

SR UNIVERSITY
AI ASSISTED CODING
ASSIGNMENT-02

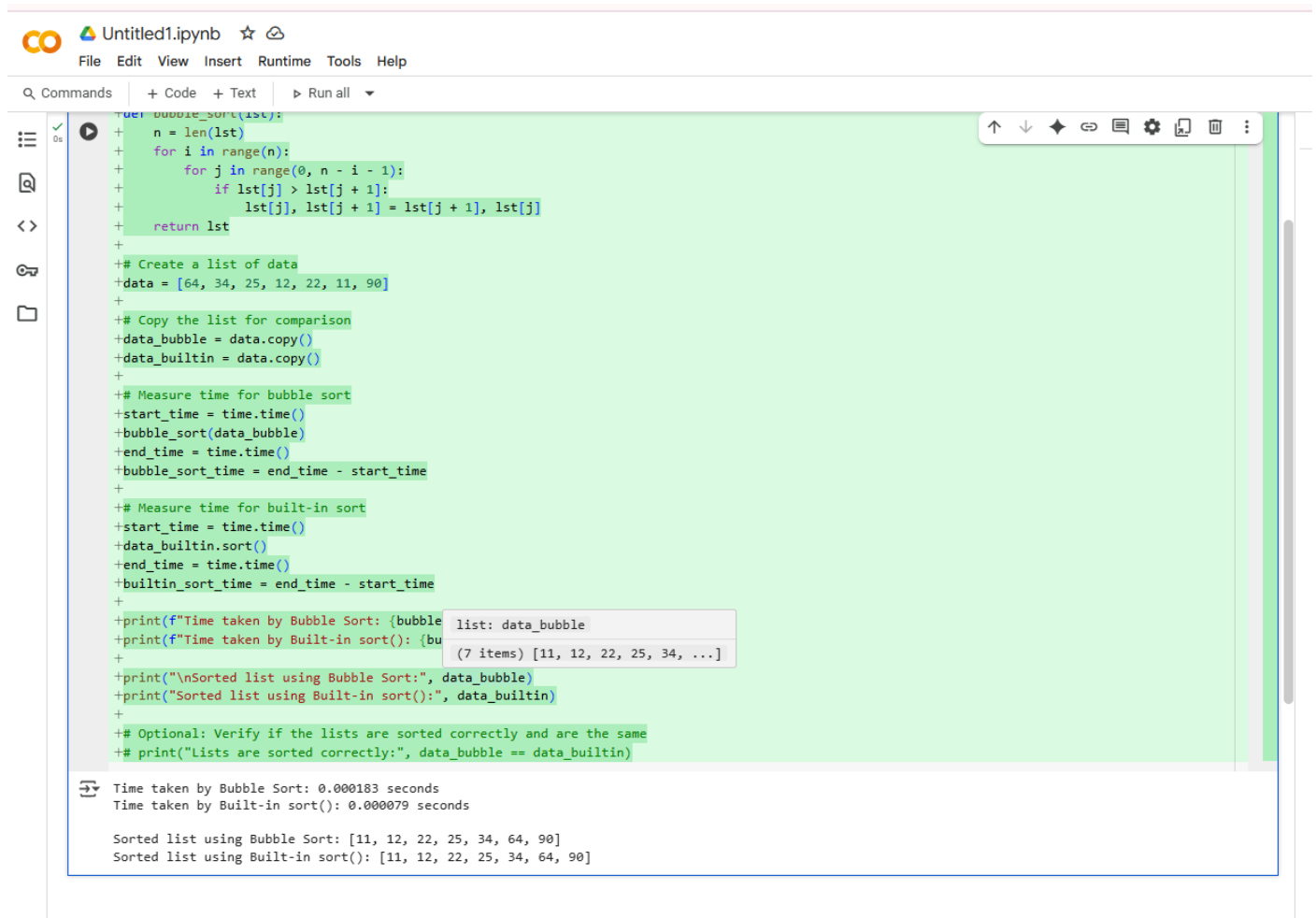
NAME:SHAIK FAHEEM
HTNO: 2503A51L39

TASK1

TASK1 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



```
def bubble_sort(lst):
    n = len(lst)
    for i in range(n):
        for j in range(0, n - i - 1):
            if lst[j] > lst[j + 1]:
                lst[j], lst[j + 1] = lst[j + 1], lst[j]
    return lst

# Create a list of data
data = [64, 34, 25, 12, 22, 11, 90]

# Copy the list for comparison
data_bubble = data.copy()
data_builtin = data.copy()

# Measure time for bubble sort
start_time = time.time()
bubble_sort(data_bubble)
end_time = time.time()
bubble_sort_time = end_time - start_time

# Measure time for built-in sort
start_time = time.time()
data_builtin.sort()
end_time = time.time()
builtin_sort_time = end_time - start_time

