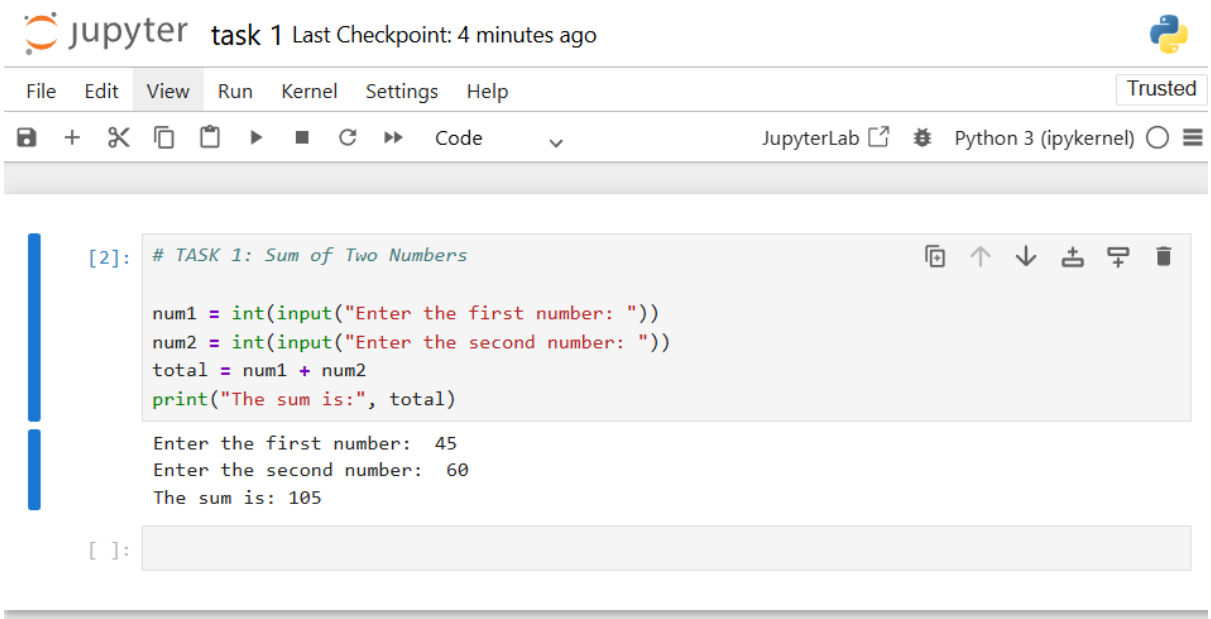
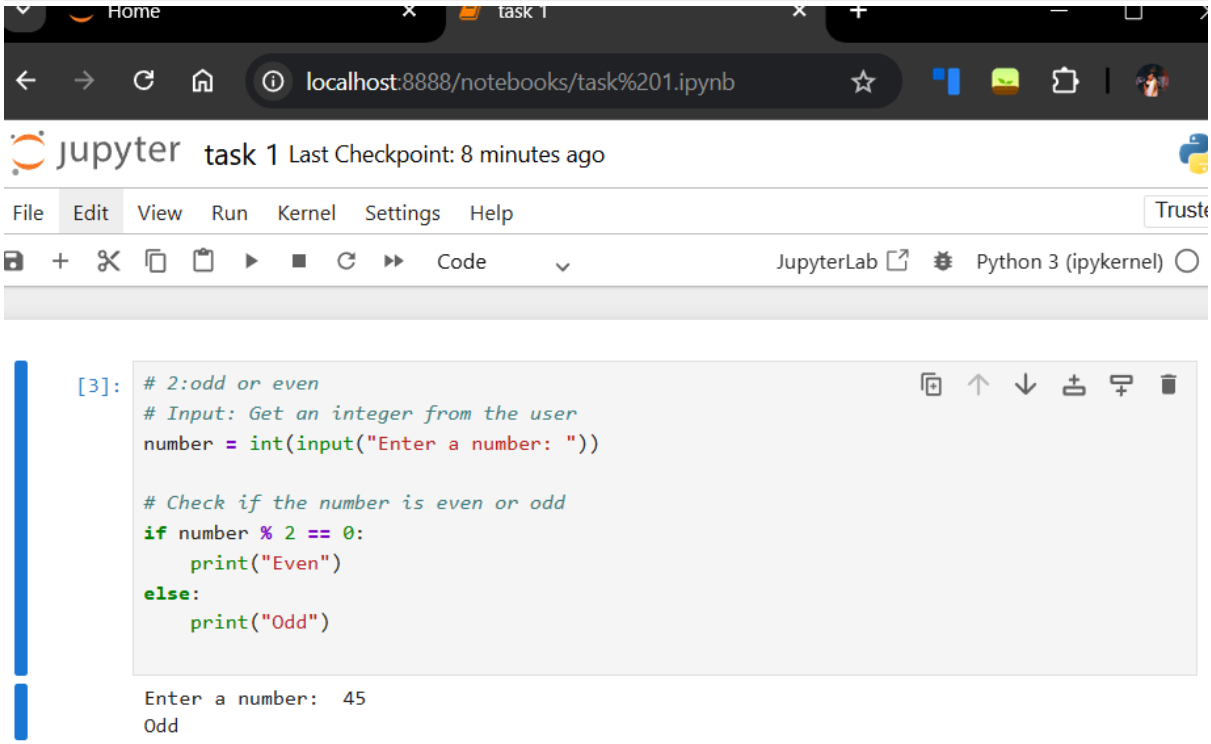
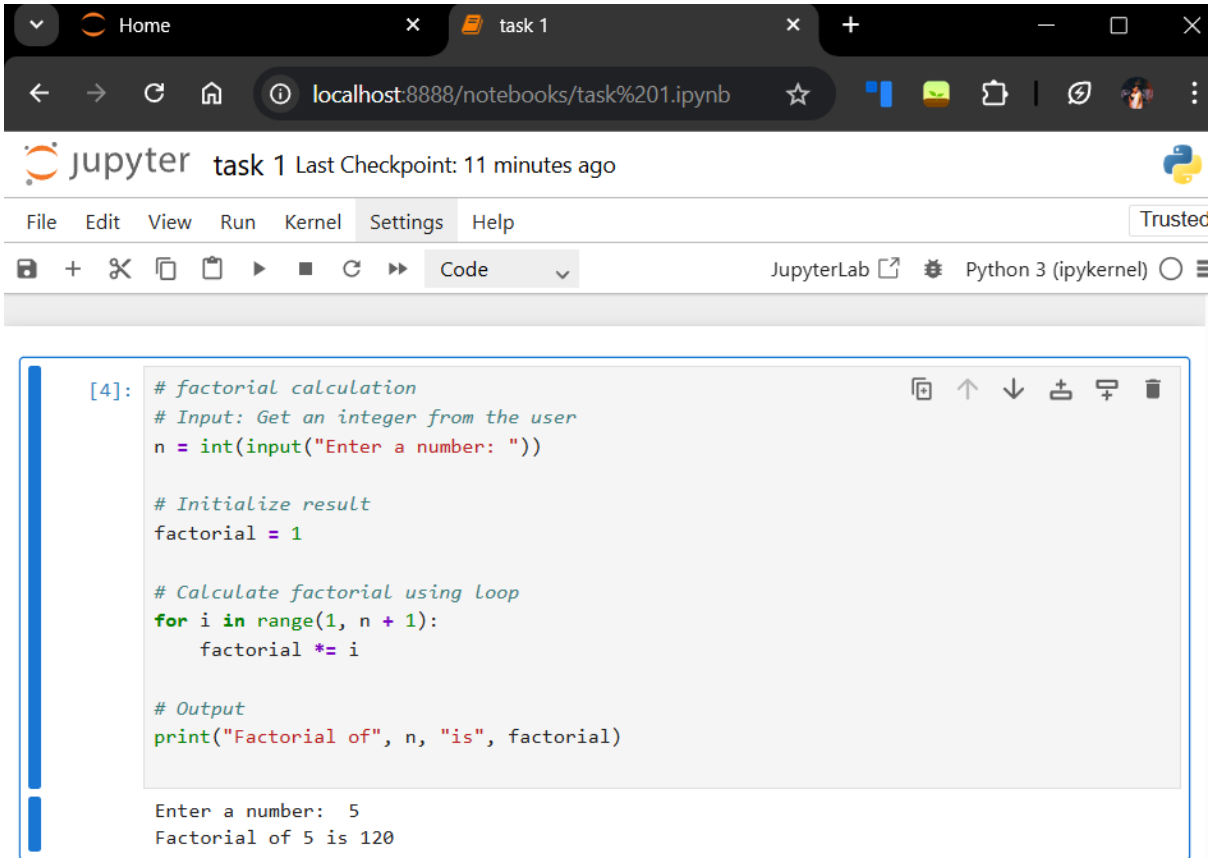
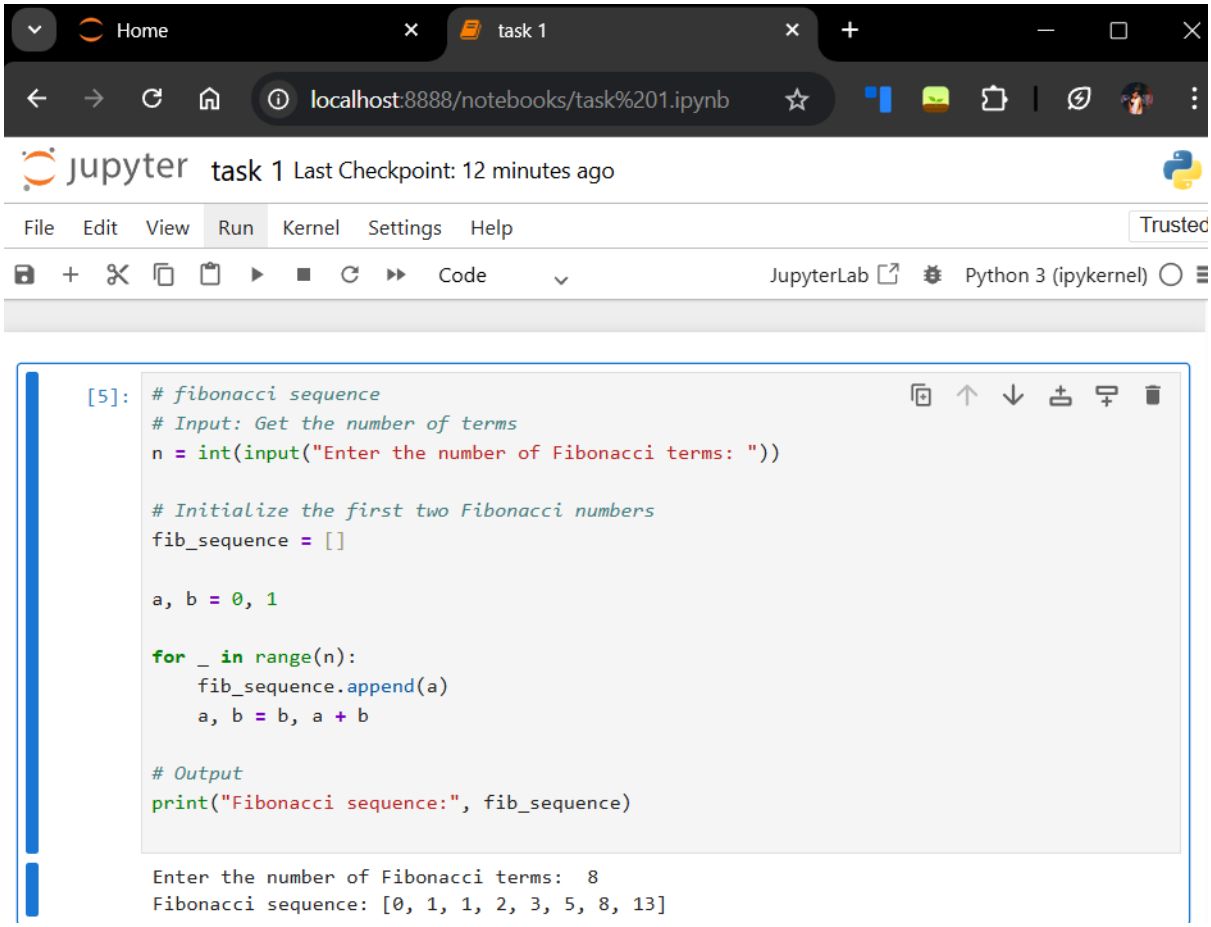
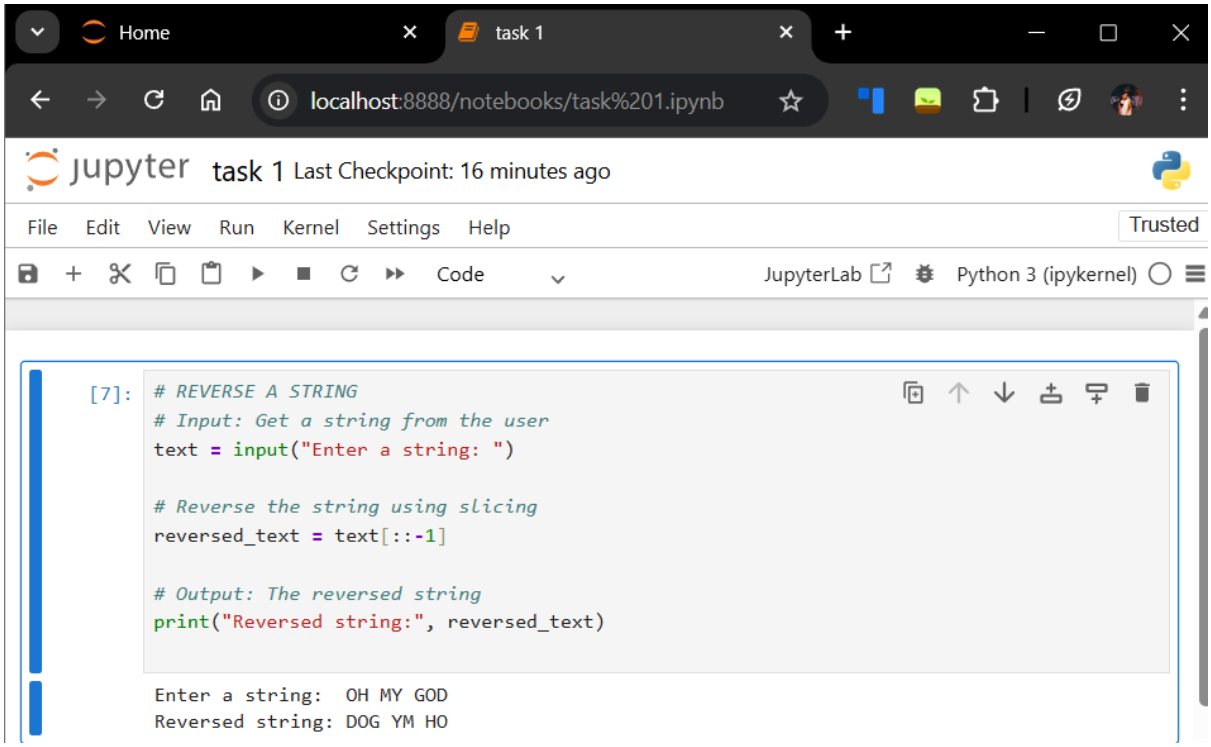

jupyter task 1 Last Checkpoint: 4 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[2]:
# TASK 1: Sum of Two Numbers
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
total = num1 + num2
print("The sum is:", total)
Enter the first number: 45
Enter the second number: 60
The sum is: 105
