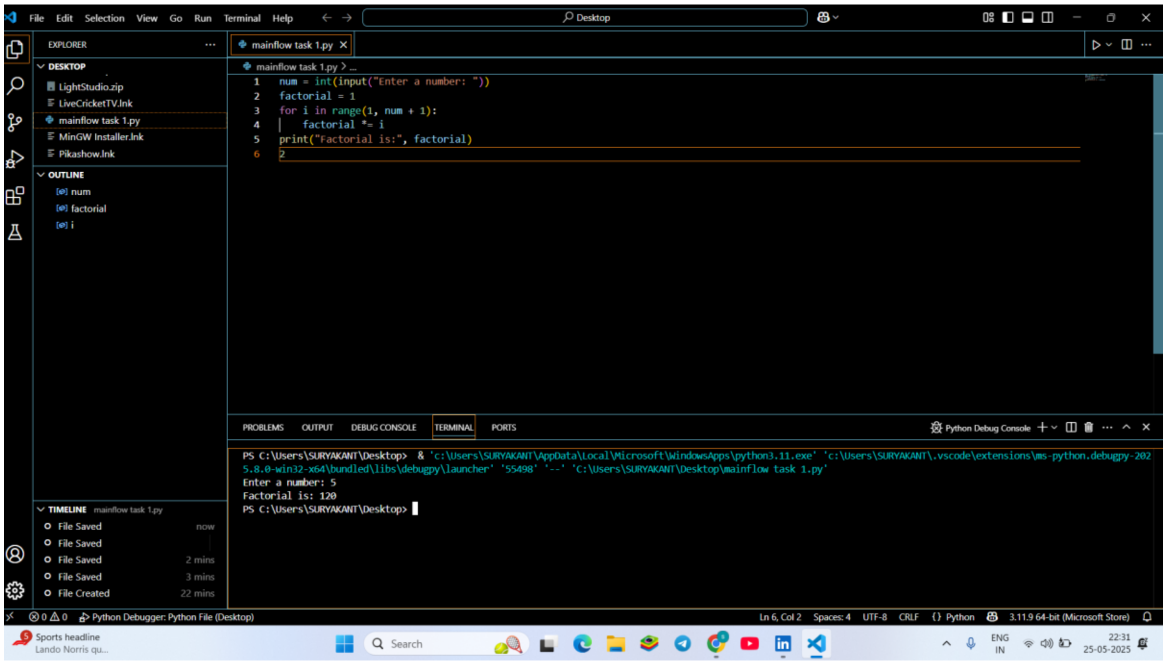
3. Factorial Calculation

Objective: Compute the factorial of a given number.

Code: import math n =

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

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



```python
num = int(input("Enter a number: "))
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print("Factorial is:", factorial)
2
```

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-202
5.8.0-win32-x64\bundled\libs\debugpy\launcher' '55498' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter a number: 5
Factorial is: 120
PS C:\Users\SURYAKANT\Desktop>
```

4. Fibonacci Sequence

Objective: Generate the first n numbers in the Fibonacci sequence.

Code: n = int(input("How many Fibonacci

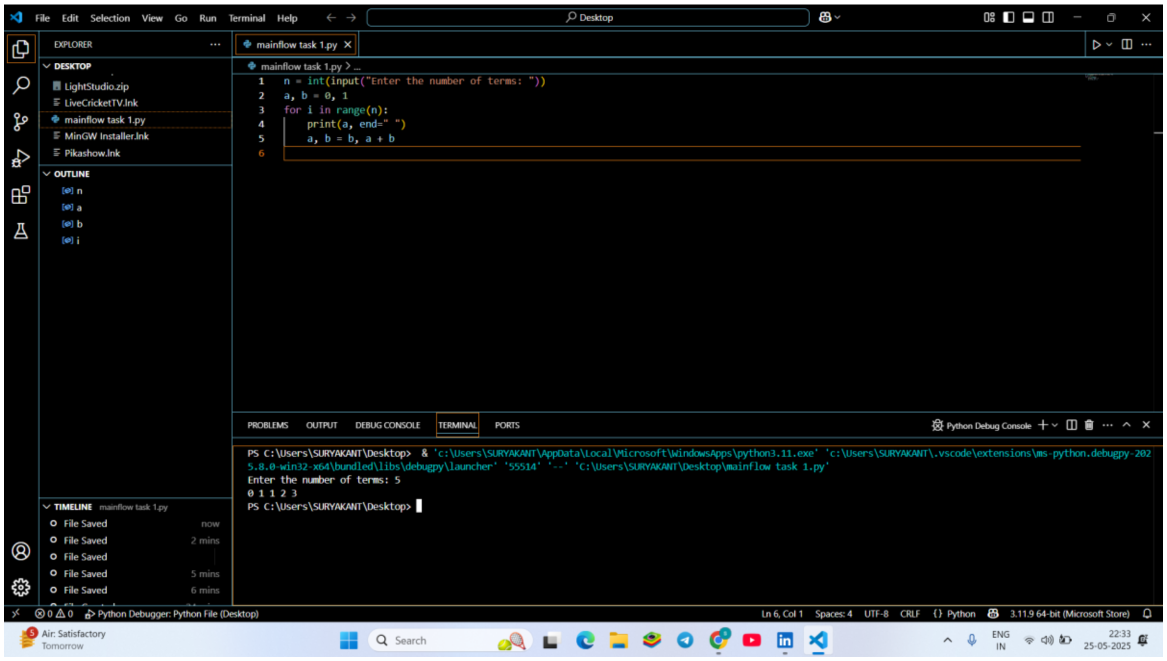
numbers? "))

a, b = 0, 1 for _ in

range(n):

print(a, end=' ')

a, b = b, a + b



