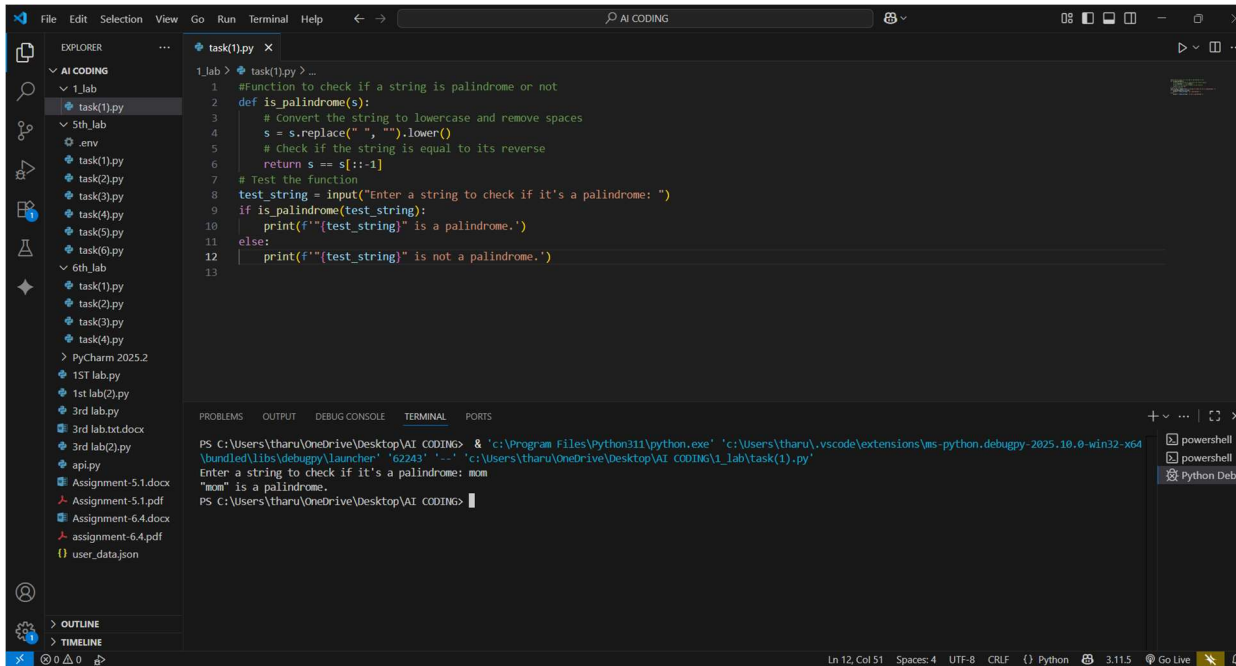


# AI Assisted Coding-ASSIGNMENT-1

## Task 1:

Write a Function to check if a string is a valid palindrome (ignoring spaces and case).



```
1 #Function to check if a string is palindrome or not
2 def is_palindrome(s):
3     # Convert the string to lowercase and remove spaces
4     s = s.replace(" ", "").lower()
5     # Check if the string is equal to its reverse
6     return s == s[::-1]
7
8 # Test the function
9 test_string = input("Enter a string to check if it's a palindrome: ")
10 if is_palindrome(test_string):
11     print(f"{test_string} is a palindrome.")
12 else:
13     print(f"{test_string} is not a palindrome.")
```

```
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> & 'c:\Program Files\Python311\python.exe' 'c:\Users\tharu\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\
bundled\libs\debugpy\launcher' '62243' '-c' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\1_lab\task(1).py'
Enter a string to check if it's a palindrome: mom
"mom" is a palindrome.
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

NAME: B.Nishant  
HT.NO: 2403A52089  
BATCH: AIB04

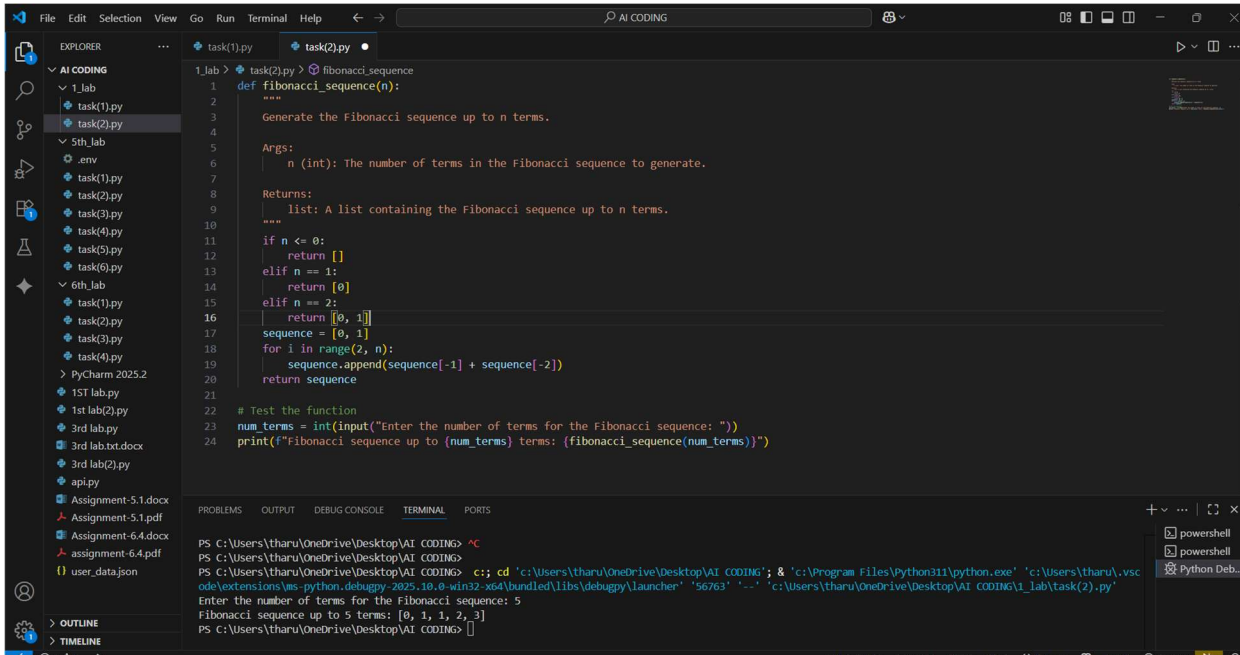
## Task 2:

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt

with only a function header and docstring

## Expected Output#2

- AI completes the function logic using loop or recursion with accurate output



