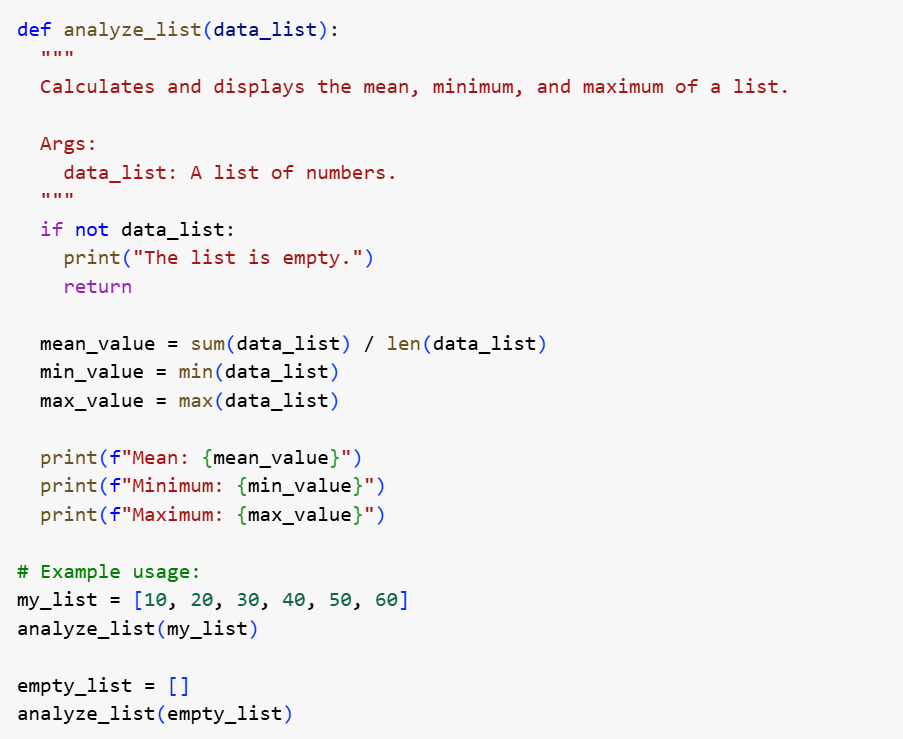
**LAB ASSIGNMENT-2**

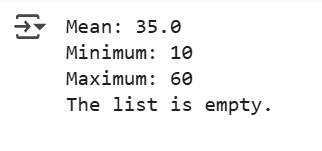
**TASK 1**: Use Google Gemini in Colab to write a Python function that reads  
a list of numbers and calculates the mean, minimum, and  
maximum values.

