LAB ASSIGNMENT - 1.1

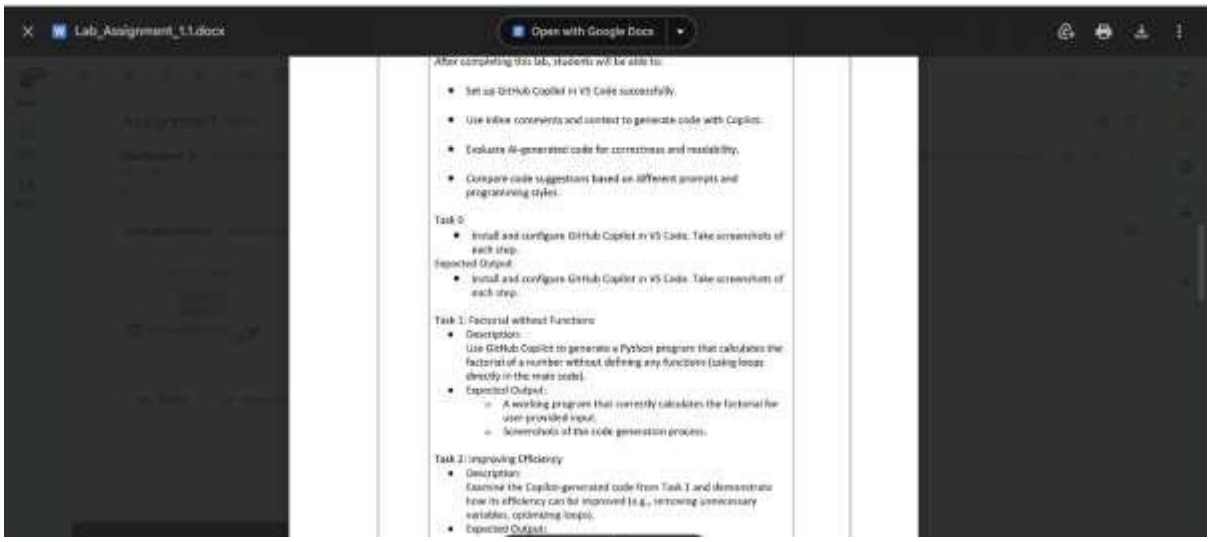
NAME : M.TANVI

ROLL NO : 2403A510A4

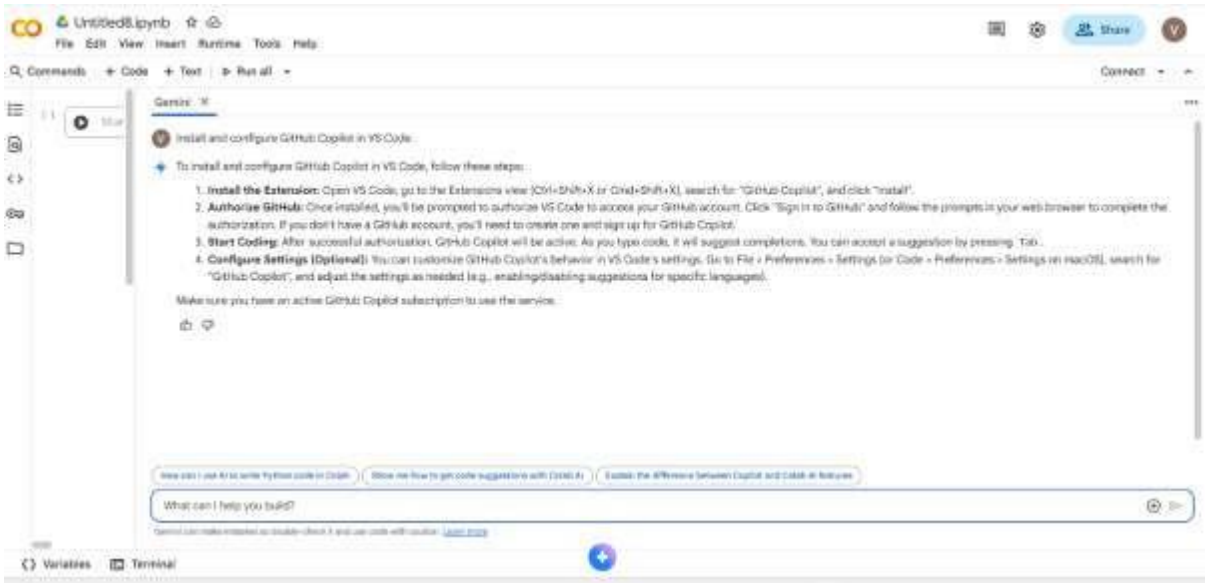