The screenshot shows a Visual Studio Code editor with a file explorer on the left containing various files like task(1).py, task(2).py, and task(3).py. The main editor window displays a Python script named task(2).py. The script defines a function `fibonacci_sequence(n)` with a docstring and logic to generate the Fibonacci sequence up to n terms using a loop. Below the function, there is a test case that prompts the user for the number of terms and prints the resulting sequence. The terminal at the bottom shows the command to run the script, the input '5', and the output '[0, 1, 1, 2, 3]'.

```
1 def fibonacci_sequence(n):
2     """
3     Generate the Fibonacci sequence up to n terms.
4
5     Args:
6         n (int): The number of terms in the Fibonacci sequence to generate.
7
8     Returns:
9         list: A list containing the Fibonacci sequence up to n terms.
10    """
11    if n <= 0:
12        return []
13    elif n == 1:
14        return [0]
15    elif n == 2:
16        return [0, 1]
17    sequence = [0, 1]
18    for i in range(2, n):
19        sequence.append(sequence[-1] + sequence[-2])
20    return sequence
21
22 # Test the function
23 num_terms = int(input("Enter the number of terms for the Fibonacci sequence: "))
24 print(f"Fibonacci sequence up to {num_terms} terms: {fibonacci_sequence(num_terms)}")
```

Terminal Output:

```
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> ^C
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> c:\Program Files\Python311\python.exe "c:\Users\tharu\OneDrive\Desktop\AI CODING\1_lab\task(2).py"
Enter the number of terms for the Fibonacci sequence: 5
Fibonacci sequence up to 5 terms: [0, 1, 1, 2, 3]
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

