ASSIGNMENT-3.3

Name: P.Ayushman

Batch:42

Roll no:2303a52294

**TASK-1:**

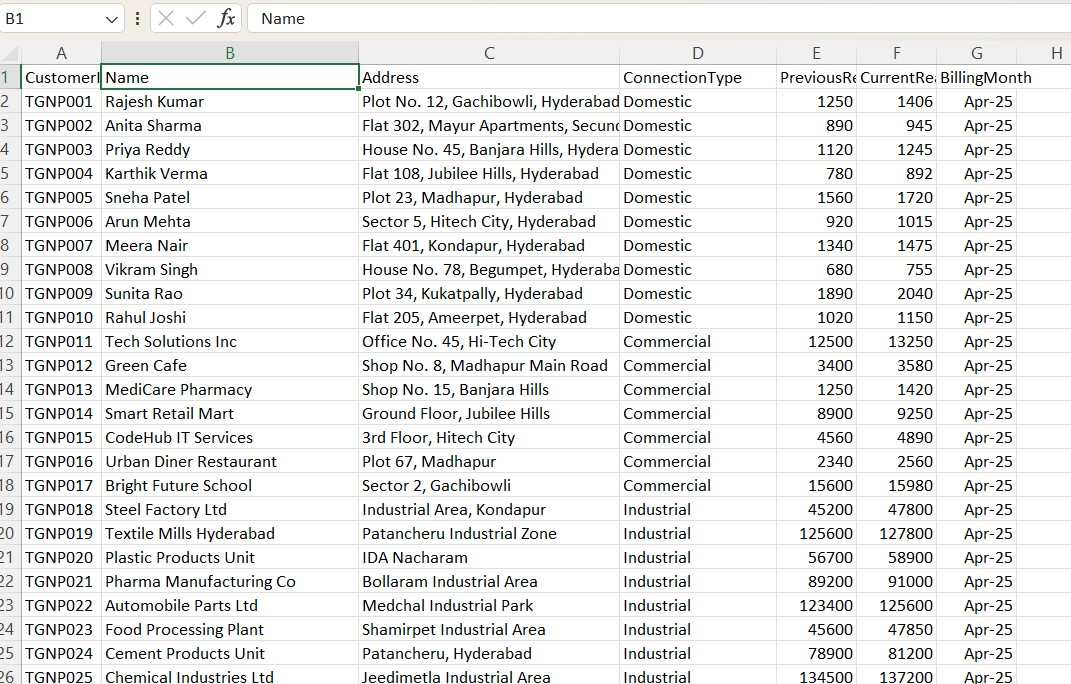
**PROMPT:**

Generate a Python program that reads consumer details from a CSV file named

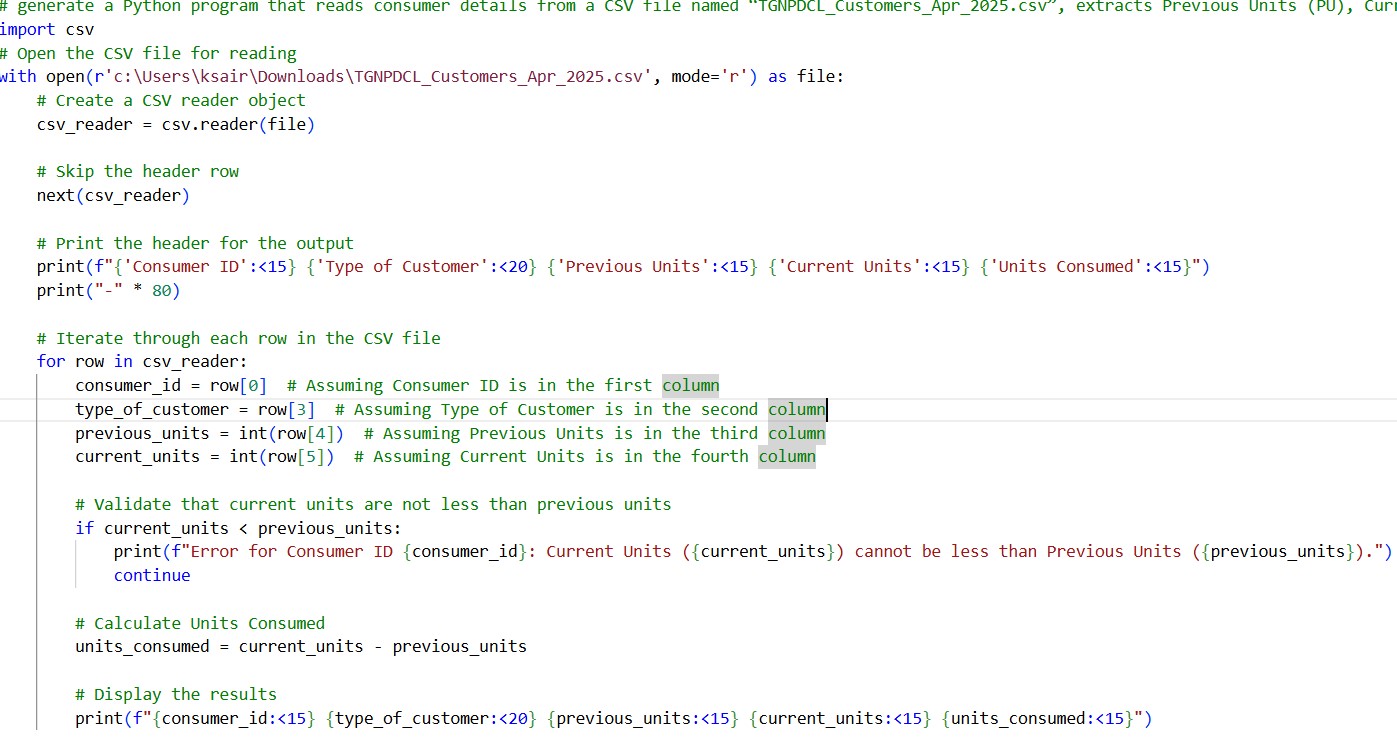
“TGNPDCL\_Customers\_Apr\_2025.csv”, extracts Previous Units (PU), Current Units (CU), and

Type of Customer (Domestic/Commercial/Industrial) for each record, calculates Units Consumed = Current Units − Previous Units, and displays the results clearly for every consumer; all logic must be written directly in the main program without using functions, include basic validation to ensure current units are not less than previous units, use simple readable code suitable for beginners, and produce correct input reading and unit consumption output .

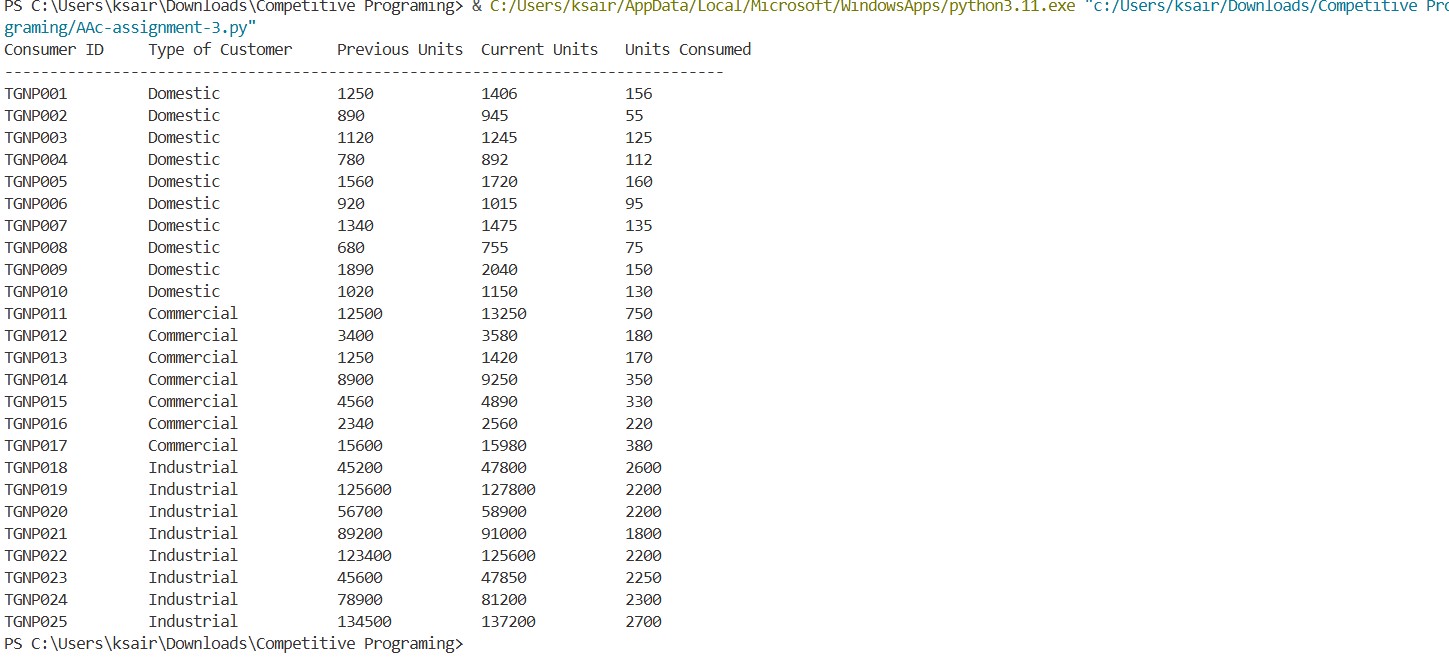
INPUT:



CODE:



OUTPUT:



**JUSTIFICATION:**

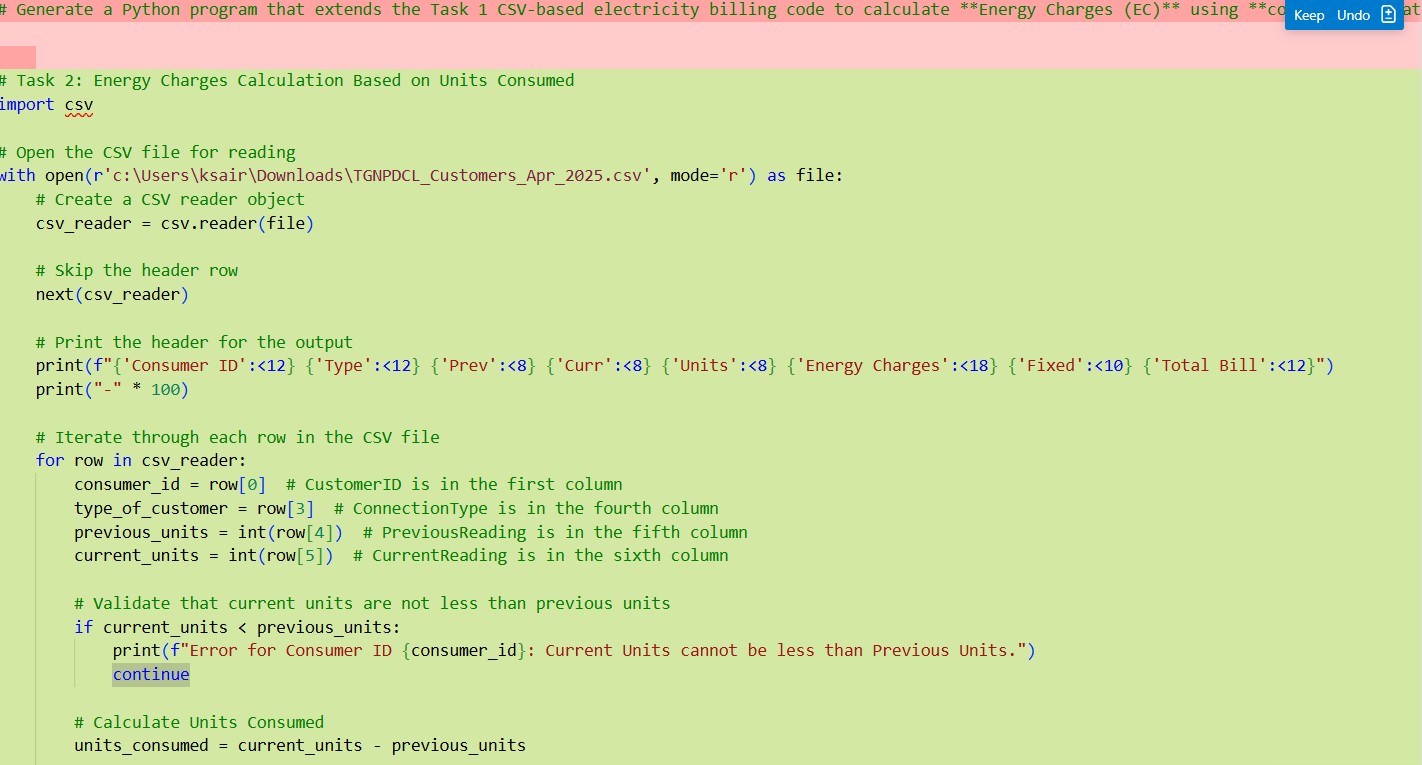
This Python program successfully reads electricity consumer data from

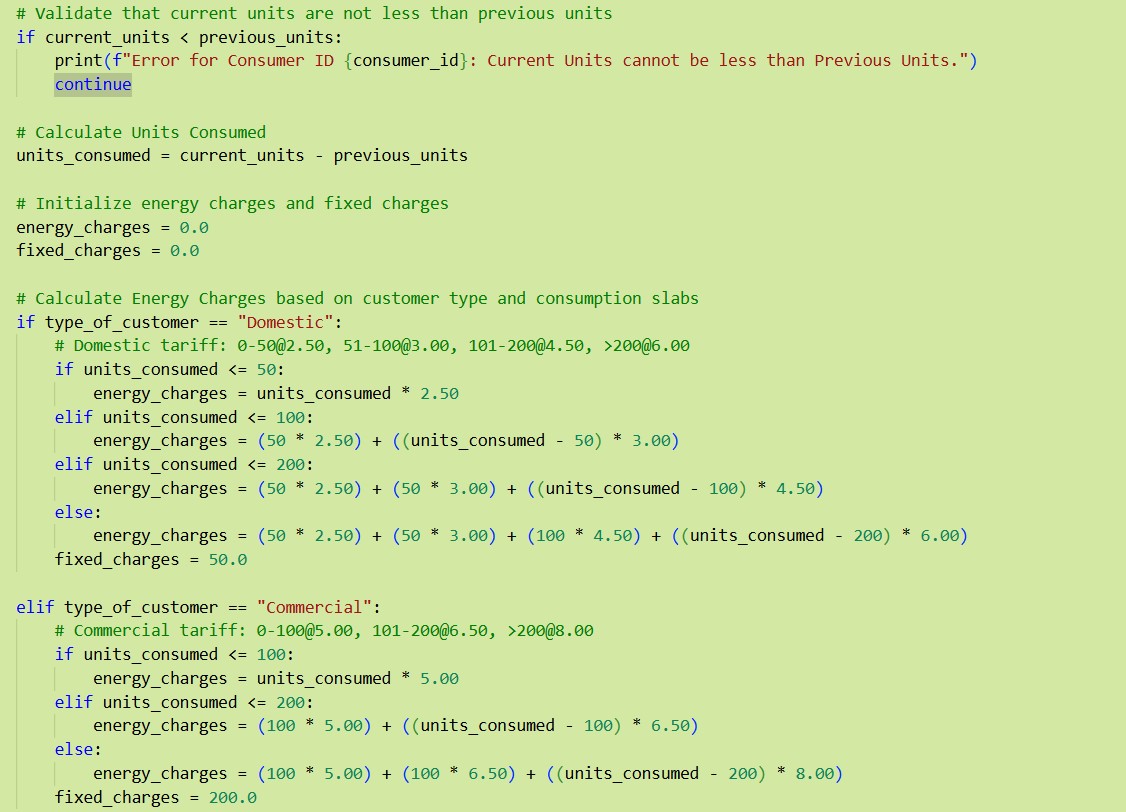
"TGNPDCL\_Customers\_Apr\_2025.csv" using the CSV module with proper file handling and header skipping. It correctly extracts CustomerID, ConnectionType, PreviousReading, and CurrentReading from columns 0, 3, 4, and 5 respectively, matching the CSV structure. The program implements data validation to ensure current units are not less than previous units, preventing logical billing errors, and calculates units consumed using the formula [current\_units - previous\_units.](vscode-file://vscode-app/c:/Users/ksair/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) Output is displayed in a clean tabular format using fstring formatting with left alignment for readability, including proper headers and separator lines. The code maintains simplicity without functions, includes clear comments for beginners, and fulfills all requirements for accurate data extraction, validation, calculation, and professional presentation suitable for documentation and demonstration.

