**AI ASSISTED CODING**

**LAB-2*: Exploring Additional AI Coding Tools – Gemini (Colab) and Cursor AI***

**Roll no:** 2503A51L34

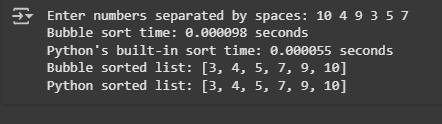
**Name:** Uzma Yasmeen

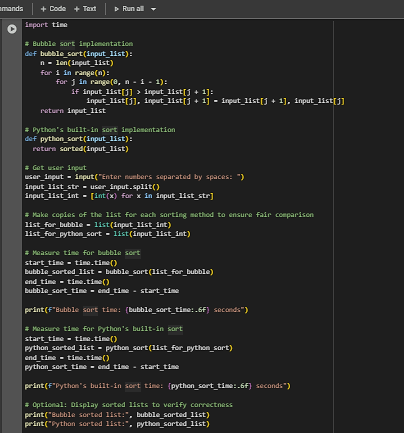
**Batch:** 25BTCAICSB20

**Task-1 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:** 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.

**Code Generated:**

**Output:**

****

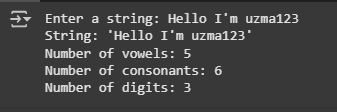
**Observation:** In this task, Gemini generated two different sorting implementations: one using the manual Bubble Sort algorithm and the other using Python’s built-in sort() function. This highlighted the difference between a step-by-step algorithmic approach and an optimized built-in method, making it clear that while algorithms are useful for learning, built-in functions provide efficiency in practical use.

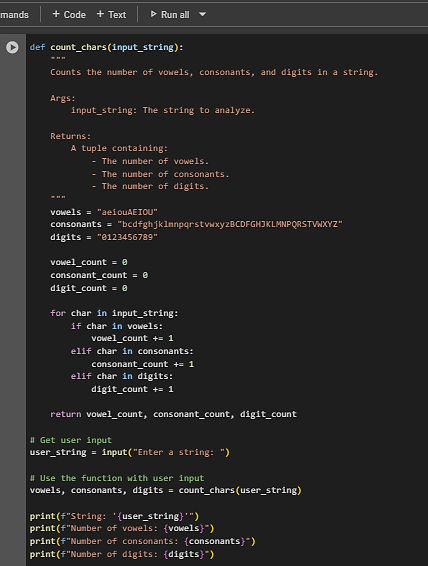
**Task-2 Description:** In Colab, use Google Gemini to generate a Python function that takes a string and returns The number of vowels, The number of consonants, The number of digits in the string

**Prompt:** 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

**Code Generated:**

**Output:**

****

****

**Observation:** In this task,I observed that Gemini was able to implement a function that processes a string and accurately counts vowels, consonants, and digits. This task demonstrated the AI’s ability to handle string manipulation and conditional logic effectively.

**Task #3:**

**Prompt:** Install and set up Cursor AI. Use it to generate a Python program that performs file handling:

1. Create a text file
2. Write sample text
3. Read and display the content.

**Code Generated:**

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.**Output:**

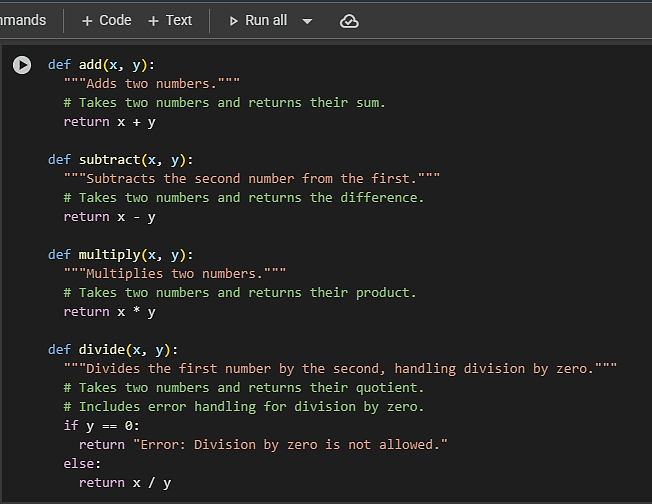
**Observation:**

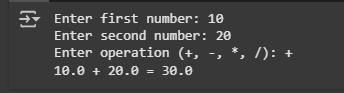
The program demonstrates basic file handling in Python by creating, writing, and reading a text file. It writes multiple lines into sample.txt and then reads back the content for display. This highlights how file operations such as open(), write(), and read() can be used to manage data storage and retrieval. Copilot generated the structure automatically, showing how it simplifies repetitive coding tasks while still producing correct results.

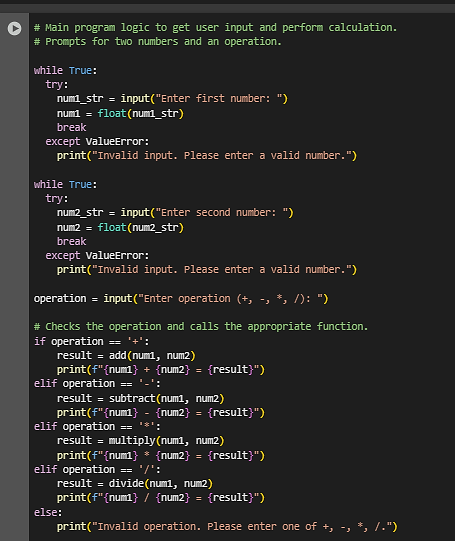
**Task-4 Description:** Ask Google Gemini to generate a Python program that implements a simple calculator using functions (add, subtract, multiply, divide). Then, ask Gemini to explain how the code works.

**Prompt:** Generate a Python program that implements a simple calculator using functions (add, subtract, multiply, divide).And explain the code.

**Code Generated:**

****

**Output:**

****

**Observation:** In this task,Gemini successfully generated a program for a simple calculator using functions for addition, subtraction, multiplication, and division. More importantly, when asked to explain the code, it provided a step-by-step breakdown, showing how AI can assist not only in code generation but also in teaching and concept explanation.

**Task #5:**

**Prompt:** Use Cursor AI to create a Python program that checks if a given year is a leap year or not. Try different prompt styles and see how Cursor modifies its code suggestions.

**Code Generated:**

A computer screen with text on it

AI-generated content may be incorrect.

**Output:**

A computer screen with text and numbers

AI-generated content may be incorrect.

**Observation:**

The program determines whether a given year is a leap year or not by applying divisibility rules. It checks if the year is divisible by 4 but not by 100, or divisible by 400, which is the correct logic for leap year calculation. This illustrates the use of conditional expressions, modular arithmetic, and function-based design in Python.