ASSIGNMENT – 3.4

NAME : s.harshavardhan

HT.NO : 2303A51563

BATCH : 10

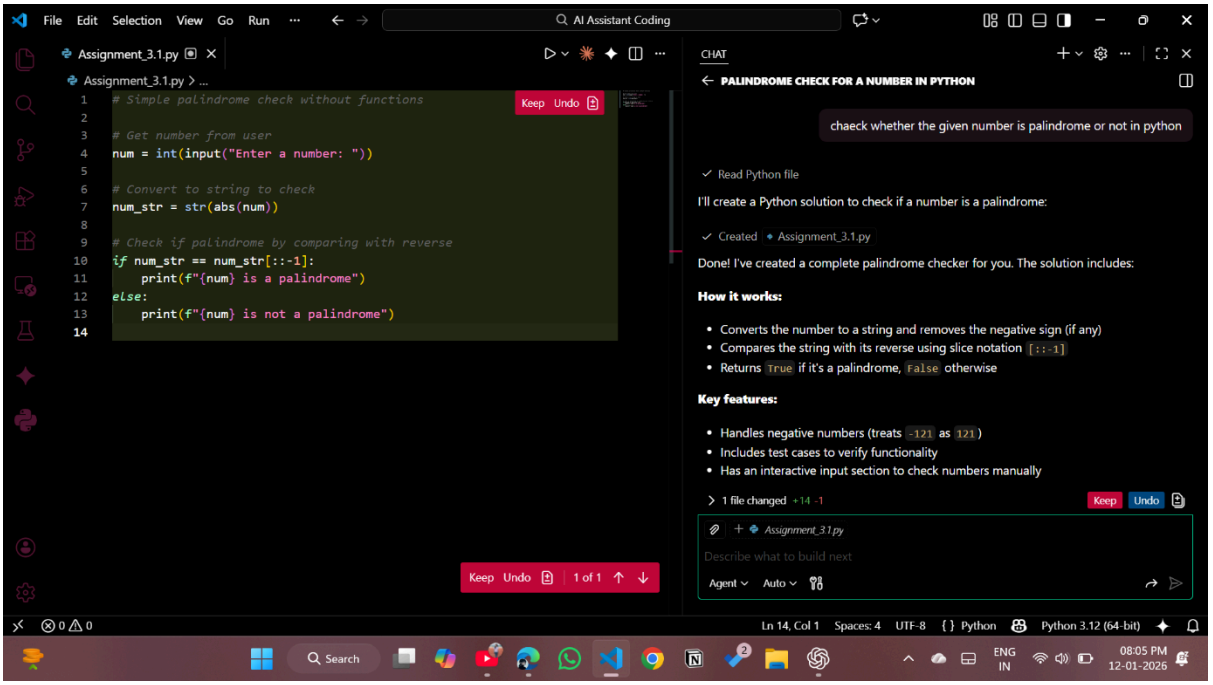
Question 1: Zero-Shot Prompting (Palindrome Number Program)

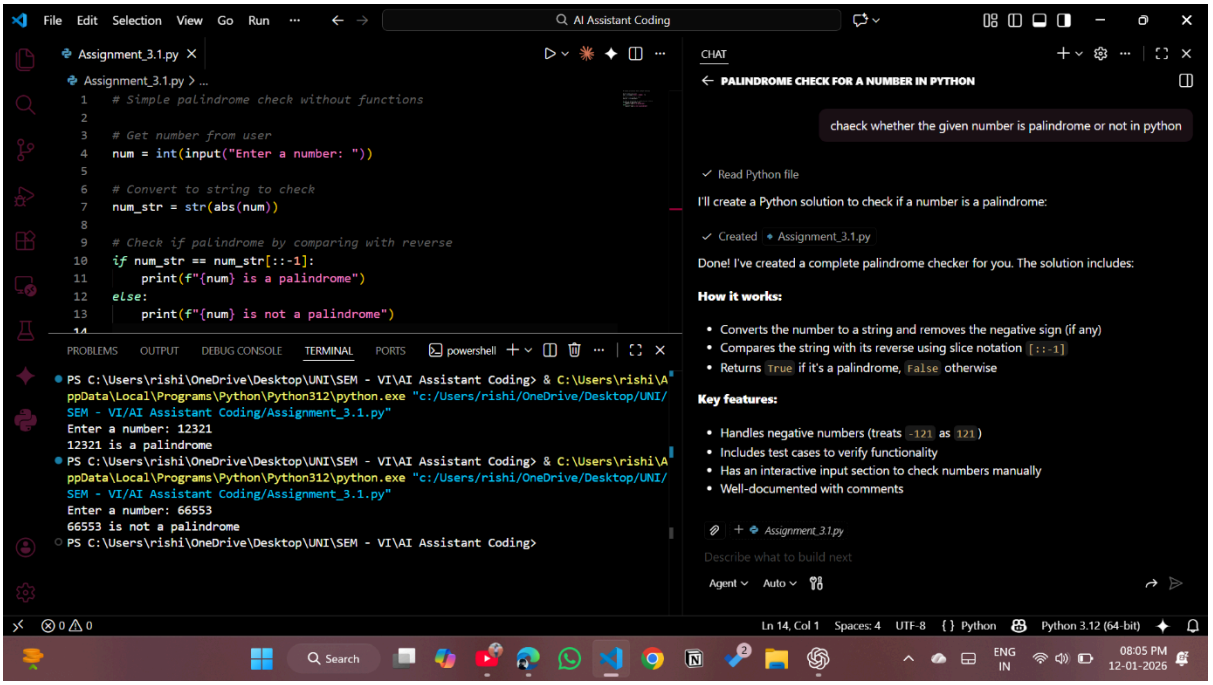
Write a zero-shot prompt (without providing any examples) to generate a Python function that checks whether a given number is a palindrome.