```python
n = int(input("Enter the number of terms: "))
a, b = 0, 1
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-202
5.8.0-win32-x64\bundled\libs\debugpy\launcher' '55514' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter the number of terms: 5
0 1 1 2 3
PS C:\Users\SURYAKANT\Desktop>
```

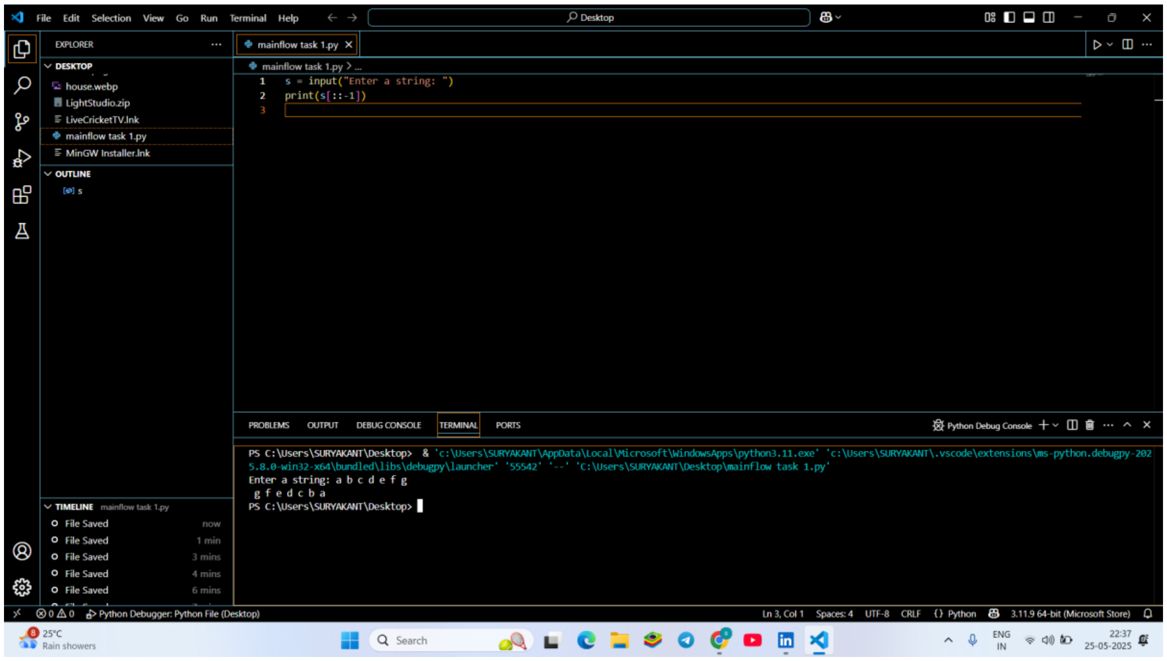
5. Reverse a String

Objective: Reverse the characters in a string.

Code:

s = input("Enter a string: ") print("Reversed

string:", s[::-1])



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

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-202
5.8.0-win32-x64\bundled\libs\debugpy\launcher' '55542' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter a string: a b c d e f g
 g f e d c b a
PS C:\Users\SURYAKANT\Desktop>
```

