**ASSIGNMENT 3.3**

NAME: md.Heena

H.NO: 2403A54067

SUBJECT: AI ASSISTANT CODING

BATCH: 01

TASK DESCRIPTION 1:

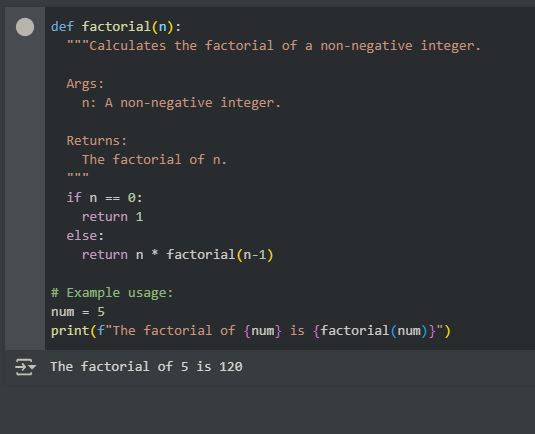
QUESTION:

Try 3 different prompts to generate a factorial function

PROMPT1:

generate a code in python factorial function

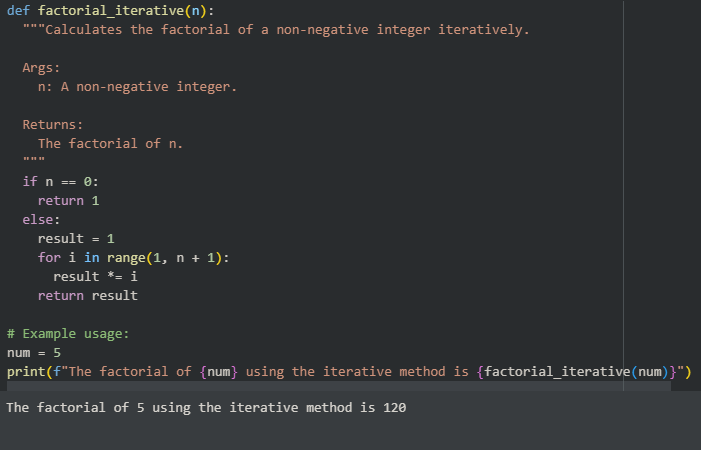
CODE WITH OUTPUT:



PROMPT2:

Write an efficient code for factorial function

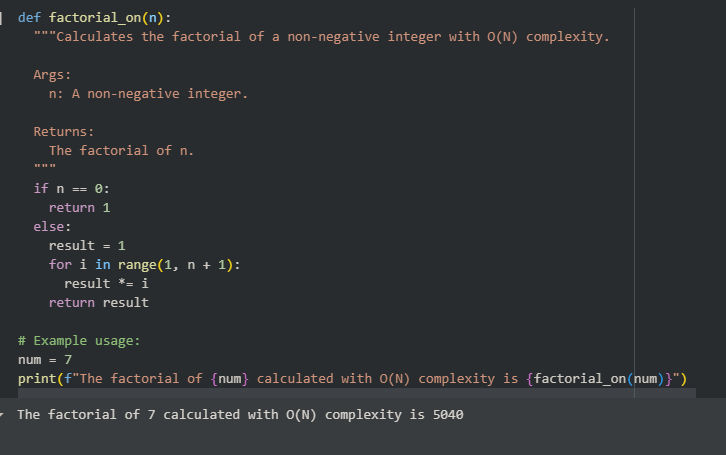
CODE WITH OUTPUT:



PROMPT3:

write a code on factorial in different approach with O(N)

CODE WITH OUTPUT:



EXPECTED OUTPUT:

Comparison of AI-generated code styles

OBSERVATION:

All three valid implementations of the factorial function produced the correct results for the examples we tested. However, the iterative approach is generally preferred in practice for its efficiency and avoidance of potential stack overflow issues with large inputs. The iterative versions are essentially the same in terms of their core logic and O(N) time complexity.

TASK DESCRIPTION 2:

QUESTION:

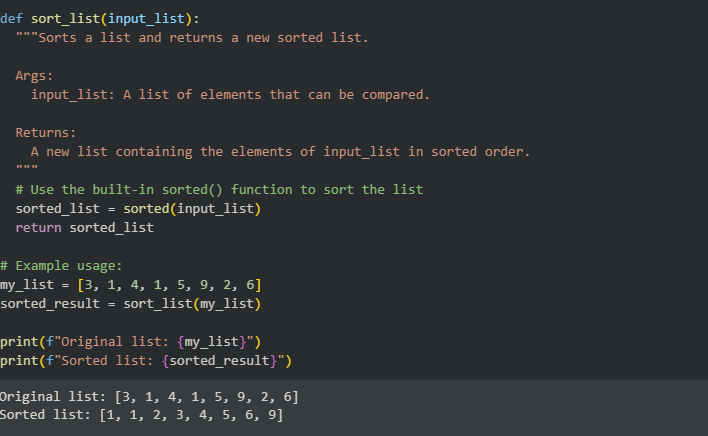
Provide a clear example input-output prompt to generate a sorting function

