AI ASSISTED CODING

Lab Assignment – 3.3

Name : Nadhiya Seelam

Roll No : 2403A510C6

Batch No : 05

Department : CSE

Task-01:

- Try 3 different prompts to generate a factorial function.

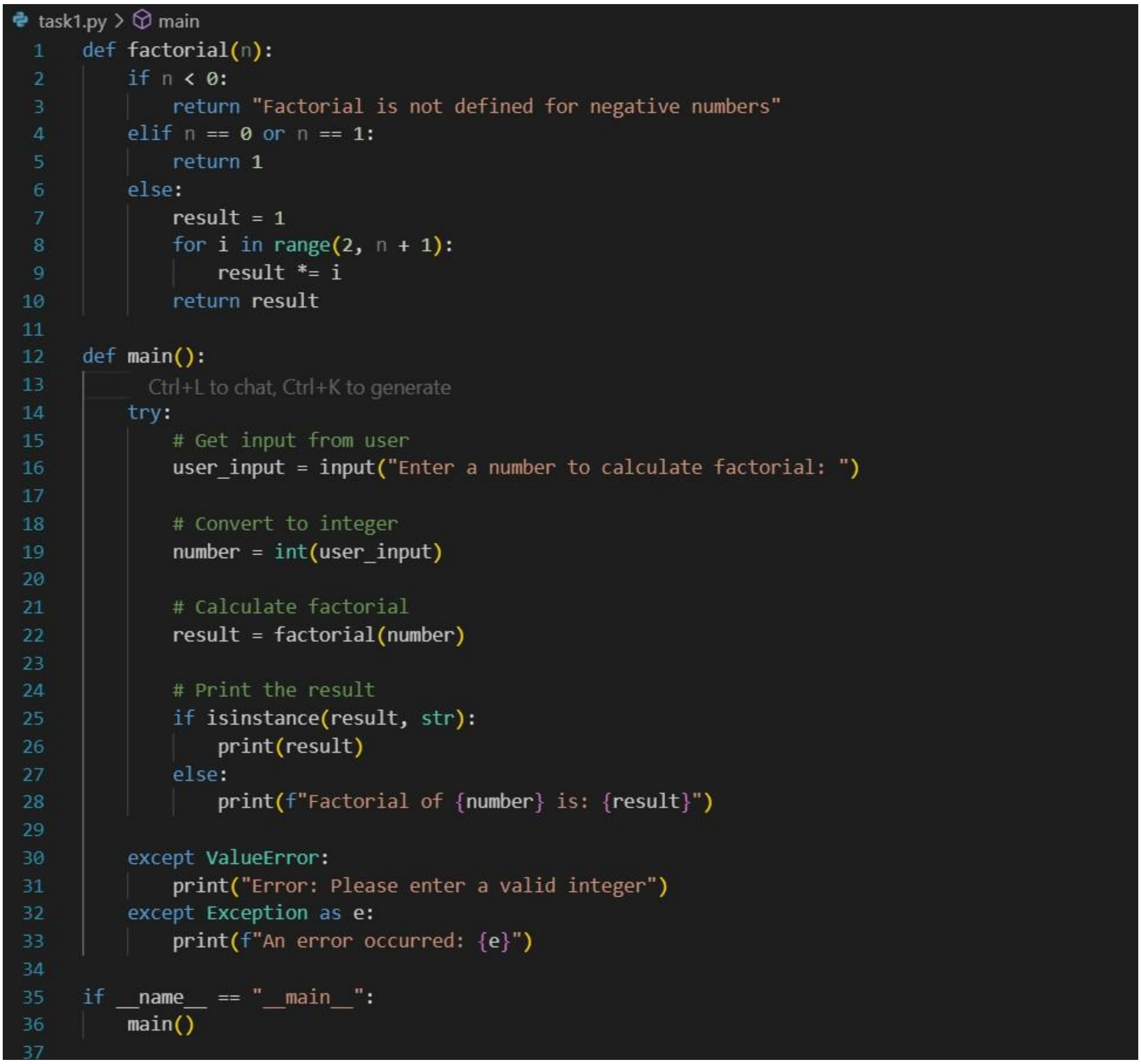
Expected Output#1

- Comparison of AI-generated code styles

Prompt-1:

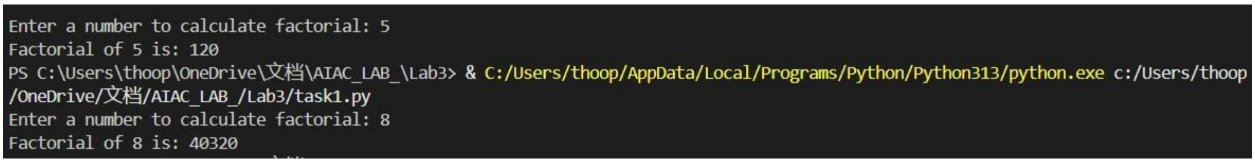
Generate a function that prints factorial of a given number

Code:



```python
def factorial(n):
    if n < 0:
        return "Factorial is not defined for negative numbers"
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result

def main():

    try:
        # Get input from user
        user_input = input("Enter a number to calculate factorial: ")

        # Convert to integer
        number = int(user_input)

        # Calculate factorial
        result = factorial(number)

        # Print the result
        if isinstance(result, str):
            print(result)
        else:
            print(f"Factorial of {number} is: {result}")

    except ValueError:
        print("Error: Please enter a valid integer")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    main()

```

Output:

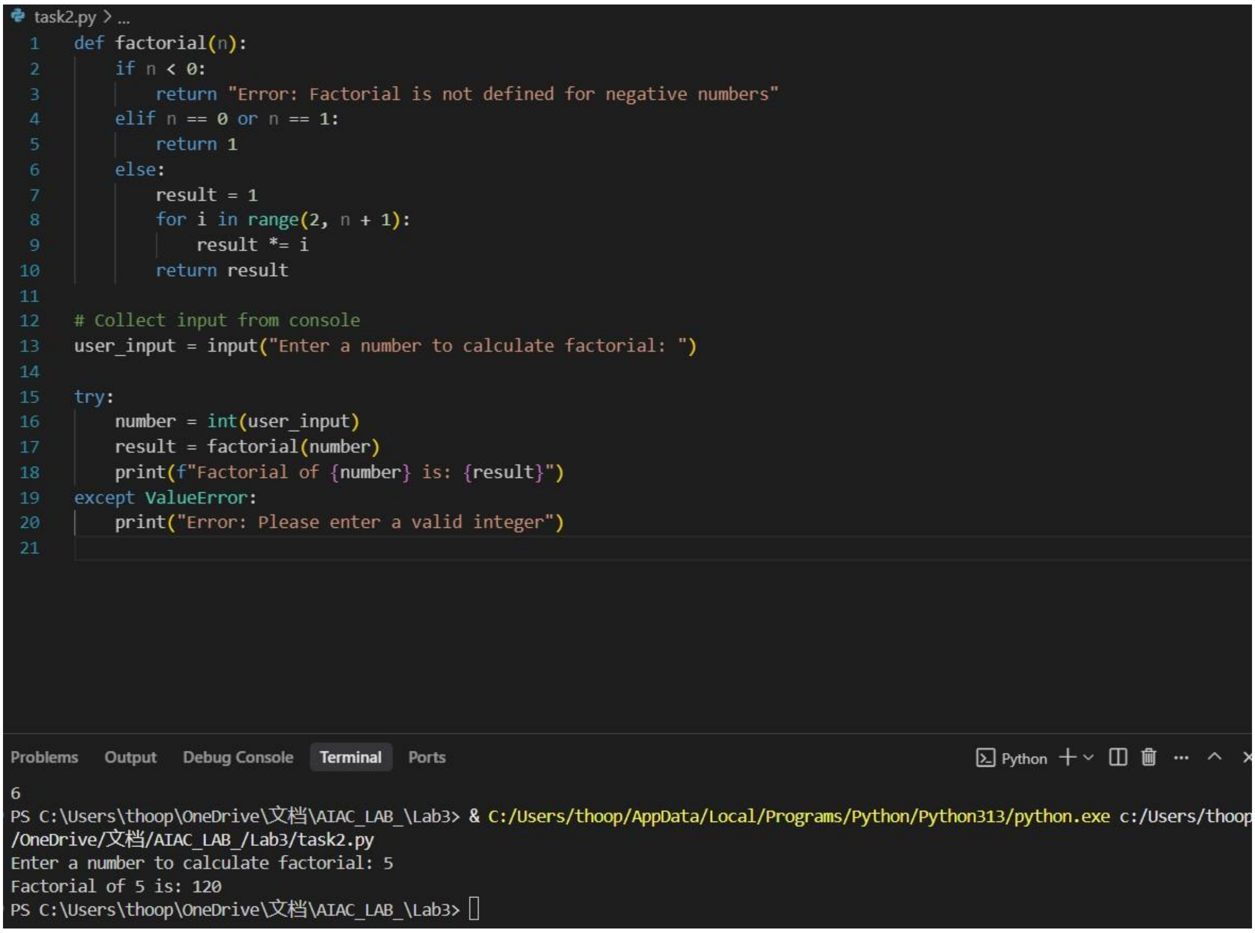


```
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop
/OneDrive/文档/AIAC_LAB_/Lab3/task1.py
Enter a number to calculate factorial: 8
Factorial of 8 is: 40320
```

Prompt 2:

Write a simple python function to find factorial of a number ,collect input from the console

Code:

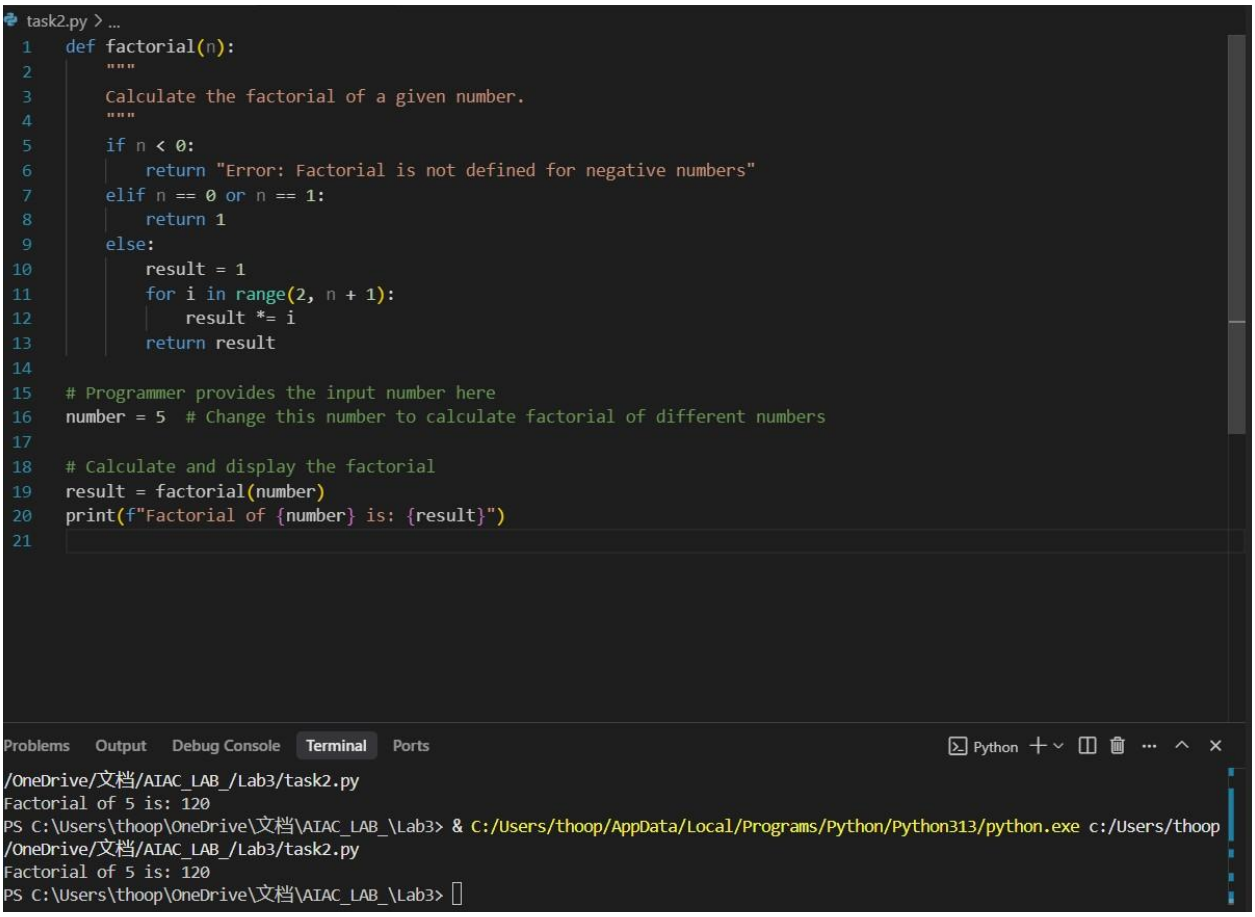


```python
def factorial(n):
    if n < 0:
        return "Error: Factorial is not defined for negative numbers"
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result

# Collect input from console
user_input = input("Enter a number to calculate factorial: ")

try:
    number = int(user_input)
    result = factorial(number)
    print(f"Factorial of {number} is: {result}")
except ValueError:
    print("Error: Please enter a valid integer")