Task:

- Record the AI-generated code.
- Test the code with multiple inputs.
- Identify any logical errors or missing edge-case handling.

PROMPT : check whether the given number is palindrome or not in python.


File Edit Selection View Go Run
AI Assistant Coding
Assignment_3.1.py
Assignment_3.1.py > ...
# Simple palindrome check without functions
# Get number from user
num = int(input("Enter a number: "))
# Convert to string to check
num_str = str(abs(num))
# Check if palindrome by comparing with reverse
if num_str == num_str[::-1]:
print(f"{num} is a palindrome")
else:
print(f"{num} is not a palindrome")
Keep Undo
Keep Undo 1 of 1
CHAT
PALINDROME CHECK FOR A NUMBER IN PYTHON
chaeck whether the given number is palindrome or not in python
Read Python file
I'll create a Python solution to check if a number is a palindrome:
Created Assignment_3.1.py
Done! I've created a complete palindrome checker for you. The solution includes:
How it works:
Converts the number to a string and removes the negative sign (if any)
Compares the string with its reverse using slice notation [::-1]
Returns True if it's a palindrome, False otherwise
Key features:
Handles negative numbers (treats -121 as 121)
Includes test cases to verify functionality
Has an interactive input section to check numbers manually
1 file changed +14 -1
Keep Undo
Assignment_3.1.py
Describe what to build next
Agent Auto
Ln 14, Col 1 Spaces: 4 UTF-8 { } Python Python 3.12 (64-bit)
Search
ENG IN
08:05 PM 12-01-2026


File Edit Selection View Go Run
AI Assistant Coding
Assignment_3.1.py
Assignment_3.1.py > ...
# Simple palindrome check without functions
# Get number from user
num = int(input("Enter a number: "))
# Convert to string to check
num_str = str(abs(num))
# Check if palindrome by comparing with reverse
if num_str == num_str[::-1]:
print(f"{num} is a palindrome")
else:
print(f"{num} is not a palindrome")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
PS C:\Users\rishi\OneDrive\Desktop\UNI\SEM - VI\AI Assistant Coding> & C:\Users\rishi\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/rishi/OneDrive/Desktop/UNI/SEM - VI/AI Assistant Coding/Assignment_3.1.py"
Enter a number: 12321
12321 is a palindrome
PS C:\Users\rishi\OneDrive\Desktop\UNI\SEM - VI\AI Assistant Coding> & C:\Users\rishi\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/rishi/OneDrive/Desktop/UNI/SEM - VI/AI Assistant Coding/Assignment_3.1.py"
Enter a number: 66553
66553 is not a palindrome
PS C:\Users\rishi\OneDrive\Desktop\UNI\SEM - VI\AI Assistant Coding>
CHAT
PALINDROME CHECK FOR A NUMBER IN PYTHON
chaeck whether the given number is palindrome or not in python
Read Python file
I'll create a Python solution to check if a number is a palindrome:
Created Assignment_3.1.py
Done! I've created a complete palindrome checker for you. The solution includes:
How it works:
Converts the number to a string and removes the negative sign (if any)
Compares the string with its reverse using slice notation [::-1]
Returns True if it's a palindrome, False otherwise
Key features:
Handles negative numbers (treats -121 as 121)
Includes test cases to verify functionality
Has an interactive input section to check numbers manually
Well-documented with comments
Assignment_3.1.py
Describe what to build next
Agent Auto
Ln 14, Col 1 Spaces: 4 UTF-8 { } Python Python 3.12 (64-bit)
Search
ENG IN
08:05 PM 12-01-2026

Question 2: One-Shot Prompting (Factorial Calculation)

Write a one-shot prompt by providing one input-output example and ask the AI to generate a Python function to compute the factorial of a given number.

Example:

Input: 5 → Output: 120

Task:

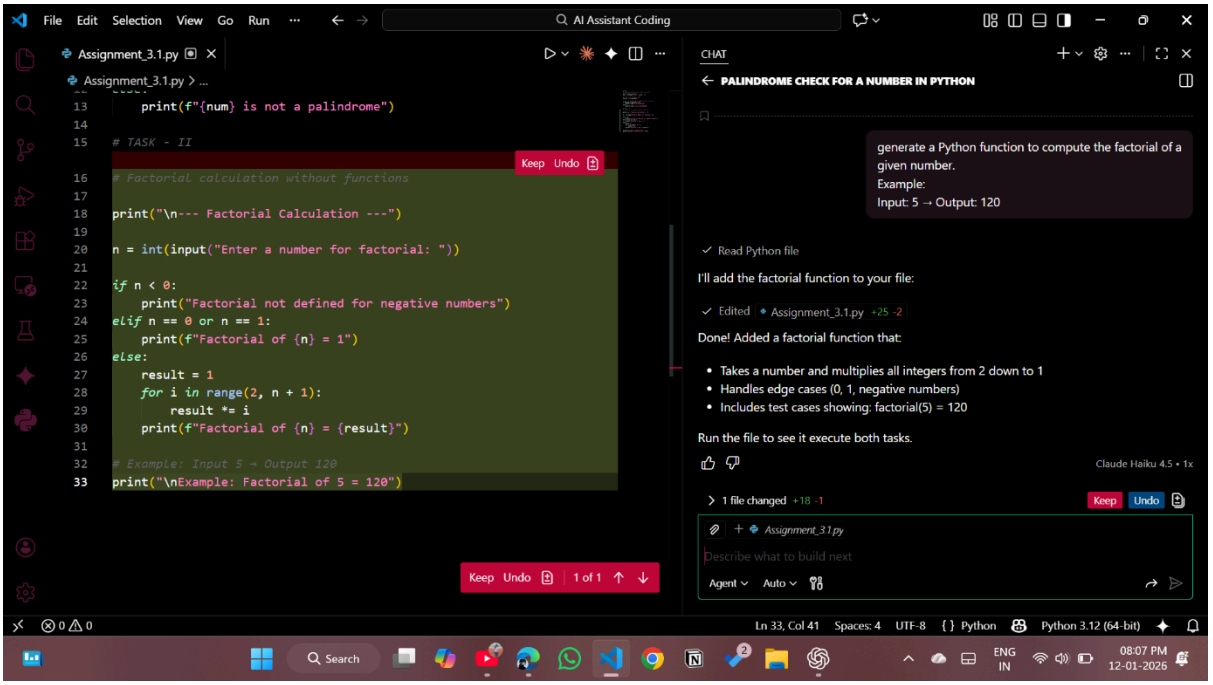
• Compare the generated code with a zero-shot solution.

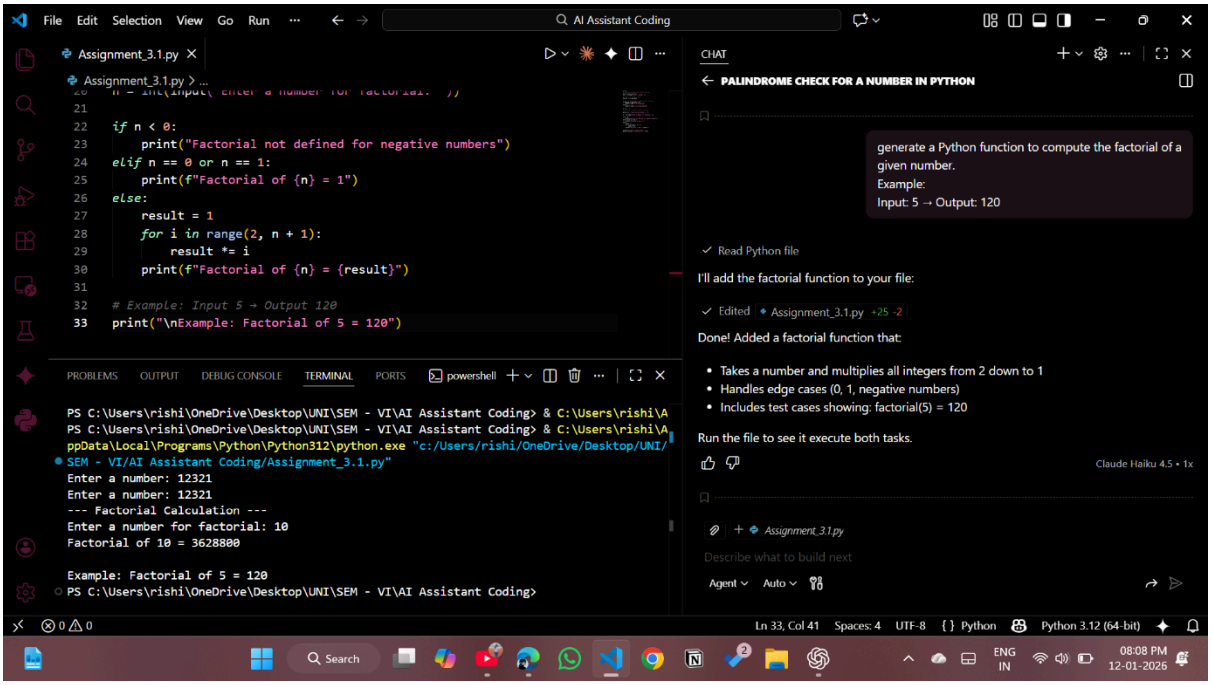
• Examine improvements in clarity and correctness.

PROMPT : Generate a Python function to compute the factorial of a given number.

Example:

Input: 5 → Output: 120





Question 3: Few-Shot Prompting (Armstrong Number Check)

Write a few-shot prompt by providing multiple input-output examples to guide the AI in generating a Python function to check whether a given number is an Armstrong number.

Examples:

• Input: 153 → Output: Armstrong Number

• Input: 370 → Output: Armstrong Number

• Input: 123 → Output: Not an Armstrong Number

Task:

• Analyze how multiple examples influence code structure and accuracy.

• Test the function with boundary values and invalid inputs.