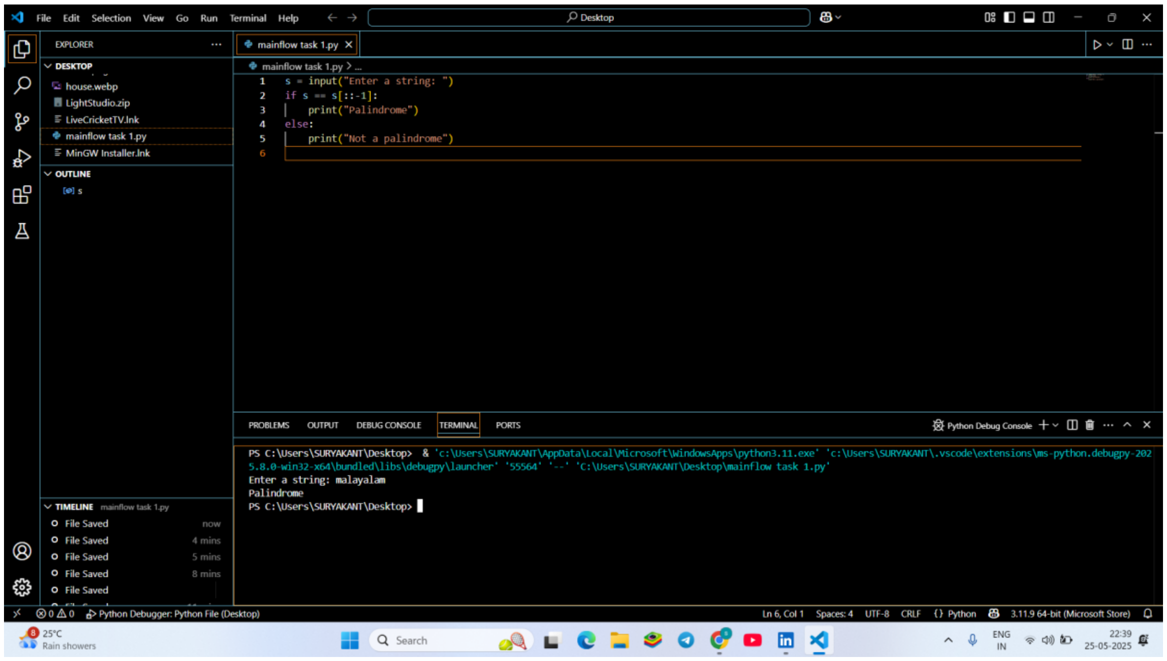
6. Palindrome Check

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

Code:

s = input("Enter a string: ") print("Palindrome:",

s == s[::-1])



```python
s = input("Enter a string: ")
if s == s[::-1]:
    print("Palindrome")
else:
    print("Not a palindrome")
```