PROMPT:  
generate a code for sorting in function where input and output is given clearly

EXPECTED OUTPUT:

Functional sorting code from AI

CODE WITH OUTPUT:



OBSERVATION:  
The code defines a function called sort\_list that takes a list of items (like numbers in our example) and sorts them from smallest to largest. It uses a built-in Python tool called sorted() to do this easily. The important thing is that it gives you a *new* list that is sorted, leaving the original list unchanged.

TASK DESCRIPTION 3:

QUESTION:

Start with the vague prompt “Generate python code to calculate power bill” and  
improve it step-by-step

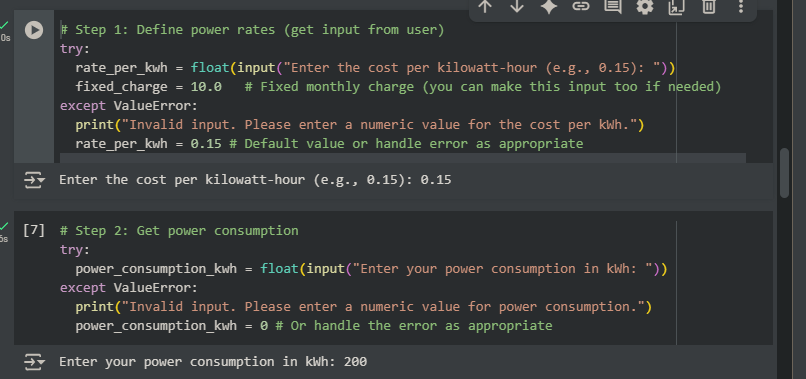
PROMPT:

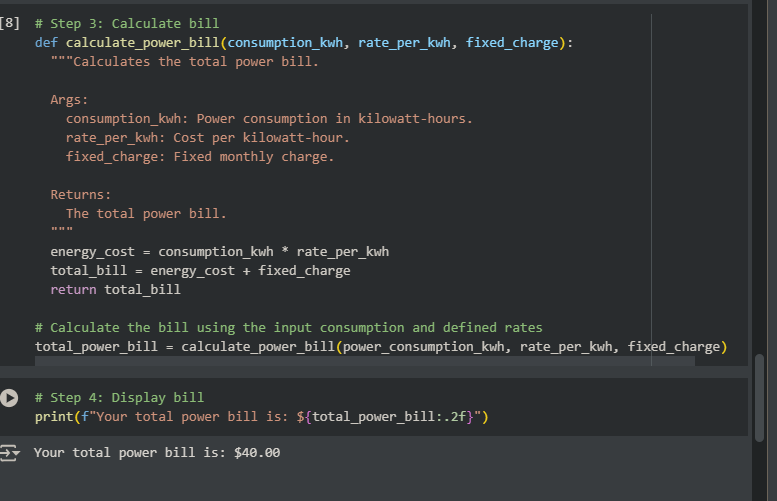
Generate python code to calculate power bill by define power rates, get power consumption, calculate bill, display bill and finish the task.

EXPECTED OUTPUT:

Enhanced AI output with clearer prompts

CODE WITH OUTPUT:





OBSERVATION:

Based on the power consumption I entered (200 kWh) and the example rates we used ($0.15 per kWh plus a $$0.15 per kWh plus a $10.00 fixed charge), the code correctly calculated and showed that the total power bill is $40.00. This means the energy cost was $$40.00. This means the energy cost was $30.00 (200 \* $0.15) plus the $$0.15) plus the $10.00 fixed charge.

TASK DESCRIPTION 4:

QUESTION:

Write structured comments to help AI generate two linked functions (e.g., login\_user()  
and register\_user()).