PROMPT : Python function to check whether a

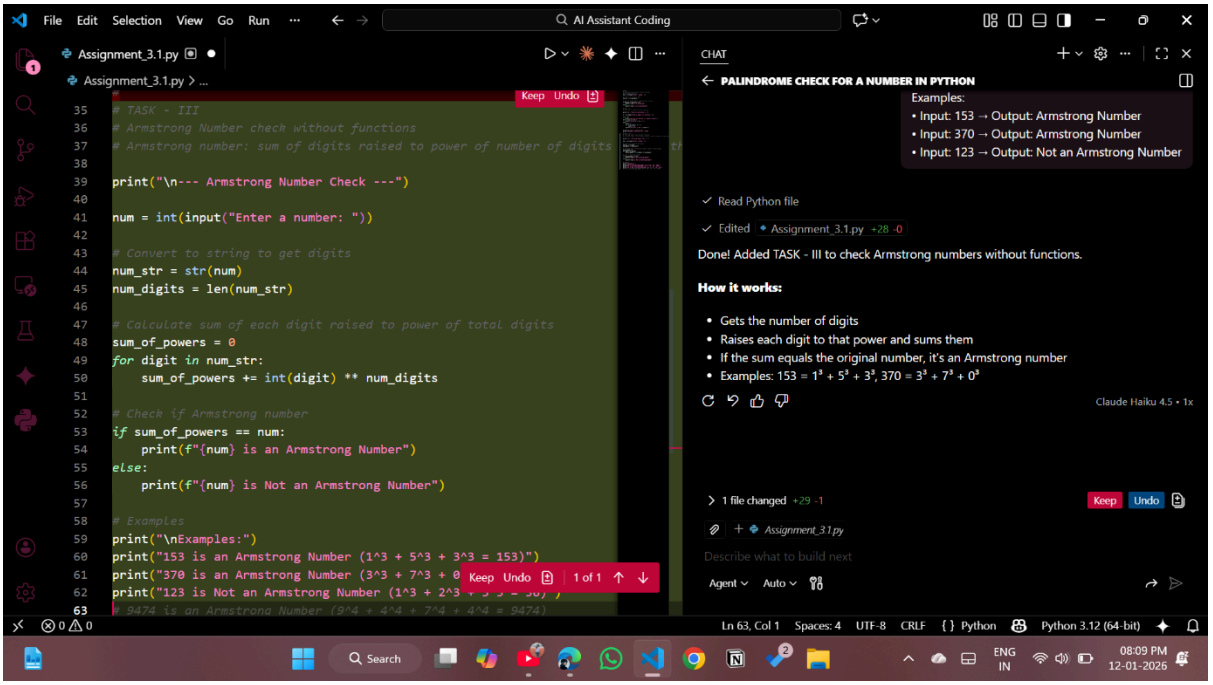
given number is an Armstrong number.

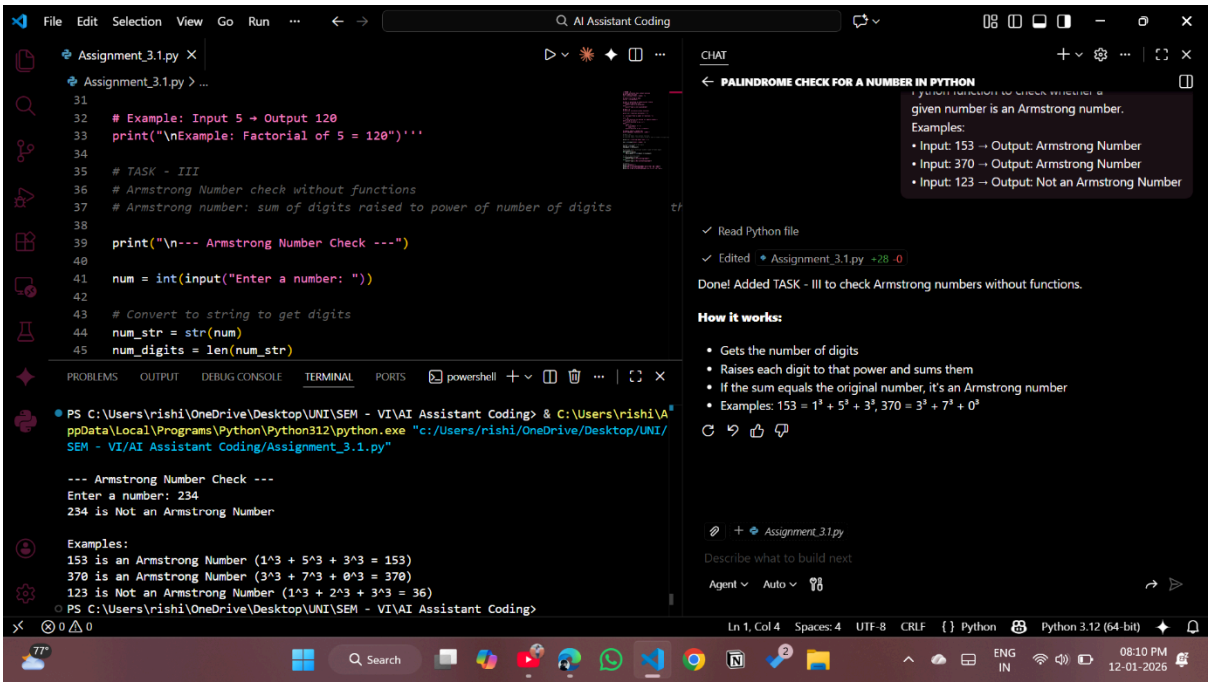
Examples:

• Input: 153 → Output: Armstrong Number

• Input: 370 → Output: Armstrong Number

• Input: 123 → Output: Not an Armstrong Number





(Optional Extension)

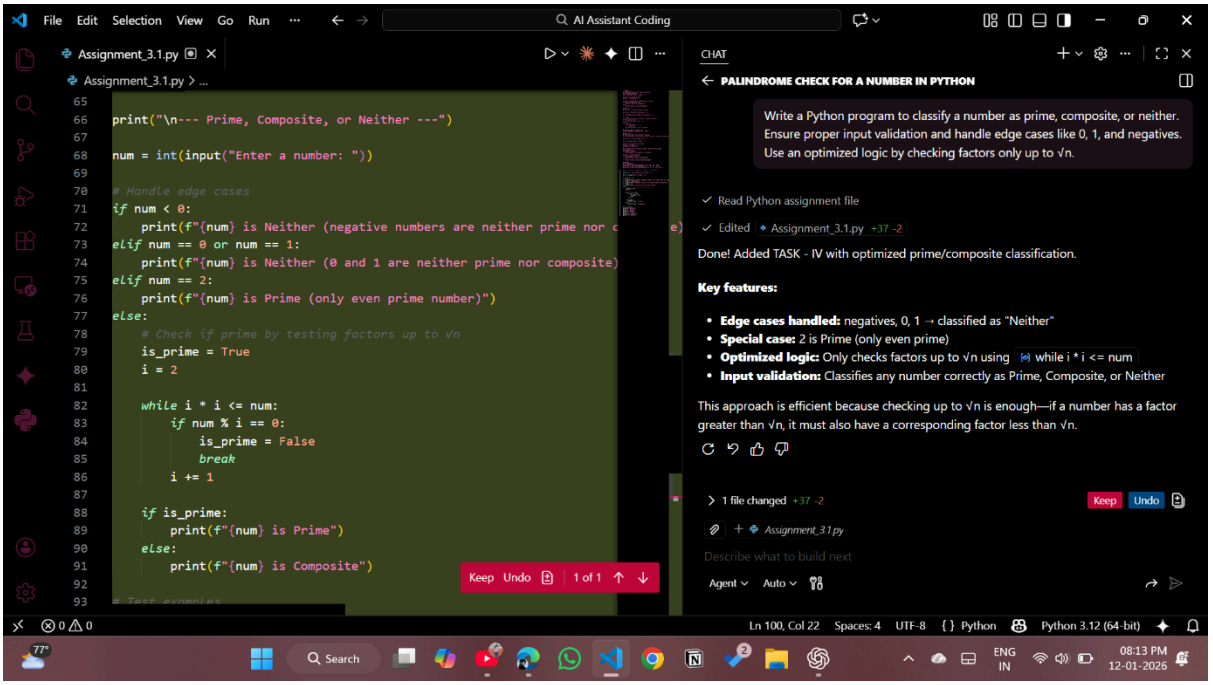
Question 4: Context-Managed Prompting (Optimized Number Classification)

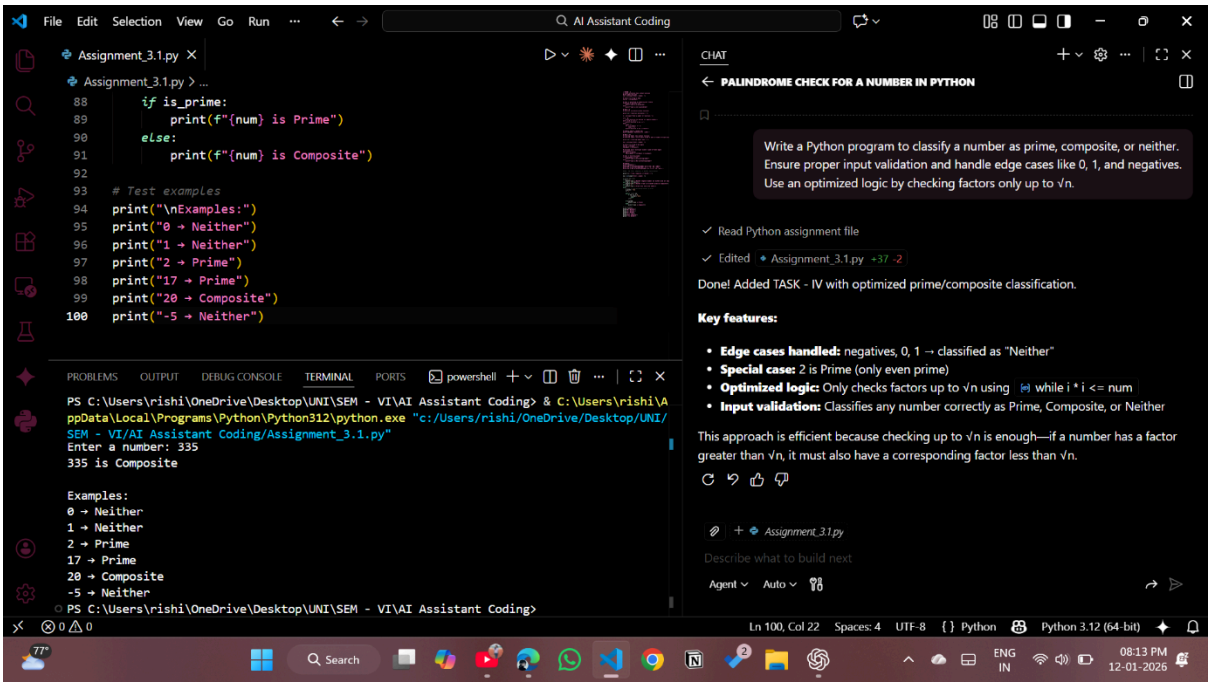
Design a context-managed prompt with clear instructions and constraints to generate an optimized Python program that classifies a number as prime, composite, or neither.