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-202
5.8.0-win32-x64\bundled\libs\debugpy\launcher' '55564' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter a string: malayalam
Palindrome
PS C:\Users\SURYAKANT\Desktop> 
```

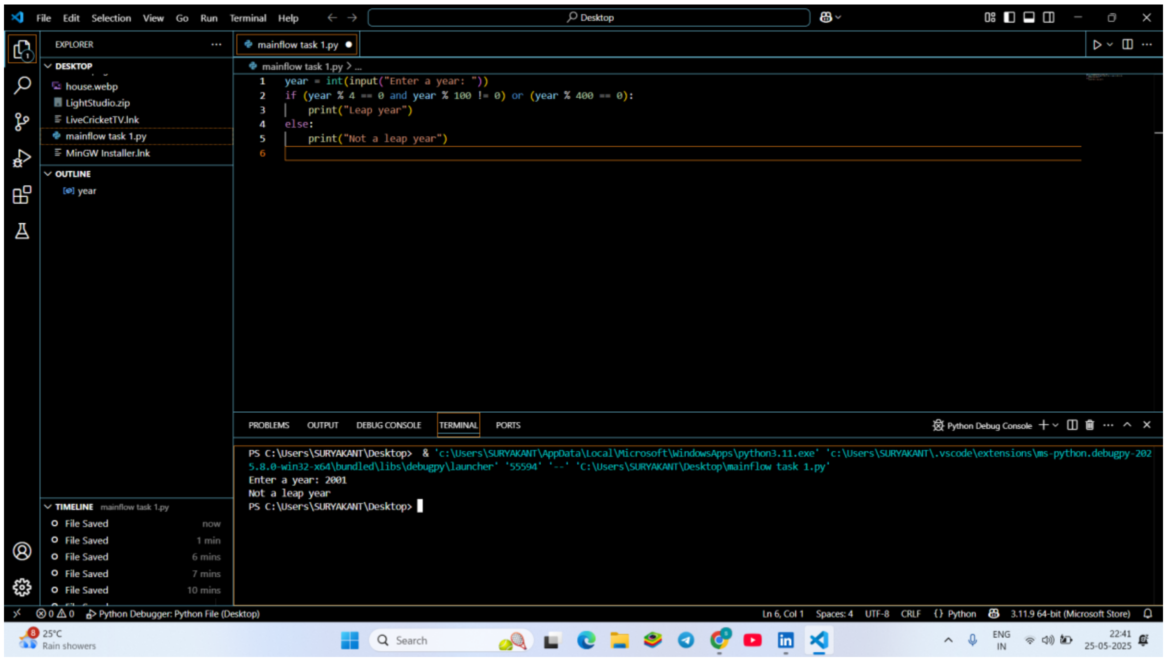
7. Leap Year Check

Objective: Determine whether a year is a leap year.

Code:

year = int(input("Enter year: ")) is_leap = (year % 4 == 0 and year

% 100 != 0) or (year % 400 == 0) print("Leap Year:", is_leap)

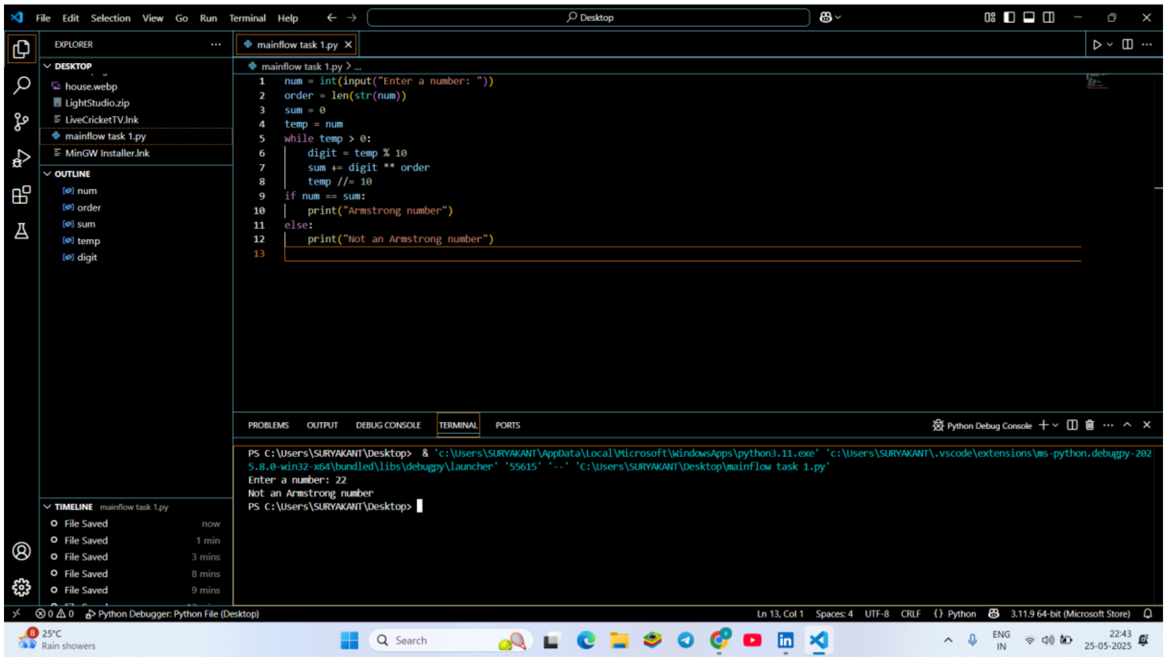


```python
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print("Leap year")
else:
    print("Not a leap year")
```

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-202
5.8.0-win32-x64\bundled\libs\debugpy\launcher' '55594' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter a year: 2001
Not a leap year
PS C:\Users\SURYAKANT\Desktop>
```

8. Armstrong Number

Objective: Check if a number equals the sum of its digits raised to the power of the number of digits. ***Code:*** num = int(input("Enter number: ")) digits = str(num) power = len(digits) sum_digits = sum(int(d)**power for d in digits) print("Armstrong Number:", num == sum_digits)



```python
num = int(input("Enter a number: "))
order = len(str(num))
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** order
    temp //= 10
if num == sum:
    print("Armstrong number")
else:
    print("Not an Armstrong number")
```

```
PS C:\Users\SURYAKANT\Desktop>  & 'c:\Users\SURYAKANT\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\SURYAKANT\.vscode\extensions\ms-python.debugpy-2025.8.0-win32-x64\bundled\libs\debugpy\launcher' '55615' '--' 'C:\Users\SURYAKANT\Desktop\mainflow task 1.py'
Enter a number: 22
Not an Armstrong number
PS C:\Users\SURYAKANT\Desktop>
```

9. Custom Encryption-Decryption System

Objective: Encrypt and decrypt messages using Caesar Cipher logic (without built-in encryption libraries).

