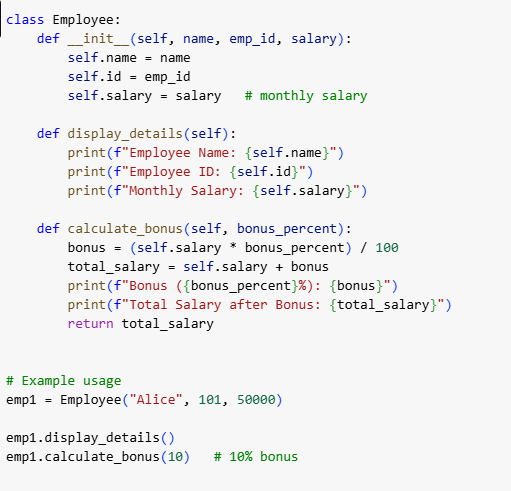
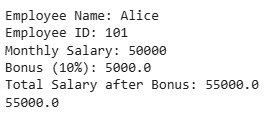
**AI Lab Assignment 6.1**

Task Description #1

Use AI to create an Employee class with attributes (name,id, salary) and a method to calculate yearly salary



Output:

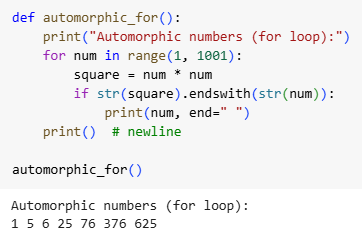


Observation:

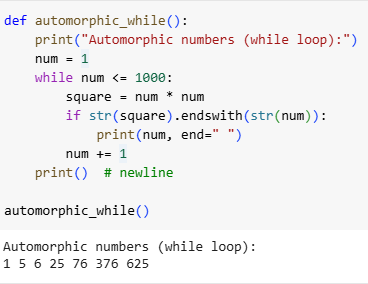
The basic code of the Employee class correctly calculated the yearly salary using the given monthly salary, but it was limited as it did not display details or handle bonuses. In the updated version, the class was improved with methods to display employee details and calculate bonuses, making it more complete and functional.

Task Description #2

Task: Prompt AI to generate a function that displays all Automorphic numbers between 1 and 1000 using a for loop.



using a while loop:



Observation:

For loop version:

More concise.

Directly iterates over the range of numbers.

Preferred for a fixed range (1–1000).

While loop version:

More flexible (you can change the stopping condition dynamically).

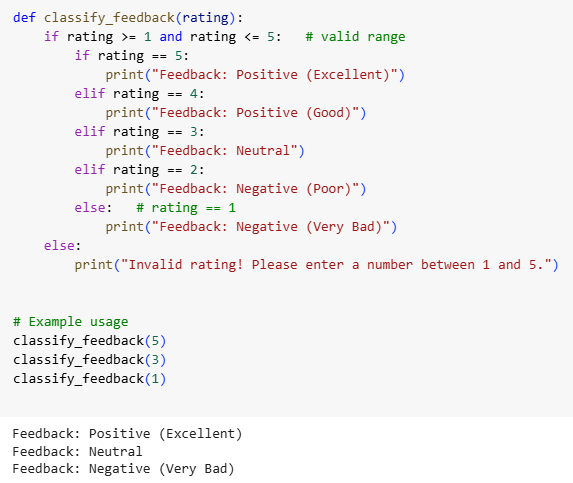
Slightly longer since you manually update num += 1.

Useful when you don’t know the exact range in advance.

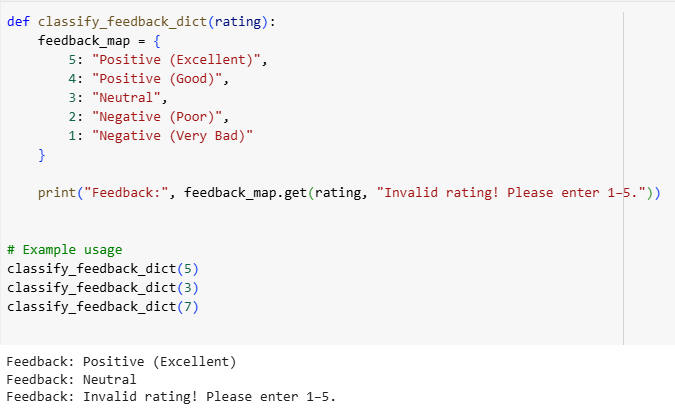
Task Description #3

Task: Ask AI to write nested if-elif-else conditions to classify online shopping feedback as Positive, Neutral, or Negative based on a numerical rating (1–5).

