Name : Maidamshetti Harish

H.no : 2303A52387

Batch : 34

Department : CSE

Specialization : AIML

Course : AI Assisted Coding

Assignment Number : 4.2

Lab Objectives

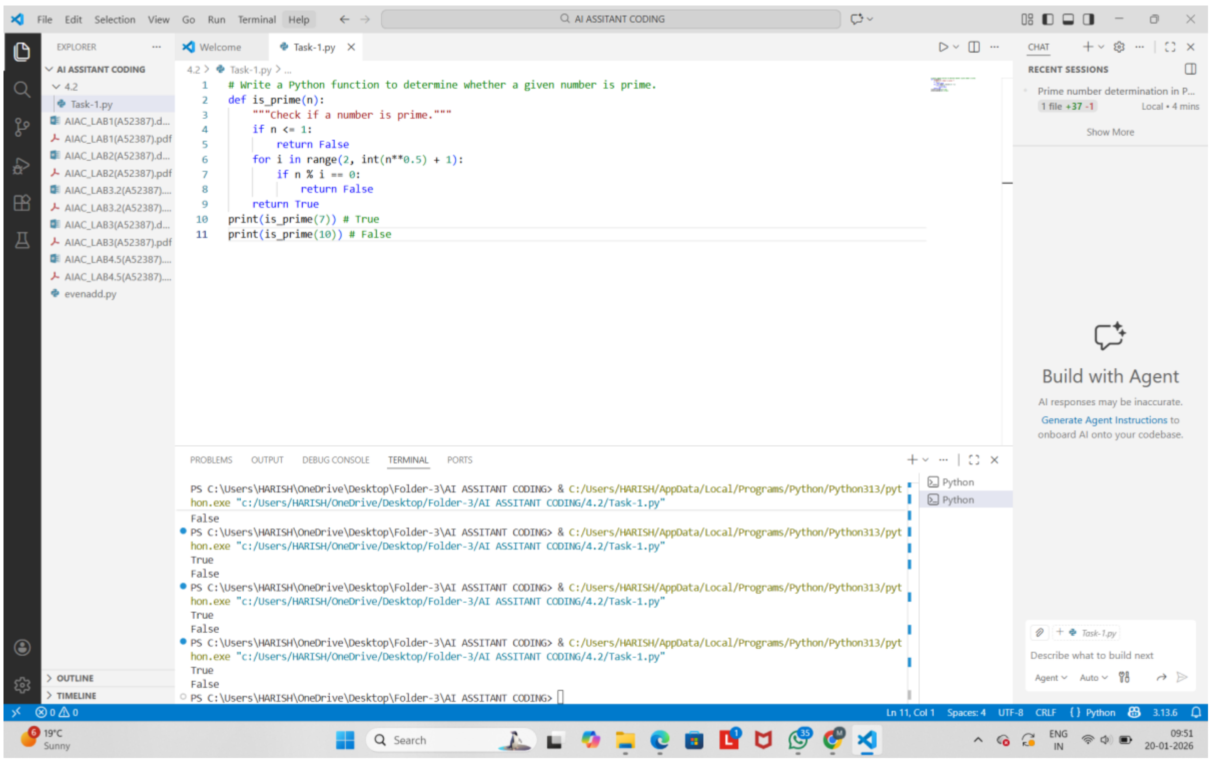
- To explore and apply different levels of prompt examples in AI-assisted code generation.
- To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality.
- To evaluate the impact of context richness and example quantity on AI performance.
- To build awareness of prompt strategy effectiveness for different problem types.

Lab Outcomes (LOs)

After completing this lab, students will be able to:

- Use zero-shot prompting to instruct AI with minimal context.
- Use one-shot prompting with a single example to guide AI code generation.
- Apply few-shot prompting using multiple examples to improve AI responses.
- Compare AI outputs across the three prompting strategies.