Code:

```python
import string



class CustomEncryptor:

    def __init__(self, shift=3):

        self.shift = shift

        self.alphabet = string.ascii_letters + string.digits + string.punctuation + ' '

        self.trans_table = self._generate_cipher()
    def _generate_cipher(self):

        shifted = self.alphabet[self.shift:] + self.alphabet[:self.shift]

        return str.maketrans(self.alphabet, shifted)



    def encrypt(self, text, layers=1):

        for _ in range(layers):

            text = text.translate(self.trans_table)[::-1]

        return text



    def decrypt(self, text, layers=1):

        reverse_table = str.maketrans(

            self.alphabet[self.shift:] + self.alphabet[:self.shift],

            self.alphabet

        )

        for _ in range(layers):

            text = text[::-1].translate(reverse_table)

        return text



encryptor = CustomEncryptor(shift=5)

original_message = "Hello, World! 123"

encrypted = encryptor.encrypt(original_message, layers=2)
```

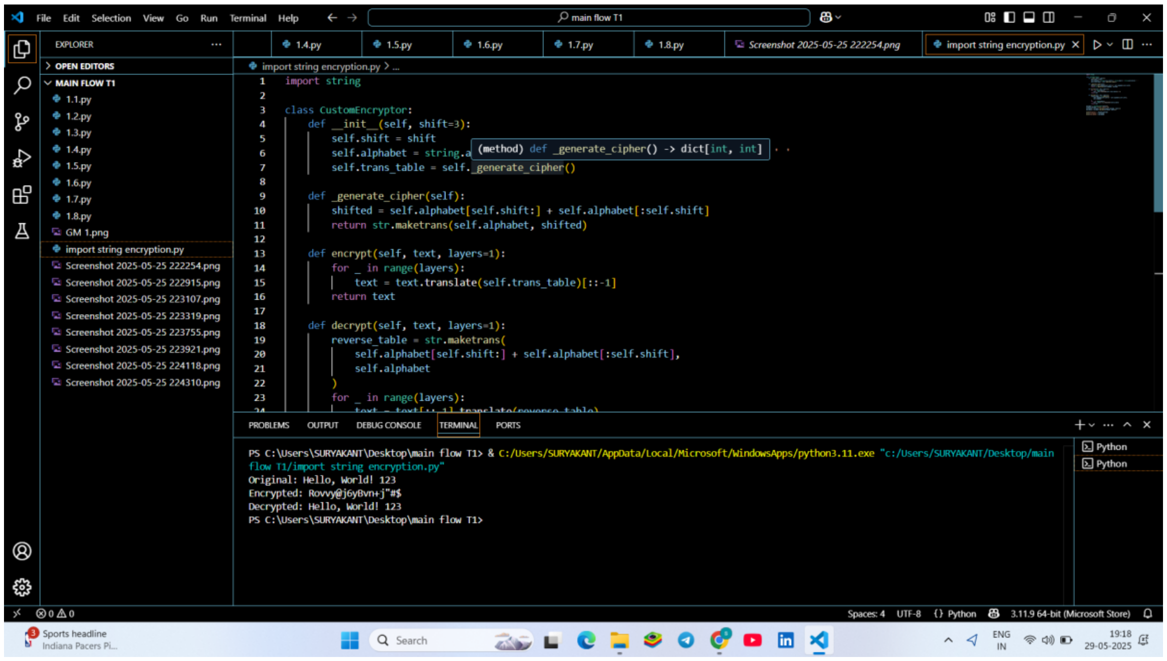
decrypted = encryptor.decrypt(encrypted, layers=2)

print("Original:", original_message)

print("Encrypted:", encrypted)

