**SR UNIVERSITY**

**AI ASSISTED CODING**

**NAME:** Mustafa hussain shaik

**HTNO**: 2503A51L43

**Lab 10: Code Review and Quality: Using AI to improve code quality and readability**

**Lab Objectives:**

* To understand the importance of code readability, maintainability, and quality.
* To explore how AI-assisted coding tools can review code and suggest improvements.
* To practice identifying code smells, redundant code, and poor naming conventions.
* To apply AI tools for refactoring and improving readability.
* To critically evaluate AI feedback and integrate it into real projects

**Lab Outcomes (LOs):**

After completing this lab, students will be able to:

* Use AI-assisted tools (e.g., GitHub Copilot, Cursor AI) to review Python code.
* Identify and correct syntax issues, code smells, and inefficient logic.
* Improve readability by applying consistent formatting, naming, and comments.
* Refactor code with AI suggestions while ensuring functionality is preserved.
* Apply best practices for writing clean, maintainable, and professional code.

**Task Description#1 AI-Assisted Code Review (Basic Errors):**

* Write python program as shown below.
* Use an AI assistant to review and suggest corrections.

**Given program to ai which was manually written:**

A screenshot of a computer

AI-generated content may be incorrect.

**Output:**

A screen shot of a computer program

AI-generated content may be incorrect.

**Given prompt to AI:**

A screenshot of a computer

AI-generated content may be incorrect.

**Given program by AI:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Output**:

A screenshot of a computer

AI-generated content may be incorrect.

**Observation:**

AI corrected the loop range to include n for proper factorial calculation.

It fixed indentation and removed the unused variable.

It also corrected the missing parenthesis in the main() call so the program runs correctly.

**Task Description#2 Automatic Inline Comments:**

* Write the Python code for Fibonacci as shown below and execute.
* Ask AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up).
* Students evaluate which suggestions improve readability most. one

**Given manually written program to AI:**

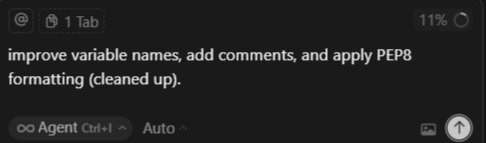
A black screen with white text

AI-generated content may be incorrect.

