SR UNIVERSITY

AI ASSIST CODING

Lab-1.2

ROLL NO:2503A51L17

NAME: Simra Tahseen

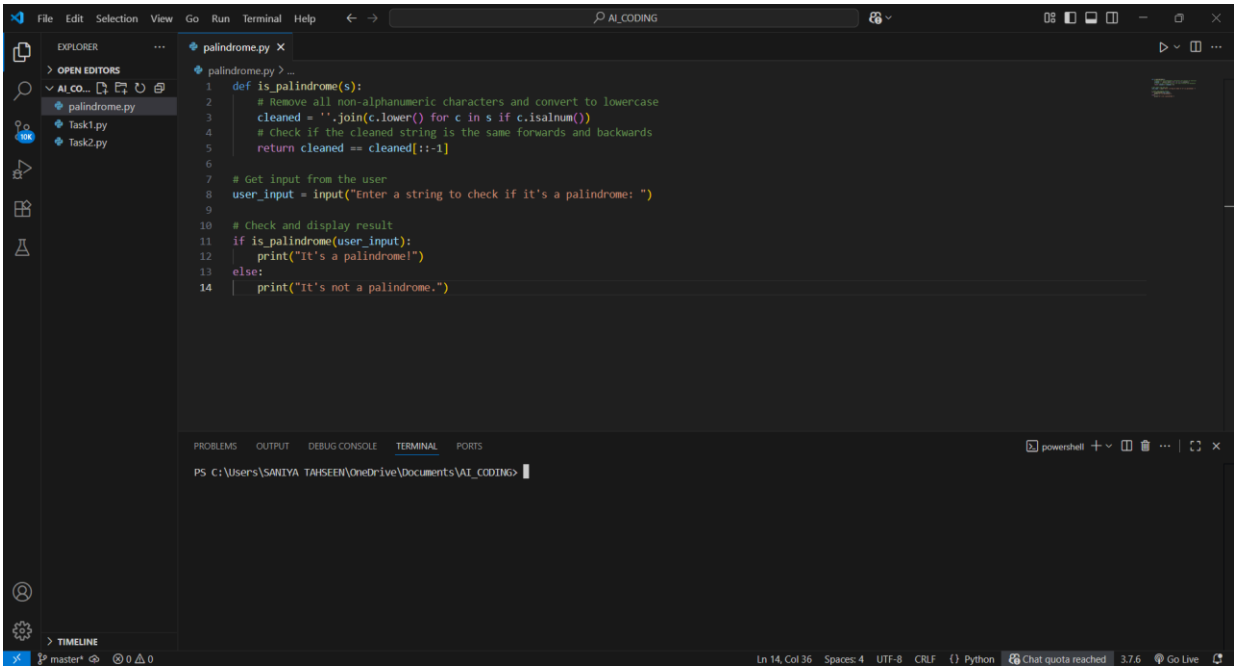
BATCH:19

TASK #1:

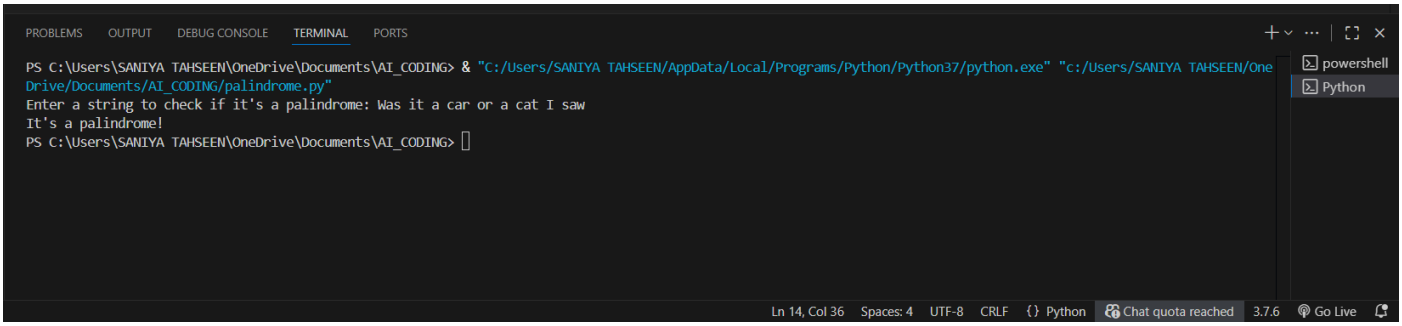
Prompt Used:

Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Code Generated:



Output After executing Code:



Observations:

- It cleans the input by removing all non-alphanumeric characters and converting everything to lowercase, ensuring accurate results regardless of punctuation, spacing, or case.

- Checks for palindrome by comparing the cleaned string to its reverse (cleaned[::-1]). If both match, the string is a palindrome.
- Works for sentences and phrases (not just single words), as spaces, punctuation, and cases are ignored in the check.

TASK #2:

Prompt Used:

Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring.

Code Generated:

fibonacci.py

```python
def fibonacci(n):
    """
    Generate and return the Fibonacci sequence up to n terms.
    """
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

# Get user input
num_terms = int(input("Enter the number of Fibonacci terms to generate: "))

print(f"Fibonacci sequence up to {num_terms} terms:")
for num in fibonacci(num_terms):
    print(num, end=' ')
print()  # for newline after printing all numbers

```

Output After executing Code:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/fibonacci.py"
Enter the number of Fibonacci terms to generate: 5
Fibonacci sequence up to 5 terms:
0 1 1 2 3
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/fibonacci.py"
Enter the number of Fibonacci terms to generate: 9
Fibonacci sequence up to 9 terms:
0 1 1 2 3 5 8 13 21
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/fibonacci.py"
Enter the number of Fibonacci terms to generate:

powershell
Python

Ln 19, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  Chat quota reached  3.7.6  Go Live
```

Observations:

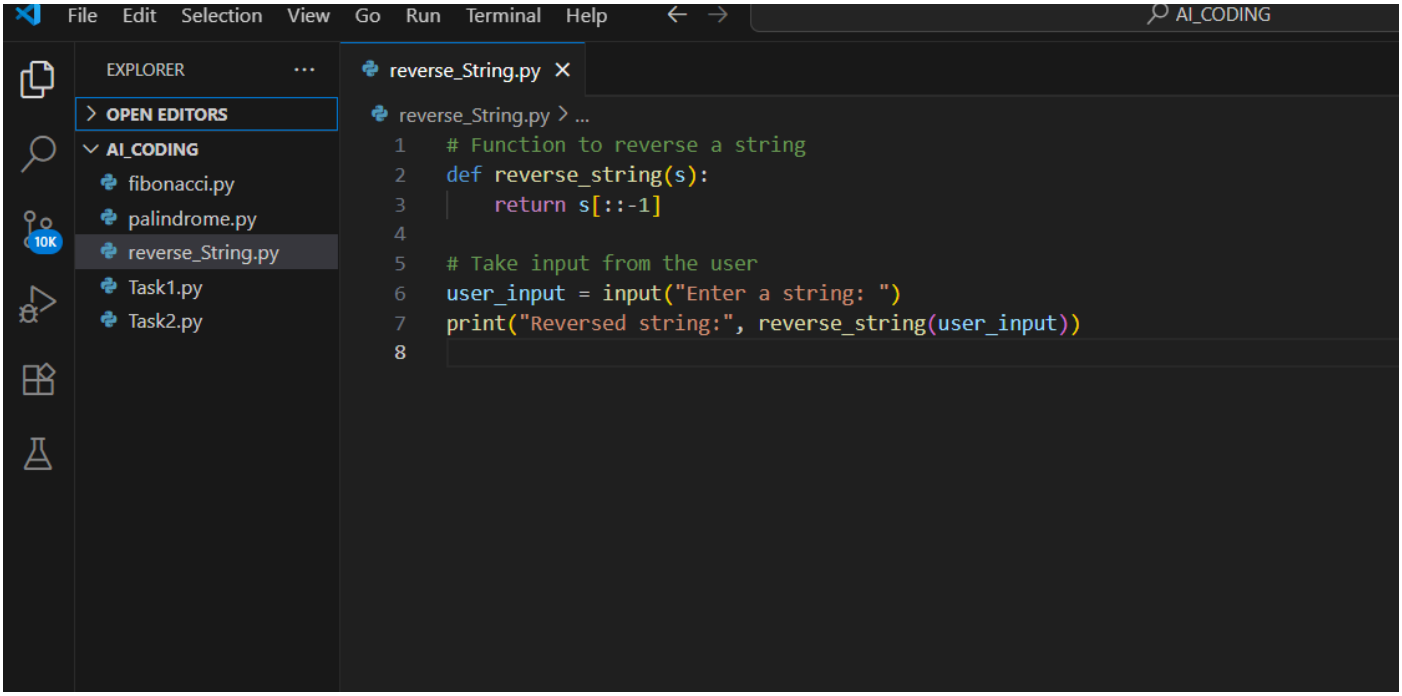
- The code generates the Fibonacci sequence up to a user-specified number of terms using a simple iterative approach.
- The sequence is generated by initializing the first two terms (a = 0, b = 1) and iteratively updating them with a, b = b, a + b, ensuring each new term is the sum of its two immediate predecessors.
- Each generated term is stored in a list, which is then returned and printed.

TASK #3:

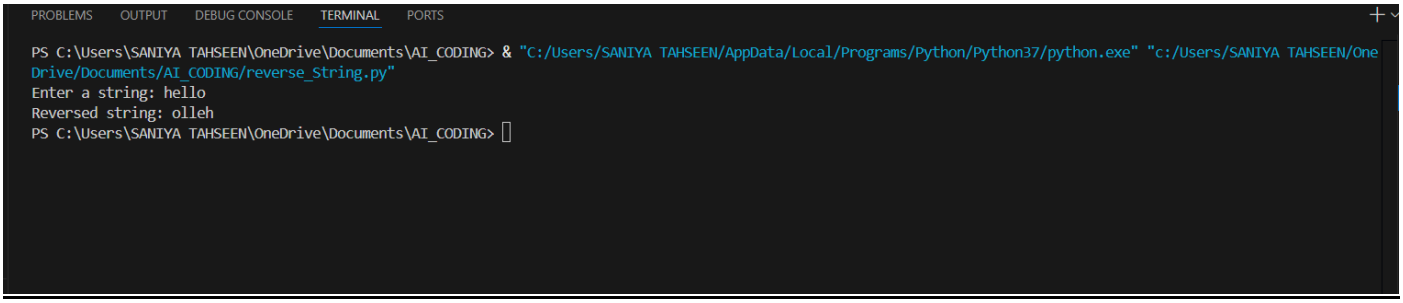
Prompt Used:

Write a comment like # Function to reverse a string and use Copilot to generate the function.

Code Generated:



Output After executing Code:



Observations:

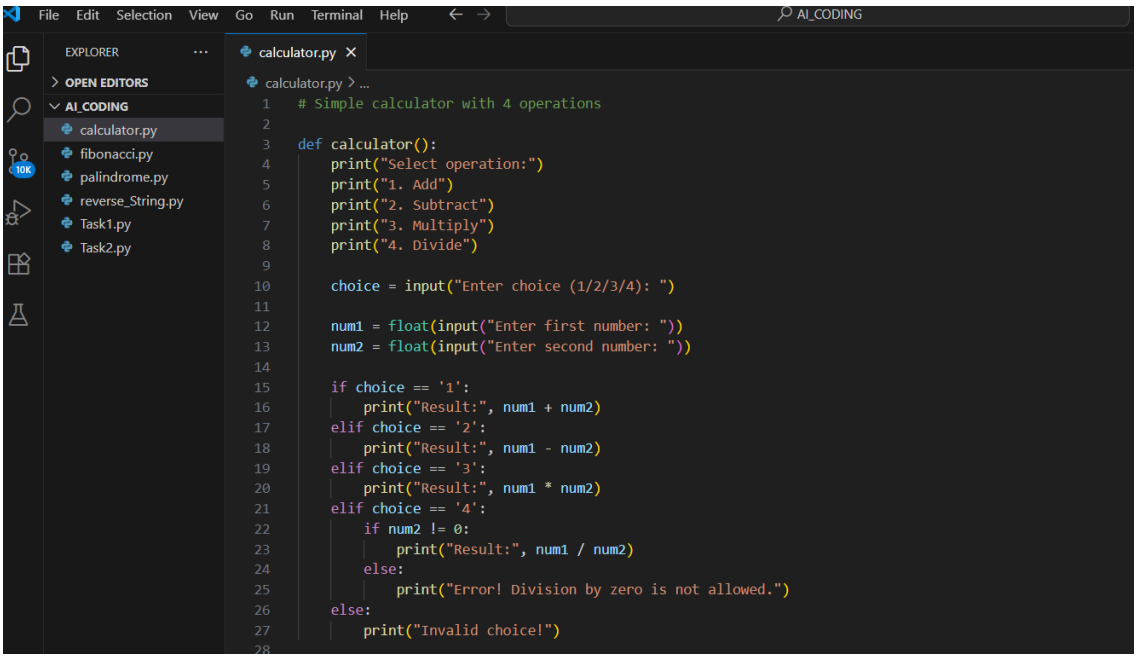
- The function uses Python's string slicing syntax s[::-1] to reverse the string.
- The slice step of -1 means characters are taken from end to start, effectively reversing the string.
- User input is taken and passed to the function, with the reversed result printed.

TASK #4:

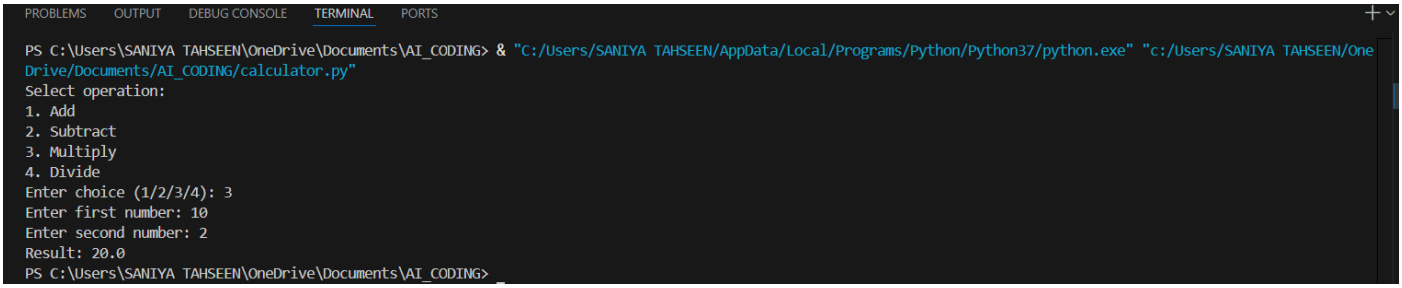
Prompt Used:

Generate a program that simulates a basic calculator (add, subtract, multiply, divide).
Write the comment: # Simple calculator with 4 operations and let AI complete it.

Code Generated:



Output After executing Code:



Observations:

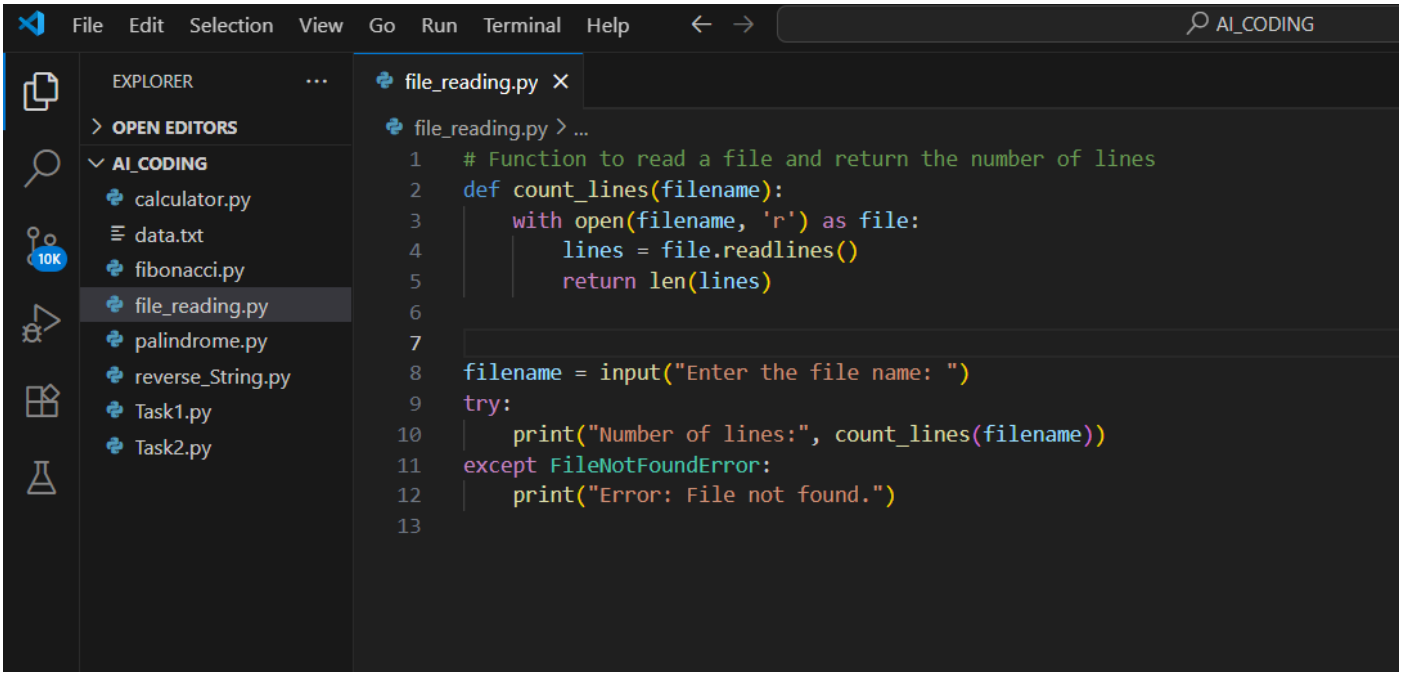
- The program allows the user to choose from four basic arithmetic operations.