Task:

- Ensure proper input validation.
- Optimize the logic for efficiency.
- Compare the output with earlier prompting strategies.

PROMPT : Write a Python program to classify a number as prime, composite, or neither. Ensure proper input validation and handle edge cases like 0, 1, and negatives. Use an optimized logic by checking factors only up to √n.





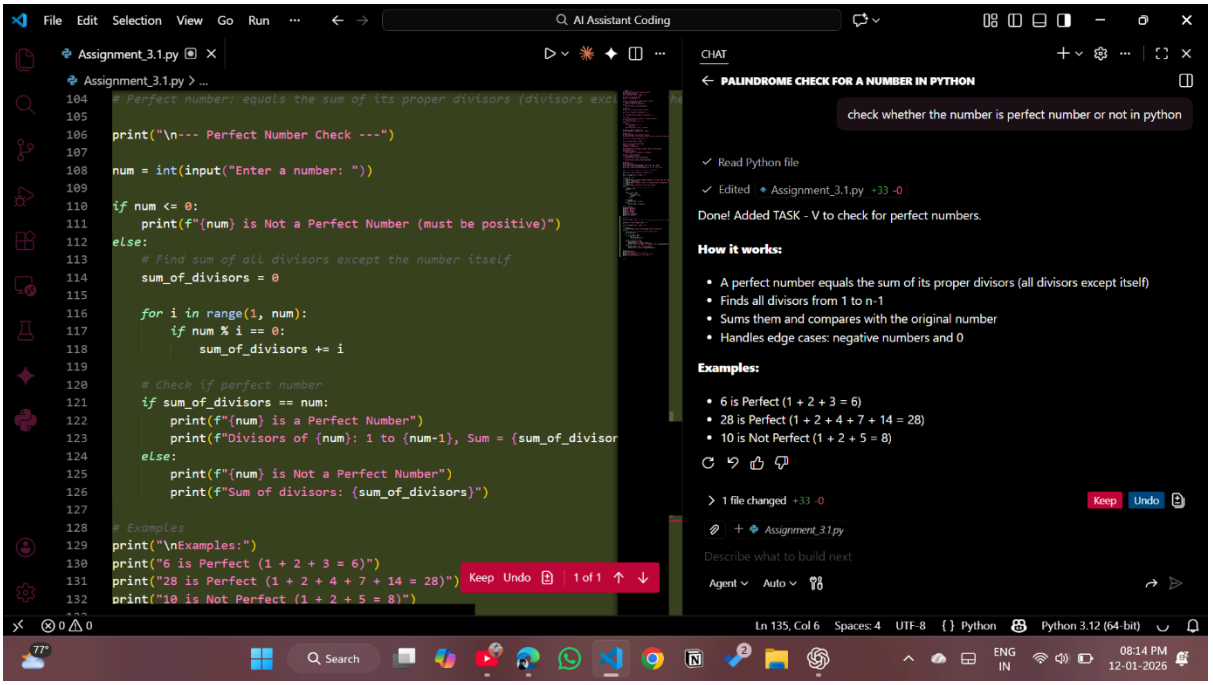
Question 5: Zero-Shot Prompting (Perfect Number Check)

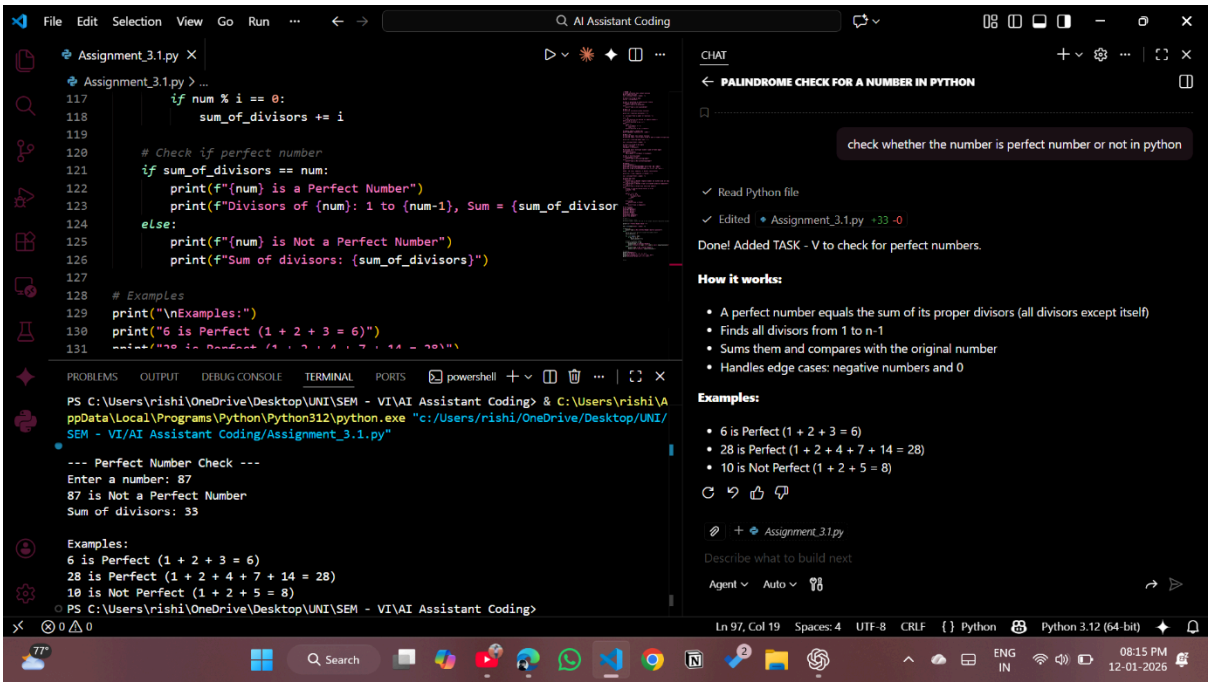
Write a zero-shot prompt (without providing any examples) to generate a Python function that checks whether a given number is a perfect number.

Task:

- Record the AI-generated code.
- Test the program with multiple inputs.
- Identify any missing conditions or inefficiencies in the logic.

PROMPT : check whether the number is perfect number or not in python.





Question 6: Few-Shot Prompting (Even or Odd Classification with Validation)

Write a few-shot prompt by providing multiple input-output examples to guide the AI in generating a

Python program that determines whether a given number is even or odd, including proper input validation.

Examples:

• Input: 8 → Output: Even

• Input: 15 → Output: Odd

• Input: 0 → Output: Even

Task:

• Analyze how examples improve input handling and output clarity.

• Test the program with negative numbers and non-integer inputs.

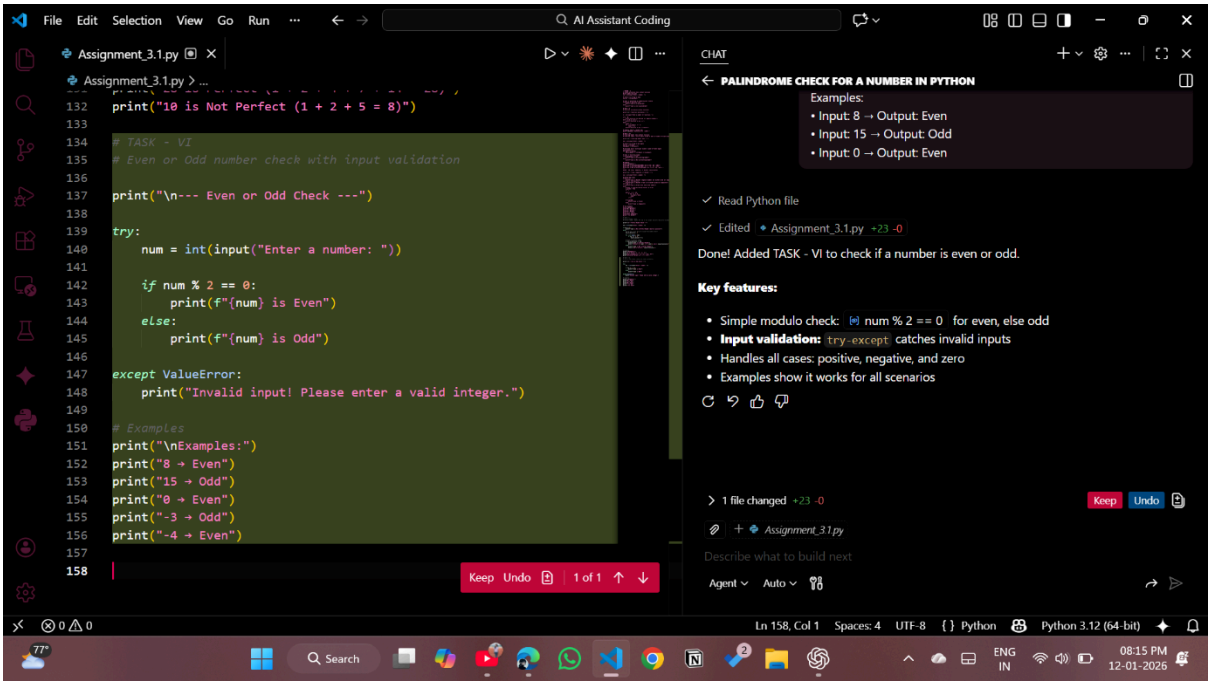
PROMPT : Python program that determines whether a given number is even or odd, including proper input validation.

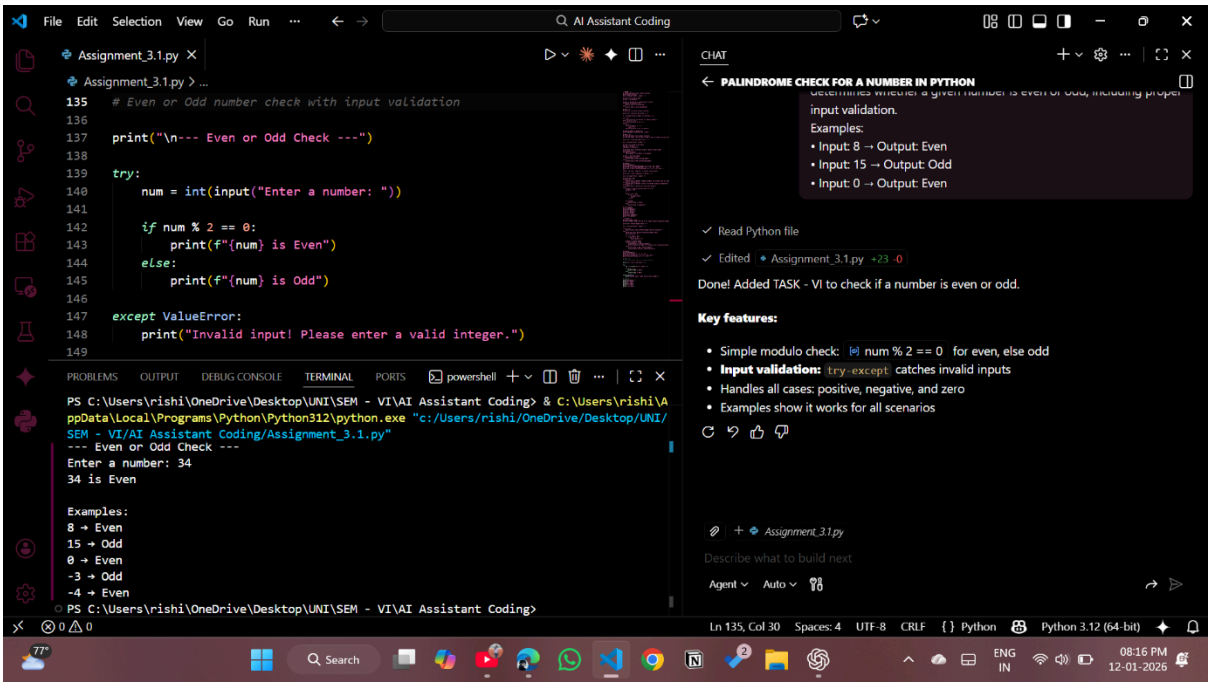
Examples:

• Input: 8 → Output: Even

• Input: 15 → Output: Odd

• Input: 0 → Output: Even


File Edit Selection View Go Run
AI Assistant Coding
Assignment_3.1.py
print("10 is Not Perfect (1 + 2 + 5 = 8)")
# TASK - VI
# Even or Odd number check with input validation
print("\n--- Even or Odd Check ---")
try:
num = int(input("Enter a number: "))
if num % 2 == 0:
print(f"{num} is Even")
else:
print(f"{num} is Odd")
except ValueError:
print("Invalid input! Please enter a valid integer.")
# Examples
print("\nExamples:")
print("8 → Even")
print("15 → Odd")
print("0 → Even")
print("-3 → Odd")
print("-4 → Even")
Keep Undo 1 of 1
CHAT
PALINDROME CHECK FOR A NUMBER IN PYTHON
Examples:
• Input: 8 → Output: Even
• Input: 15 → Output: Odd
• Input: 0 → Output: Even
Read Python file
Edited Assignment_3.1.py +23 -0
Done! Added TASK - VI to check if a number is even or odd.
Key features:
• Simple modulo check: num % 2 == 0 for even, else odd
• Input validation: try-except catches invalid inputs
• Handles all cases: positive, negative, and zero
• Examples show it works for all scenarios
1 file changed +23 -0
Keep Undo
Assignment_3.1.py
Describe what to build next
Agent Auto
Ln 158, Col 1 Spaces: 4 UTF-8 { } Python Python 3.12 (64-bit)
Search
ENG IN
08:15 PM 12-01-2026


File Edit Selection View Go Run
AI Assistant Coding
Assignment_3.1.py
# Even or Odd number check with input validation
print("\n--- Even or Odd Check ---")
try:
num = int(input("Enter a number: "))
if num % 2 == 0:
print(f"{num} is Even")
else:
print(f"{num} is Odd")
except ValueError:
print("Invalid input! Please enter a valid integer.")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
PS C:\Users\rishi\OneDrive\Desktop\UNI\SEM - VI\AI Assistant Coding> & C:\Users\rishi\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/rishi/OneDrive/Desktop/UNI/SEM - VI/AI Assistant Coding/Assignment_3.1.py"
--- Even or Odd Check ---
Enter a number: 34
34 is Even
Examples:
8 → Even
15 → Odd
0 → Even
-3 → Odd
-4 → Even
PS C:\Users\rishi\OneDrive\Desktop\UNI\SEM - VI\AI Assistant Coding>
CHAT
PALINDROME CHECK FOR A NUMBER IN PYTHON
input validation.
Examples:
• Input: 8 → Output: Even
• Input: 15 → Output: Odd
• Input: 0 → Output: Even
Read Python file
Edited Assignment_3.1.py +23 -0
Done! Added TASK - VI to check if a number is even or odd.
Key features:
• Simple modulo check: num % 2 == 0 for even, else odd
• Input validation: try-except catches invalid inputs
• Handles all cases: positive, negative, and zero
• Examples show it works for all scenarios
Assignment_3.1.py
Describe what to build next
Agent Auto
Ln 135, Col 30 Spaces: 4 UTF-8 CRLF { } Python Python 3.12 (64-bit)
Search
ENG IN
08:16 PM 12-01-2026