[ ]:


Home
task 1
localhost:8888/notebooks/task%201.ipynb
jupyter task 1 Last Checkpoint: 8 minutes ago
File Edit View Run Kernel Settings Help
Code
JupyterLab
Python 3 (ipykernel)
[3]:
# 2:odd or even
# Input: Get an integer from the user
number = int(input("Enter a number: "))
# Check if the number is even or odd
if number % 2 == 0:
print("Even")
else:
print("Odd")
Enter a number: 45
Odd

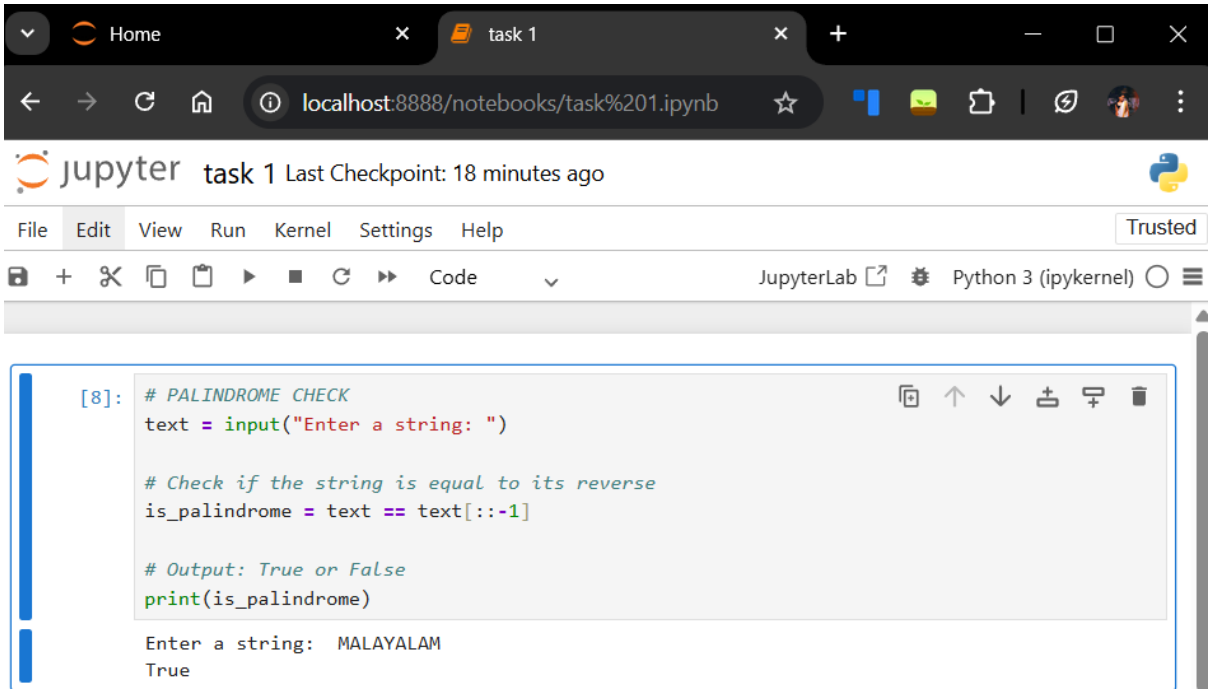

Home
task 1
localhost:8888/notebooks/task%201.ipynb