- ## Task Description – 1
- **Prompt:**
  "Write a Python function to determine whether a given number is prime."



```python
# Write a Python function to determine whether a given number is prime.
def is_prime(n):
    """Check if a number is prime."""
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
print(is_prime(7)) # True
print(is_prime(10)) # False
```

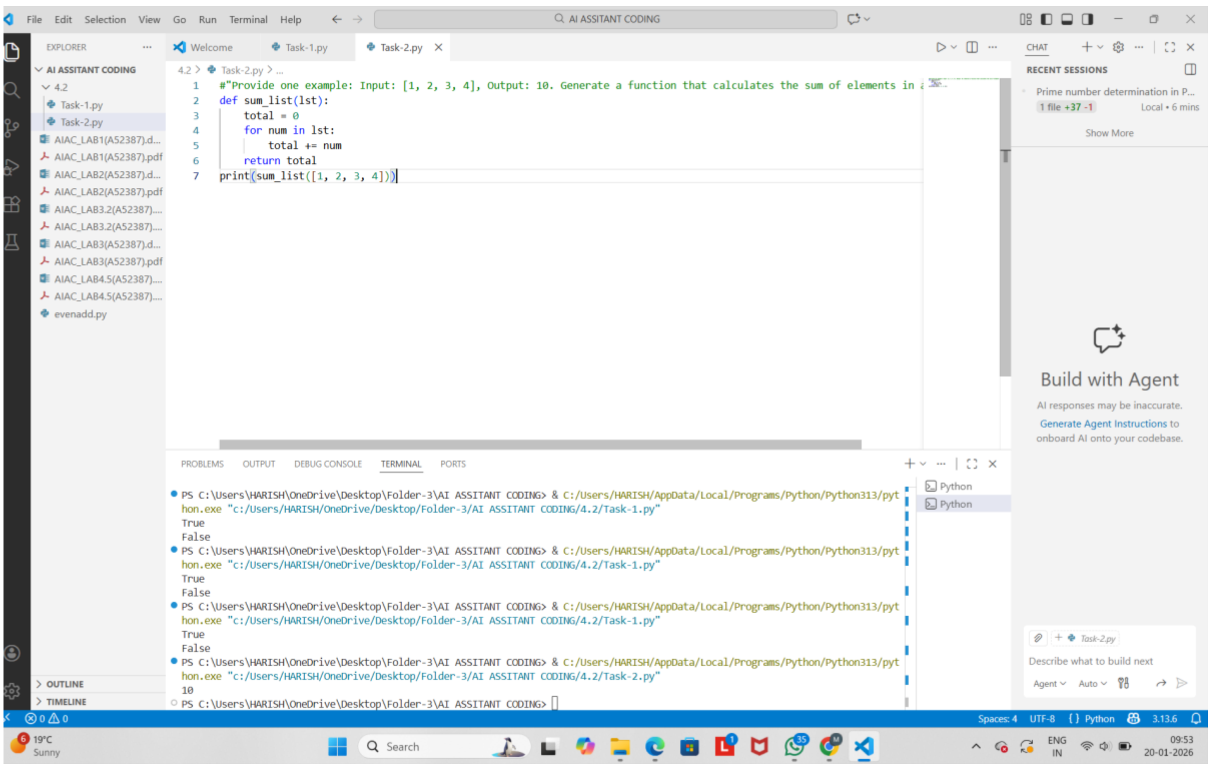
```
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING>
```

Explanation

This function checks if a number is prime by:

- Returning False for numbers less than or equal to 1.
- Iterating from 2 to the square root of the number.
- If any divisor is found, it returns False; otherwise, it returns True.

- ## Task Description – 2
- **Prompt:**
  "Provide one example: Input: [1, 2, 3, 4], Output: 10. Generate a function that calculates the sum of elements in a list."



```python
#"Provide one example: Input: [1, 2, 3, 4], Output: 10. Generate a function that calculates the sum of elements in a
def sum_list(lst):
    total = 0
    for num in lst:
        total += num
    return total
print(sum_list([1, 2, 3, 4]))
```