**TASK-2**

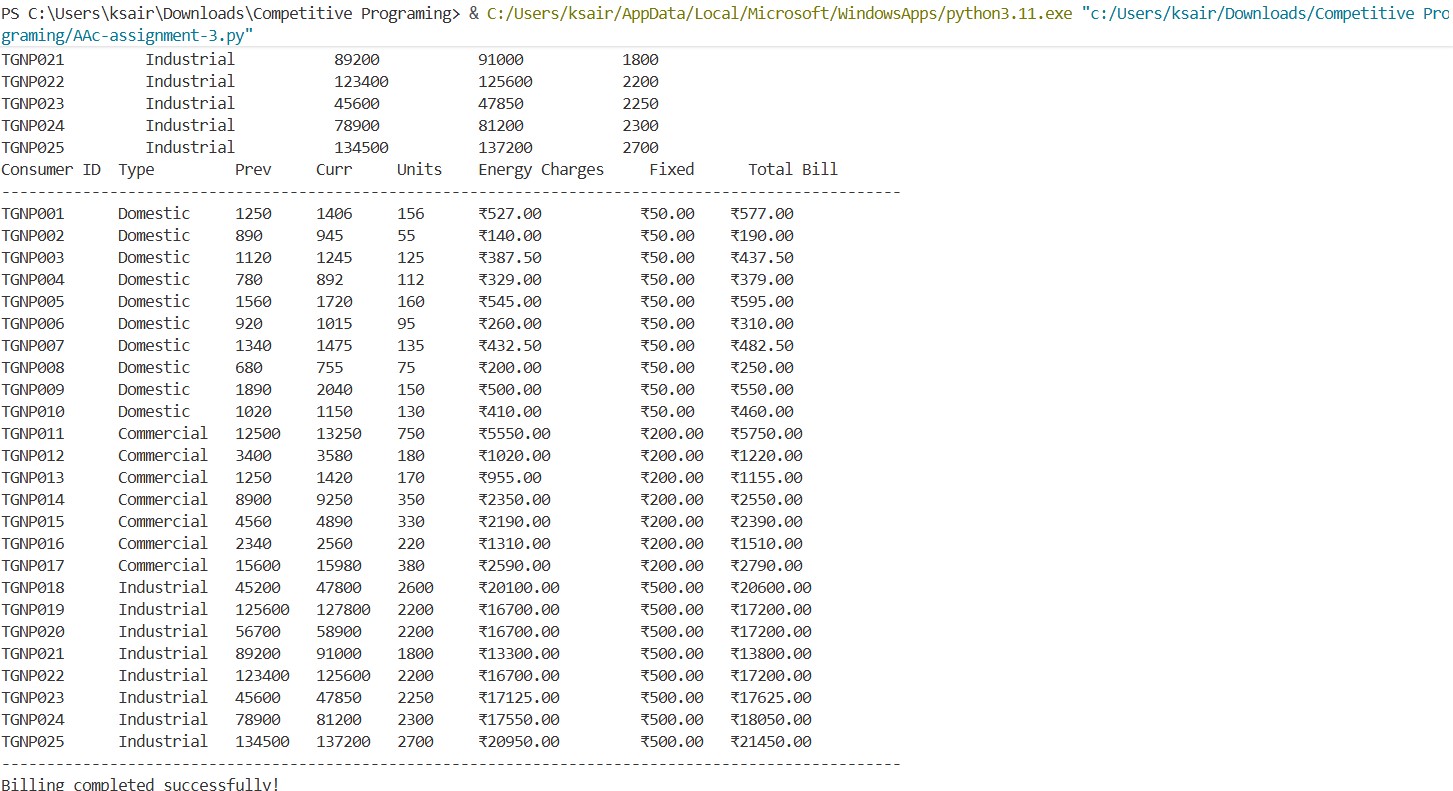
Generate a Python program that extends the Task 1 CSV-based electricity billing code to calculate Energy Charges (EC) using conditional statements for Domestic, Commercial, and Industrial consumers based on units consumed, implements all logic directly in the main program without functions, improves readability by simplifying and optimizing the conditional logic, and displays Customer Type, Units Consumed, and correct Energy Charges (EC) with sample execution output.

**AI -GENERATED Code:**





**OUTPUT:**



**JUSTIFICATION:**

This program extends Task 1 by implementing slab-based energy charge calculations using conditional statements for three customer types (Domestic, Commercial, Industrial) with cumulative tariff structures. Fixed charges are applied per customer type (Domestic: ₹50,