COURSE : AI ASSISTED CODING

BATCH : 01

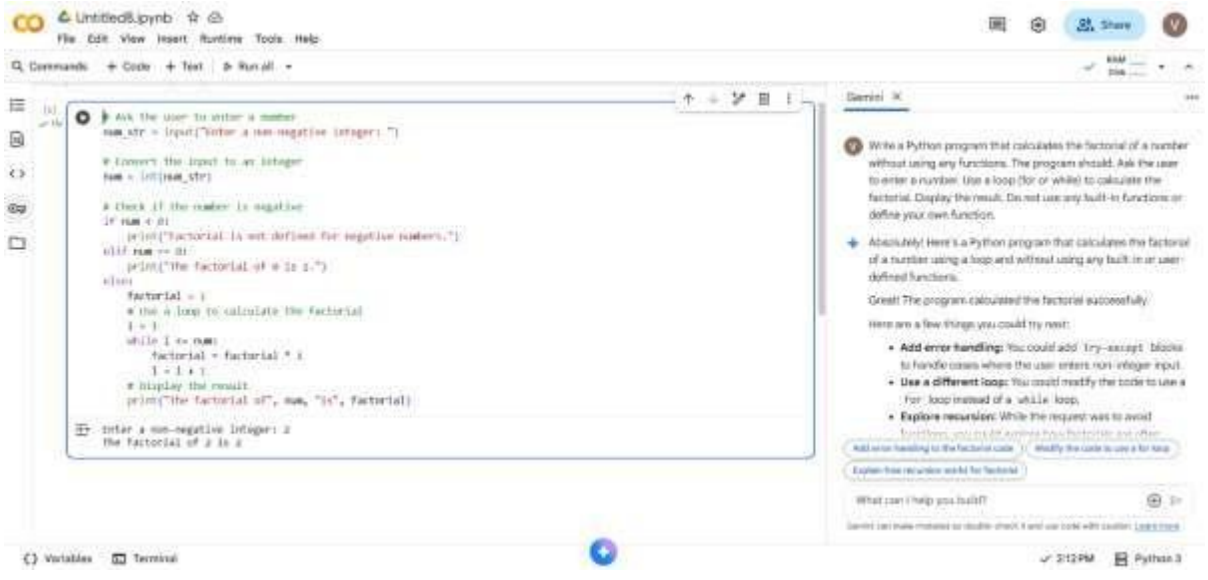
QUESTIONS



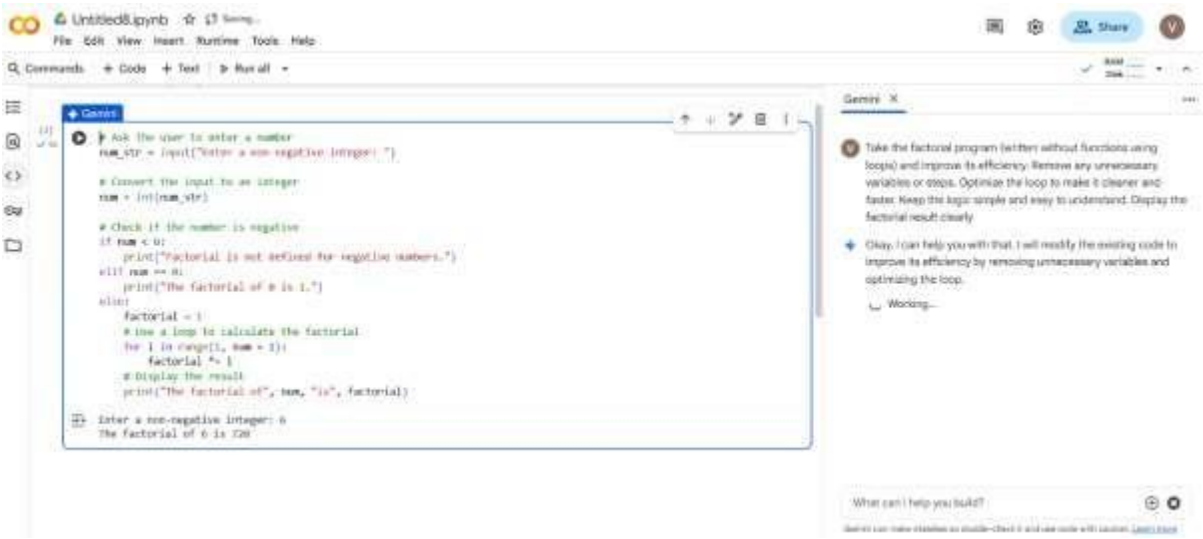
TASK 0:



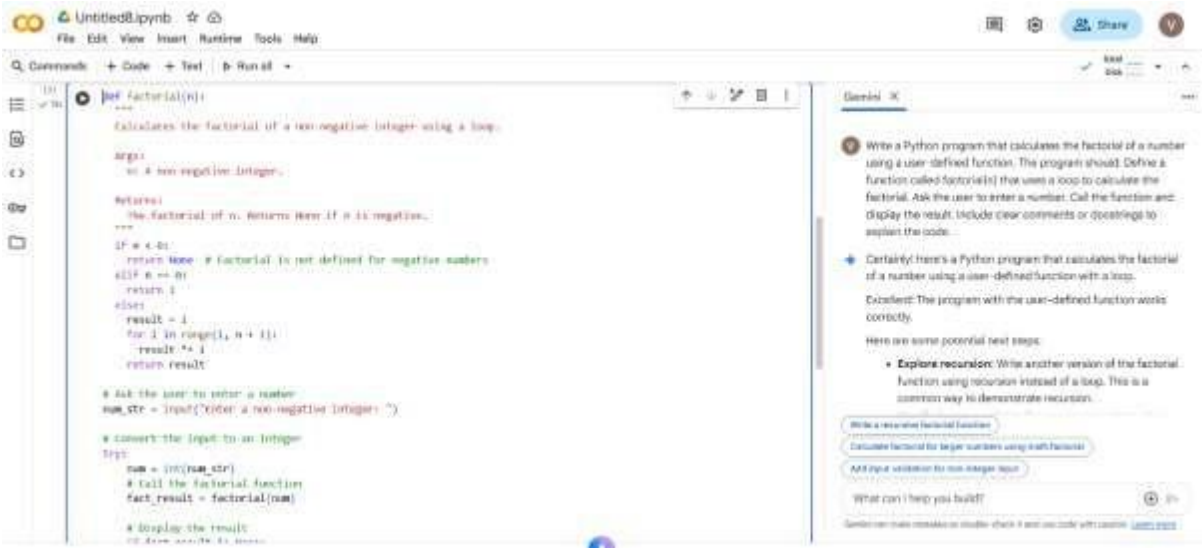
TASK 1 :