jupyter task 1 Last Checkpoint: 11 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[4]:
# factorial calculation
# Input: Get an integer from the user
n = int(input("Enter a number: "))
# Initialize result
factorial = 1
# Calculate factorial using loop
for i in range(1, n + 1):
factorial *= i
# Output
print("Factorial of", n, "is", factorial)
Enter a number: 5
Factorial of 5 is 120

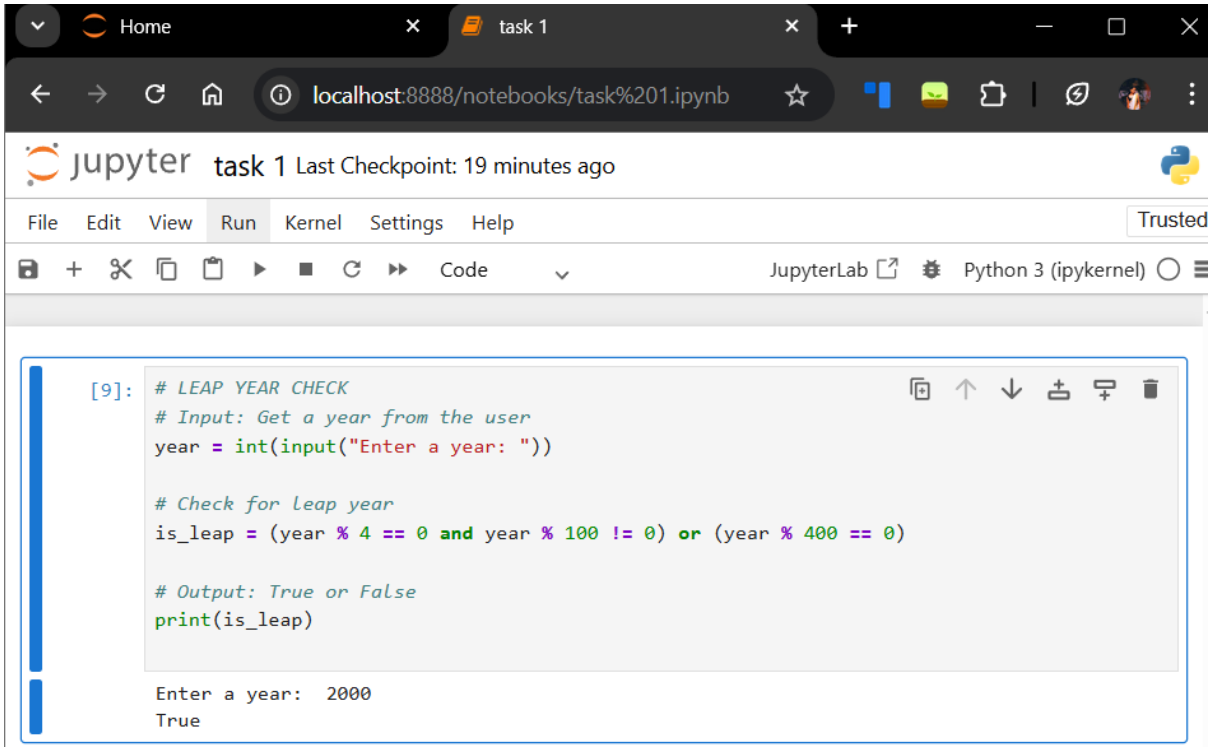

Home
task 1
localhost:8888/notebooks/task%201.ipynb