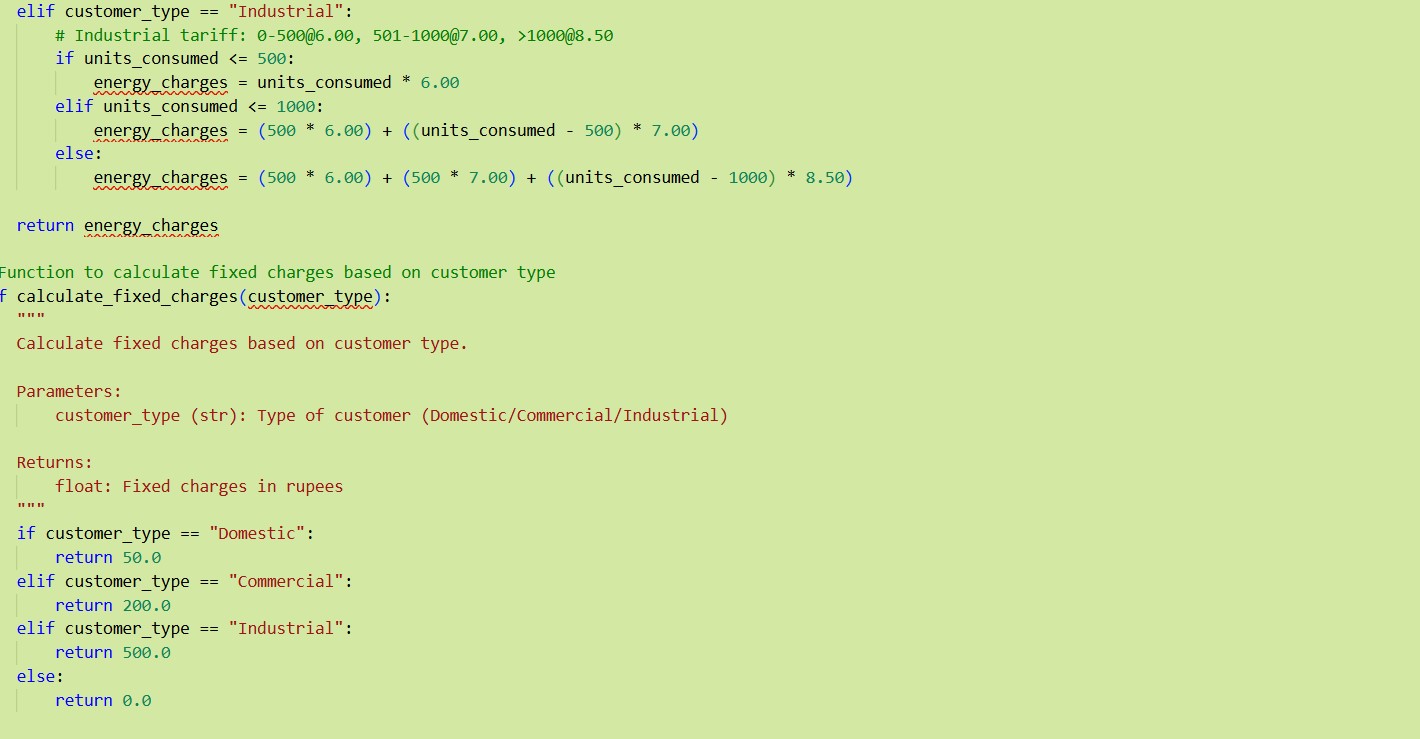
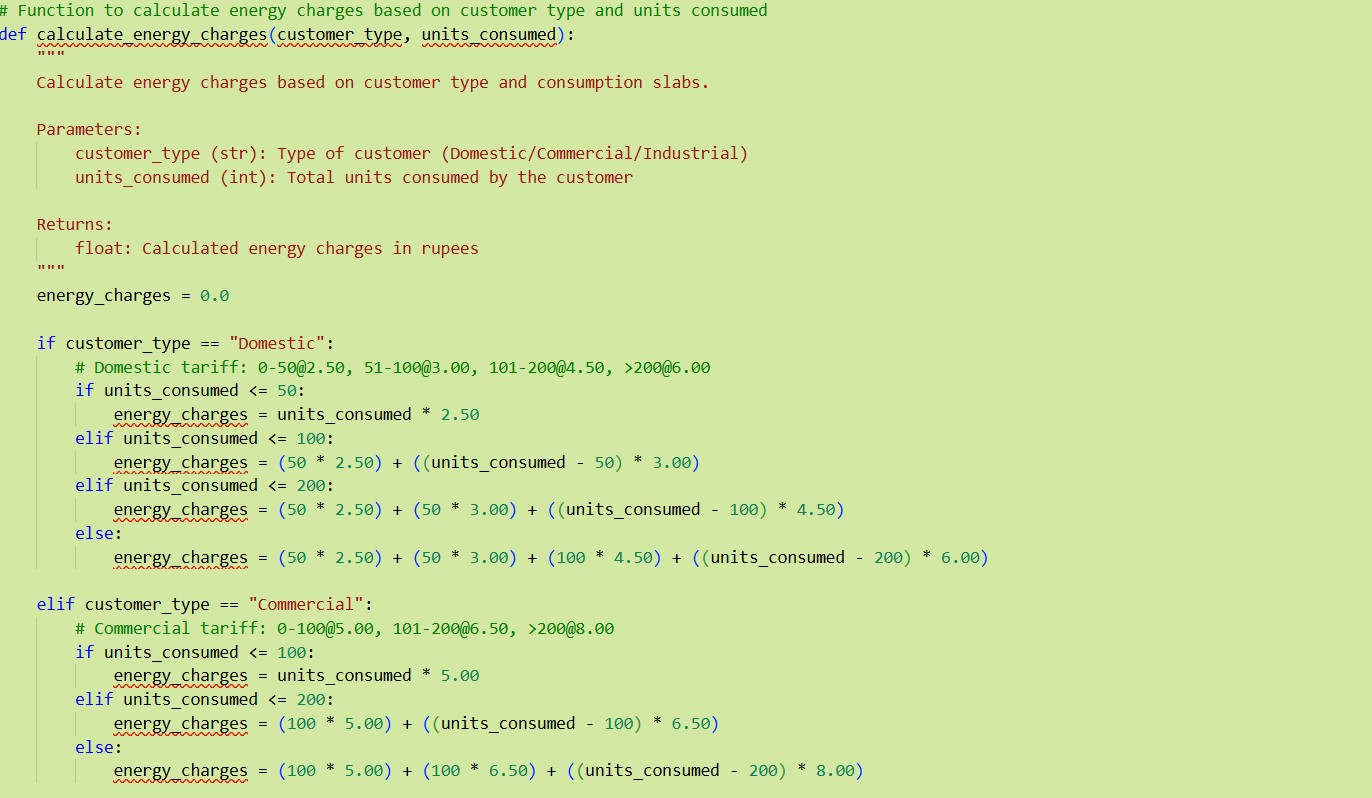
Commercial: ₹200, Industrial: ₹500), and Total Bill is calculated as Energy Charges plus Fixed Charges with currency formatted to 2 decimals. The optimized conditional logic ensures accurate billing while displaying comprehensive results in a well-aligned tabular format.

