

# AI ASSISTANT CODING ASSIGNMENT - 2

---

**NAME:** Deekshith Kowdagani

**HT.NO:** 2303A51414

**BATCH:** 21

---

## **LAB 2:**

**Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab)  
and Cursor AI**

### **Task 1: Cleaning Sensor Data**

#### **❖ Scenario:**

❖ You are cleaning IoT sensor data where negative values are invalid.

#### **❖ Task:**

Use Gemini in Colab to generate a function that filters out all negative numbers from a list.

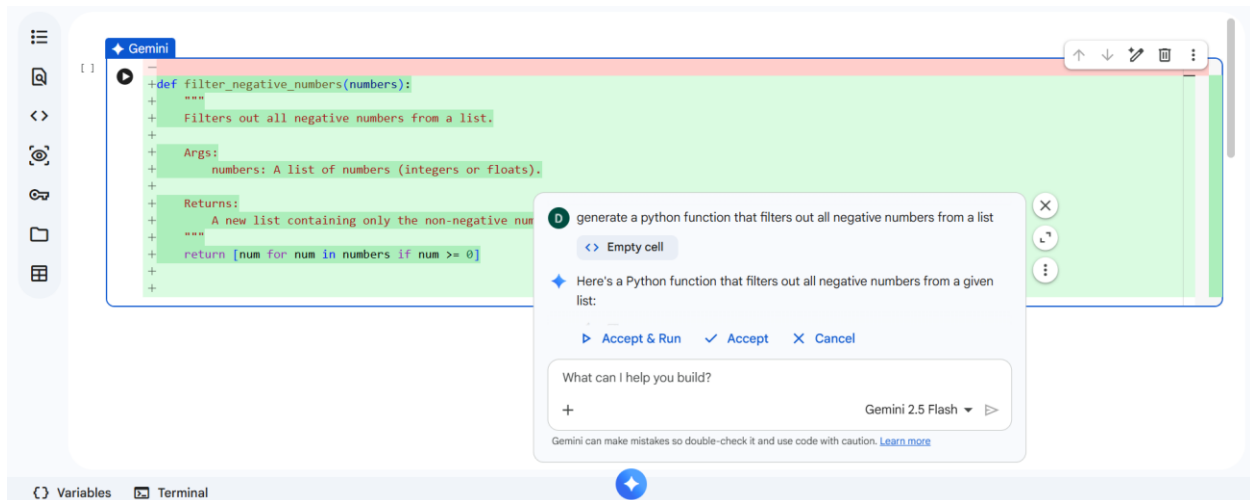
#### **❖ Expected Output:**

➤ Before/after list

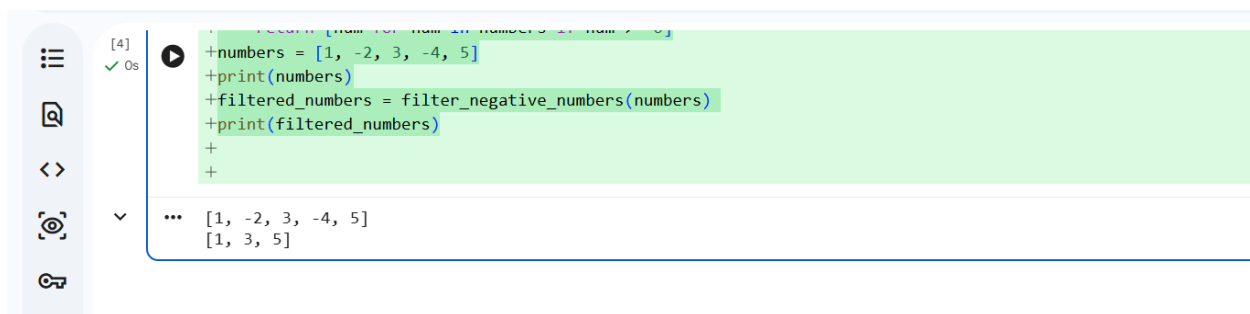
➤ Screenshot of Colab execution

---

## CODE :



## OUTPUT:



## Task 2: String Character Analysis

### ❖ Scenario:

You are building a text-analysis feature.

### ❖ Task:

Use Gemini to generate a Python function that counts vowels, consonants, and digits in a string.

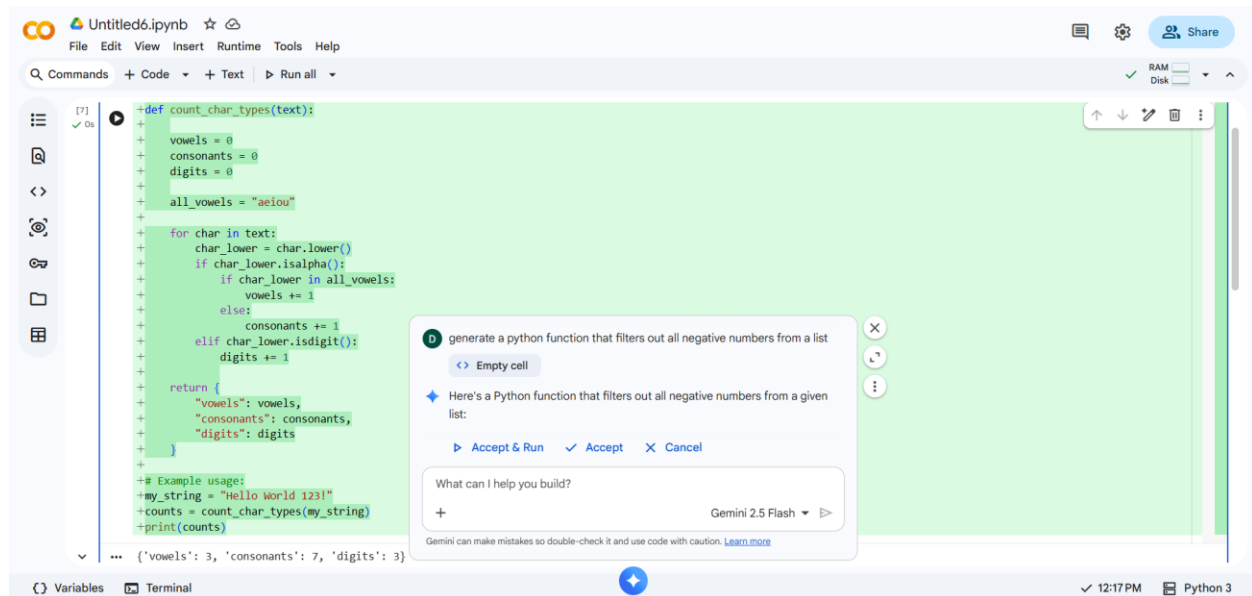
## ❖ Expected Output:

### ➤ Working function

### ➤ Sample inputs and outputs

---

## CODE :

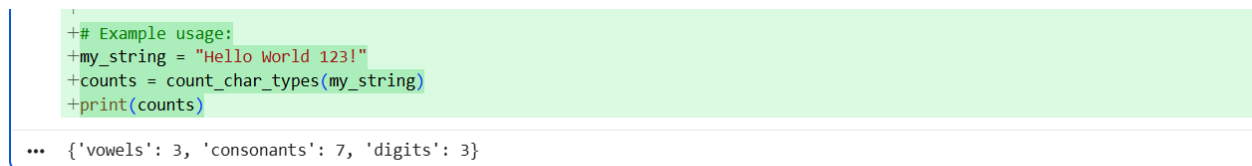


The screenshot shows a Jupyter Notebook titled 'Untitled6.ipynb'. The code in the cell defines a function `count_char_types(text)` that counts vowels, consonants, and digits in a string. The function initializes counters to 0 and a string of vowels 'aeiou'. It iterates through each character in the text, converting it to lowercase. If it's an alphabet, it checks if it's a vowel or consonant. If it's a digit, it increments the digit counter. Finally, it returns a dictionary with the counts. An example usage is provided: `my_string = "Hello world 123!"`, `counts = count_char_types(my_string)`, and `print(counts)`. The output of the cell is `{'vowels': 3, 'consonants': 7, 'digits': 3}`. A Gemini AI chat window is overlaid on the right, showing a prompt to generate a Python function for filtering negative numbers and a response providing a function for that purpose.

```
[?] Untitled6.ipynb ☆ ☁  
File Edit View Insert Runtime Tools Help  
Q Commands + Code + Text ▶ Run all  
[?] 0s  
def count_char_types(text):  
+  
+ vowels = 0  
+ consonants = 0  
+ digits = 0  
+  
+ all_vowels = "aeiou"  
+  
+ for char in text:  
+     char_lower = char.lower()  
+     if char_lower.isalpha():  
+         if char_lower in all_vowels:  
+             vowels += 1  
+         else:  
+             consonants += 1  
+     elif char_lower.isdigit():  
+         digits += 1  
+  
+ return {  
+     "vowels": vowels,  
+     "consonants": consonants,  
+     "digits": digits  
+ }  
+  
+ # Example usage:  
+ my_string = "Hello world 123!"  
+ counts = count_char_types(my_string)  
+ print(counts)  
... {'vowels': 3, 'consonants': 7, 'digits': 3}  
{ } Variables Terminal  
12:17 PM Python 3
```

generate a python function that filters out all negative numbers from a list  
<> Empty cell  
Here's a Python function that filters out all negative numbers from a given list:  
Accept & Run ✓ Accept ✕ Cancel  
What can I help you build?  
+ Gemini 2.5 Flash ▶  
Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

## OUTPUT:



The screenshot shows the output of the Python code, which is a dictionary: `{'vowels': 3, 'consonants': 7, 'digits': 3}`.

```
+ # Example usage:  
+ my_string = "Hello world 123!"  
+ counts = count_char_types(my_string)  
+ print(counts)  
... {'vowels': 3, 'consonants': 7, 'digits': 3}
```

## Task 3: Palindrome Check – Tool Comparison

### ❖ Scenario:

You must decide which AI tool is clearer for string logic.

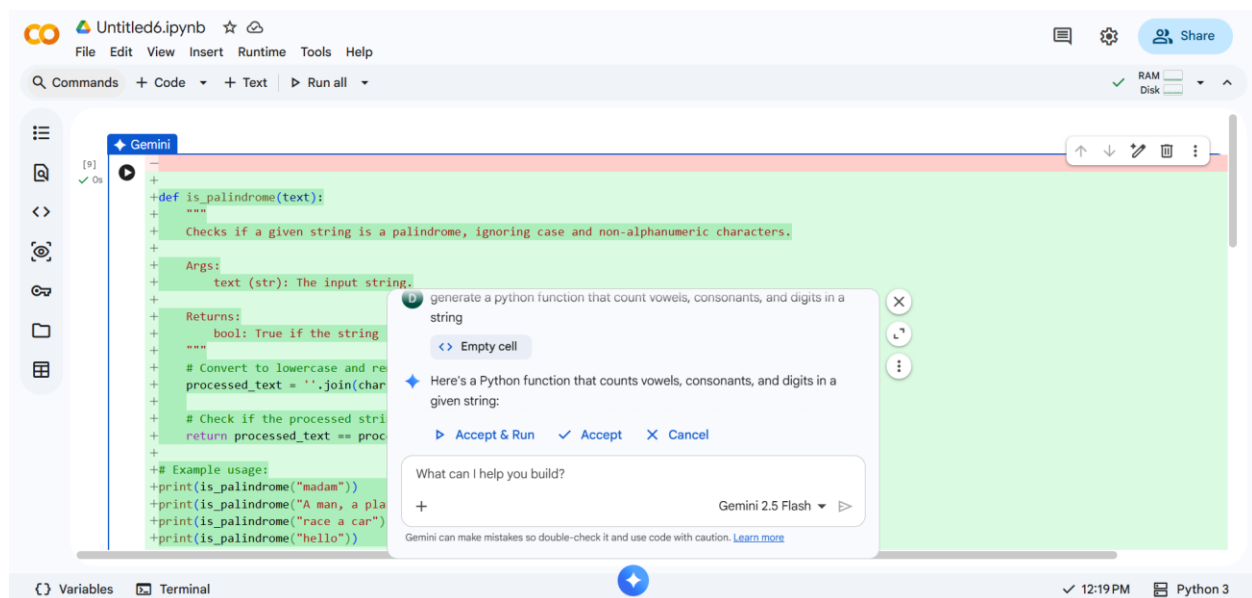
## ❖ Task:

Generate a palindrome-checking function using Gemini and Copilot, then compare the results.

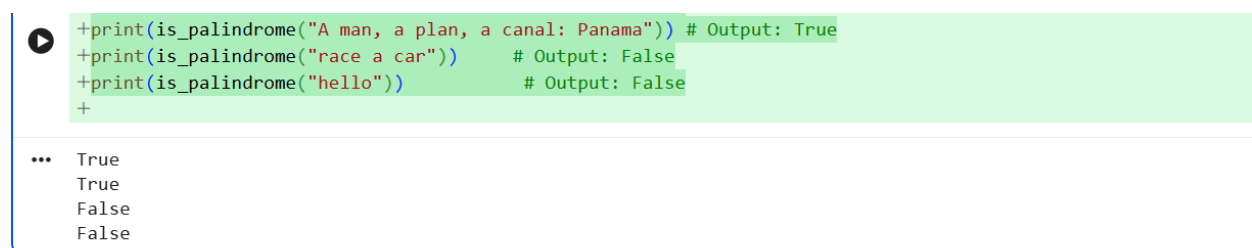
## ❖ Expected Output:

- Side-by-side code comparison
- Observations on clarity and structure

## CODE:



## OUTPUT:



## Task 4: Code Explanation Using AI

### ❖ Scenario:

You are reviewing unfamiliar code written by another developer.

❖ Task:

Ask Gemini to explain a Python function (prime check OR palindrome check) line by line.

❖ Expected Output:

➤ Code snippet

➤ AI explanation

➤ Student comments on understanding

---

OUTPUT:

The screenshot shows a Gemini AI interface. At the top, a prompt says "generate a python function that filters out all negative numbers from a list". Below this is a code editor with a blue "Empty cell" button. A blue diamond icon indicates a generated response: "Here's a Python function that filters out all negative numbers from a given list:". Below the text are three buttons: "Accept & Run", "Accept", and "Cancel". A text input field contains "What can I help you build?". To the right of the input field is a dropdown menu set to "Gemini 2.5 Flash" and a "Send" button. At the bottom, a disclaimer reads: "Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)".

1. `def is_palindrome(text):`  
    ◦ This line defines a Python function named `is_palindrome` that accepts one argument: `text` (which is expected to be a string).

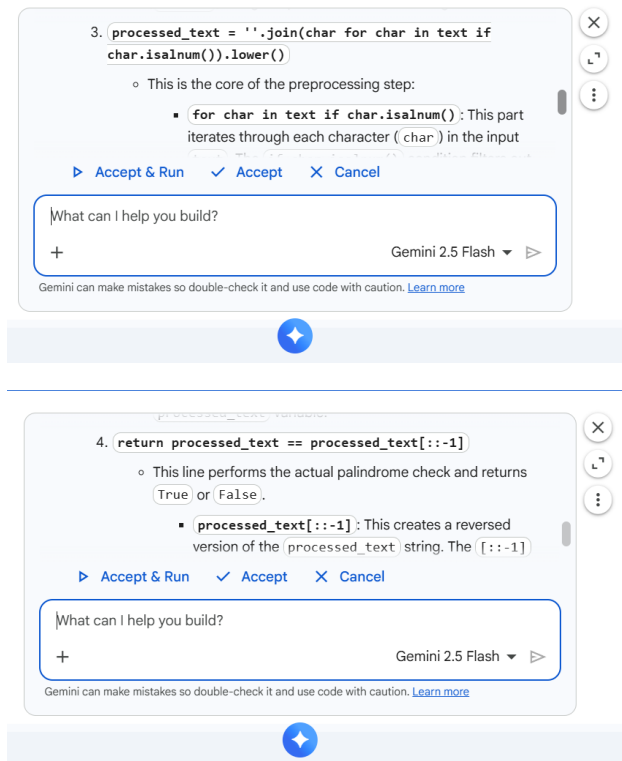
2. `"""..."""`

Buttons: Accept & Run, Accept, Cancel

What can I help you build?

+ Gemini 2.5 Flash Send

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)



### My own experience using both Gemini and GitHub Copilot:

While using Gemini in Google Colab, I found the explanations to be very clear and helpful in understanding the logic behind the code. Gemini was especially useful for learning and analyzing Python programs step by step. GitHub Copilot, was faster in generating code directly inside the editor and helped me complete tasks quickly. Copilot felt more suitable for continuous coding, while Gemini was better for conceptual clarity. Overall, using both tools together improved my coding efficiency and understanding.