jupyter task 1 Last Checkpoint: 12 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[5]:
# fibonacci sequence
# Input: Get the number of terms
n = int(input("Enter the number of Fibonacci terms: "))
# Initialize the first two Fibonacci numbers
fib_sequence = []
a, b = 0, 1
for _ in range(n):
fib_sequence.append(a)
a, b = b, a + b
# Output
print("Fibonacci sequence:", fib_sequence)
Enter the number of Fibonacci terms: 8
Fibonacci sequence: [0, 1, 1, 2, 3, 5, 8, 13]

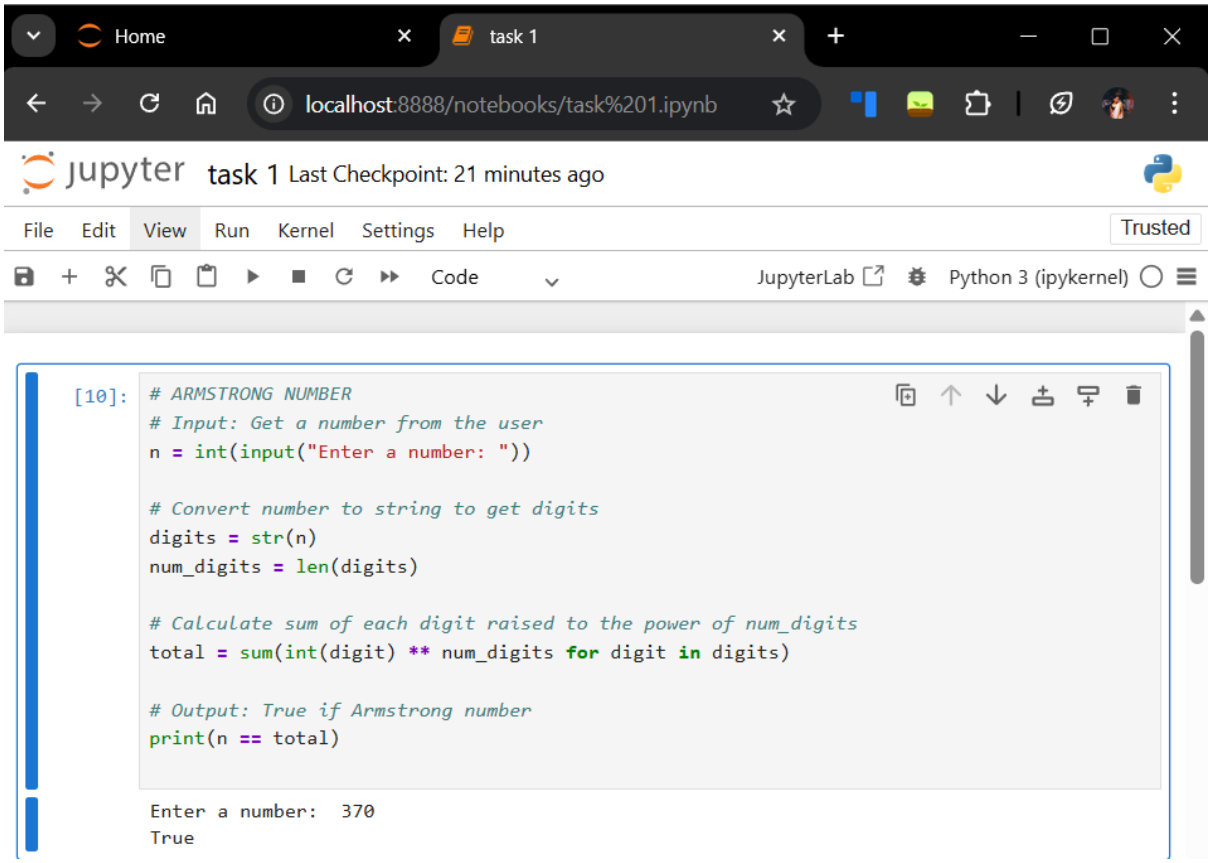