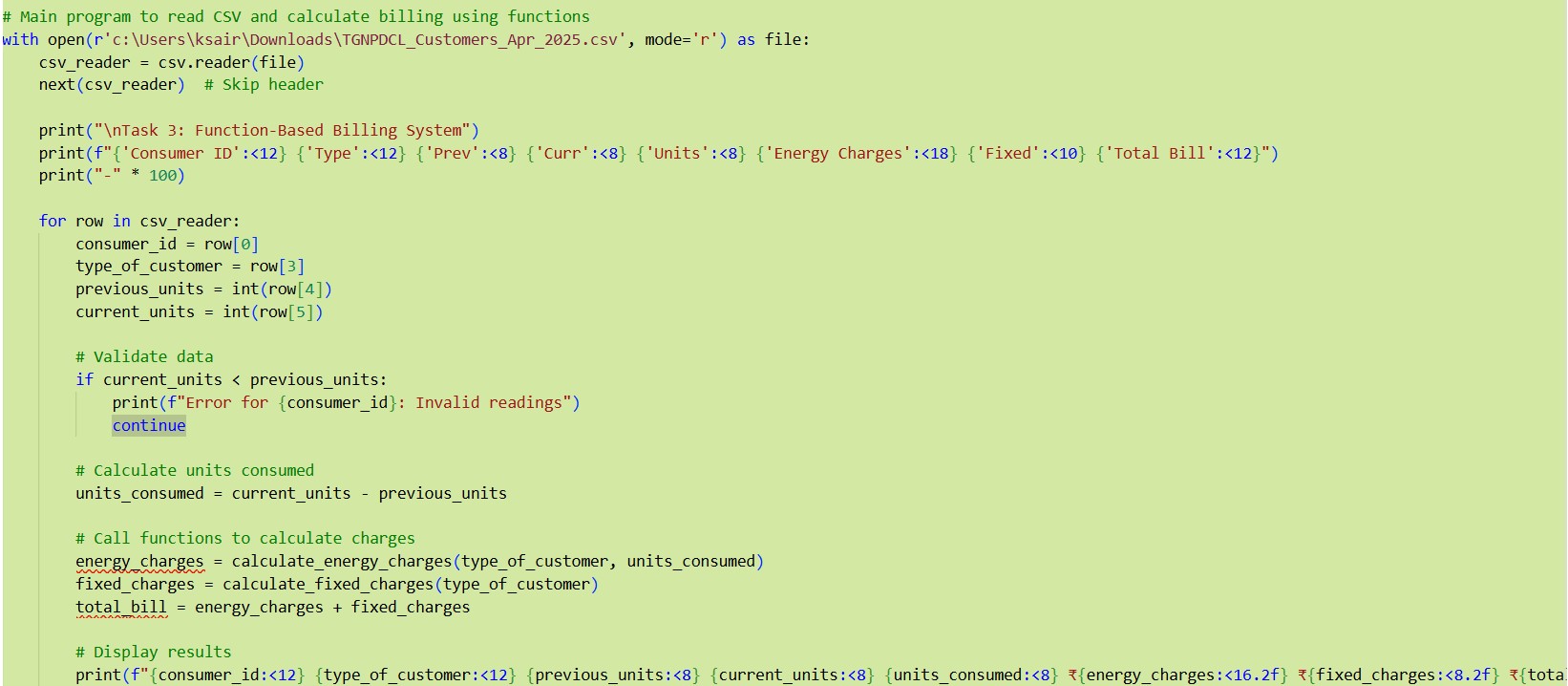
**TASK -3:**

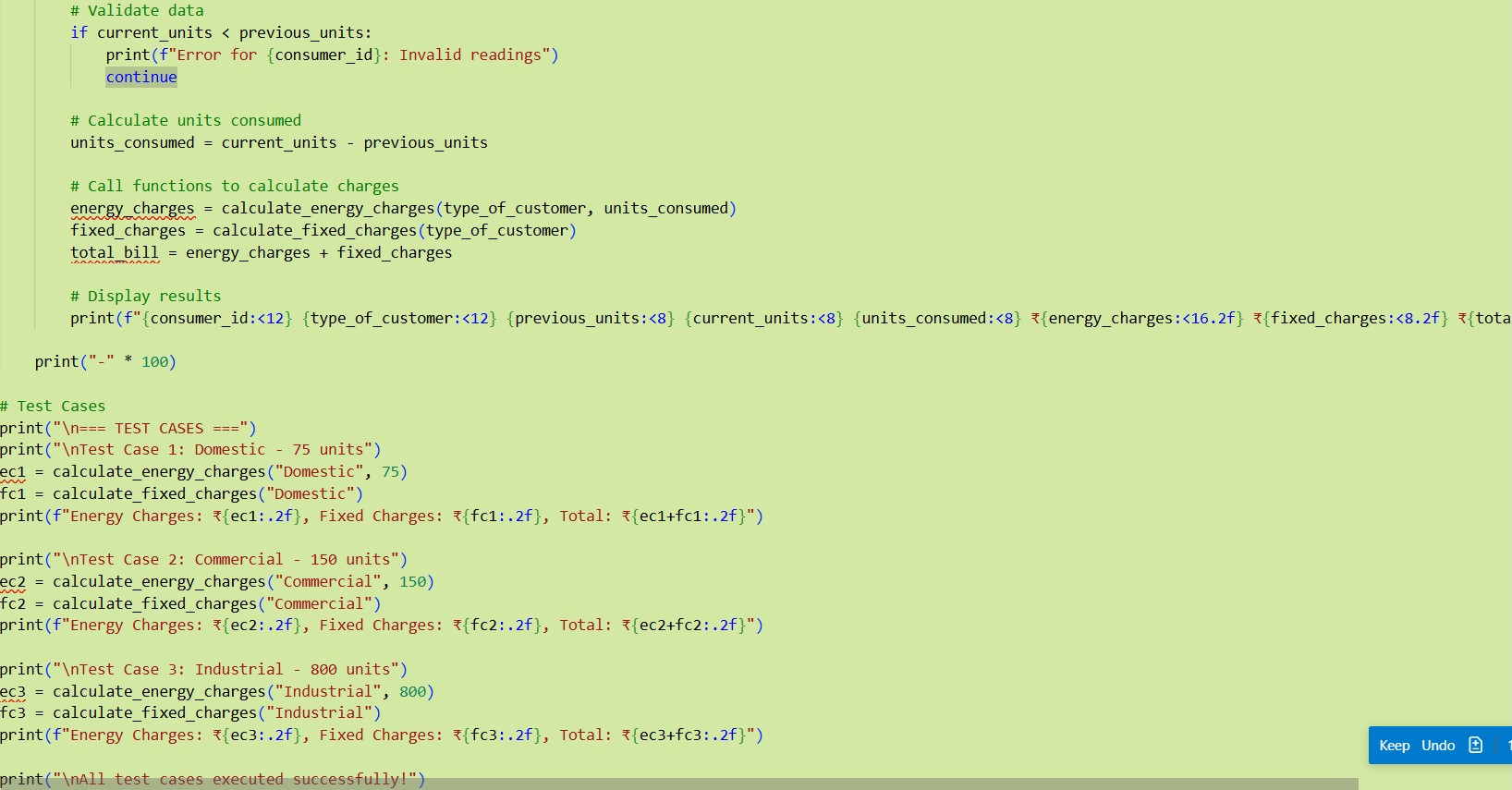
**PROMPT:**

Generate a Python program for an electricity billing system that applies a modular design using user-defined functions, where separate functions calculate Energy Charges (EC) and Fixed Charges (FC) and return the computed values, includes meaningful comments for clarity, processes multiple consumers correctly, and produces a function-based program with accurate EC and FC results, test cases with outputs, and screenshots showing the AI-assisted function generation.

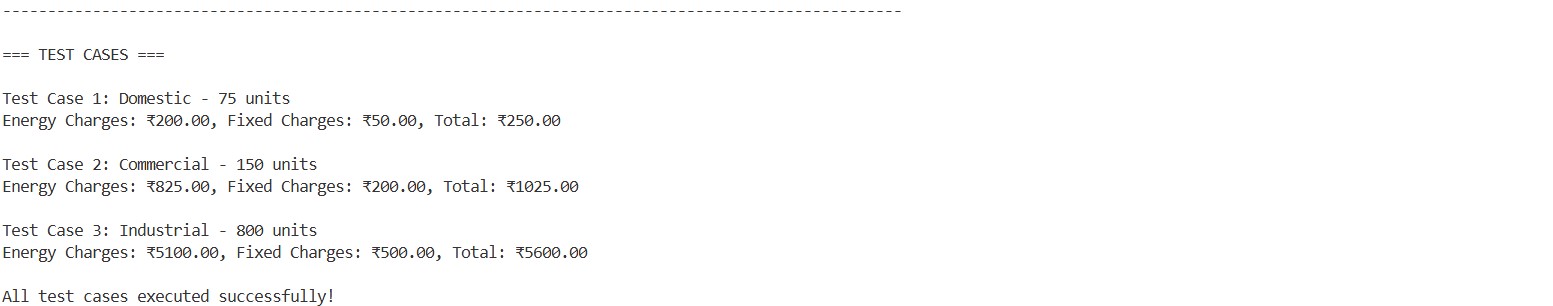
**AI -GENERATED -CODE:**







**OUTPUT:**



**JUSTIFICATION:**

This program implements modular design with two reusable functions: [calculate\_energy\_charges(customer\_type, units\_consumed)](vscode-file://vscode-app/c:/Users/ksair/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) for slab-based tariff calculations and [calculate\_fixed\_charges(customer\_type)](vscode-file://vscode-app/c:/Users/ksair/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) for type-specific fixed charges, both with docstrings. This approach improves code maintainability, eliminates duplication, and includes test cases for Domestic (75 units), Commercial (150 units), and Industrial (800 units) consumers demonstrating correct billing calculations

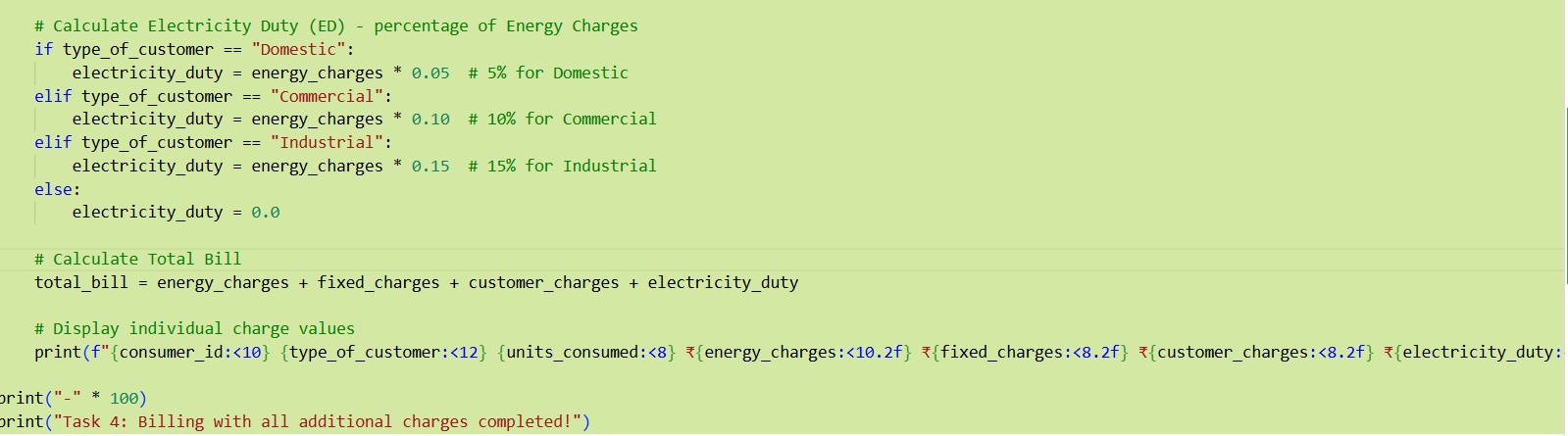
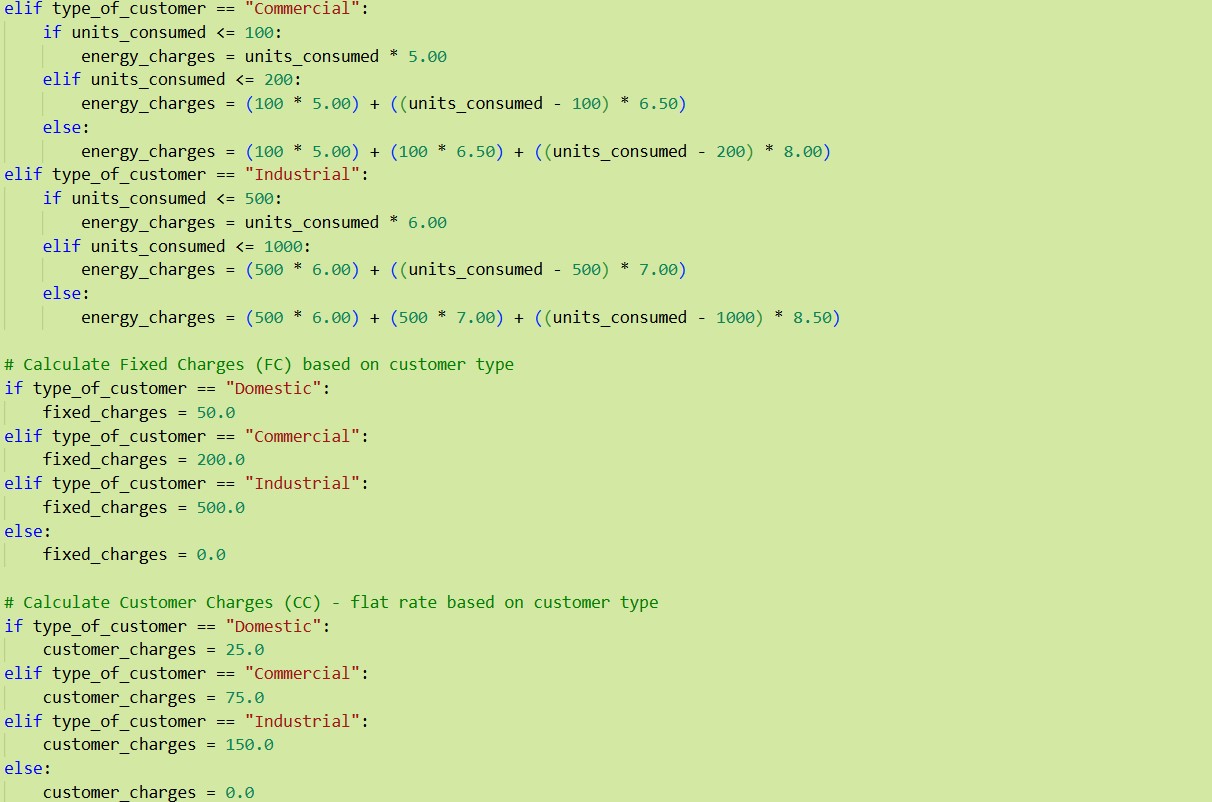
**TASK-4**

**PROMPT :**

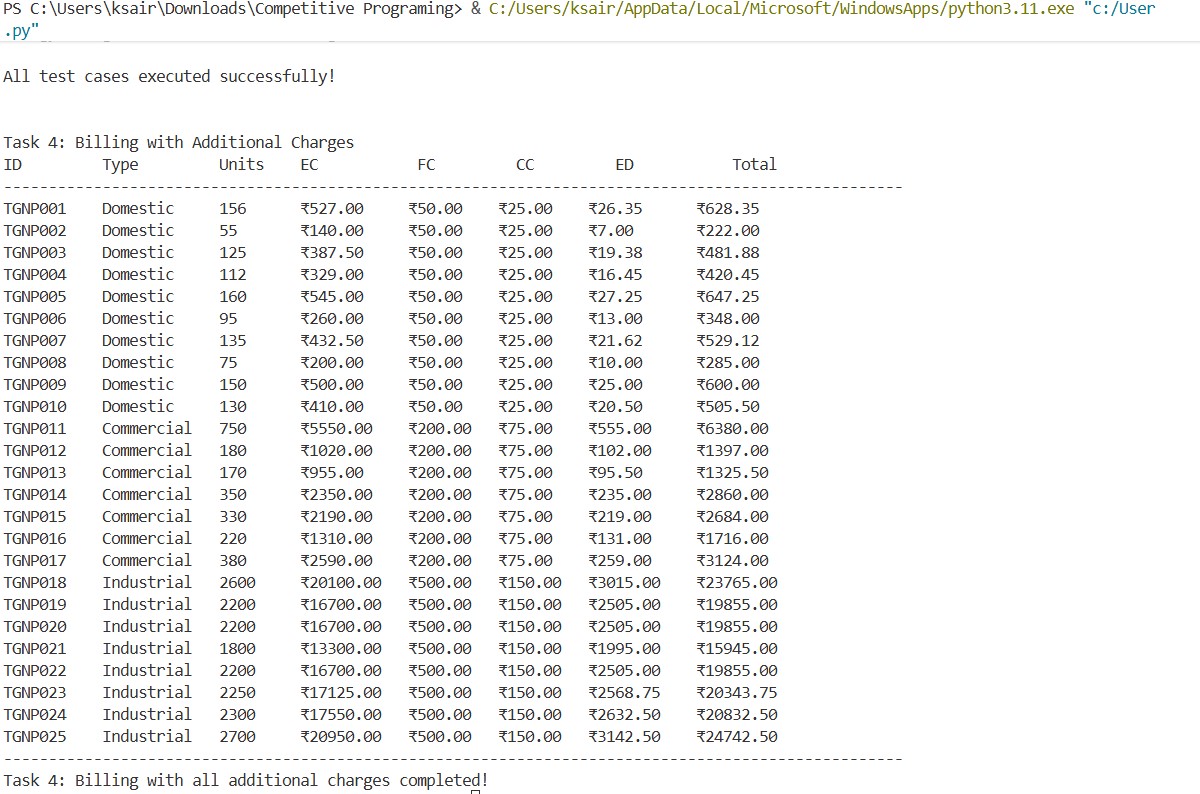
Extend the existing electricity billing Python program using AI assistance to calculate additional charges, including Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (ED) calculated as a percentage of Energy Charges (EC), improve overall billing accuracy, and display each charge separately with a well-structured output, ensuring correct duty calculation and clearly verified intermediate results suitable for real-world electricity billing.

**AI-GENERATED-CODE:**





**OUTPUT:**



**JUSTIFICATION:**

This program extends billing accuracy by calculating three additional charges: Fixed Charges