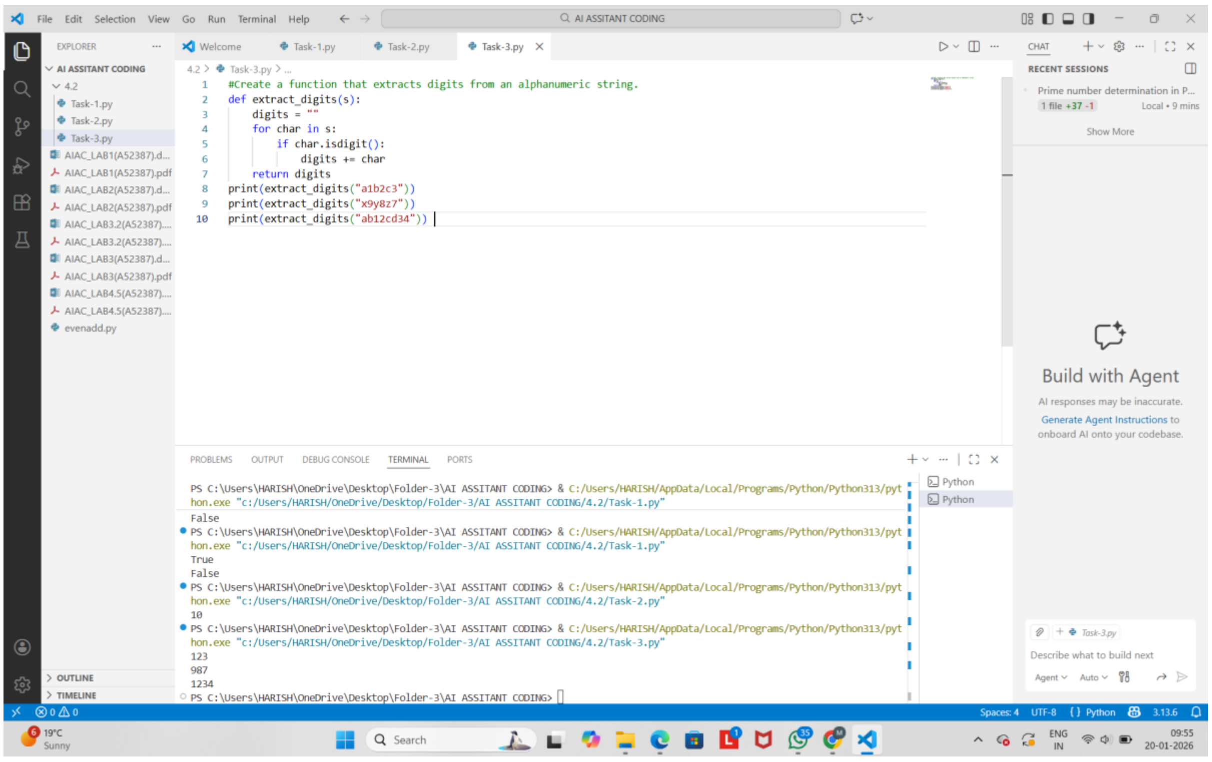
```
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-2.py"
10
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING>
```

Explanation

The function:

- Initializes a variable total to 0.
- Iterates through each number in the list.
- Adds each number to total.
- Returns the final sum.

- ## Task Description – 3
- **Prompt**
  "Create a function that extracts digits from an alphanumeric string."



```python
#Create a function that extracts digits from an alphanumeric string.
def extract_digits(s):
    digits = ""
    for char in s:
        if char.isdigit():
            digits += char
    return digits
print(extract_digits("a1b2c3"))
print(extract_digits("x9y8z7"))
print(extract_digits("ab12cd34"))
```

```
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-1.py"
True
False
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-2.py"
10
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-3.py"
123
987
1234
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING>
```

