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.

```python
def clean_dynamic_data():
    # Prompt user for input
    user_input = input("Enter sensor readings separated by commas (e.g., 23, -5, 10, -1): ")

    try:
        # Convert the string input into a list of floats
        raw_data = [float(x.strip()) for x in user_input.split(',')]

        # Filter out negatives
        cleaned_data = [val for val in raw_data if val >= 0]

        print(f"\n✅ Processed {len(raw_data)} readings.")
        print(f"📊 Cleaned Data: {cleaned_data}")

    except ValueError:
        print("❌ Error: Please ensure you only enter numbers and commas.")

clean_dynamic_data()
```

```
Enter sensor readings separated by commas (e.g., 23, -5, 10, -1): 23,-10,220.5,33.0,-250,362,-56,-70

✅ Processed 8 readings.
📊 Cleaned Data: [23.0, 220.5, 33.0, 362.0]
```

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.

```python
def analyze_string(text):
    """
    Counts vowels, consonants, and digits in a given string.
    """
    vowels_set = "aeiouAEIOU"
    counts = {
        "vowels": 0,
        "consonants": 0,
        "digits": 0
    }

    for char in text:
        if char.isdigit():
            counts["digits"] += 1
        elif char.isalpha():
            if char in vowels_set:
                counts["vowels"] += 1
            else:
                counts["consonants"] += 1

    return counts

# Dynamic Input for Colab
user_text = input("Enter a string to analyze: ")
results = analyze_string(user_text)

print("-" * 20)
print(f"Analysis for: '{user_text}'")
print(f"Vowels: {results['vowels']}")
print(f"Consonants: {results['consonants']}")
print(f"Digits: {results['digits']}")
```

```
Enter a string to analyze: bhanuprasadbelamkonda
--------------------
Analysis for: 'bhanuprasadbelamkonda'
Vowels: 8
Consonants: 13
Digits: 0
```

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.

```python
def is_palindrome(text: str) -> bool:
    # Normalize: remove non-alphanumeric and lowercase
    cleaned = ""
    for ch in text:
        if ch.isalnum():
            cleaned += ch.lower()

    # Compare with reversed string
    reversed_text = cleaned[::-1]
    return cleaned == reversed_text
print(is_palindrome("Madam, I'm Adam"))
```

```
True
```

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.

```python
import math

def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True
print(is_prime(29))
```

```
True
```