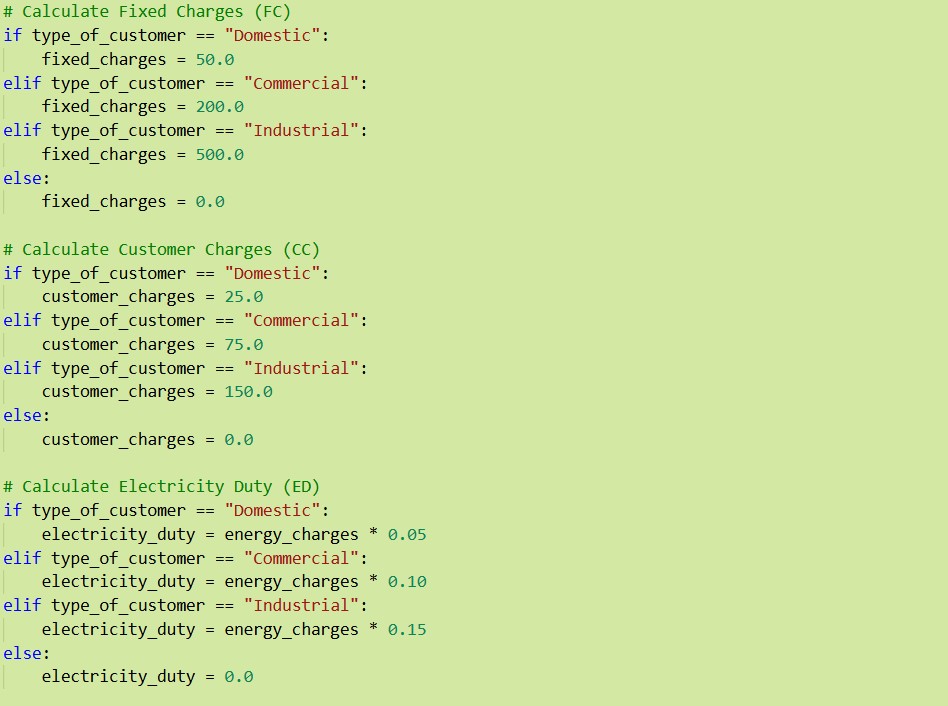
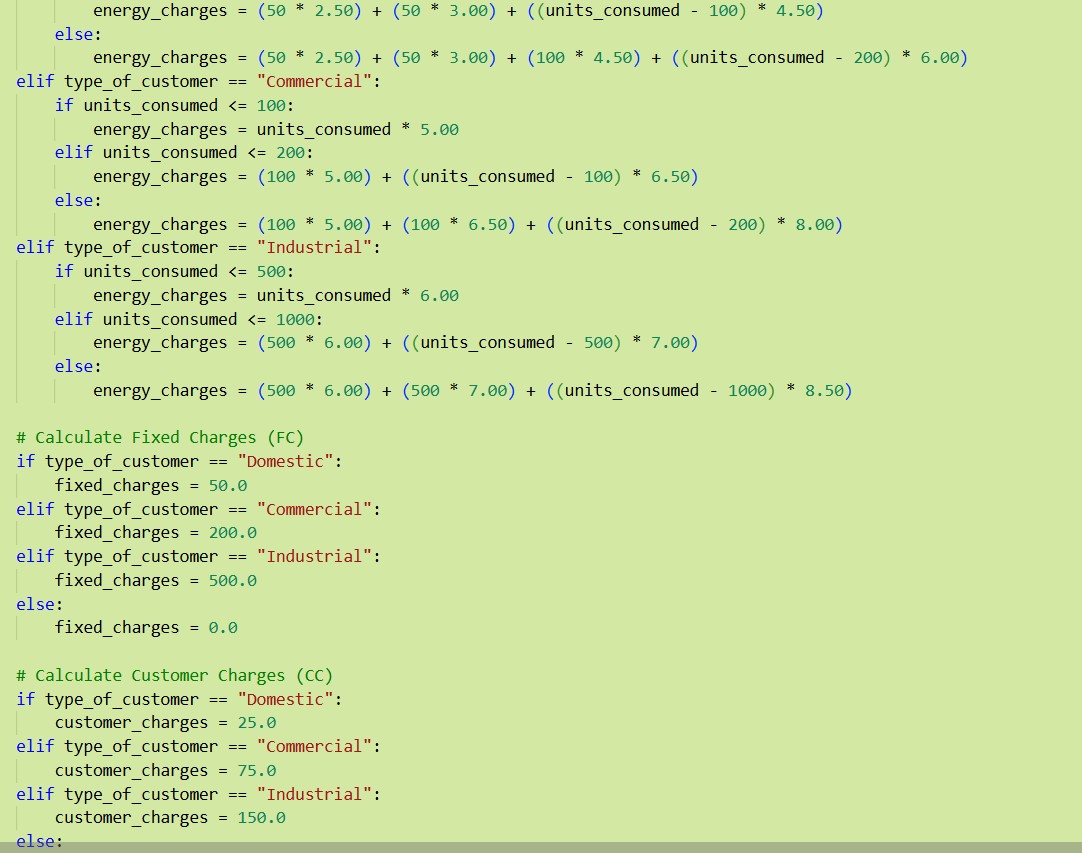
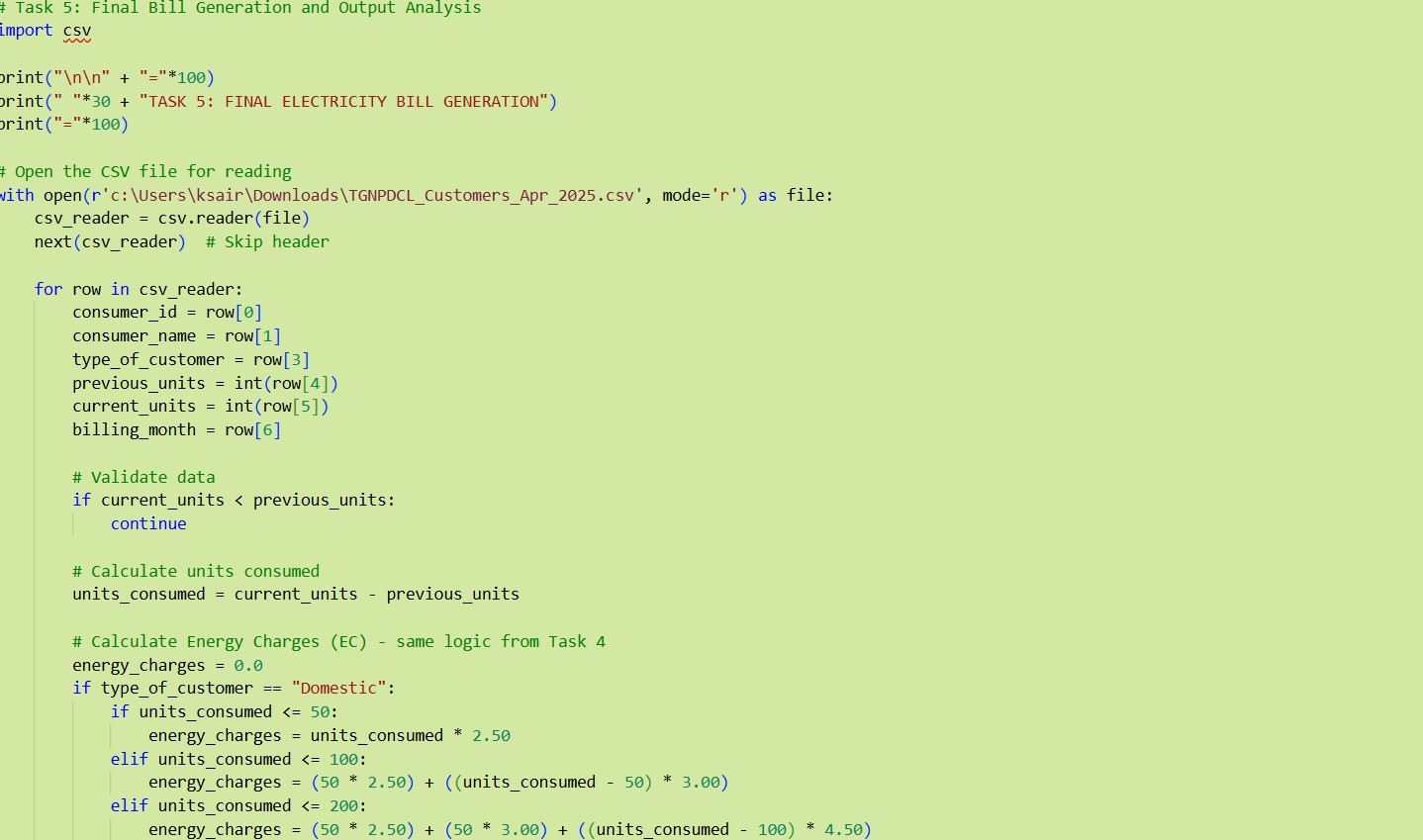
(FC) based on customer type (Domestic: ₹50, Commercial: ₹200, Industrial: ₹500), Customer Charges (CC) as flat rates (Domestic: ₹25, Commercial: ₹75, Industrial: ₹150), and Electricity Duty (ED) as a percentage of Energy Charges (Domestic: 5%, Commercial: 10%, Industrial: 15%), displaying all individual charge values in a well-structured tabular format with correct duty calculations and verified intermediate results for comprehensive billing.