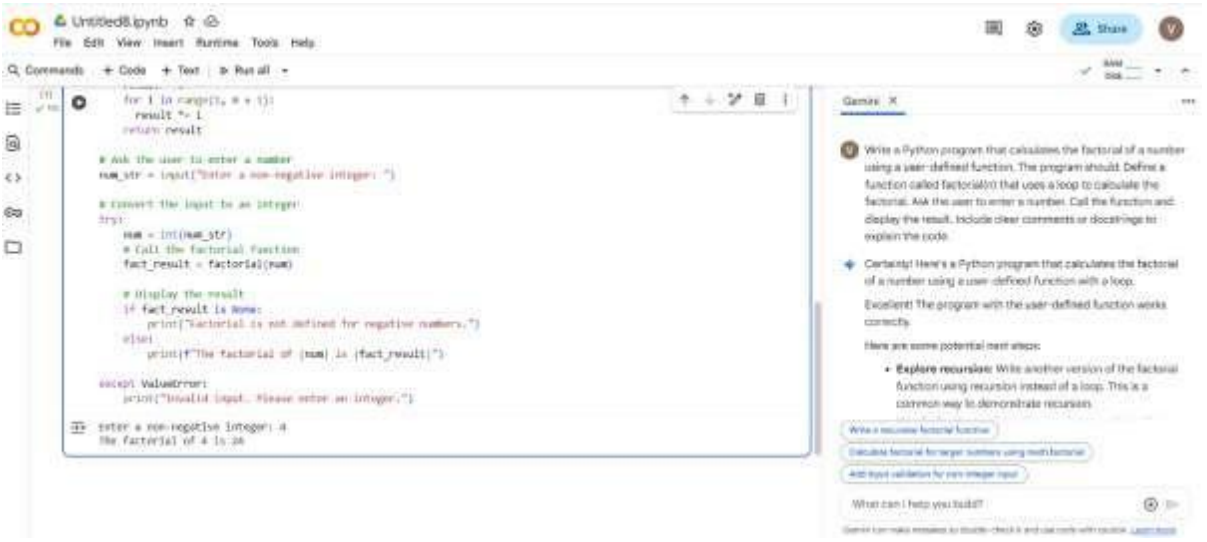


TASK 2 :

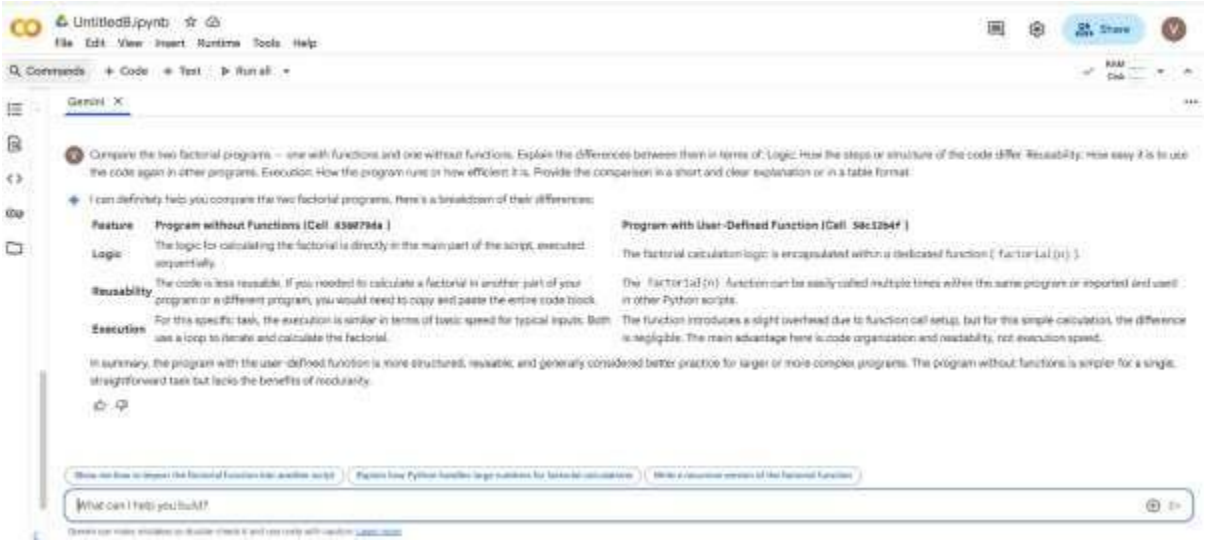

Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Gemini
# Ask the user to enter a number
num_str = input("Enter a non-negative integer: ")
# Convert the input to an integer
num = int(num_str)
# Check if the number is negative
if num < 0:
print("Factorial is not defined for negative numbers.")
elif num == 0:
print("The factorial of 0 is 1.")
else:
factorial = 1
# Use a loop to calculate the factorial
for i in range(1, num + 1):
factorial *= i
# Display the result
print("The factorial of", num, "is", factorial)
Enter a non-negative integer: 6
The factorial of 6 is 720
Gemini
Take the factorial program (written without functions using loops) and improve its efficiency. Remove any unnecessary variables or steps. Optimize the loop to make it cleaner and faster. Keep the logic simple and easy to understand. Display the factorial result clearly.
Okay, I can help you with that. I will modify the existing code to improve its efficiency by removing unnecessary variables and optimizing the loop.
Working...
What can I help you build?

TASK 3 :

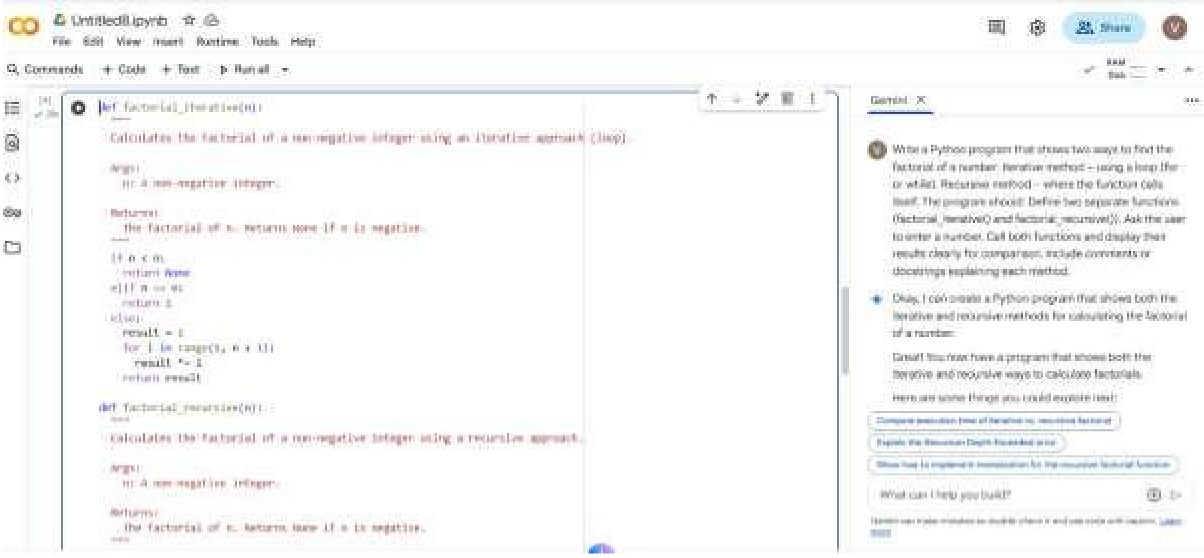

Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
def factorial(n):
"""
Calculates the factorial of a non-negative integer using a loop.
Args:
n: A non-negative integer.
Returns:
The factorial of n. Returns None if n is negative.
"""
if n < 0:
return None # Factorial is not defined for negative numbers
elif n == 0:
return 1
else:
result = 1
for i in range(1, n + 1):
result *= i
return result
# Ask the user to enter a number
num_str = input("Enter a non-negative integer: ")
# Convert the input to an integer
try:
num = int(num_str)
# Call the factorial function
fact_result = factorial(num)
# Display the result
Gemini
Write a Python program that calculates the factorial of a number using a user-defined function. The program should: Define a function called factorial(n) that uses a loop to calculate the factorial. Ask the user to enter a number. Call the function and display the result. Include clear comments or docstrings to explain the code.
Certainly! Here's a Python program that calculates the factorial of a number using a user-defined function with a loop.
Excellent! The program with the user-defined function works correctly.
Here are some potential next steps:
Explore recursion: Write another version of the factorial function using recursion instead of a loop. This is a common way to demonstrate recursion.
What can I help you build?

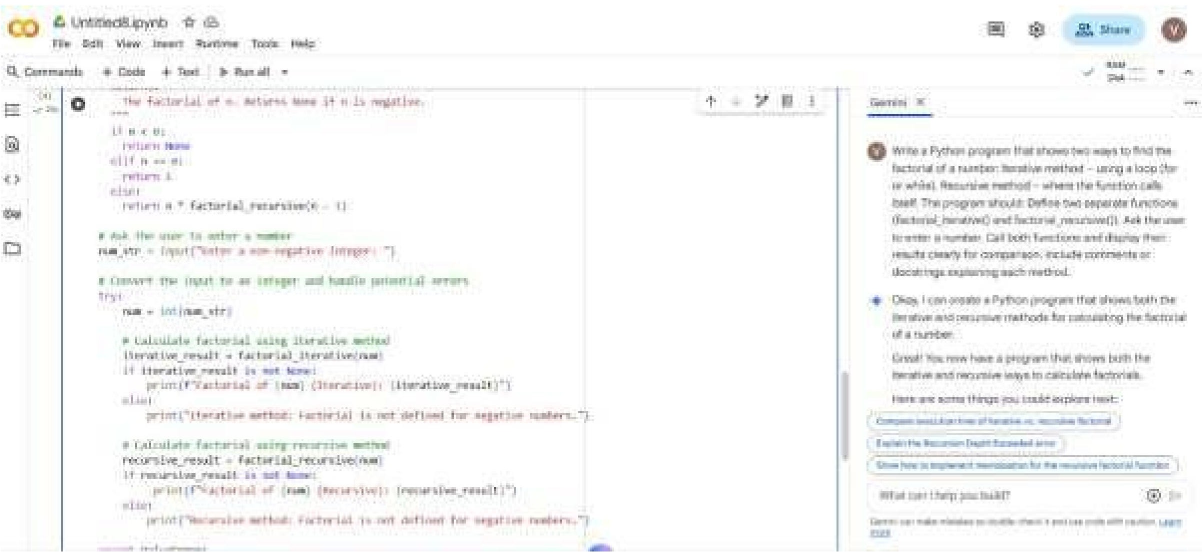

Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
for i in range(1, n + 1):
result *= i
return result
# Ask the user to enter a number
num_str = input("Enter a non-negative integer: ")
# Convert the input to an integer
try:
num = int(num_str)
# Call the factorial function
fact_result = factorial(num)
# Display the result
if fact_result is None:
print("Factorial is not defined for negative numbers.")
else:
print(f"The factorial of {num} is {fact_result}")
except ValueError:
print("Invalid input. Please enter an integer.")
Enter a non-negative integer: 4
The factorial of 4 is 24
Gemini
Write a Python program that calculates the factorial of a number using a user-defined function. The program should: Define a function called factorial(n) that uses a loop to calculate the factorial. Ask the user to enter a number. Call the function and display the result. Include clear comments or docstrings to explain the code.
Certainly! Here's a Python program that calculates the factorial of a number using a user-defined function with a loop.
Excellent! The program with the user-defined function works correctly.
Here are some potential next steps:
Explore recursion: Write another version of the factorial function using recursion instead of a loop. This is a common way to demonstrate recursion.
What can I help you build?

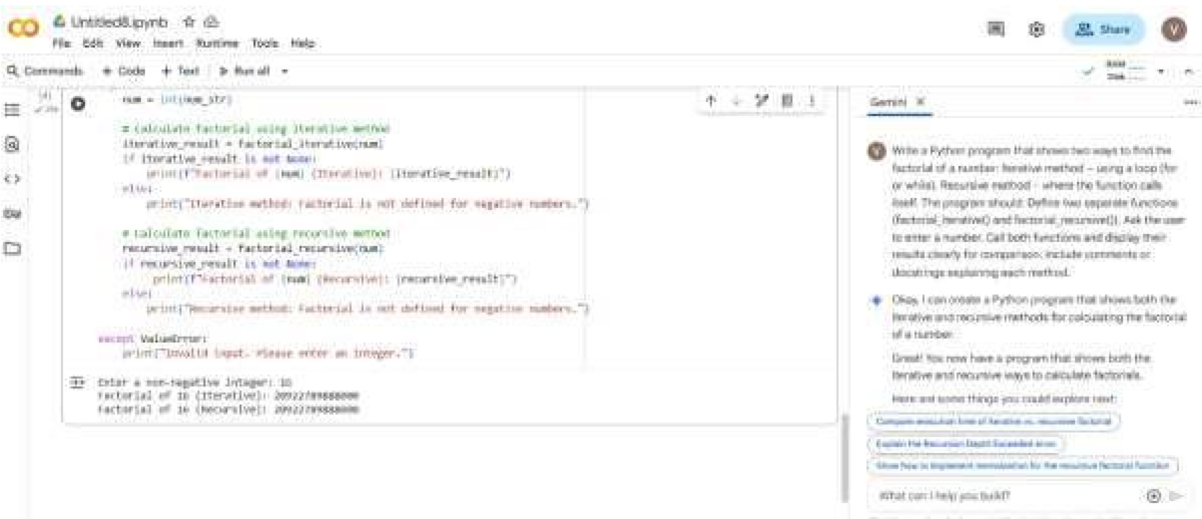
TASK 4 :


Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Gemini
Compare the two factorial programs – one with functions and one without functions. Explain the differences between them in terms of: Logic: How the steps or structure of the code differ. Reusability: How easy it is to use the code again in other programs. Execution: How the program runs or how efficient it is. Provide the comparison in a short and clear explanation or in a table format.
I can definitely help you compare the two factorial programs. Here's a breakdown of their differences:
Feature
Program without Functions (Cell 630879da)
Program with User-Defined Function (Cell 58c12b4f)
Logic
The logic for calculating the factorial is directly in the main part of the script, executed sequentially.
The factorial calculation logic is encapsulated within a dedicated function (factorial(n)).
Reusability
The code is less reusable. If you needed to calculate a factorial in another part of your program or a different program, you would need to copy and paste the entire code block.
The factorial(n) function can be easily called multiple times within the same program or imported and used in other Python scripts.
Execution
For this specific task, the execution is similar in terms of basic speed for typical inputs. Both use a loop to iterate and calculate the factorial.
The function introduces a slight overhead due to function call setup, but for this simple calculation, the difference is negligible. The main advantage here is code organization and reusability, not execution speed.
In summary, the program with the user-defined function is more structured, reusable, and generally considered better practice for larger or more complex programs. The program without functions is simpler for a single, straightforward task but lacks the benefits of modularity.
What can I help you build?

TASK 5 :


Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
def factorial_iterative(n):
"""
Calculates the factorial of a non-negative integer using an iterative approach (loop).
Args:
n: A non-negative integer.
Returns:
The factorial of n. Returns None if n is negative.
"""
if n < 0:
return None
elif n == 0:
return 1
else:
result = 1
for i in range(1, n + 1):
result *= i
return result
def factorial_recursive(n):
"""
Calculates the factorial of a non-negative integer using a recursive approach.
Args:
n: A non-negative integer.
Returns:
The factorial of n. Returns None if n is negative.


Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
The factorial of n. Returns None if n is negative.
"""
if n < 0:
return None
elif n == 0:
return 1
else:
return n * factorial_recursive(n - 1)
# Ask the user to enter a number
num_str = input("Enter a non-negative integer: ")
# Convert the input to an integer and handle potential errors
try:
num = int(num_str)
# Calculate factorial using iterative method
iterative_result = factorial_iterative(num)
if iterative_result is not None:
print(f"Factorial of {num} (Iterative): {iterative_result}")
else:
print("Iterative method: Factorial is not defined for negative numbers.")
# Calculate factorial using recursive method
recursive_result = factorial_recursive(num)
if recursive_result is not None:
print(f"Factorial of {num} (Recursive): {recursive_result}")
else:
print("Recursive method: Factorial is not defined for negative numbers.")


Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
num = int(num_str)
# Calculate factorial using iterative method
iterative_result = factorial_iterative(num)
if iterative_result is not None:
print(f"Factorial of {num} (Iterative): {iterative_result}")
else:
print("Iterative method: Factorial is not defined for negative numbers.")
# Calculate factorial using recursive method
recursive_result = factorial_recursive(num)
if recursive_result is not None:
print(f"Factorial of {num} (Recursive): {recursive_result}")
else:
print("Recursive method: Factorial is not defined for negative numbers.")
except ValueError:
print("Invalid input. Please enter an integer.")
Enter a non-negative integer: 16
Factorial of 16 (Iterative): 20922789888000
Factorial of 16 (Recursive): 20922789888000