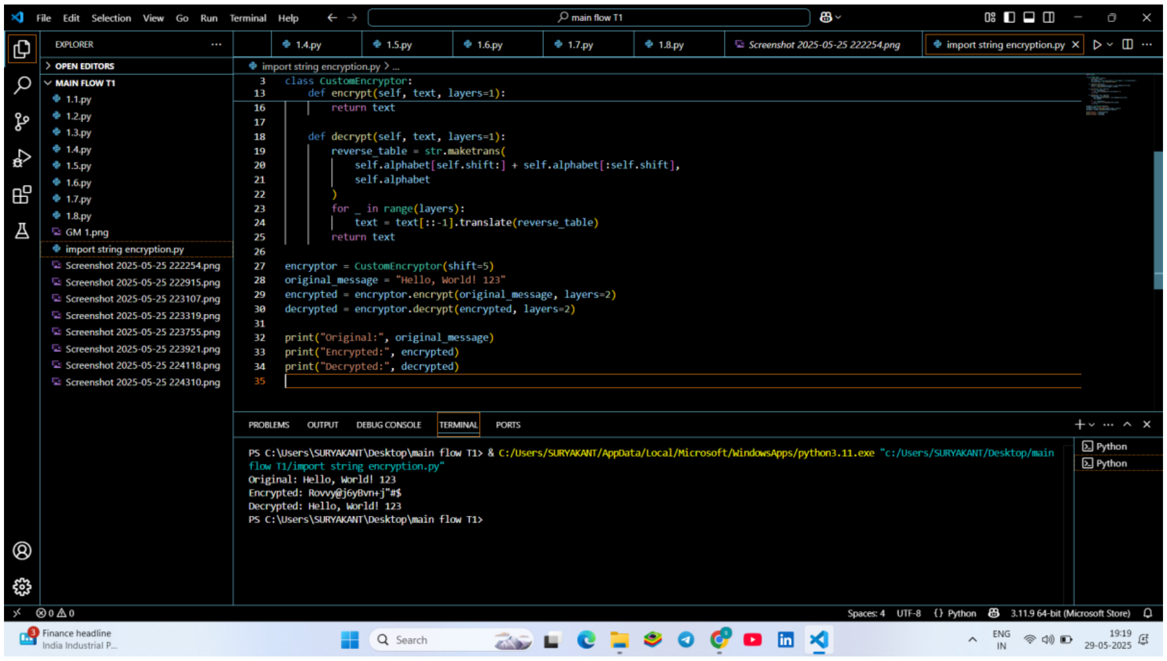
print("Decrypted:", decrypted)

Screenshot of Output:



```python
import string

class CustomEncryptor:
    def __init__(self, shift=3):
        self.shift = shift
        self.alphabet = string.a
        self.trans_table = self._generate_cipher()

    def _generate_cipher(self):
        shifted = self.alphabet[self.shift:] + self.alphabet[:self.shift]
        return str.maketrans(self.alphabet, shifted)

    def encrypt(self, text, layers=1):
        for _ in range(layers):
            text = text.translate(self.trans_table)[::-1]
        return text

    def decrypt(self, text, layers=1):
        reverse_table = str.maketrans(
            self.alphabet[self.shift:] + self.alphabet[:self.shift],
            self.alphabet
        )
        for _ in range(layers):
```

```
PS C:\Users\SURYAKANT\Desktop\main flow T1> & C:/Users/SURYAKANT/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/SURYAKANT/Desktop/main
flow T1/import string encryption.py"
Original: Hello, World! 123
Encrypted: Rovvy@j6yBvn+j"#$
Decrypted: Hello, World! 123
PS C:\Users\SURYAKANT\Desktop\main flow T1>
```



```python
class CustomEncryptor:
    def encrypt(self, text, layers=1):
        return text

    def decrypt(self, text, layers=1):
        reverse_table = str.maketrans(
            self.alphabet[self.shift:] + self.alphabet[:self.shift],
            self.alphabet
        )
        for _ in range(layers):
            text = text[::-1].translate(reverse_table)
        return text

encryptor = CustomEncryptor(shift=5)
original_message = "Hello, World! 123"
encrypted = encryptor.encrypt(original_message, layers=2)
decrypted = encryptor.decrypt(encrypted, layers=2)

print("Original:", original_message)
print("Encrypted:", encrypted)
print("Decrypted:", decrypted)
```

```
PS C:\Users\SURYAKANT\Desktop\main flow T1> & C:/Users/SURYAKANT/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/SURYAKANT/Desktop/main flow T1/import string encryption.py"
Original: Hello, World! 123
Encrypted: Rovvy@j6yBvn+j"#$
Decrypted: Hello, World! 123
PS C:\Users\SURYAKANT\Desktop\main flow T1>
```