NAME: B.Nishant

HT.NO: 2403A52089

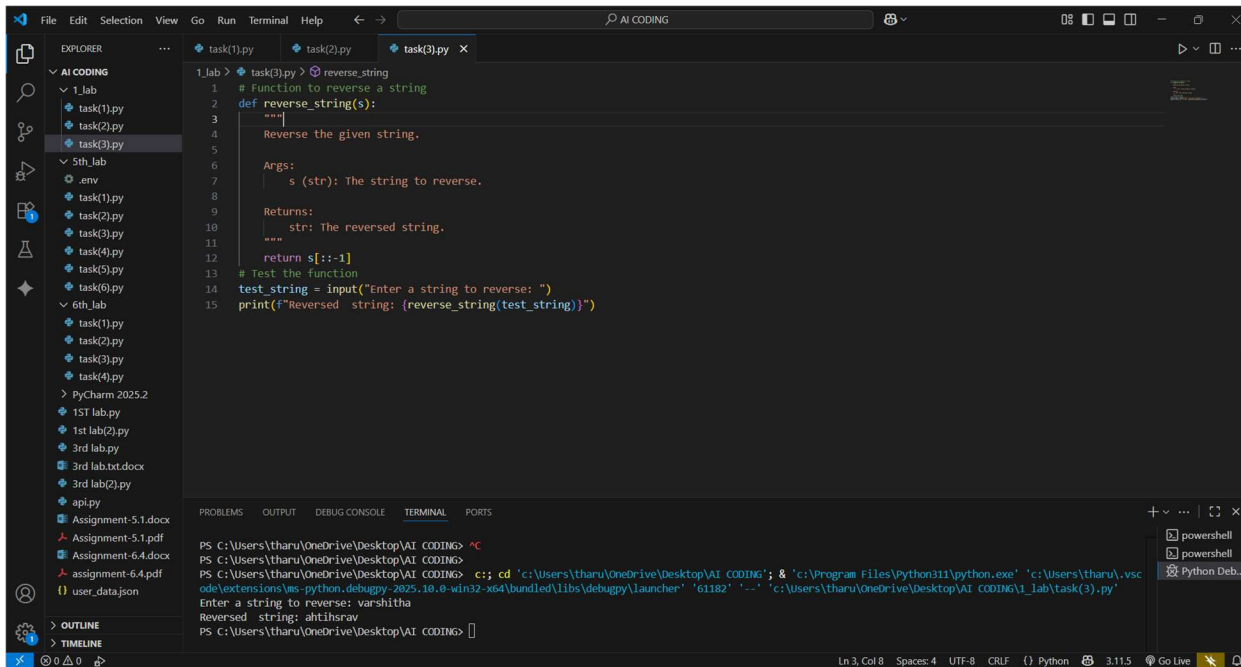
BATCH: AIB04

### Task 3:

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

### Expected Output#3

- Auto-completed reverse function



The screenshot shows the Visual Studio Code interface with a Python file named `task(3).py` open. The code defines a function `reverse_string` and tests it with a sample input.

```
1_lab > task(3).py > reverse_string
1 # Function to reverse a string
2 def reverse_string(s):
3     """
4     Reverse the given string.
5
6     Args:
7         s (str): The string to reverse.
8
9     Returns:
10        str: The reversed string.
11    """
12    return s[::-1]
13 # Test the function
14 test_string = input("Enter a string to reverse: ")
15 print(f"Reversed string: {reverse_string(test_string)}")
```