- It accepts user input as float, enabling both integer and decimal calculations.
- Division includes error handling to avoid division by zero.

TASK #5:

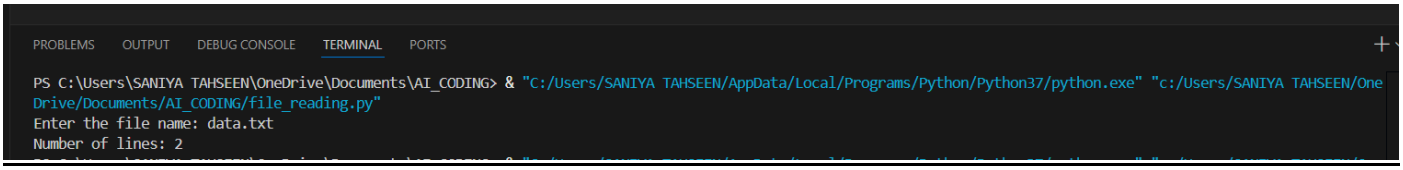
Prompt Used:

Use a comment to instruct AI to write a function that reads a file and returns the number of lines

Code Generated:



Output After executing Code:



Observations:

- The program defines a function count_lines() that opens a file and counts the number of lines using readlines().
- It uses with open() to ensure the file is properly handled and closed automatically.
- The program takes the filename as user input, making it flexible for different files.
- Error handling with try-except prevents crashes in case the file is not found.