print(f"Time taken by Bubble Sort: {bubble_sort_time} seconds")
print(f"Time taken by Built-in sort(): {builtin_sort_time} seconds")

print("\nSorted list using Bubble Sort:", data_bubble)
print("Sorted list using Built-in sort():", data_builtin)

# Optional: Verify if the lists are sorted correctly and are the same
print("Lists are sorted correctly:", data_bubble == data_builtin)
```

Time taken by Bubble Sort: 0.000183 seconds
Time taken by Built-in sort(): 0.000079 seconds

Sorted list using Bubble Sort: [11, 12, 22, 25, 34, 64, 90]
Sorted list using Built-in sort(): [11, 12, 22, 25, 34, 64, 90]

TASK2

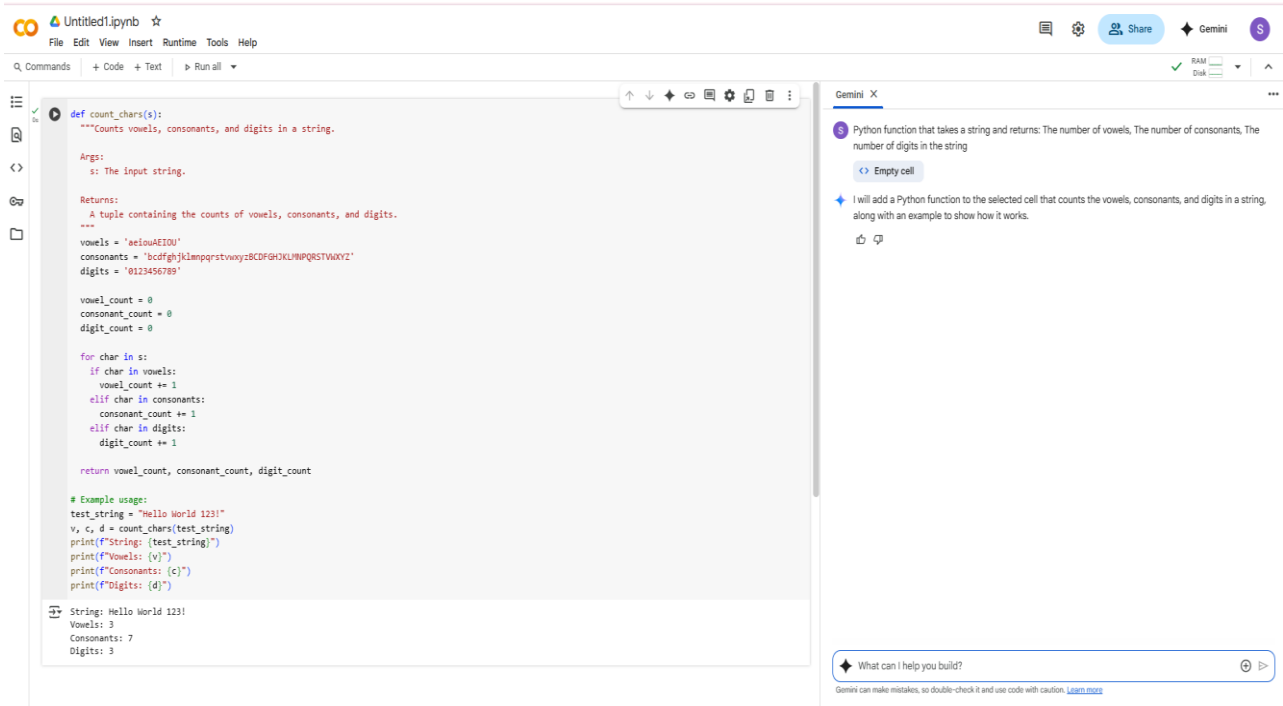
TASK2 DESCRIPTION:- In Colab, use Google Gemini to generate a Python function that takes a string and returns The number of vowels, The number of consonants, The number of digits in the string

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

SR UNIVERSITY
AI ASSISTED CODING
ASSIGNMENT-02

NAME:SHAIK FAHEEM
HTNO: 2503A51L39

CODE



```
def count_chars(s):  
    """Counts vowels, consonants, and digits in a string.  
  
    Args:  
        s: The input string.  
  
    Returns:  
        A tuple containing the counts of vowels, consonants, and digits.  
    """  
    vowels = 'aeiouAEIOU'  
    consonants = 'bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ'  
    digits = '0123456789'  
  
    vowel_count = 0  
    consonant_count = 0  
    digit_count = 0  
  
    for char in s:  
        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  
  
# Example usage:  
test_string = "Hello World 123!"  
v, c, d = count_chars(test_string)  
print(f"String: {test_string}")  
print(f"Vowels: {v}")  
print(f"Consonants: {c}")  
print(f"Digits: {d}")
```

String: Hello World 123!
Vowels: 3
Consonants: 7
Digits: 3

Python function that takes a string and returns: The number of vowels, The number of consonants, The number of digits in the string

Empty cell

I will add a Python function to the selected cell that counts the vowels, consonants, and digits in a string, along with an example to show how it works.

What can I help you build?

TASK4

TASK4 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.

SR UNIVERSITY
AI ASSISTED CODING
ASSIGNMENT-02

NAME:SHAIK FAHEEM
HTNO: 2503A51L39

CODE

```
Untitled2.ipynb
File Edit View Insert Runtime Tools Help

Q Commands + Code + Text > Run all

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 zero."""
    if y == 0:
        return "Error: Division by zero"
    return x / y

while True:
    print("Select operation:")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")
    print("5. Exit")

    choice = input("Enter choice(1/2/3/4/5): ")

    if choice in ('1', '2', '3', '4'):
        try:
            num1 = float(input("Enter first number: "))
            num2 = float(input("Enter second number: "))
        except ValueError:
            print("Invalid input. Please enter numbers.")
            continue

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))
        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))
        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))
        elif choice == '4':
            result = divide(num1, num2)
            print(num1, "/", num2, "=", result)
        elif choice == '5':
            print("Exiting calculator.")
            break
    else:
        print("Invalid Input")

""" Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
"""
```

LINE BY LINE EXPLANATION

EXPLAIN THIS CODE LINE BY LINE

✦ Okay, I can explain the code in cell `sdCnIw1FBcrj` line by line.

The code is empty in this cell, and the output shows a menu for a calculator program. However, based on the output, I can tell you what the code *likely* does and how it works:

1. `print("Select operation: ")`: This line probably prints the header "Select operation:" to the console, prompting the user to choose an operation.
2. `print("1. Add")`: This line likely prints the option "1. Add" to the console.
3. `print("2. Subtract")`: This line likely prints the option "2. Subtract" to the console.
4. `print("3. Multiply")`: This line likely prints the option "3. Multiply" to the console.
5. `print("4. Divide")`: This line likely prints the option "4. Divide" to the console.
6. `print("5. Exit")`: This line likely prints the option "5. Exit" to the console, allowing the user to quit the program.

After this, the code probably takes user input to select one of these options and then performs the corresponding calculation based on the previously defined variables `num1`, `num2`, `num3`, and `num4`. However, the code itself is not visible in the provided cell.



✦ What can I help you build?



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

SR UNIVERSITY
AI ASSISTED CODING
ASSIGNMENT-02

NAME:SHAIK FAHEEM
HTNO: 2503A51L39

OBSERVATION:- I observed how Google Gemini can generate Python programs when provided with clear prompts and how different problem-solving approaches can be compared and analyzed.

- **In Task 1**, 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.
- **In Task 2**, 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.
- **In Task 4**, 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.