**NAME: P.Heman                                                                                            ROLL NO: 2403A510F5**

**LAB ASSIGNMENT 2.1:**

**TASK 1:**

**Code:**

A screenshot of a computer program

AI-generated content may be incorrect.

A black rectangle with white text

AI-generated content may be incorrect.

**OUTPUT:**

A black background with white text

AI-generated content may be incorrect.

**OBSERVATION:**

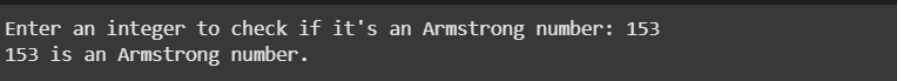
Observation: The code correctly handles empty input and invalid values, making it user-friendly. It also demonstrates how built-in functions simplify finding mean, min, and max

**ww**

This Python code defines a function calculate\_stats to find the average (mean), smallest (minimum), and largest (maximum) numbers in a list. It first checks if the list is empty; if so, it returns nothing. Otherwise, it calculates the sum and length to find the mean, and uses built-in functions min() and max() for the minimum and maximum. The code then prompts the user to enter numbers separated by spaces. It attempts to convert the input string into a list of floating-point numbers, handling potential errors if non-numeric input is given. Finally, it calls the calculate\_stats function with the user's list and prints the resulting mean, minimum, and maximum values, or indicates if no valid numbers were entered.

**TASK 2:**

**OUTPUT OF THE GEMINI:**



**OUTPUT OF THE CO PILOT:**



Side by Side Comparision:

A screenshot of a computer

AI-generated content may be incorrect.

**TASK 3:**

**THE FOLLOWING IS THE GEMINI EXPLANATION LINE BY LINE:**

Let's go through the is\_prime(n) function step by step:

* **def is\_prime(n):**: This line defines a function named is\_prime that takes one argument, n.
* **"""Checks if a number is prime."""**: This is a docstring, which explains what the function does.
* **if n <= 1:**: This checks if the input number n is less than or equal to 1.
* **return False # Numbers less than or equal to 1 are not prime**: If n is less than or equal to 1, the function immediately returns False because prime numbers are defined as being greater than 1. The comment clarifies why.
* **`for i in range(2, int(n**0.5) + 1):\*\*: This line starts aforloop. It iterates through a range of numbers starting from 2 up to the integer part of the square root ofn, plus 1. We only need to check for divisors up to the square root ofnbecause if a numbern` has a divisor greater than its square root, it must also have a divisor smaller than its square root.
* **if n % i == 0:**: Inside the loop, this line checks if n is divisible by the current number i using the modulo operator (%). If the remainder of the division is 0, it means n is divisible by i.
* **return False # If divisible by any number in this range, it's not prime**: If n is divisible by any number i in the specified range, it means n is not a prime number, so the function returns False. The comment explains this.
* **return True # If not divisible by any number in the range, it's prime**: If the loop finishes without finding any divisors (i.e., the function hasn't returned False yet), it means n is not divisible by any number in the range, and therefore it must be a prime number. The function then returns True. The comment clarifies this.

**TASK 4**:

**CURSOR SETUP**:   
A screenshot of a computer

AI-generated content may be incorrect.

**THIS GIVEN CODE IS FROM CURSOR**:

A screenshot of a computer program

AI-generated content may be incorrect.

**OUTPUT:**

A black background with white letters and numbers

AI-generated content may be incorrect.

**OBSERVATION** :

The cursor setup and code demonstrate how AI-based coding assistants can generate functioning code quickly, but human verification is still needed to ensure correctness.

**TASK 5:**

**TASK 5:**

A screen shot of a computer program

AI-generated content may be incorrect.

**OUTPUT:**

A black background with white text

AI-generated content may be incorrect.