The terminal at the bottom shows the command prompt execution:

```
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> ^C
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> c:; cd 'c:\Users\tharu\OneDrive\Desktop\AI CODING'; & 'c:\Program Files\Python11\python.exe' 'c:\Users\tharu\vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '61182' '-.' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\1_lab\task(3).py'
Enter a string to reverse: varshitha
Reversed string: ahtsihsrav
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

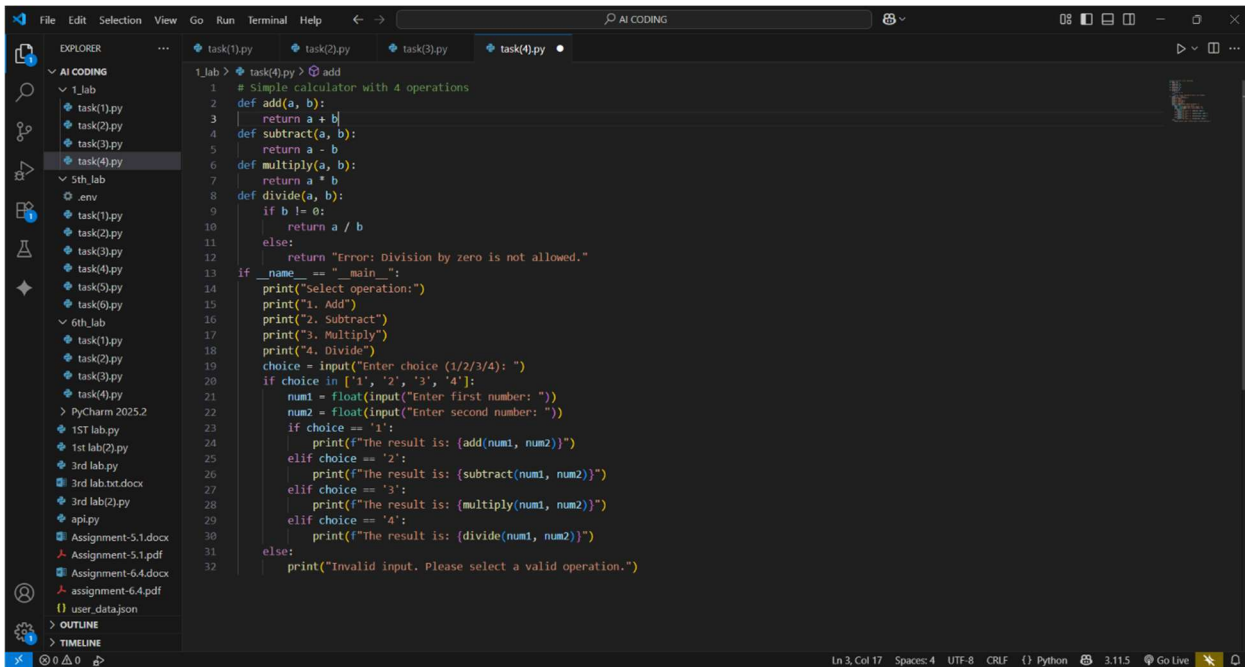
## Task 4:

Generate a program that simulates a basic calculator (add, subtract, multiply, divide).

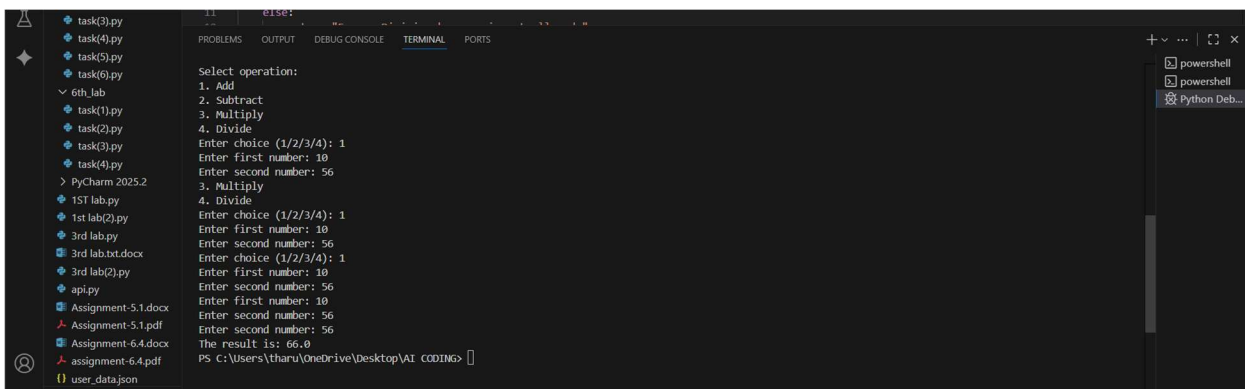
Write the comment: # Simple calculator with 4 operations and let AI complete it.

### Expected Output#4

- Fully working calculator with input/output and operator selection logic



```
1 # Simple calculator with 4 operations
2 def add(a, b):
3     return a + b
4 def subtract(a, b):
5     return a - b
6 def multiply(a, b):
7     return a * b
8 def divide(a, b):
9     if b != 0:
10        return a / b
11    else:
12        return "Error: Division by zero is not allowed."
13 if __name__ == "__main__":
14     print("Select operation:")
15     print("1. Add")
16     print("2. Subtract")
17     print("3. Multiply")
18     print("4. Divide")
19     choice = input("Enter choice (1/2/3/4): ")
20     if choice in ['1', '2', '3', '4']:
21         num1 = float(input("Enter first number: "))
22         num2 = float(input("Enter second number: "))
23         if choice == '1':
24             print(f"The result is: {add(num1, num2)}")
25         elif choice == '2':
26             print(f"The result is: {subtract(num1, num2)}")
27         elif choice == '3':
28             print(f"The result is: {multiply(num1, num2)}")
29         elif choice == '4':
30             print(f"The result is: {divide(num1, num2)}")
31     else:
32         print("Invalid input. Please select a valid operation.")
```