if-elif-else conditions:



Dictionary-based:

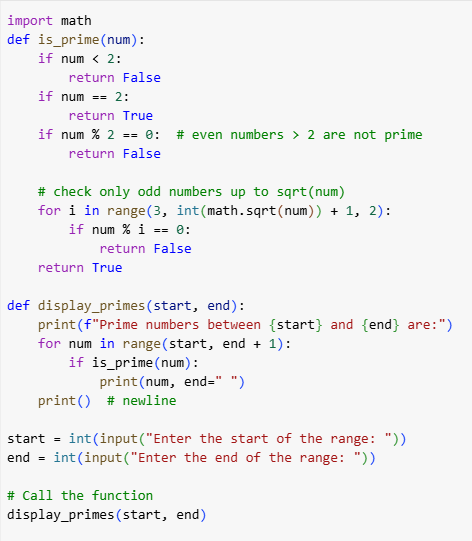


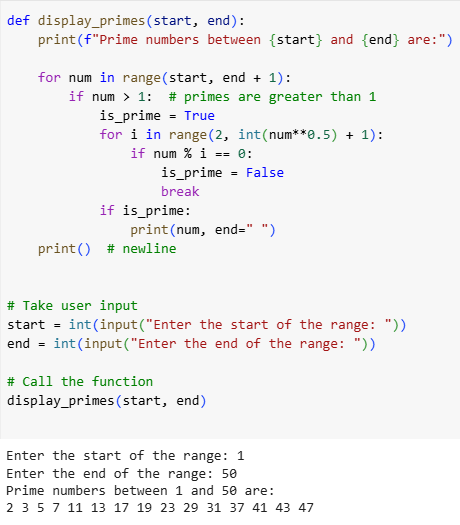
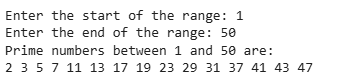
Observation:

The feedback classification function using nested if-elif-else correctly maps numerical ratings (1–5) to categories: **Positive (4–5), Neutral (3), Negative (1–2)**, and handles invalid ratings appropriately. The nested structure is easy to understand for small rating scales but contains some repetition, making it slightly verbose.

Task Description #4

Task: Generate a function using AI that displays all prime numbers within a user-specified range (e.g., 1 to 500).

 Optimized:



Observation:

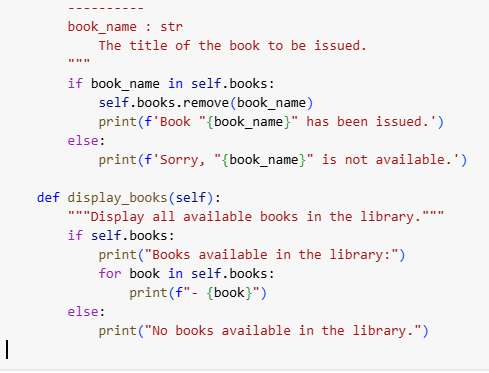
**Square root method** reduces checks:

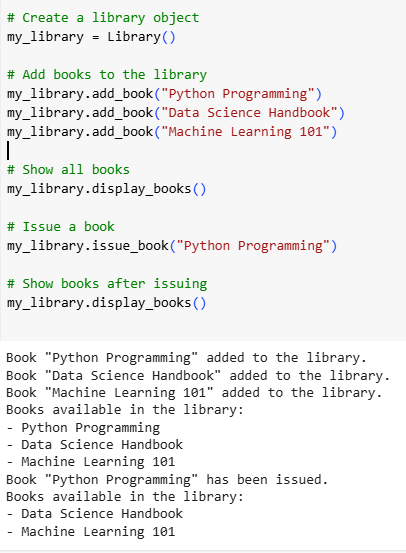
If a number n is not divisible by any number up to √n, then it is prime.

This cuts down checks dramatically.

Task Description #5

Task: Use AI to build a Library class with methods to add\_book(), issue\_book(), and display\_books().





Observation:

The Library class successfully manages a collection of books with methods to add, issue, and display books. The add\_book() method correctly adds new books, while issue\_book() properly handles issuing available books and gracefully manages cases where a book is unavailable. The display\_books() method shows all current books and appropriately handles an empty library. Inline comments and documentation make the class easy to understand and maintain. Overall, the class demonstrates correct functionality, good readability, and effective handling of edge cases.