reverse a string

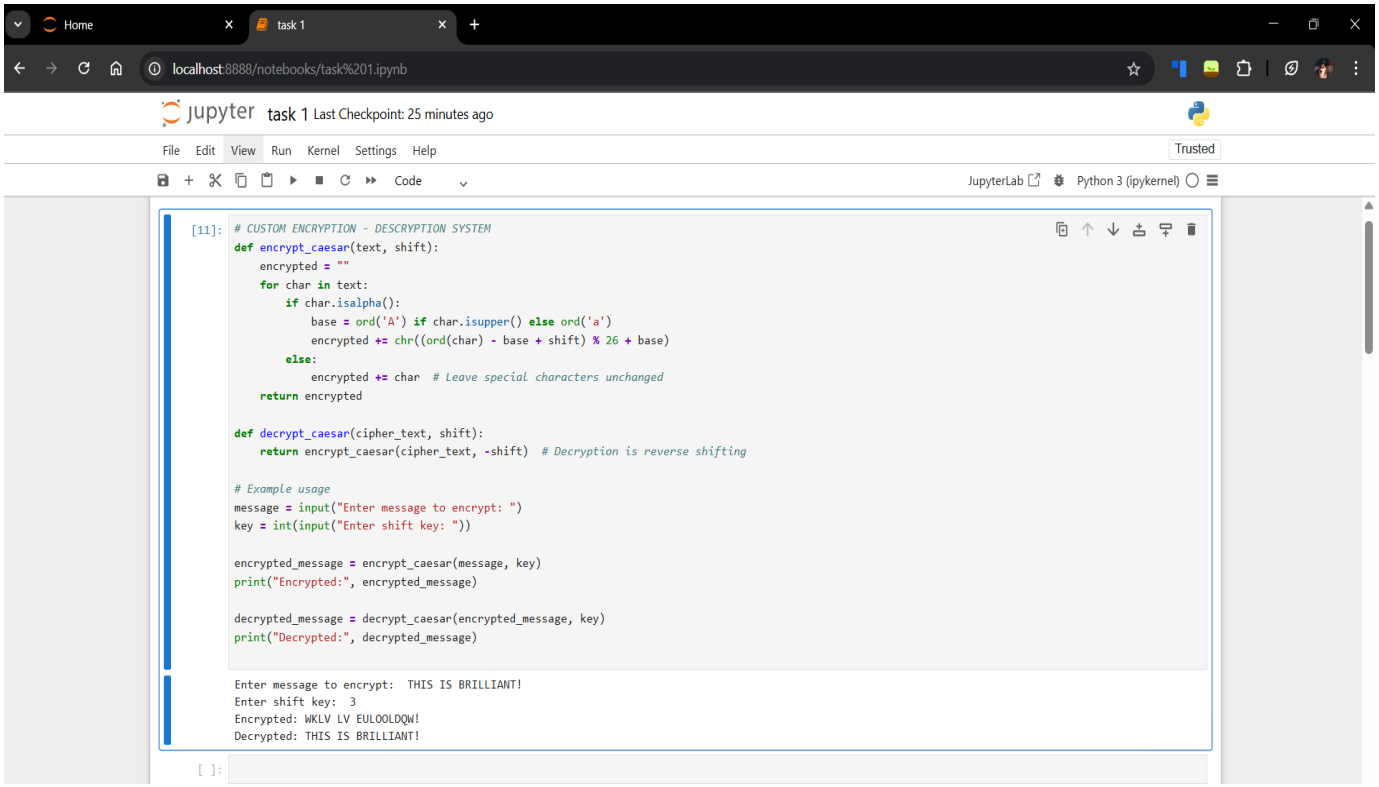


palindrome




Home
task 1
localhost:8888/notebooks/task%201.ipynb
jupyter task 1 Last Checkpoint: 19 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[9]:
# LEAP YEAR CHECK
# Input: Get a year from the user
year = int(input("Enter a year: "))
# Check for leap year
is_leap = (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
# Output: True or False
print(is_leap)
Enter a year: 2000
True


Home
task 1