```

```
6
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop
/OneDrive/文档/AIAC_LAB_/Lab3/task2.py
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> 
```

Prompt-3:

Create a simple python function that gives factorial of a number,use input from the programmer



```python
def factorial(n):
    """
    Calculate the factorial of a given number.
    """
    if n < 0:
        return "Error: Factorial is not defined for negative numbers"
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result

# Programmer provides the input number here
number = 5  # Change this number to calculate factorial of different numbers

# Calculate and display the factorial
result = factorial(number)
print(f"Factorial of {number} is: {result}")
```

```
/OneDrive/文档/AIAC_LAB_/Lab3/task2.py
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop
/OneDrive/文档/AIAC_LAB_/Lab3/task2.py
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3>
```

Task-02:

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

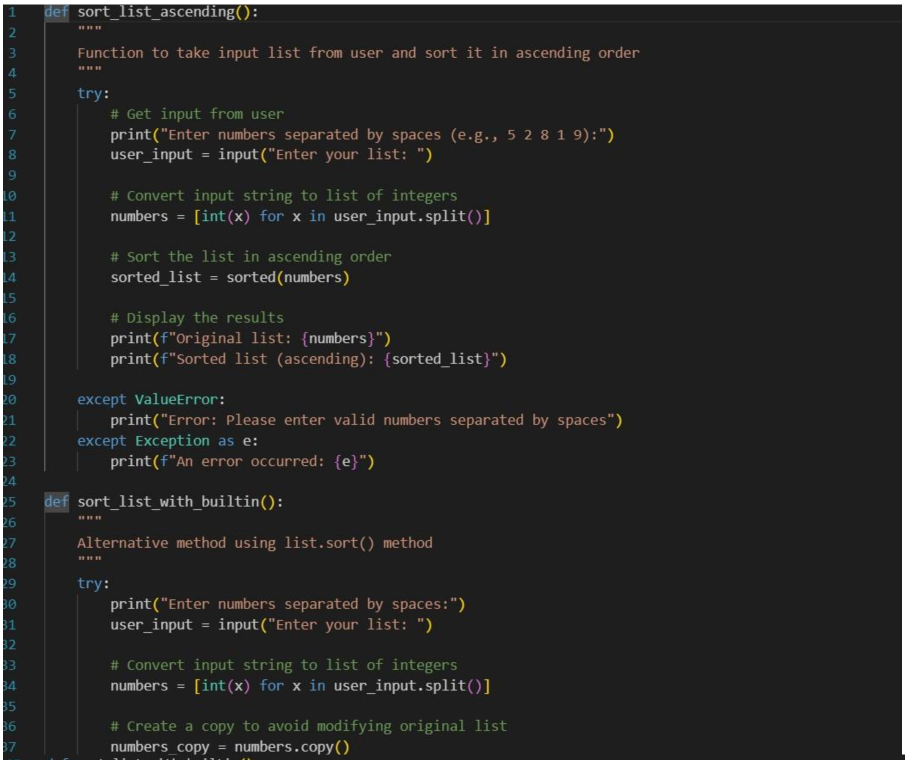
Expected Output#2

- Functional sorting code from AI

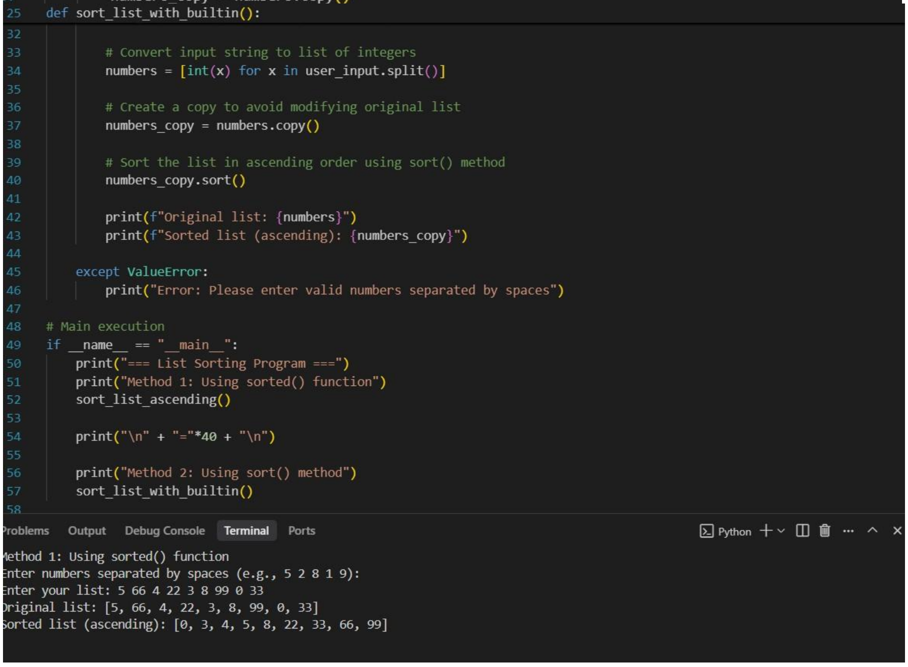
Prompt:

Generate a python code that takes a input list from the user and sort the list Ascending order