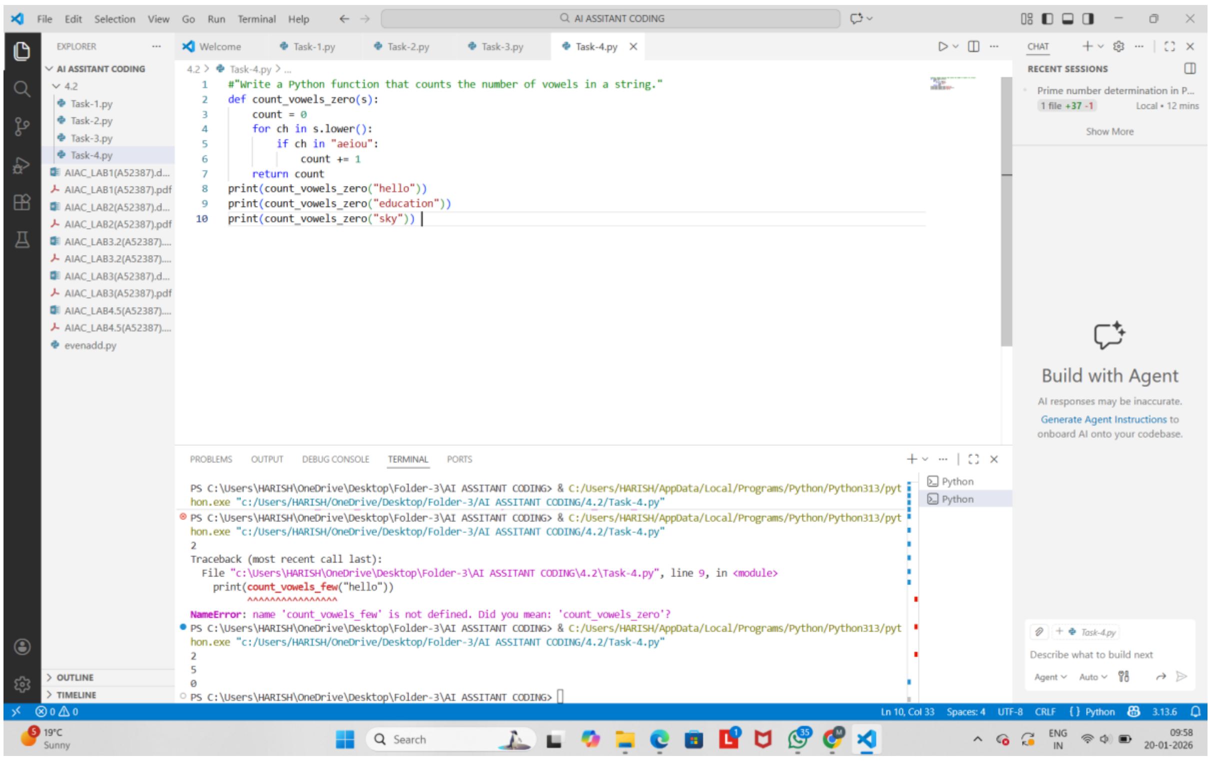
Explanation

The function:

- Initializes an empty string digits.
- Iterates through each character in the string.
- Appends the character if it is a digit.
- Returns the final digit-only string.

- ## Task Description – 4
- **Prompt:**
  "Write a Python function that counts the number of vowels in a string."

-



```python
#"Write a Python function that counts the number of vowels in a string."
def count_vowels_zero(s):
    count = 0
    for ch in s.lower():
        if ch in "aeiou":
            count += 1
    return count
print(count_vowels_zero("hello"))
print(count_vowels_zero("education"))
print(count_vowels_zero("sky"))
```