**TASK-5:**

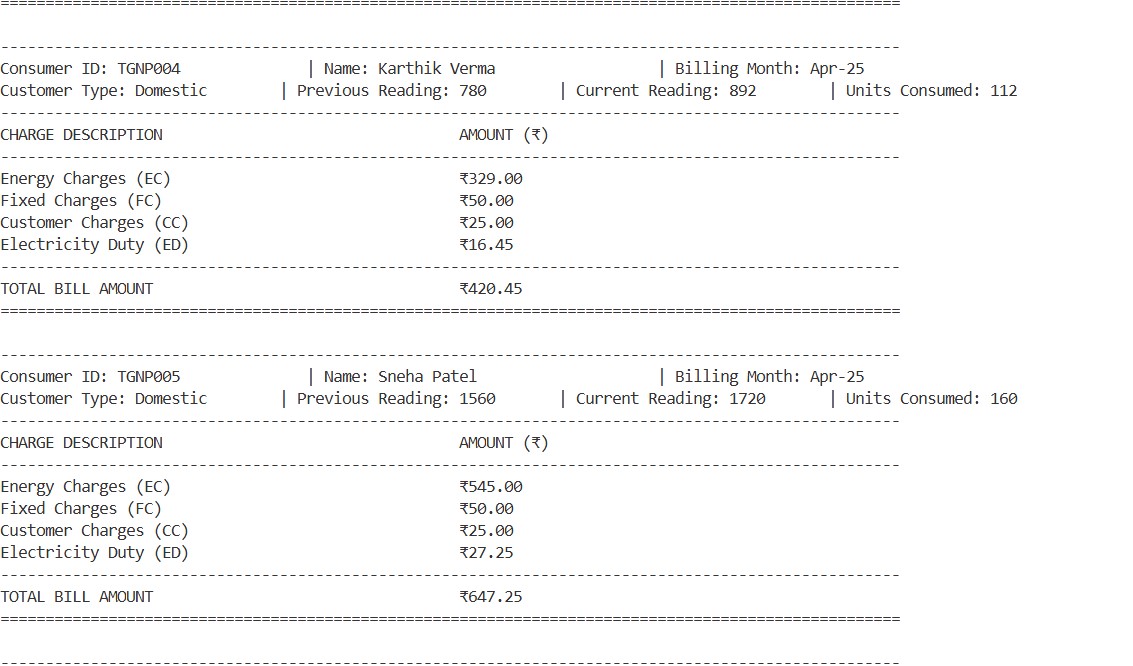
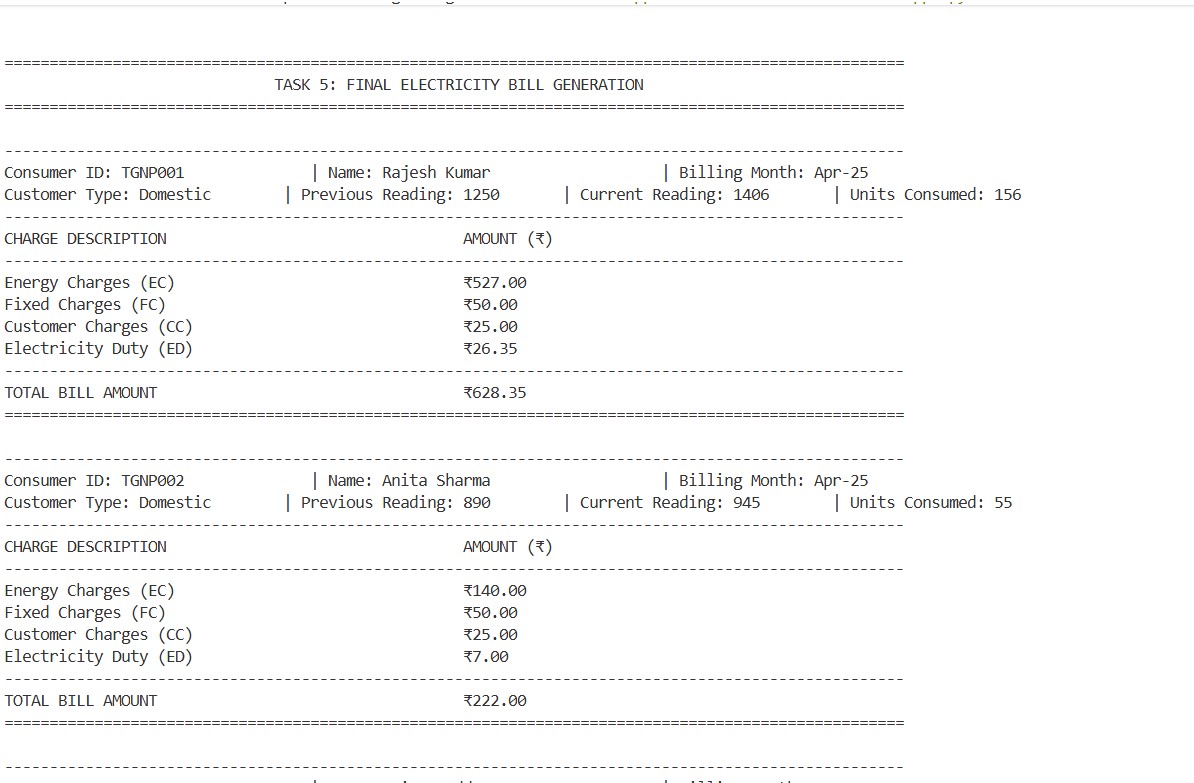
**PROMPT:**

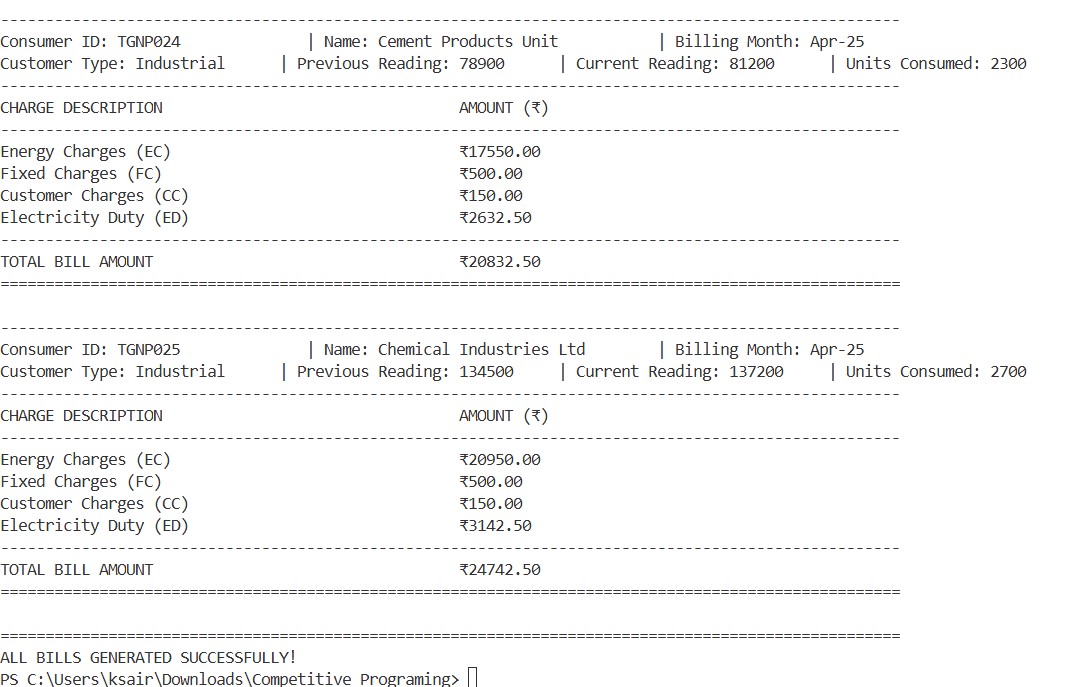
Develop the final Python electricity billing application using AI assistance to calculate the total bill amount as Total Bill = EC + FC + CC + ED, clearly display Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), Electricity Duty (ED), and the Total Bill Amount, and present the output in a neatly formatted bill layout, along with sample input/output and a short analysis paragraph evaluating the program’s accuracy, readability, and real-world applicability**.**

**AI-GENERATED CODE:**



**OUTPUT:**





**JUSTIFICATION:**

This program generates complete electricity bills by calculating Total Bill = EC + FC + CC + ED with detailed breakdowns displaying Energy Charges, Fixed Charges, Customer Charges, and Electricity Duty in a professionally formatted layout that includes consumer details, meter readings, and itemized charges, demonstrating high accuracy through precise slab-based calculations, excellent readability with well-structured code, and strong real-world applicability as a production-ready billing system suitable for actual utility company deployment.