**PROMPT**: Write a function in python to display mean,minimum,maximum from the given list

**CODE:**

****

**OUTPUT:**



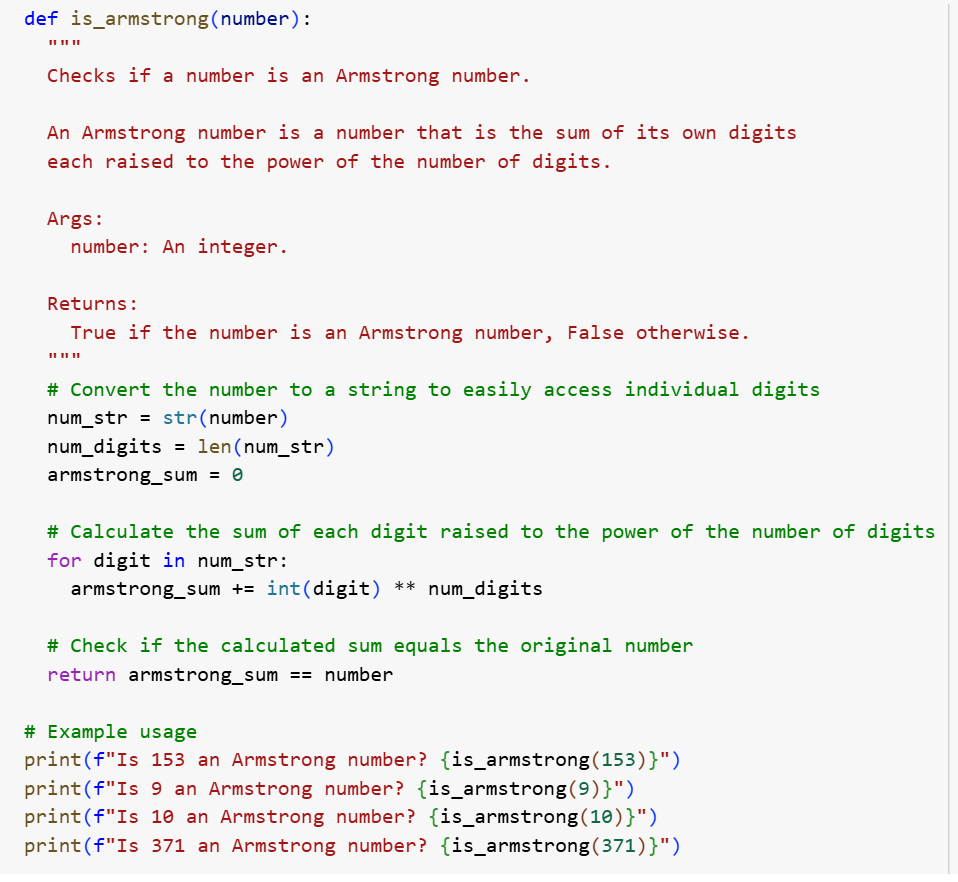
**EXPLANATION**:

Certainly! The provided Python code defines a function analyze\_list that takes a list of numbers and calculates its mean, minimum, and maximum values. It first checks if the list is empty, printing a message if it is. Otherwise, it computes the sum and length of the list to find the mean, and uses built-in functions min() and max() to find the smallest and largest values, respectively. Finally, it prints these calculated statistics in a formatted output. The code then demonstrates the usage of this function with both a sample list and an empty list.

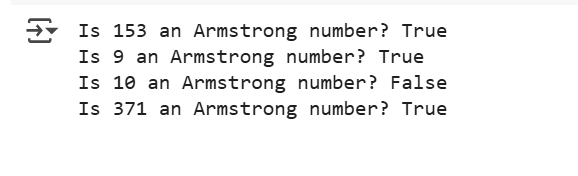
**TASK-2**: Compare Gemini and Copilot outputs for a Python function that checks whether a number is an Armstrong number. Document the steps, prompts, and outputs