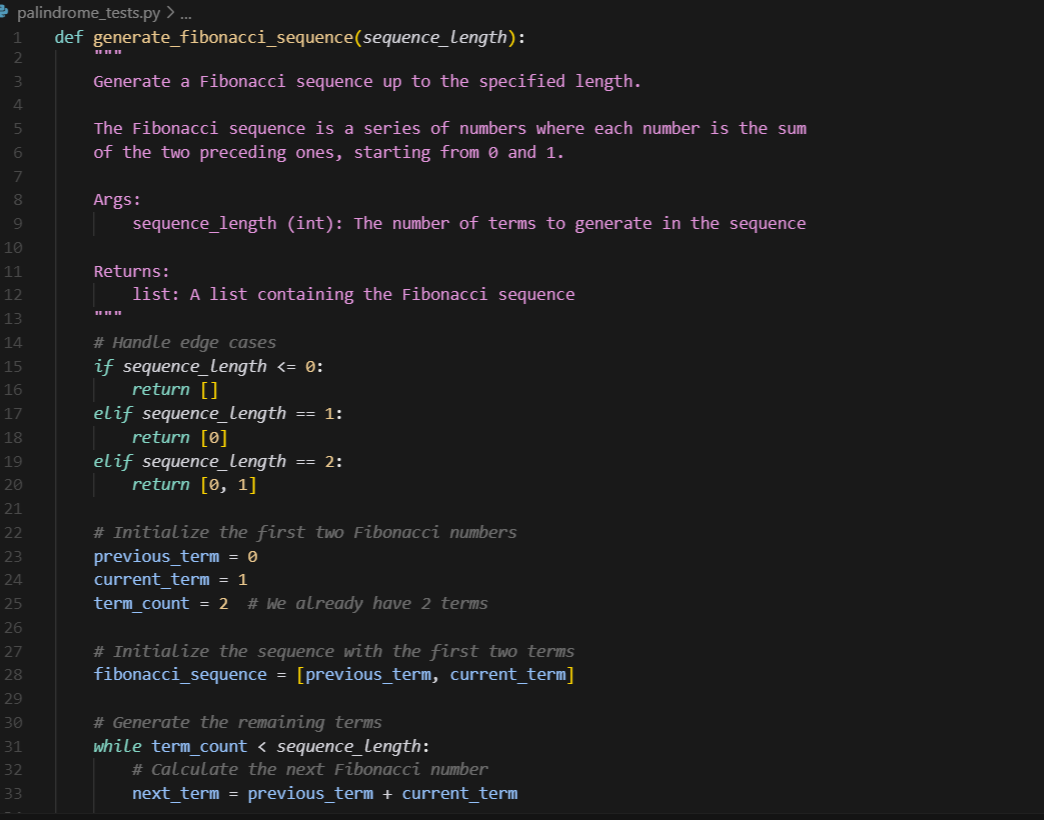
**Output: A screen shot of a computer

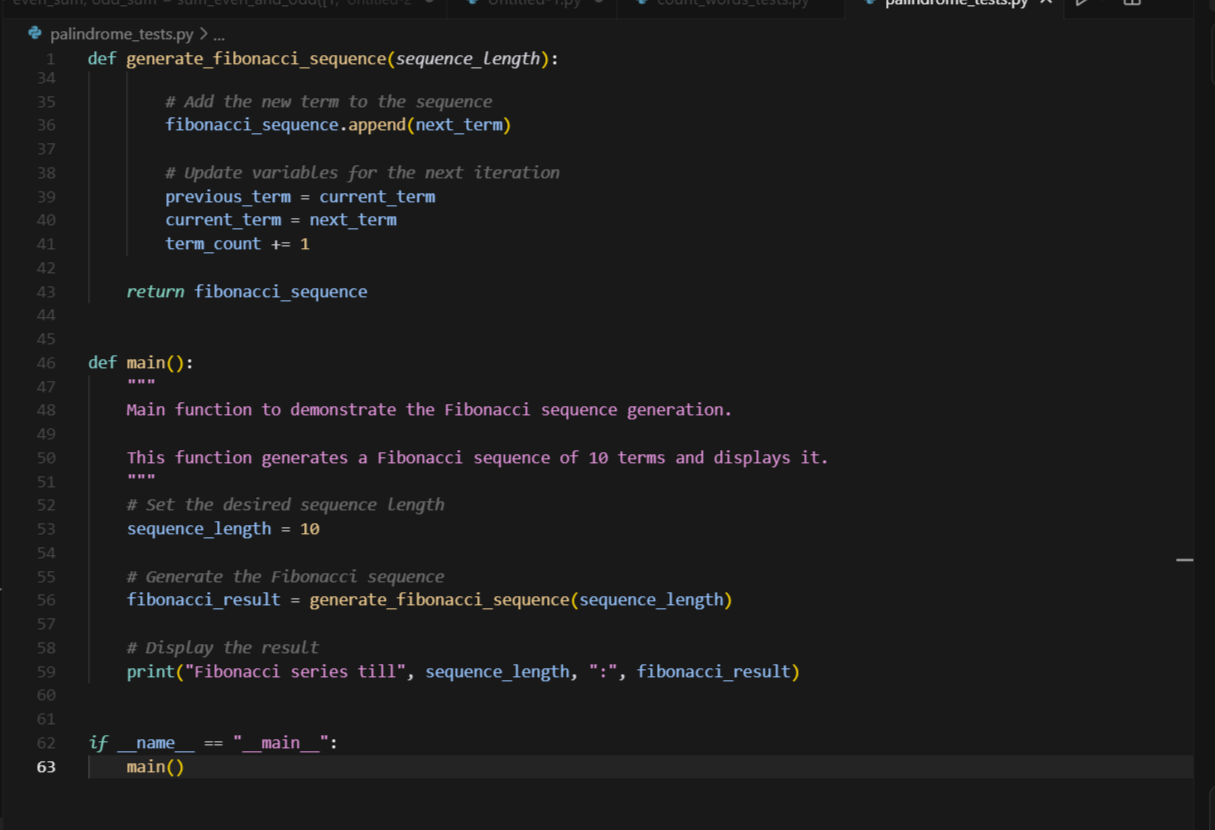
AI-generated content may be incorrect.**

**Given prompt:**

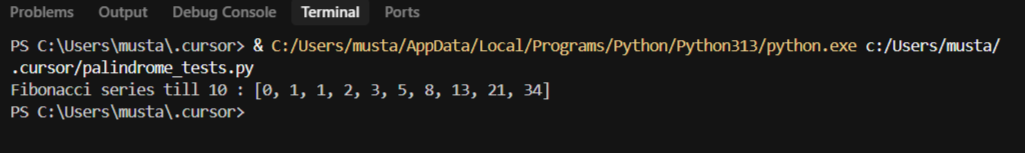
****

**Given program by AI:**

****

****

**Output**:



**Observation**:

AI improved the variable names to make the code more readable and meaningful.