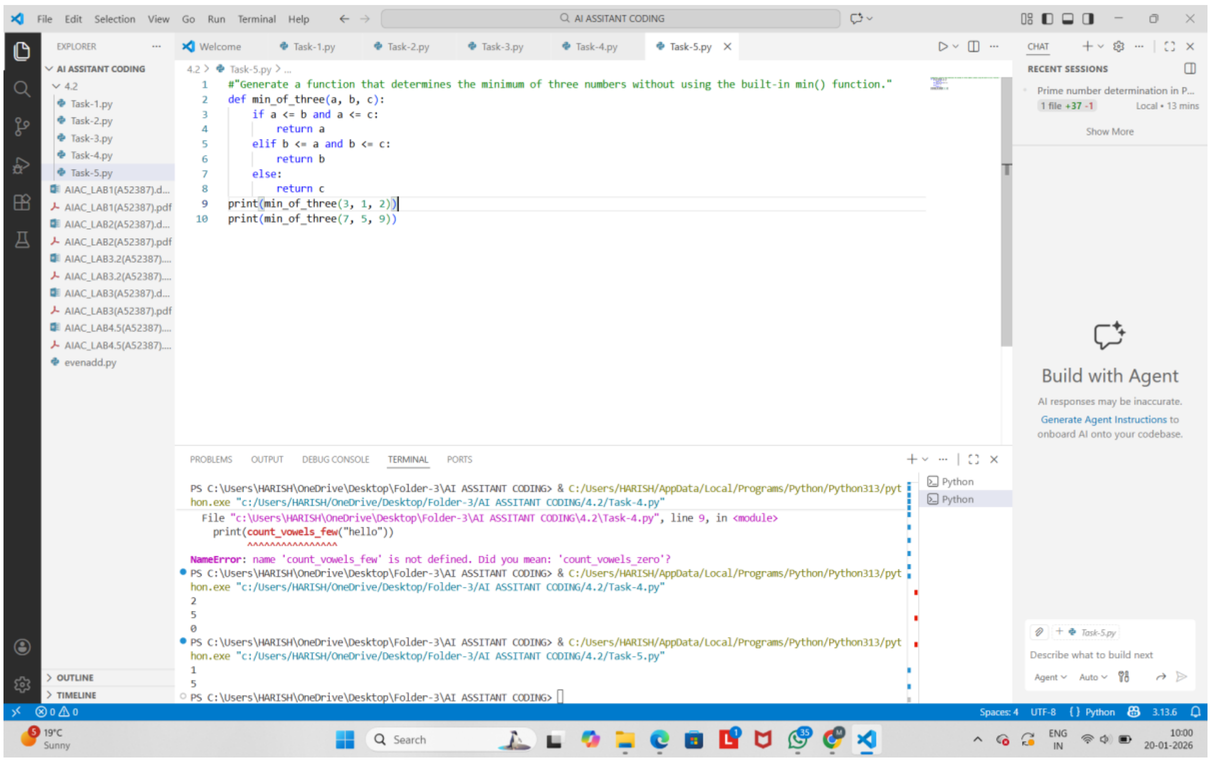
```
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-4.py"
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-4.py"
2
Traceback (most recent call last):
  File "c:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING\4.2\Task-4.py", line 9, in <module>
    print(count_vowels_few("hello"))
          ^^^^^^^^^^^^^^^^
NameError: name 'count_vowels_few' is not defined. Did you mean: 'count_vowels_zero'?
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-4.py"
2
5
0
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING>
```

Explanation

- The zero-shot version worked correctly with minimal instruction.
- The few-shot version produced a similar solution but was slightly clearer and more structured.
- Providing examples helped reinforce correct handling of edge cases like strings with no vowels.
- Few-shot prompting improves confidence in output accuracy for more complex or ambiguous tasks.

• Task Description – 5

- "Generate a function that determines the minimum of three numbers without using the built-in min() function."



```python
#"Generate a function that determines the minimum of three numbers without using the built-in min() function."
def min_of_three(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c
print(min_of_three(3, 1, 2))
print(min_of_three(7, 5, 9))
```

```
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-4.py"
  File "c:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING\4.2\Task-4.py", line 9, in <module>
    print(count_vowels_few("hello"))
          ^^^^^^^^^^^^^^^^
NameError: name 'count_vowels_few' is not defined. Did you mean: 'count_vowels_zero'?
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-4.py"
2
5
0
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING> & C:/Users/HARISH/AppData/Local/Programs/Python/Python313/pyt
hon.exe "c:/Users/HARISH/OneDrive/Desktop/Folder-3/AI ASSITANT CODING/4.2/Task-5.py"
1
5
PS C:\Users\HARISH\OneDrive\Desktop\Folder-3\AI ASSITANT CODING>
```

Explanation

The function:

- Compares a with b and c.
- If a is smallest, it returns a.
- Otherwise, it checks if b is smallest.
- If not, it returns c.
- This logic handles all possible cases correctly.

Conclusion

- Zero-shot prompting works well for simple and well-defined problems.
- One-shot prompting guides the AI toward the desired logic using a single example.
- Few-shot prompting significantly improves accuracy and clarity for pattern-based tasks.

- Providing examples helps the model understand edge cases and expected behavior more effectively.