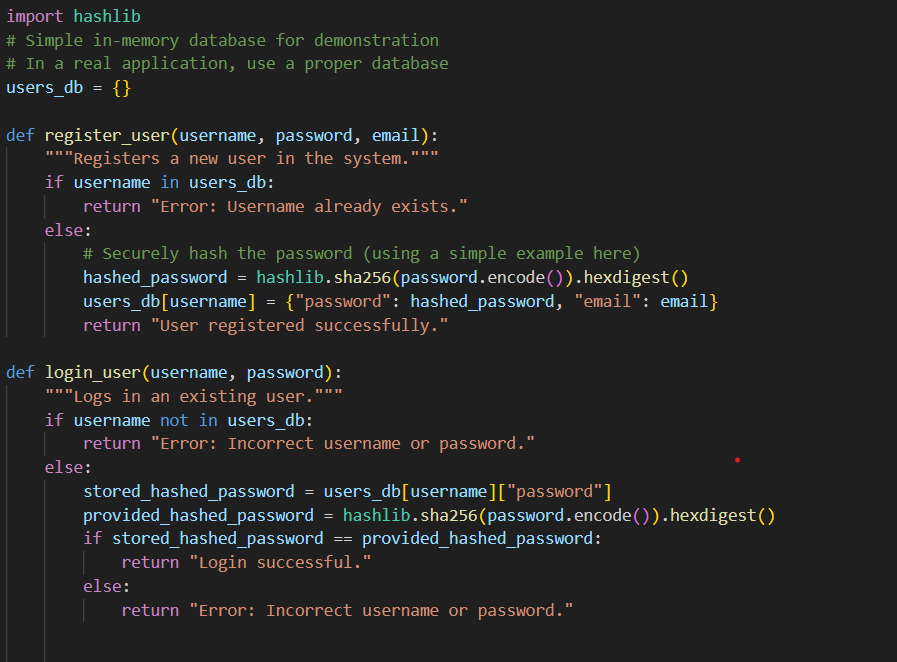
PROMPT:

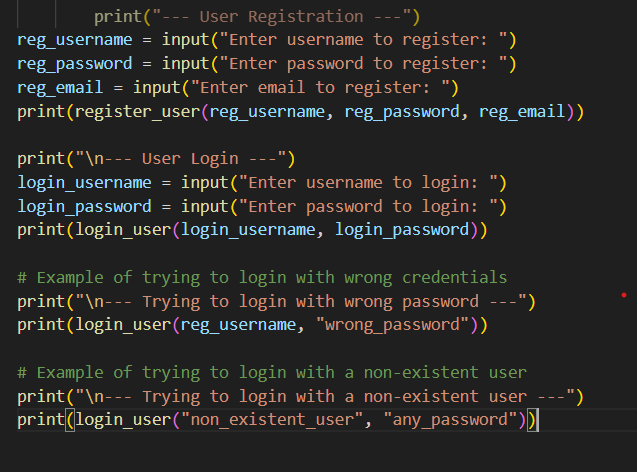
# Function: register\_user # Goal: Register a new user in the system. # Inputs: # - username: The desired username (string). # - password: The user's password (string). # - email: The user's email address (string). # Outputs: # - Success message if registration is successful (string). # - Error message if registration fails (e.g., username already exists) (string). # Steps: # 1. Check if the username already exists in the database. # 2. If username exists, return an error message. # 3. If username does not exist, securely hash the password. # 4. Store the username, hashed password, and email in the database. # 5. Return a success message. # Linked Function: This function is linked to login\_user(). A user needs to register before they can log in.

EXPECTED OUTPUT:

Consistent functions with shared logic

CODE WITH OUTPUT:







OBSERVATION:

**The main observation is that the code correctly distinguishes between valid and invalid login attempts.**

* When you provide the **correct username and password** that you registered, the code outputs a "Login successful" message.
* When you provide an **incorrect password** for a registered user, or try to log in with a **username that doesn't exist**, the code correctly outputs an "Error: Incorrect username or password" message.

TASK DESCRIPTION 5:

QUESTION:

Analyzing Prompt Specificity: Improving Temperature Conversion Function with  
Clear Instructions

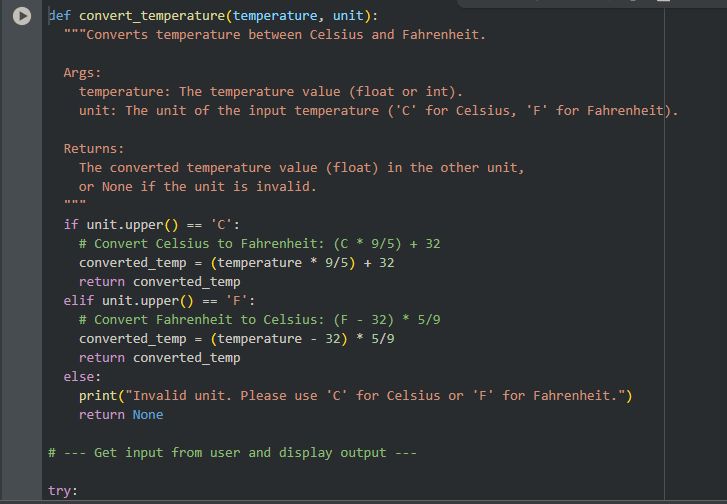
PROMPT1:

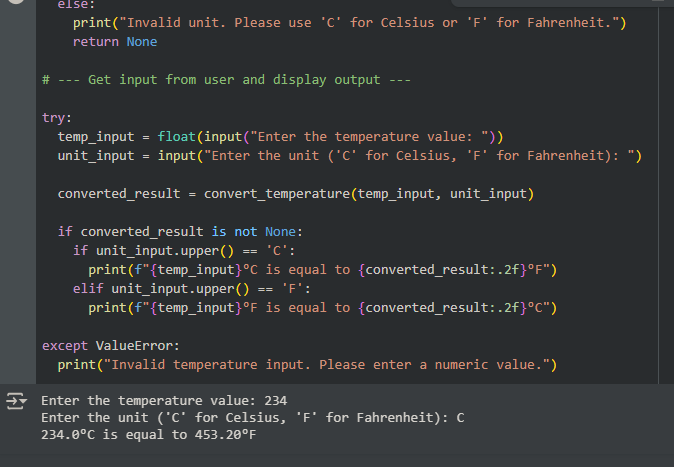
Generate a code Temperature Conversion Function with Clear Instructions with user’s input and output.

EXPECTED OUTPUT:

Code quality difference analysis for various prompts

CODE WITH OUTPUT:

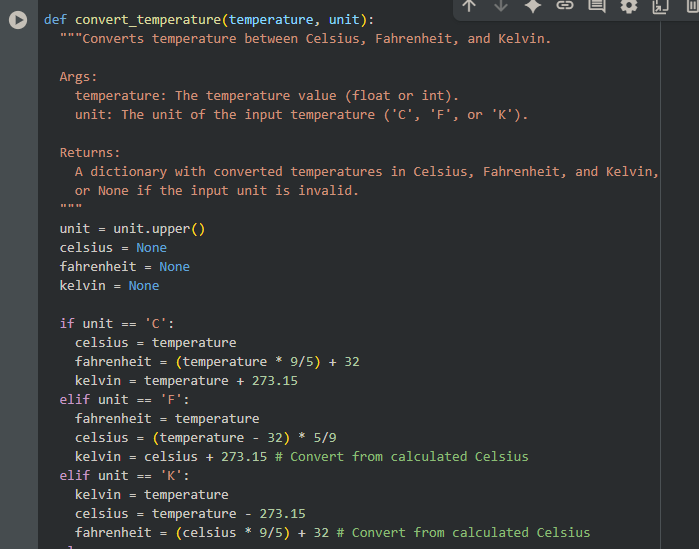


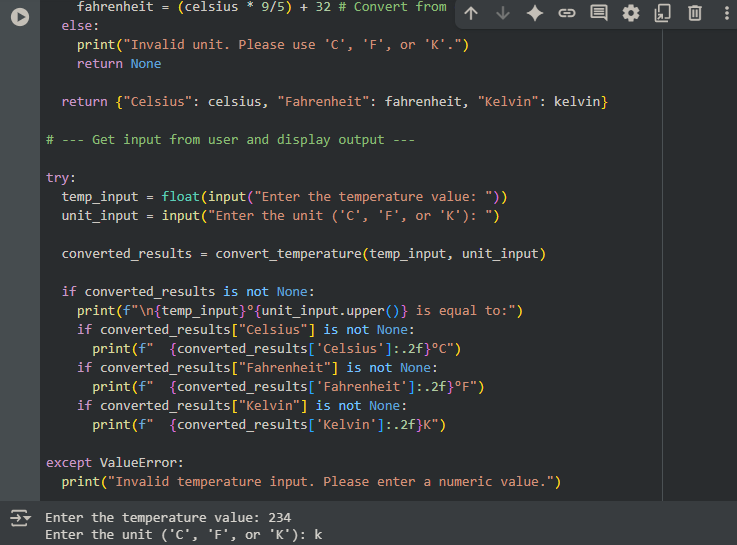


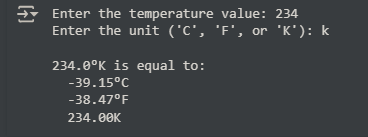
PROMPT 2:

Generate a code Temperature Conversion Function with kelvin,celsiun,fahrenheit.

CODE WITH OUTPUT:







OBSERVATION:

The key difference is the **scope of conversion** and the **output format**. The first version did a single conversion and returned one value, while the second version does multiple conversions and returns a dictionary of results, providing a more complete view of the temperature across different scales. The second version is more versatile as it includes Kelvin, which is important in scientific contexts.