It applied PEP8 formatting with proper indentation and spacing.

It also added comments and a docstring to clearly explain the function and its purpose.

**Task Description#3**

* Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
* Incorporate manual **docstring** in code with NumPy Style
* Use AI assistance to generate a module-level docstring + individual function docstrings.
* Compare the AI-generated docstring with your manually written one.

**Given program with multiple functions and manual docstring in code:**

A screenshot of a computer program

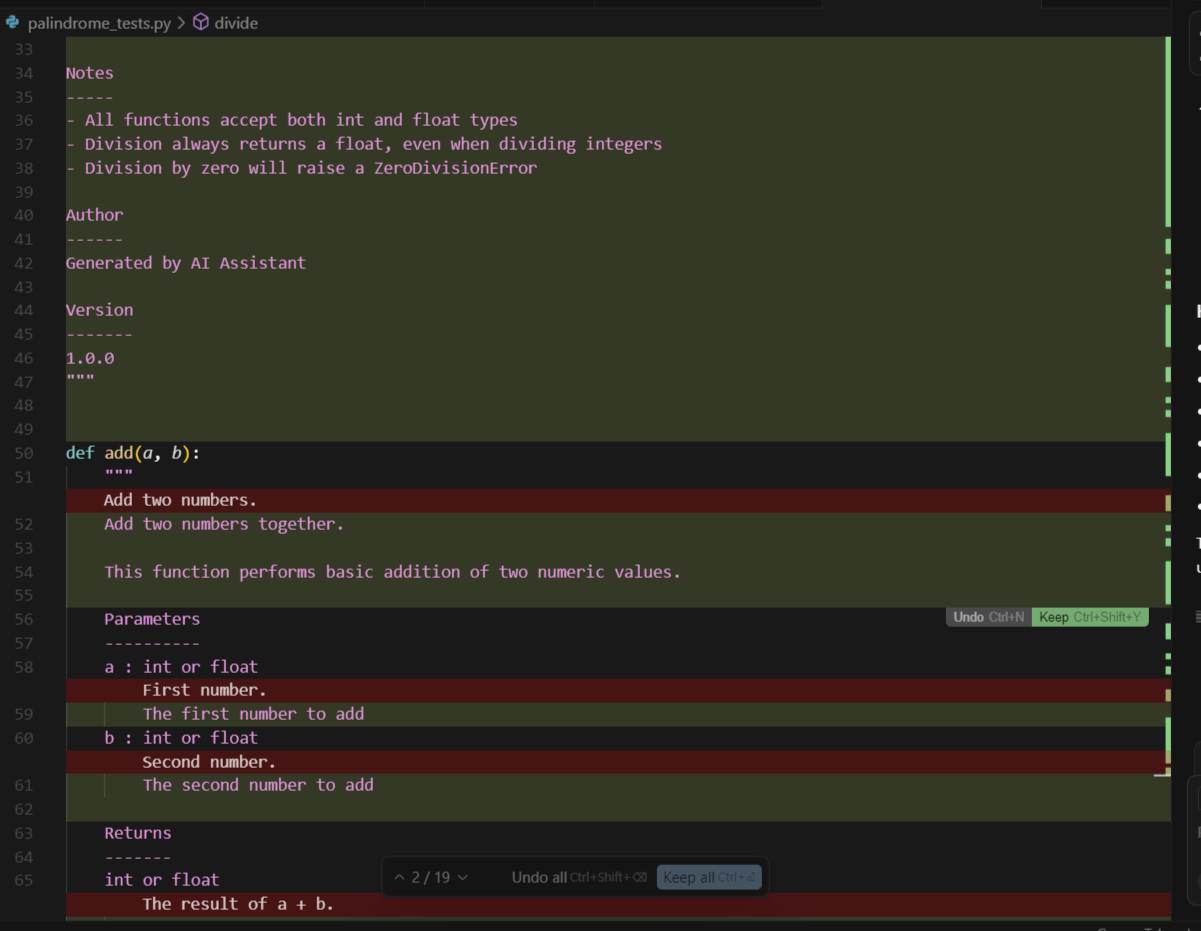
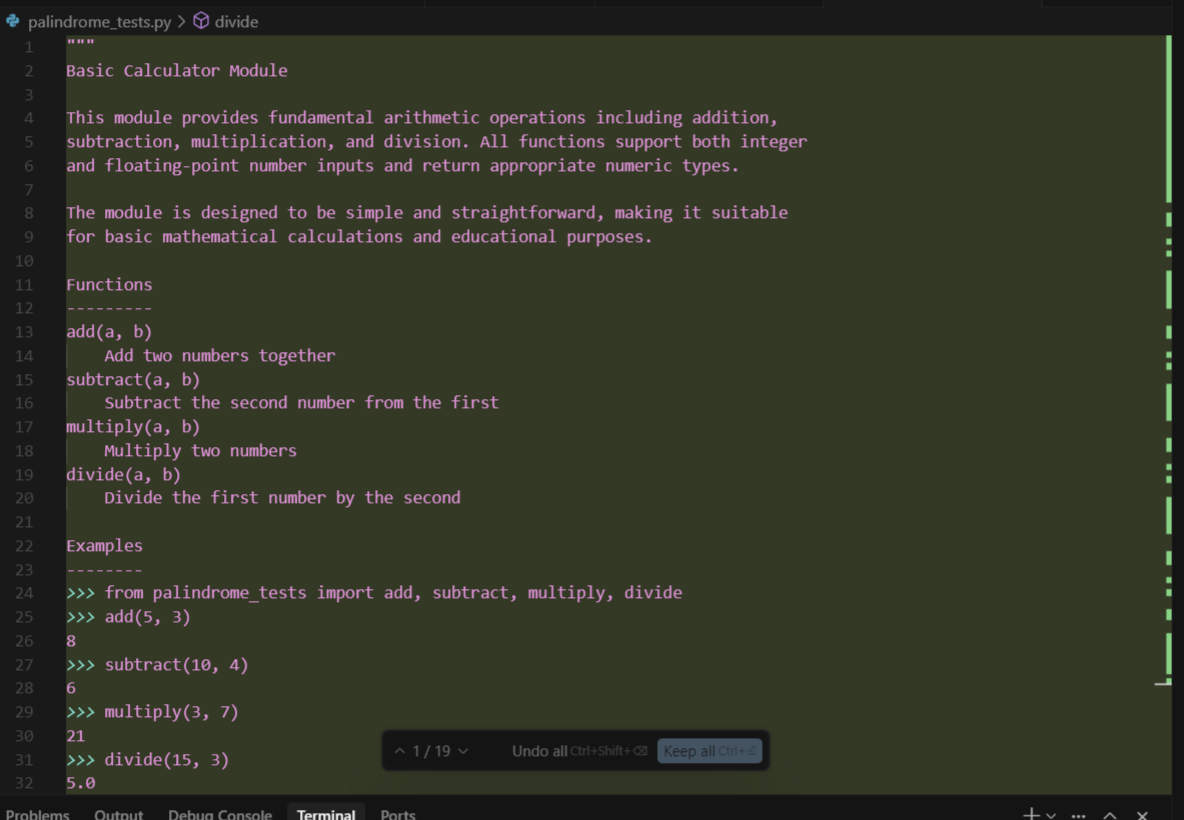
AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Now, the given prompt to the AI: A black background with white text