localhost:8888/notebooks/task%201.ipynb
jupyter task 1 Last Checkpoint: 21 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[10]:
# ARMSTRONG NUMBER
# Input: Get a number from the user
n = int(input("Enter a number: "))
# Convert number to string to get digits
digits = str(n)
num_digits = len(digits)
# Calculate sum of each digit raised to the power of num_digits
total = sum(int(digit) ** num_digits for digit in digits)
# Output: True if Armstrong number
print(n == total)
Enter a number: 370
True


Home
task 1
localhost:8888/notebooks/task%201.ipynb
jupyter task 1 Last Checkpoint: 25 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[11]:
# CUSTOM ENCRYPTION - DESCRYPTION SYSTEM
def encrypt_caesar(text, shift):
encrypted = ""
for char in text:
if char.isalpha():
base = ord('A') if char.isupper() else ord('a')
encrypted += chr((ord(char) - base + shift) % 26 + base)
else:
encrypted += char # Leave special characters unchanged
return encrypted
def decrypt_caesar(cipher_text, shift):
return encrypt_caesar(cipher_text, -shift) # Decryption is reverse shifting
# Example usage
message = input("Enter message to encrypt: ")
key = int(input("Enter shift key: "))
encrypted_message = encrypt_caesar(message, key)
print("Encrypted:", encrypted_message)
decrypted_message = decrypt_caesar(encrypted_message, key)
print("Decrypted:", decrypted_message)
Enter message to encrypt: THIS IS BRILLIANT!
Enter shift key: 3
Encrypted: WKLV LV EULOOLDQW!
Decrypted: THIS IS BRILLIANT!
[ ]: