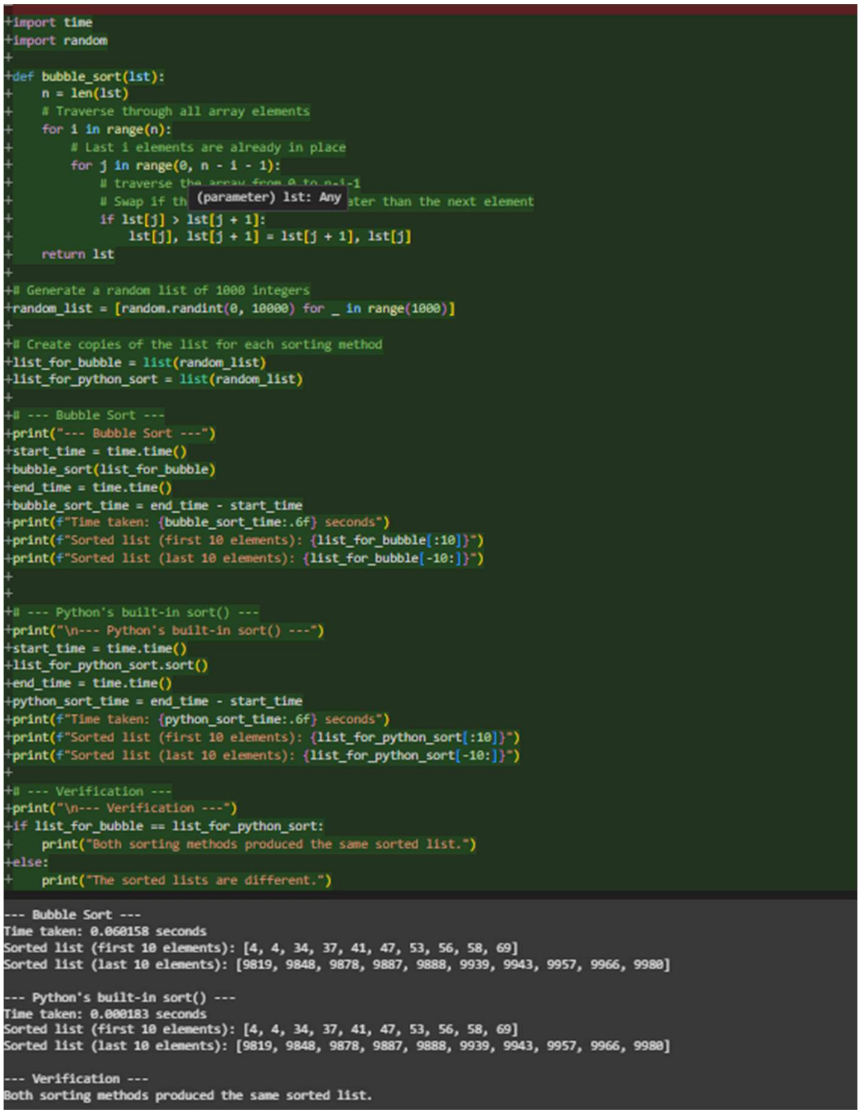
LAB NAME : AI ASSISTED CODING
LAB NUMBER :02
ROLL NO :2503A51L31
BRANCH : CSE
NAME : CH. ABISHEK

TASK 1

TASK DESCRIPTION: Open Google Colab and use Google Gemini to generate Python code that performs sorting of a list using both the bubble sort algorithm and Python’s built-in sort() function. Compare the two implementations.

PROMPT: Generate Python code that sorts a list of integers using two methods: (1) the bubble sort algorithm implemented manually, and (2) Python’s built-in `sort()` function. Then compare the performance and output of both methods using a randomly generated list of 1000 integers. Include timing analysis using the `time` module and print the sorted results to verify correctness.

CODE:



```
+import time
+import random
+
+def bubble_sort(lst):
+    n = len(lst)
+    # Traverse through all array elements
+    for i in range(n):
+        # Last i elements are already in place
+        for j in range(0, n - i - 1):
+            # traverse the array from 0 to n-i-1
+            # Swap if th (parameter) lst: Any ater than the next element
+            if lst[j] > lst[j + 1]:
+                lst[j], lst[j + 1] = lst[j + 1], lst[j]
+    return lst
+
+# Generate a random list of 1000 integers
+random_list = [random.randint(0, 10000) for _ in range(1000)]
+
+# Create copies of the list for each sorting method
+list_for_bubble = list(random_list)
+list_for_python_sort = list(random_list)
+
+# --- Bubble Sort ---
+print("--- Bubble Sort ---")
+start_time = time.time()
+bubble_sort(list_for_bubble)
+end_time = time.time()
+bubble_sort_time = end_time - start_time
+print(f"Time taken: {bubble_sort_time:.6f} seconds")
+print(f"Sorted list (first 10 elements): {list_for_bubble[:10]}")
+print(f"Sorted list (last 10 elements): {list_for_bubble[-10:]}")
+
+
+# --- Python's built-in sort() ---
+print("\n--- Python's built-in sort() ---")
+start_time = time.time()
+list_for_python_sort.sort()
+end_time = time.time()
+python_sort_time = end_time - start_time
+print(f"Time taken: {python_sort_time:.6f} seconds")
+print(f"Sorted list (first 10 elements): {list_for_python_sort[:10]}")
+print(f"Sorted list (last 10 elements): {list_for_python_sort[-10:]}")
+
+# --- Verification ---
+print("\n--- Verification ---")
+if list_for_bubble == list_for_python_sort:
+    print("Both sorting methods produced the same sorted list.")
+else:
+    print("The sorted lists are different.")
```

```
--- Bubble Sort ---
Time taken: 0.060158 seconds
Sorted list (first 10 elements): [4, 4, 34, 37, 41, 47, 53, 56, 58, 69]
Sorted list (last 10 elements): [9819, 9848, 9878, 9887, 9888, 9939, 9943, 9957, 9966, 9980]

--- Python's built-in sort() ---
Time taken: 0.000183 seconds
Sorted list (first 10 elements): [4, 4, 34, 37, 41, 47, 53, 56, 58, 69]
Sorted list (last 10 elements): [9819, 9848, 9878, 9887, 9888, 9939, 9943, 9957, 9966, 9980]

--- Verification ---
Both sorting methods produced the same sorted list.
```

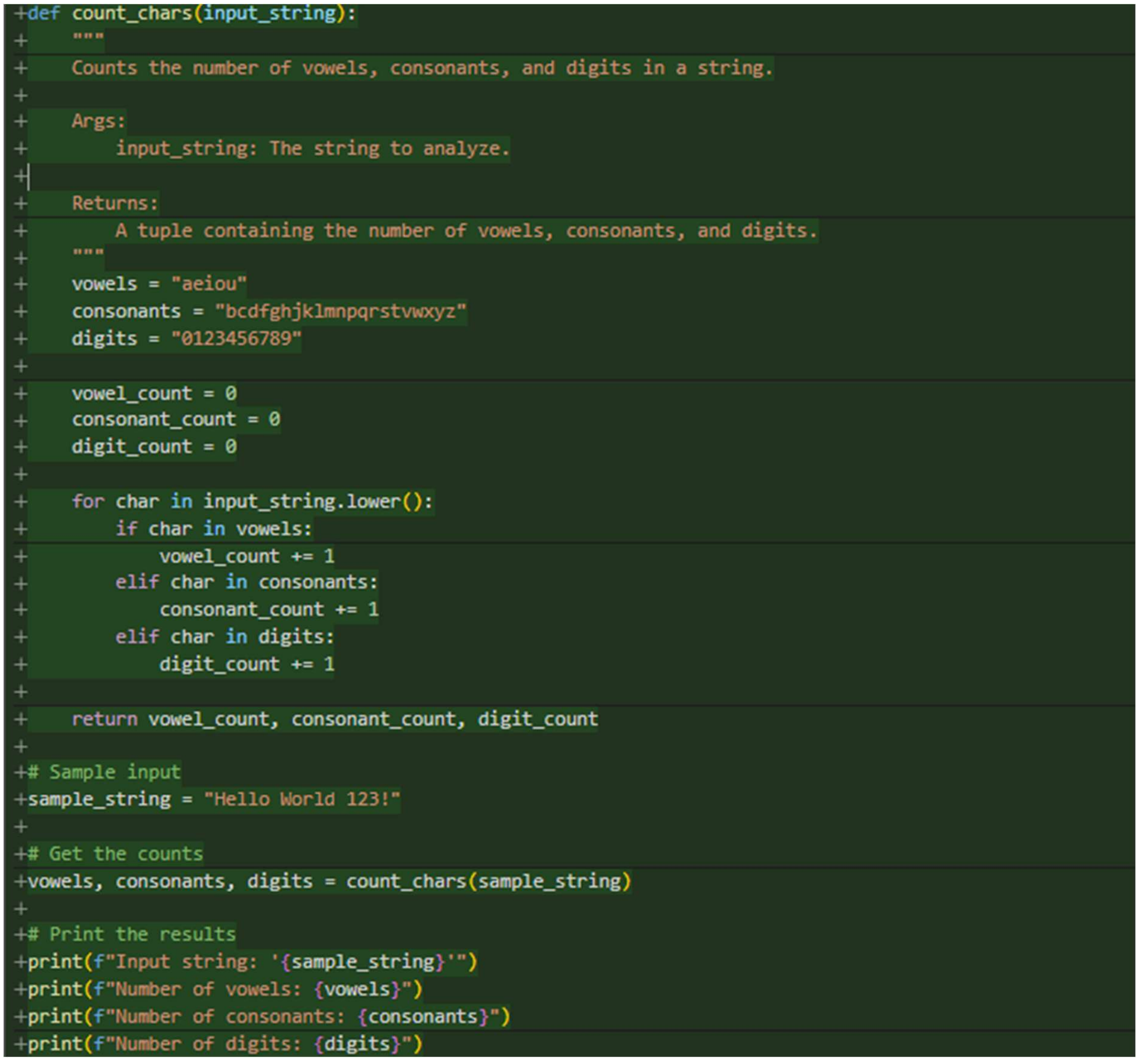
TASK 2

TASK DESCRIPTION: In Colab, use Google Gemini to generate a Python function that takes a string and returns:

The number of vowels, The number of consonants, The number of digits in the string

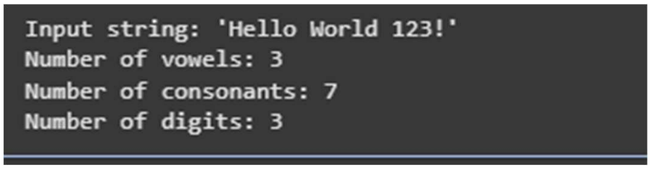
PROMPT: Write a Python function that takes a string as input and returns three values: the number of vowels, the number of consonants, and the number of digits in the string. Ignore spaces and special characters. Make sure the function handles both uppercase and lowercase letters. Include a sample input and output to demonstrate the function

CODE



```
+def count_chars(input_string):
+    """
+    Counts the number of vowels, consonants, and digits in a string.
+
+    Args:
+        input_string: The string to analyze.
+
+    Returns:
+        A tuple containing the number of vowels, consonants, and digits.
+    """
+    vowels = "aeiou"
+    consonants = "bcdfghjklmnpqrstvwxyz"
+    digits = "0123456789"
+
+    vowel_count = 0
+    consonant_count = 0
+    digit_count = 0
+
+    for char in input_string.lower():
+        if char in vowels:
+            vowel_count += 1
+        elif char in consonants:
+            consonant_count += 1
+        elif char in digits:
+            digit_count += 1
+
+    return vowel_count, consonant_count, digit_count
+
+# Sample input
+sample_string = "Hello World 123!"
+
+# Get the counts
+vowels, consonants, digits = count_chars(sample_string)
+
+# Print the results
+print(f"Input string: '{sample_string}'")
+print(f"Number of vowels: {vowels}")
+print(f"Number of consonants: {consonants}")
+print(f"Number of digits: {digits}")
```

OUTPUT:



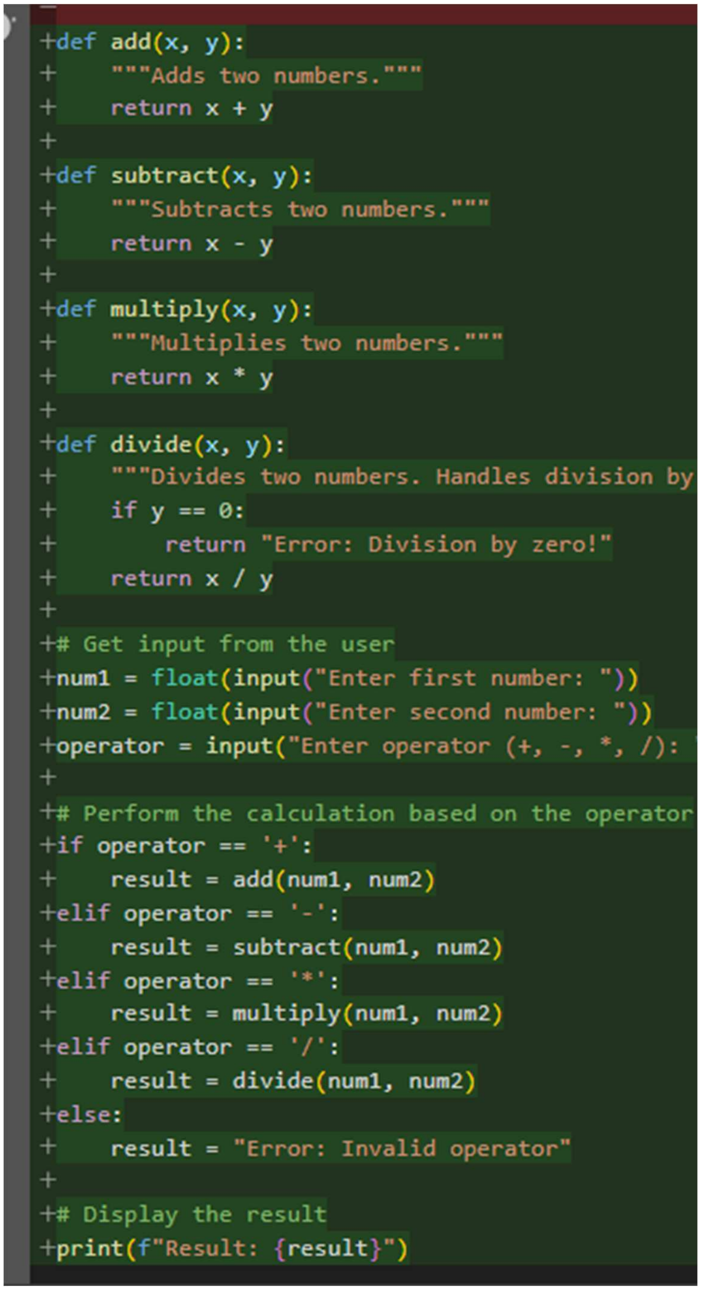
```
Input string: 'Hello World 123!'
Number of vowels: 3
Number of consonants: 7
Number of digits: 3
```

TASK 4

TASK DESCRITPION: Ask Google Gemini to generate a Python program that implements a simple calculator using functions (add, subtract, multiply, divide). Then, ask Gemini to explain how the code works.

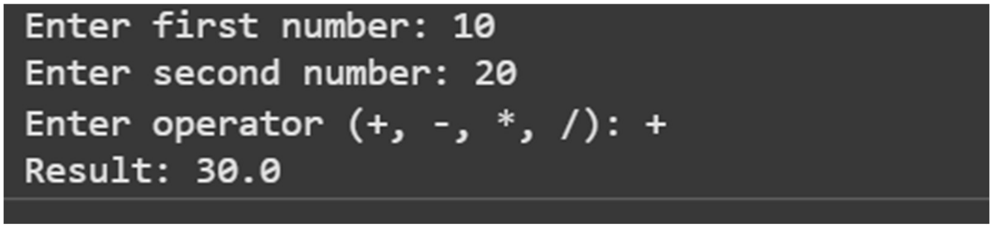
PROMPT: Generate a Python program that implements a simple calculator using functions for addition, subtraction, multiplication, and division. The program should take two numbers and an operator as input, then call the appropriate function and display the result. After generating the code, explain how each part of the program works, including how the functions are defined and called.

CODE:



```
+def add(x, y):
+    """Adds two numbers."""
+    return x + y
+
+def subtract(x, y):
+    """Subtracts two numbers."""
+    return x - y
+
+def multiply(x, y):
+    """Multiplies two numbers."""
+    return x * y
+
+def divide(x, y):
+    """Divides two numbers. Handles division by
+    if y == 0:
+        return "Error: Division by zero!"
+    return x / y
+
+# Get input from the user
+num1 = float(input("Enter first number: "))
+num2 = float(input("Enter second number: "))
+operator = input("Enter operator (+, -, *, /): 
+
+# Perform the calculation based on the operator
+if operator == '+':
+    result = add(num1, num2)
+elif operator == '-':
+    result = subtract(num1, num2)
+elif operator == '*':
+    result = multiply(num1, num2)
+elif operator == '/':
+    result = divide(num1, num2)
+else:
+    result = "Error: Invalid operator"
+
+# Display the result
+print(f"Result: {result}")
```

OUTPUT:



```
Enter first number: 10
Enter second number: 20
Enter operator (+, -, *, /): +
Result: 30.0
```