Code along with output:



```python
def sort_list_ascending():
    """
    Function to take input list from user and sort it in ascending order
    """
    try:
        # Get input from user
        print("Enter numbers separated by spaces (e.g., 5 2 8 1 9):")
        user_input = input("Enter your list: ")

        # Convert input string to list of integers
        numbers = [int(x) for x in user_input.split()]

        # Sort the list in ascending order
        sorted_list = sorted(numbers)

        # Display the results
        print(f"Original list: {numbers}")
        print(f"Sorted list (ascending): {sorted_list}")

    except ValueError:
        print("Error: Please enter valid numbers separated by spaces")
    except Exception as e:
        print(f"An error occurred: {e}")

def sort_list_with_builtin():
    """
    Alternative method using list.sort() method
    """
    try:
        print("Enter numbers separated by spaces:")
        user_input = input("Enter your list: ")

        # Convert input string to list of integers
        numbers = [int(x) for x in user_input.split()]

        # Create a copy to avoid modifying original list
        numbers_copy = numbers.copy()
```



```python
def sort_list_with_builtin():

        # Convert input string to list of integers
        numbers = [int(x) for x in user_input.split()]

        # Create a copy to avoid modifying original list
        numbers_copy = numbers.copy()

        # Sort the list in ascending order using sort() method
        numbers_copy.sort()

        print(f"Original list: {numbers}")
        print(f"Sorted list (ascending): {numbers_copy}")

    except ValueError:
        print("Error: Please enter valid numbers separated by spaces")

# Main execution
if __name__ == "__main__":
    print("=== List Sorting Program ===")
    print("Method 1: Using sorted() function")
    sort_list_ascending()

    print("\n" + "="*40 + "\n")

    print("Method 2: Using sort() method")
    sort_list_with_builtin()
```

```
ethod 1: Using sorted() function
nter numbers separated by spaces (e.g., 5 2 8 1 9):
nter your list: 5 66 4 22 3 8 99 0 33
riginal list: [5, 66, 4, 22, 3, 8, 99, 0, 33]
orted list (ascending): [0, 3, 4, 5, 8, 22, 33, 66, 99]
```

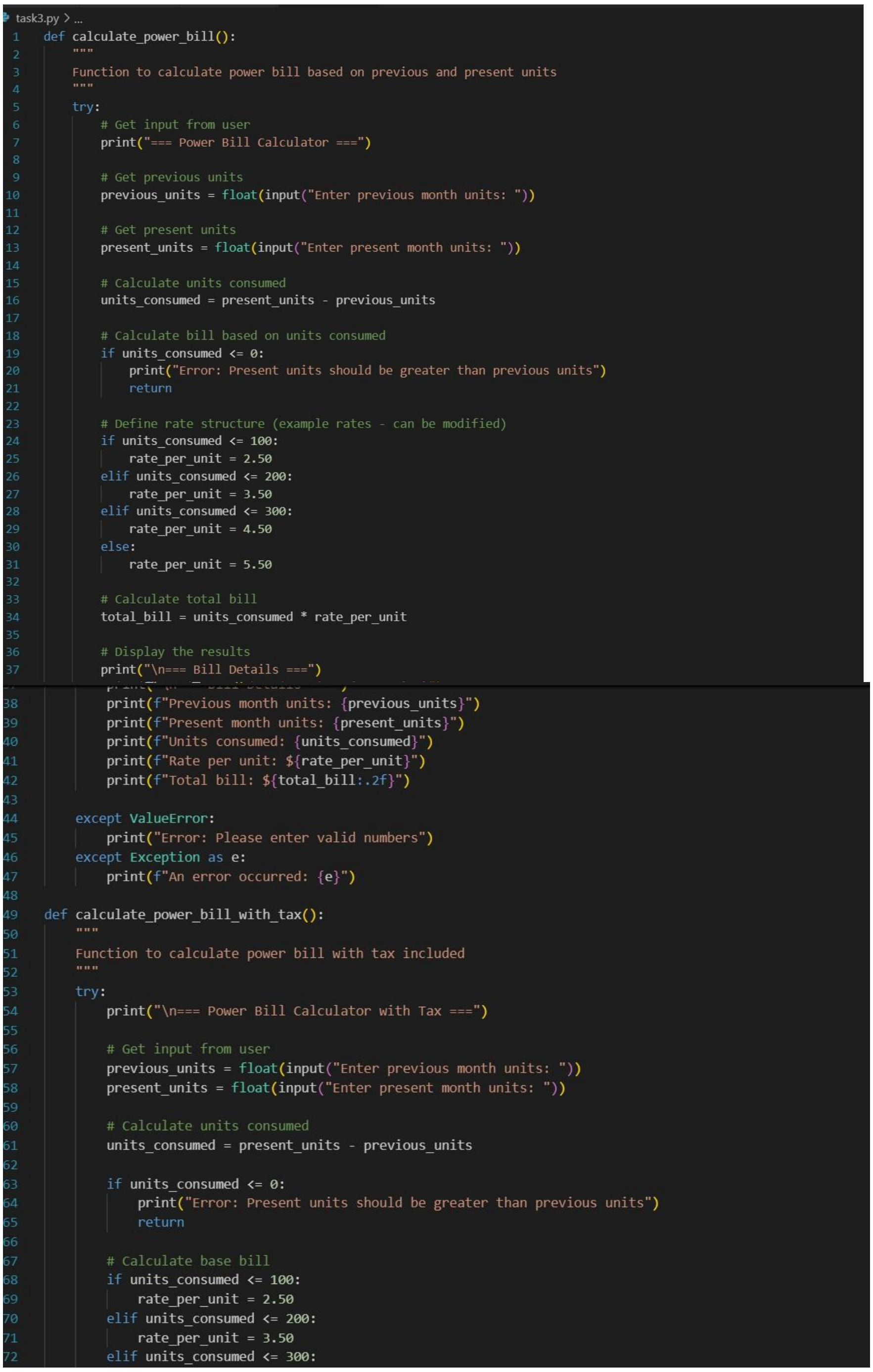
Task-03

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

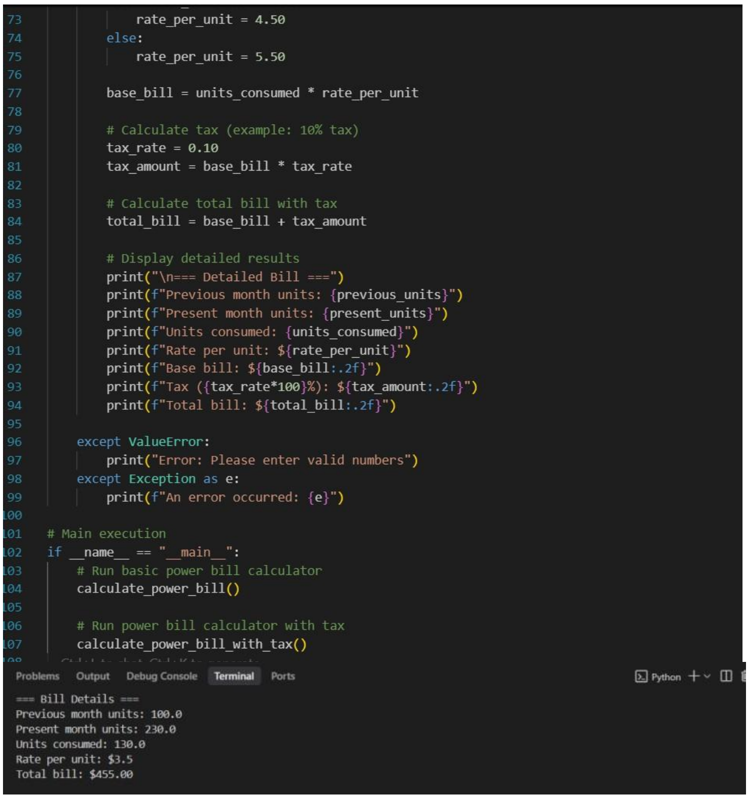
prompt:

generate a python code to calculate powerbill containing previous units ,present units ,read input units from the console

code:



```python
def calculate_power_bill():
    """
    Function to calculate power bill based on previous and present units
    """
    try:
        # Get input from user
        print("=== Power Bill Calculator ===")

        # Get previous units
        previous_units = float(input("Enter previous month units: "))

        # Get present units
        present_units = float(input("Enter present month units: "))

        # Calculate units consumed
        units_consumed = present_units - previous_units

        # Calculate bill based on units consumed
        if units_consumed <= 0:
            print("Error: Present units should be greater than previous units")
            return

        # Define rate structure (example rates - can be modified)
        if units_consumed <= 100:
            rate_per_unit = 2.50
        elif units_consumed <= 200:
            rate_per_unit = 3.50
        elif units_consumed <= 300:
            rate_per_unit = 4.50
        else:
            rate_per_unit = 5.50

        # Calculate total bill
        total_bill = units_consumed * rate_per_unit

        # Display the results
        print("\n=== Bill Details ===")
        print(f"Previous month units: {previous_units}")
        print(f"Present month units: {present_units}")
        print(f"Units consumed: {units_consumed}")
        print(f"Rate per unit: ${rate_per_unit}")
        print(f"Total bill: ${total_bill:.2f}")

    except ValueError:
        print("Error: Please enter valid numbers")
    except Exception as e:
        print(f"An error occurred: {e}")

def calculate_power_bill_with_tax():
    """
    Function to calculate power bill with tax included
    """
    try:
        print("\n=== Power Bill Calculator with Tax ===")

        # Get input from user
        previous_units = float(input("Enter previous month units: "))
        present_units = float(input("Enter present month units: "))

        # Calculate units consumed
        units_consumed = present_units - previous_units

        if units_consumed <= 0:
            print("Error: Present units should be greater than previous units")
            return

        # Calculate base bill
        if units_consumed <= 100:
            rate_per_unit = 2.50
        elif units_consumed <= 200:
            rate_per_unit = 3.50
        elif units_consumed <= 300:
```



```python
            rate_per_unit = 4.50
        else:
            rate_per_unit = 5.50

        base_bill = units_consumed * rate_per_unit

        # Calculate tax (example: 10% tax)
        tax_rate = 0.10
        tax_amount = base_bill * tax_rate

        # Calculate total bill with tax
        total_bill = base_bill + tax_amount

        # Display detailed results
        print("\n=== Detailed Bill ===")
        print(f"Previous month units: {previous_units}")
        print(f"Present month units: {present_units}")
        print(f"Units consumed: {units_consumed}")
        print(f"Rate per unit: ${rate_per_unit}")
        print(f"Base bill: ${base_bill:.2f}")
        print(f"Tax ({tax_rate*100}%): ${tax_amount:.2f}")
        print(f"Total bill: ${total_bill:.2f}")

    except ValueError:
        print("Error: Please enter valid numbers")
    except Exception as e:
        print(f"An error occurred: {e}")

# Main execution
if __name__ == "__main__":
    # Run basic power bill calculator
    calculate_power_bill()

    # Run power bill calculator with tax
    calculate_power_bill_with_tax()
```

```
=== Bill Details ===
Previous month units: 100.0
Present month units: 230.0
Units consumed: 130.0
Rate per unit: $3.5
Total bill: $455.00
```

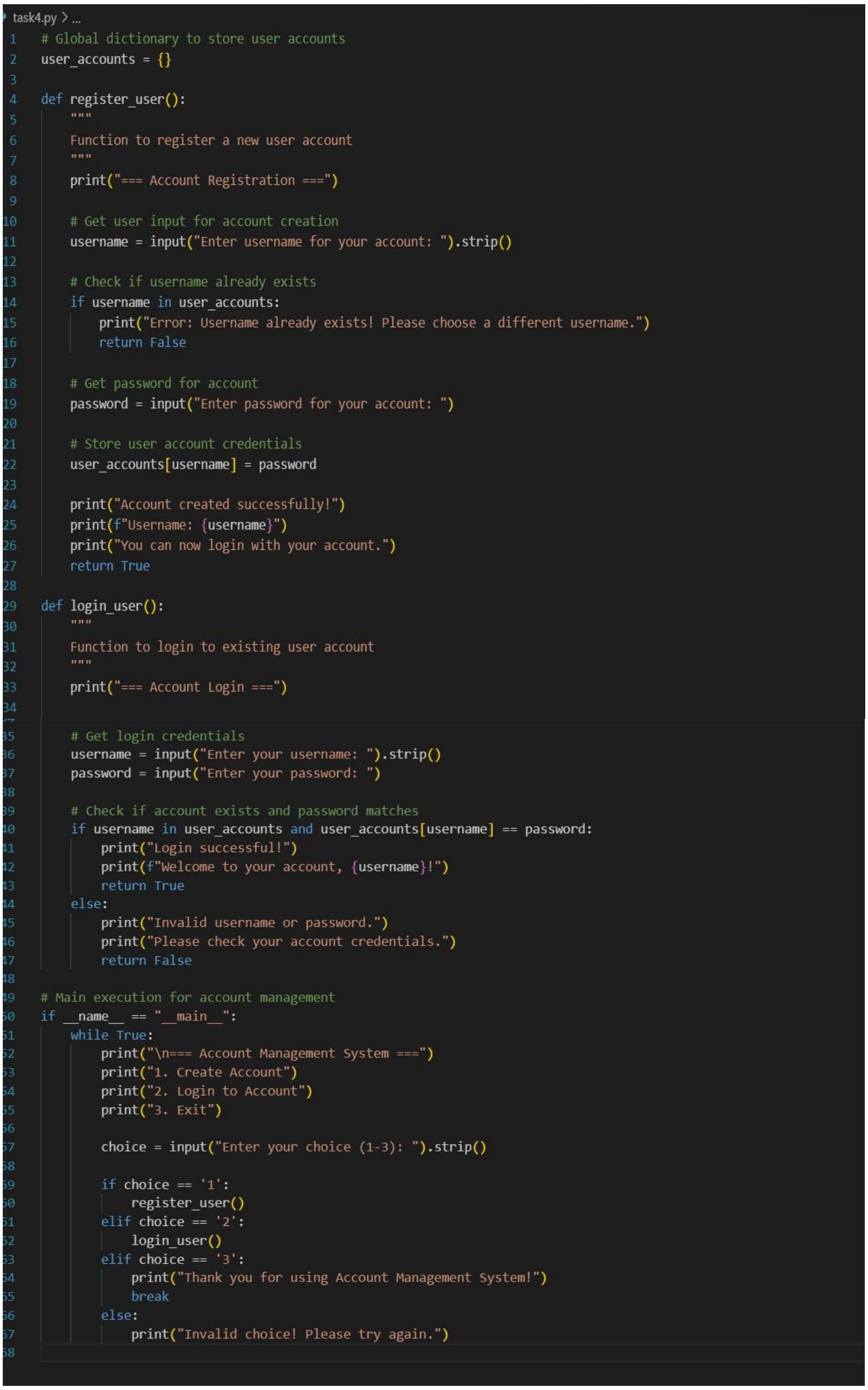
Task-04

- Write structured comments to help AI generate two linked functions (e.g., login_user() and register_user()).

Prompt:

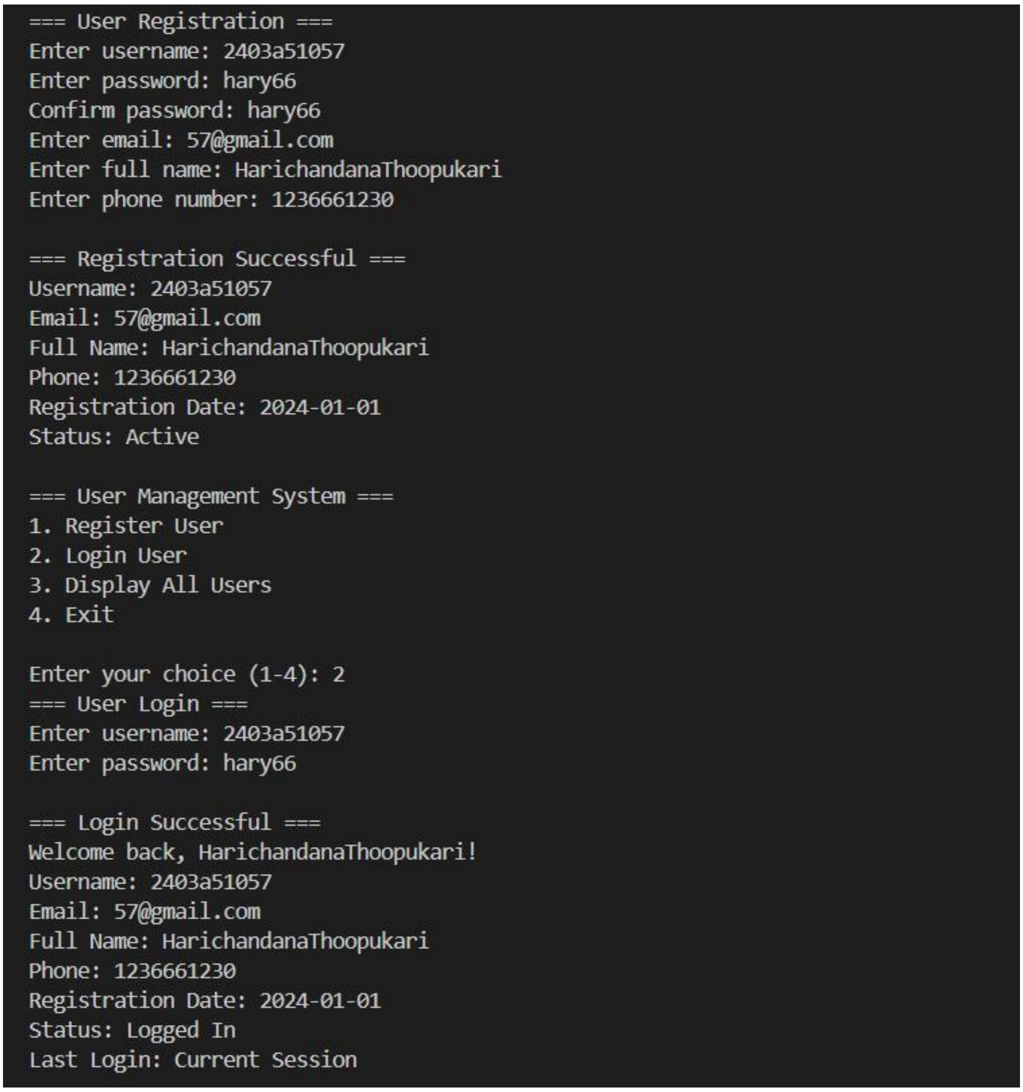
Generate simple two python linked functions one is for register user and login user for account creation

Code:



```python
# Global dictionary to store user accounts
user_accounts = {}

def register_user():
    """
    Function to register a new user account
    """
    print("=== Account Registration ===")

    # Get user input for account creation
    username = input("Enter username for your account: ").strip()

    # Check if username already exists
    if username in user_accounts:
        print("Error: Username already exists! Please choose a different username.")
        return False

    # Get password for account
    password = input("Enter password for your account: ")

    # Store user account credentials
    user_accounts[username] = password

    print("Account created successfully!")
    print(f"Username: {username}")
    print("You can now login with your account.")
    return True

def login_user():
    """
    Function to login to existing user account
    """
    print("=== Account Login ===")

    # Get login credentials
    username = input("Enter your username: ").strip()
    password = input("Enter your password: ")

    # Check if account exists and password matches
    if username in user_accounts and user_accounts[username] == password:
        print("Login successful!")
        print(f"Welcome to your account, {username}!")
        return True
    else:
        print("Invalid username or password.")
        print("Please check your account credentials.")
        return False

# Main execution for account management
if __name__ == "__main__":
    while True:
        print("\n=== Account Management System ===")
        print("1. Create Account")
        print("2. Login to Account")
        print("3. Exit")

        choice = input("Enter your choice (1-3): ").strip()

        if choice == '1':
            register_user()
        elif choice == '2':
            login_user()
        elif choice == '3':
            print("Thank you for using Account Management System!")
            break
        else:
            print("Invalid choice! Please try again.")

```

Output:



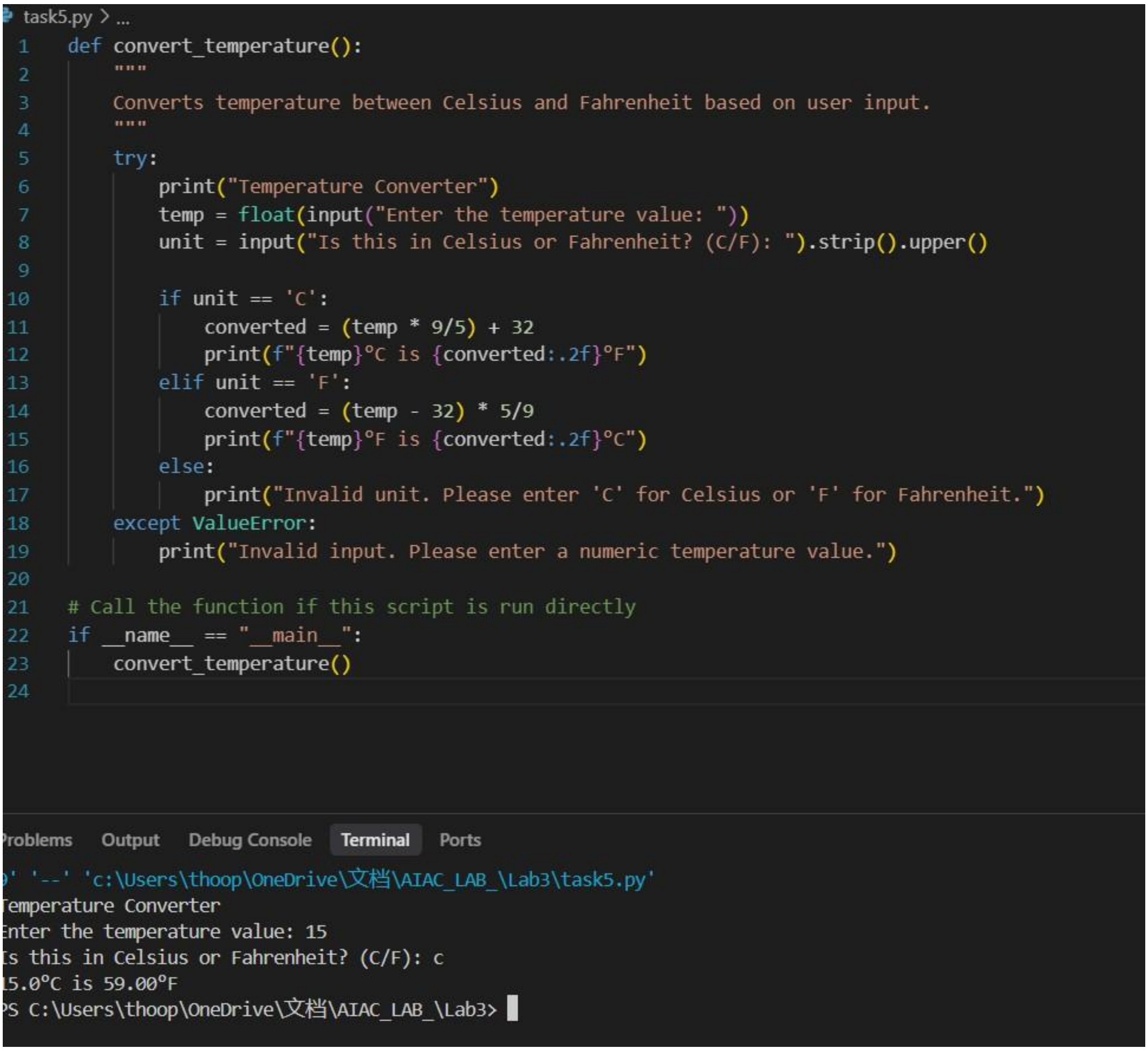
```
=== User Registration ===
Enter username: 2403a51057
Enter password: hary66
Confirm password: hary66
Enter email: 57@gmail.com
Enter full name: HarichandanaThoopukari
Enter phone number: 1236661230

=== Registration Successful ===
Username: 2403a51057
Email: 57@gmail.com
Full Name: HarichandanaThoopukari
Phone: 1236661230
Registration Date: 2024-01-01
Status: Active

=== User Management System ===
1. Register User
2. Login User
3. Display All Users
4. Exit

Enter your choice (1-4): 2
=== User Login ===
Enter username: 2403a51057
Enter password: hary66

=== Login Successful ===
Welcome back, HarichandanaThoopukari!
Username: 2403a51057
Email: 57@gmail.com
Full Name: HarichandanaThoopukari
Phone: 1236661230
Registration Date: 2024-01-01
Status: Logged In
Last Login: Current Session
```

Task -05

- Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

Prompt1:
Write a Python function to convert temperature from Celsius to Fahrenheit or vice-versa as for the user`s selection

Code:



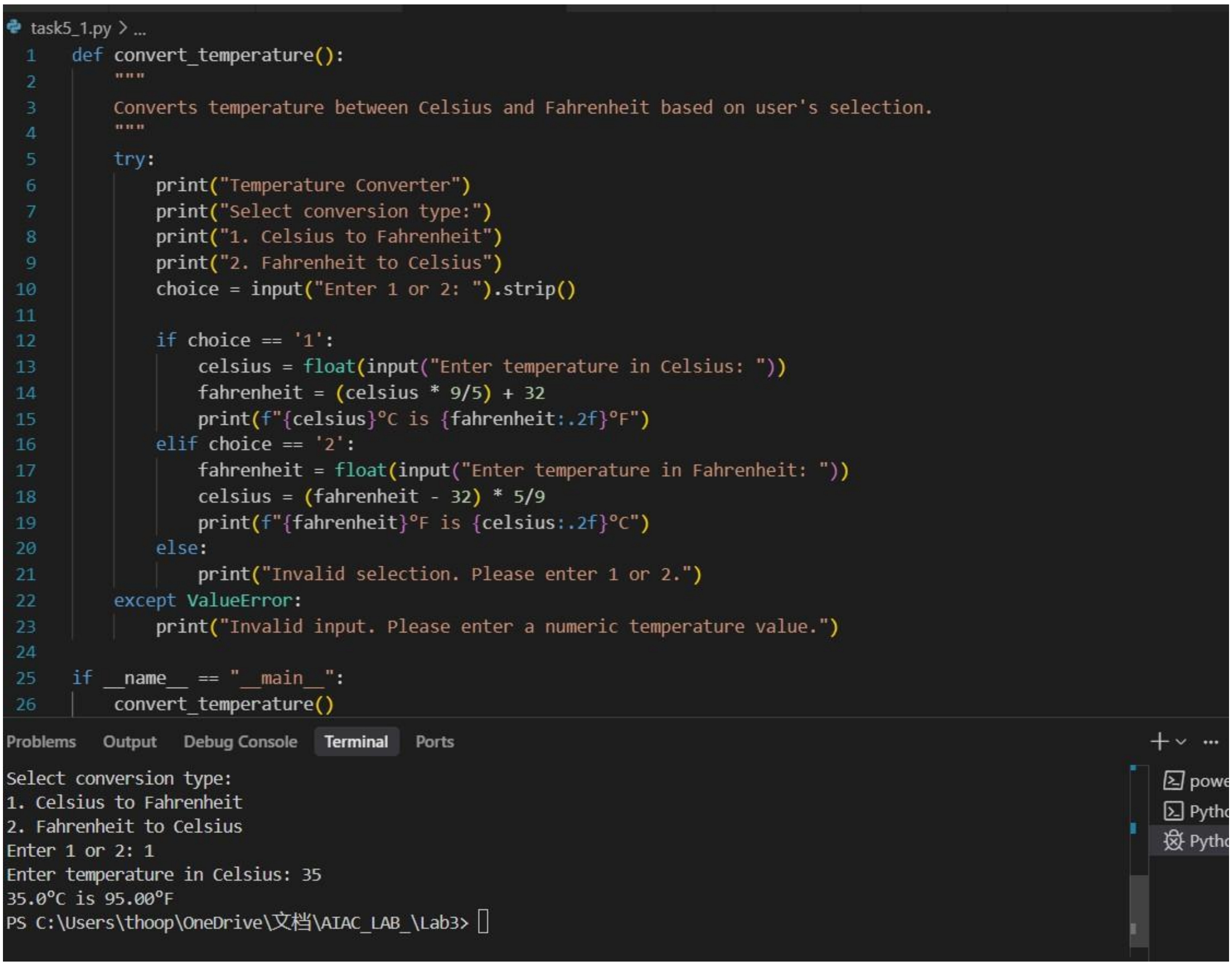
```python
def convert_temperature():
    """
    Converts temperature between Celsius and Fahrenheit based on user input.
    """
    try:
        print("Temperature Converter")
        temp = float(input("Enter the temperature value: "))
        unit = input("Is this in Celsius or Fahrenheit? (C/F): ").strip().upper()

        if unit == 'C':
            converted = (temp * 9/5) + 32
            print(f"{temp}°C is {converted:.2f}°F")
        elif unit == 'F':
            converted = (temp - 32) * 5/9
            print(f"{temp}°F is {converted:.2f}°C")
        else:
            print("Invalid unit. Please enter 'C' for Celsius or 'F' for Fahrenheit.")
    except ValueError:
        print("Invalid input. Please enter a numeric temperature value.")

# Call the function if this script is run directly
if __name__ == "__main__":
    convert_temperature()

```

```
' '--' 'c:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3\task5.py'
emperature Converter
nter the temperature value: 15
s this in Celsius or Fahrenheit? (C/F): c
5.0°C is 59.00°F
S C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3>
```

Prompt2:

Write a Python function to convert temperature from Celsius to Fahrenheit or vice-versa as for the user`s selection.

Code along with output:



```python
def convert_temperature():
    """
    Converts temperature between Celsius and Fahrenheit based on user's selection.
    """
    try:
        print("Temperature Converter")
        print("Select conversion type:")
        print("1. Celsius to Fahrenheit")
        print("2. Fahrenheit to Celsius")
        choice = input("Enter 1 or 2: ").strip()

        if choice == '1':
            celsius = float(input("Enter temperature in Celsius: "))
            fahrenheit = (celsius * 9/5) + 32
            print(f"{celsius}°C is {fahrenheit:.2f}°F")
        elif choice == '2':
            fahrenheit = float(input("Enter temperature in Fahrenheit: "))
            celsius = (fahrenheit - 32) * 5/9
            print(f"{fahrenheit}°F is {celsius:.2f}°C")
        else:
            print("Invalid selection. Please enter 1 or 2.")
    except ValueError:
        print("Invalid input. Please enter a numeric temperature value.")

if __name__ == "__main__":
    convert_temperature()
```

```
Select conversion type:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter 1 or 2: 1
Enter temperature in Celsius: 35
35.0°C is 95.00°F
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> 
```