```
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 1
Enter first number: 10
Enter second number: 56
3. Multiply
4. Divide
Enter choice (1/2/3/4): 1
Enter first number: 10
Enter second number: 56
Enter choice (1/2/3/4): 1
Enter first number: 10
Enter second number: 56
The result is: 66.0
PS C:\Users\tharu\OneDrive\Desktop\VAI CODING>
```

NAME: B.Nishant

HT.NO: 2403A52089

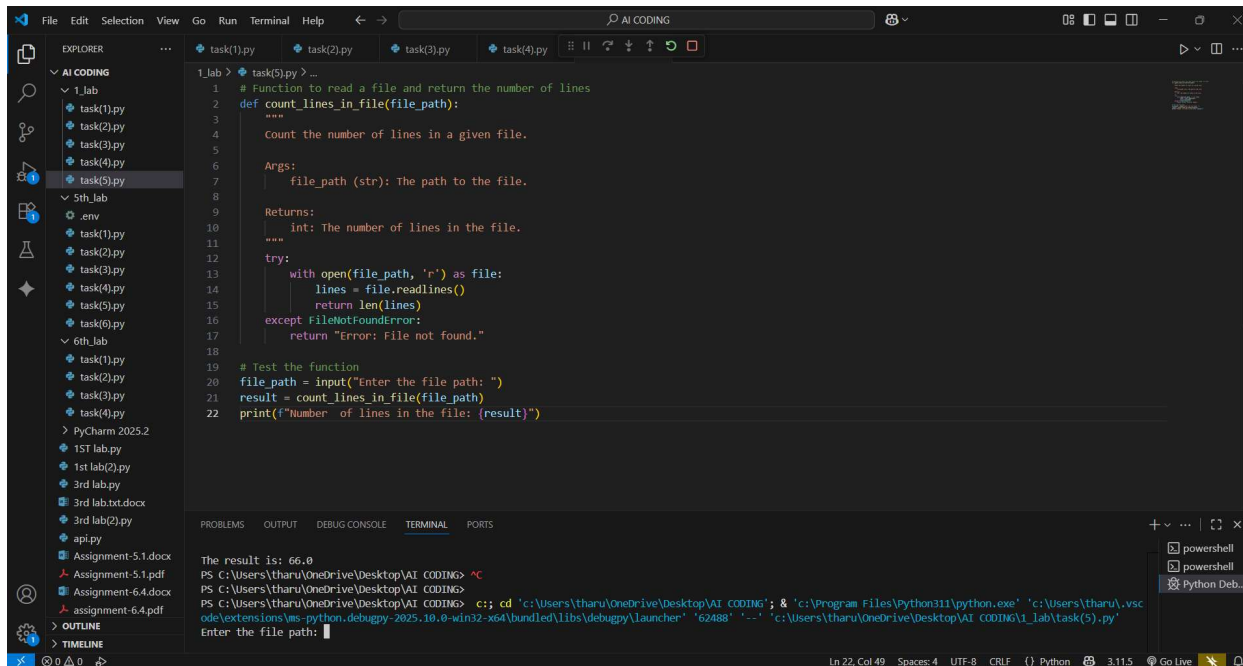
BATCH: AIB04

## Task 5:

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

## Expected Output#5

- Functional implementation using open() or with open() and readlines()



The screenshot shows the PyCharm IDE with a Python script named `task(5).py` open. The script defines a function `count_lines_in_file` that takes a file path as an argument and returns the number of lines in the file. The function uses `with open(file_path, 'r') as file:` to open the file and `lines = file.readlines()` to read all lines. The number of lines is then returned as `len(lines)`. The script also includes a test section where the user is prompted to enter a file path, and the function is called to count the lines. The terminal output shows the result: `The result is: 66.0`. The terminal also displays the command used to run the script: `PS C:\Users\tharu\OneDrive\Desktop\AI CODING> cd 'c:\Users\tharu\OneDrive\Desktop\AI CODING'; & 'c:\Program Files\Python11\python.exe' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\1_lab\task(5).py'`.

```
1 # Function to read a file and return the number of lines
2 def count_lines_in_file(file_path):
3     """
4     Count the number of lines in a given file.
5
6     Args:
7         file_path (str): The path to the file.
8
9     Returns:
10        int: The number of lines in the file.
11    """
12    try:
13        with open(file_path, 'r') as file:
14            lines = file.readlines()
15            return len(lines)
16    except FileNotFoundError:
17        return "Error: File not found."
18
19 # Test the function
20 file_path = input("Enter the file path: ")
21 result = count_lines_in_file(file_path)
22 print(f"Number of lines in the file: {result}")
```

The result is: 66.0

PS C:\Users\tharu\OneDrive\Desktop\AI CODING> cd 'c:\Users\tharu\OneDrive\Desktop\AI CODING'; & 'c:\Program Files\Python11\python.exe' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\1\_lab\task(5).py'

NAME: B.Nishant

HT.NO: 2403A52089

BATCH: AIB04