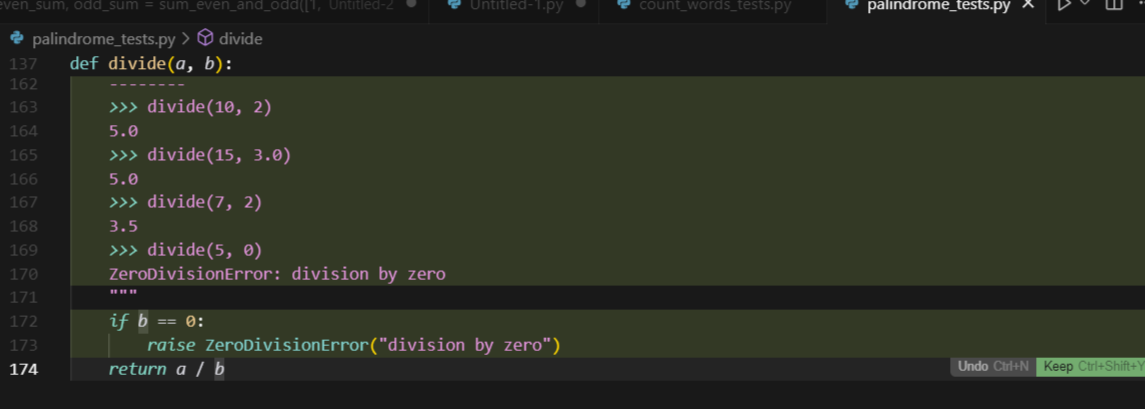
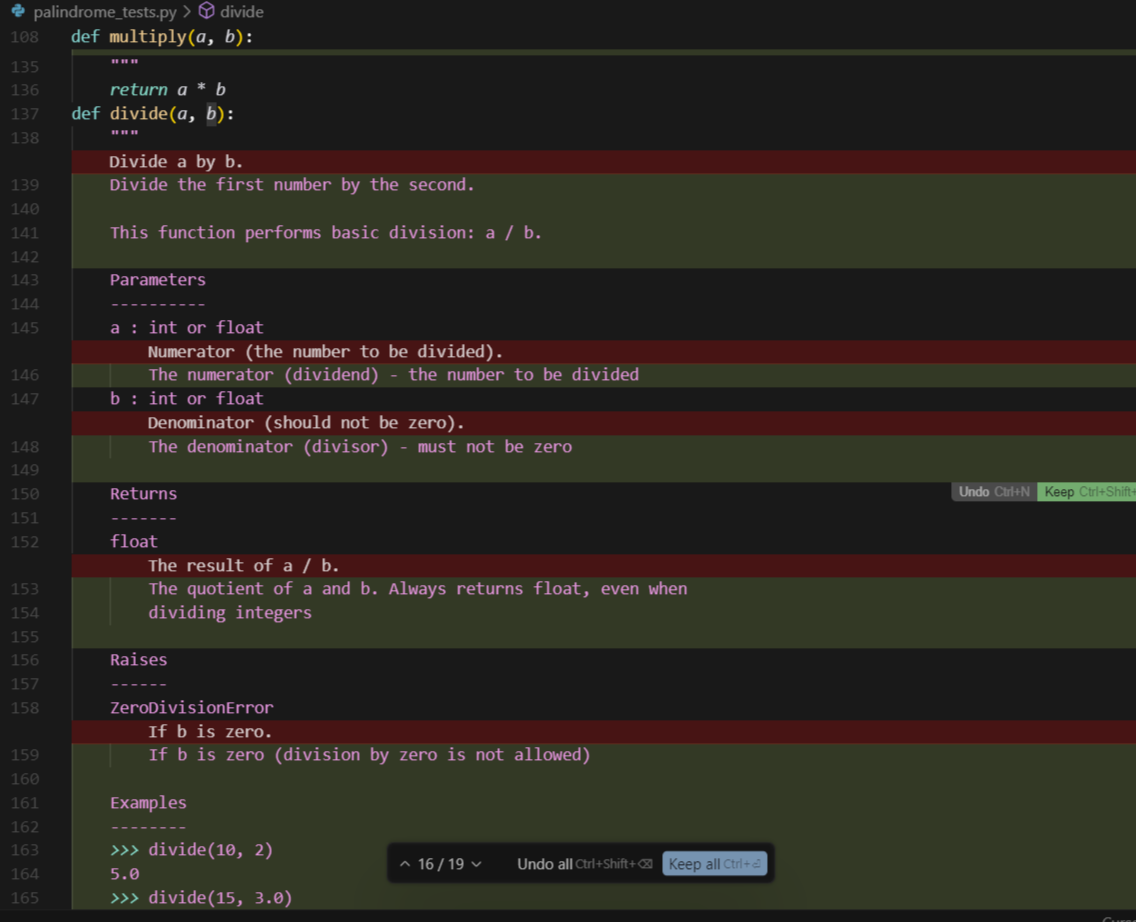
AI-generated content may be incorrect.**

**Given code by AI:  
**

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.



**Comparison**:

**AI-Generated Docstring:**

* Very formal and polished.
* Uses full sentences like *“This module provides basic arithmetic operations…”*.
* Adds sections like *Functions* with descriptions.
* Looks professional but can feel a bit too “perfect” or lengthy.

**Manually Written Docstring:**

* Shorter and to the point.
* Uses simple wording like *“Add two numbers”*, *“Subtract b from a”*.
* Easier to read and less formal.
* More in line with how a student would normally explain code.