**PROMPT**: write a function in python which checks whether a number is an Armstrong number

**CODE**:



**OUTPUT:**

** EXPLANATION:**

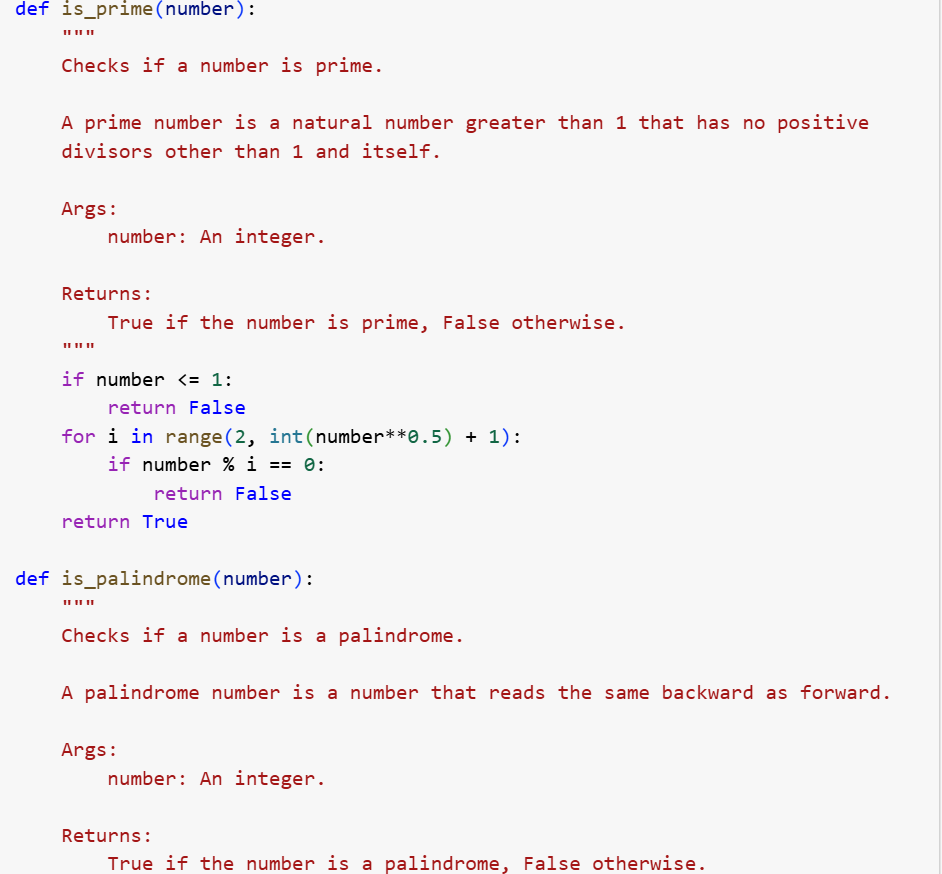
Certainly. The Python code you see defines a function called is\_armstrong that determines if a given integer is an Armstrong number. It works by converting the input number to a string to count its digits and then iterating through each digit. For every digit, it calculates the digit raised to the power of the total number of digits and adds this result to a running sum. Finally, it compares this calculated sum to the original number. If they are equal, the function returns True, indicating it's an Armstrong number; otherwise, it returns False. The code also includes example calls to the function with different numbers to demonstrate its usage and output.

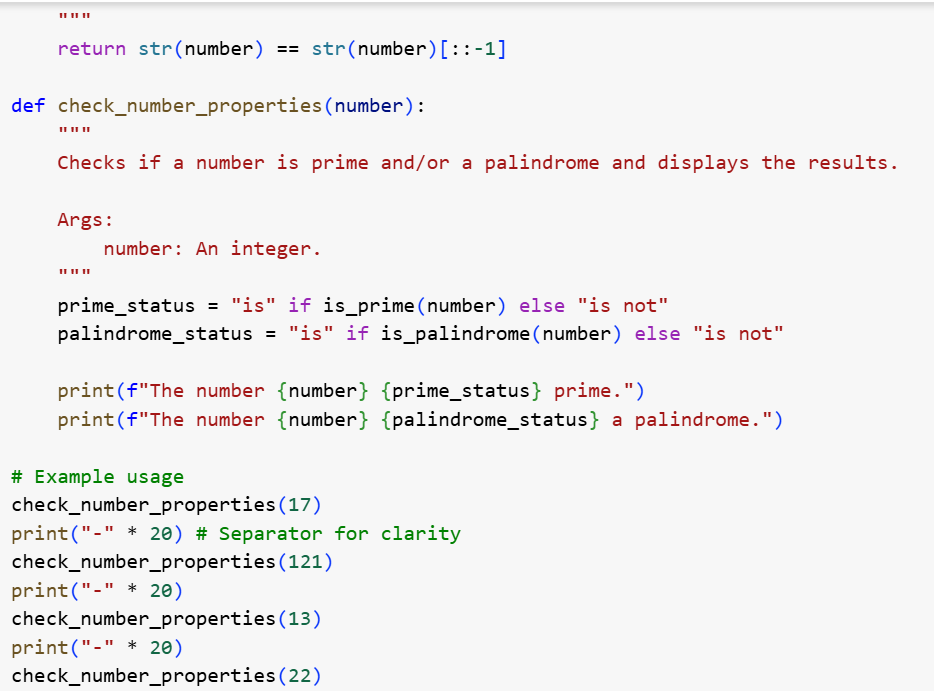
**TASK-3:** Ask Gemini to explain a Python function (e.g., is\_prime(n) or is\_palindrome(s)) line by line.

Choose either a prime-checking or palindrome-checking function and document the explanation provided by Gemini.

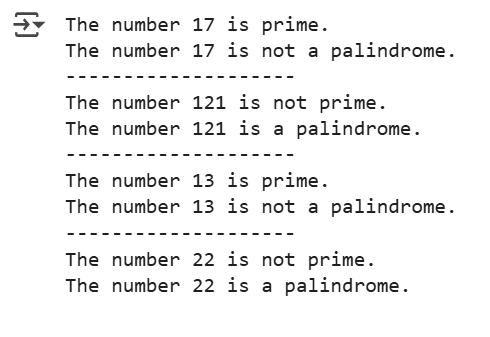
**PROMPT:** write a function in python which checks the number is prime or not using is\_prime() and also checks if the number is palindrome or not using is\_palindrome() and display the output

**CODE:**





**OUTPUT:**

****

**EXPLANATION:**

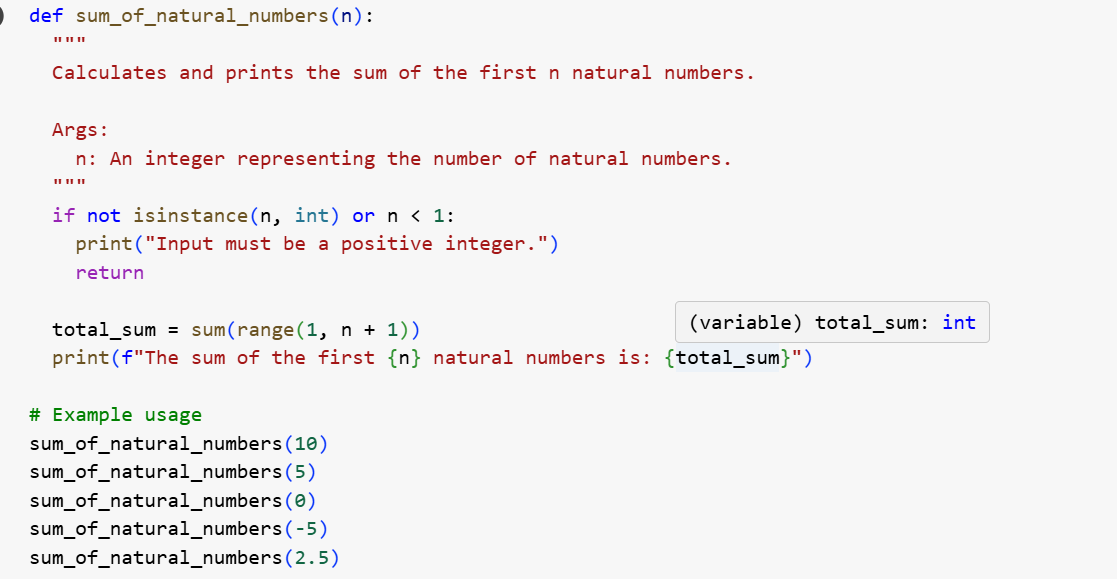
# Certainly. The code you provided defines three Python functions: is\_prime, is\_palindrome, and check\_number\_properties. The is\_prime function checks if a given number is a prime number by iterating from 2 up to the square root of the number and checking for divisibility. The is\_palindrome function determines if a number reads the same forwards and backward by converting it to a string and comparing it to its reverse. The main function, check\_number\_properties, takes an integer as input, calls both is\_prime and is\_palindrome on it, and then prints whether the number is prime and whether it is a palindrome based on the results of these checks. The code also includes example calls to check\_number\_properties to demonstrate its usage with different numbers.

**TASK-4:** Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of the first N natural numbers) and test its output.

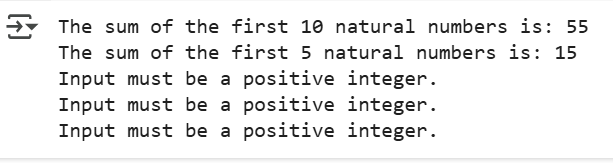
Optionally, compare Cursor AI’s generated code with Gemini’s output

**PROMPT:** write a function python which takes input and prints the sum of first n natural numbers

**CODE:**



**OUTPUT:**

****

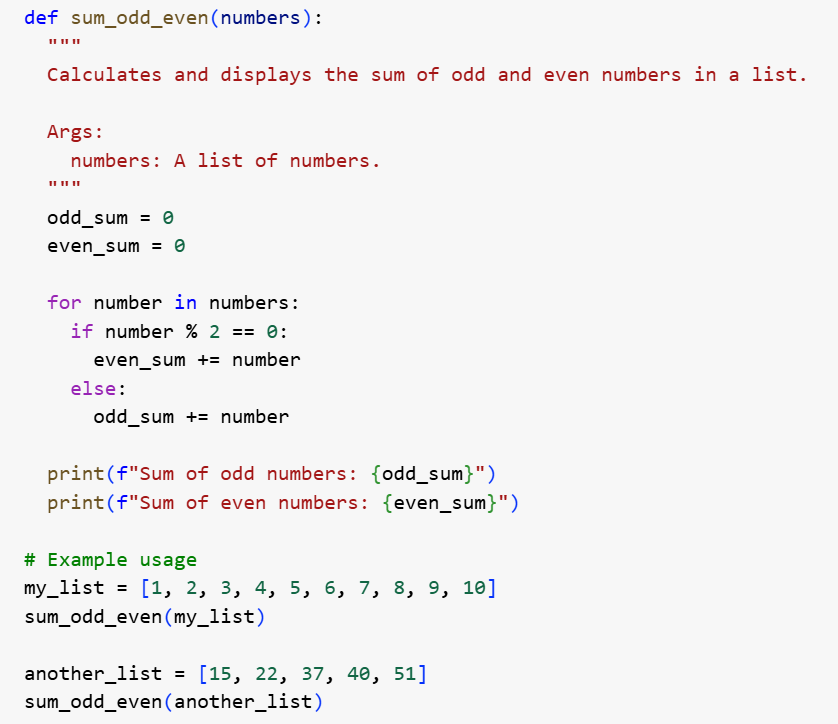
**EXPLANATION:**

Certainly. The Python code in cell 6bda51f7 defines a function called sum\_of\_natural\_numbers that takes a single integer argument n. This function first checks if the input n is a positive integer; if not, it prints an error message and exits. If n is a positive integer, it calculates the sum of all natural numbers from 1 up to and including n using the built-in sum() function with a range() object. Finally, it prints the calculated sum in a formatted string. The code also includes example calls to demonstrate the function's behavior with valid and invalid inputs.

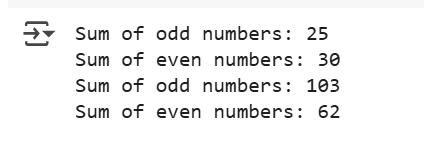
* **TASK-5:** Students need to write a Python program to calculate the sum of odd numbers and even numbers in a given tuple.
* Refactor the code to improve logic and readability.

**PROMPT:** write a python program to calculate sum of odd and even numbers separated and display it

**CODE:**



**OUTPUT:**

****

**EXPLANATION:**

Certainly. The Python code in cell bd28972f defines a function called sum\_odd\_even that takes a list of numbers as input. It initializes two variables, odd\_sum and even\_sum, to zero. The function then iterates through each number in the input list. Inside the loop, it checks if the current number is even using the modulo operator (%). If the number divided by 2 has a remainder of 0, it's even, and its value is added to even\_sum; otherwise, it's odd, and its value is added to odd\_sum. After processing all numbers in the list, the function prints the final calculated sum of odd numbers and the sum of even numbers. The code also includes example calls to the function with different lists to demonstrate its usage.