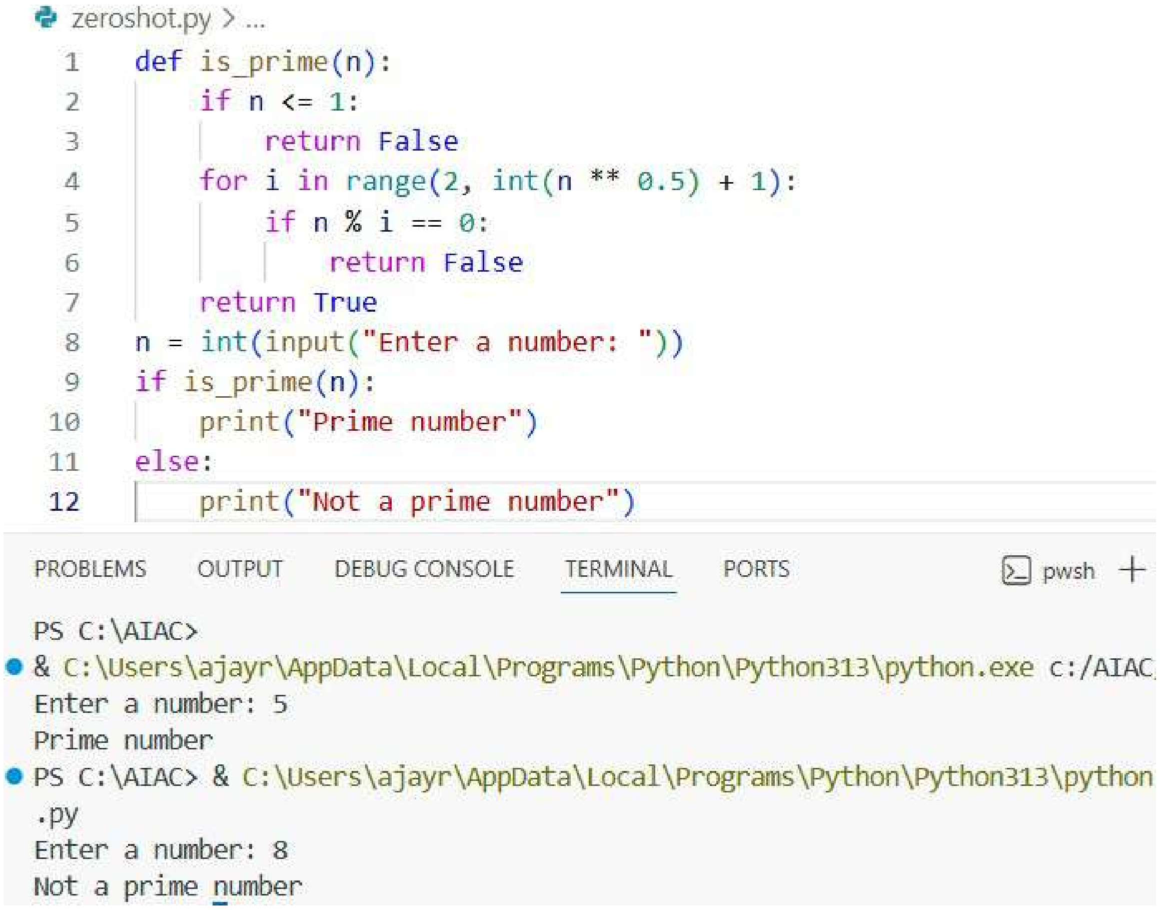
AI - Assisted Coding-Assignment -4

HTNO:2303A52228
Batch:37

Lab 4: Advanced Prompt Engineering–Zero-shot,One-shot,Few-shot Techniques.

Task Description-1
Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime.
Expected Output:
A basic Python function to check if a number is prime, demonstrating correct logical conditions without relying on examples or additional context.
Prompt:
Write a Python function to determine whether a given number is prime.
CODE AND OUTPUT:



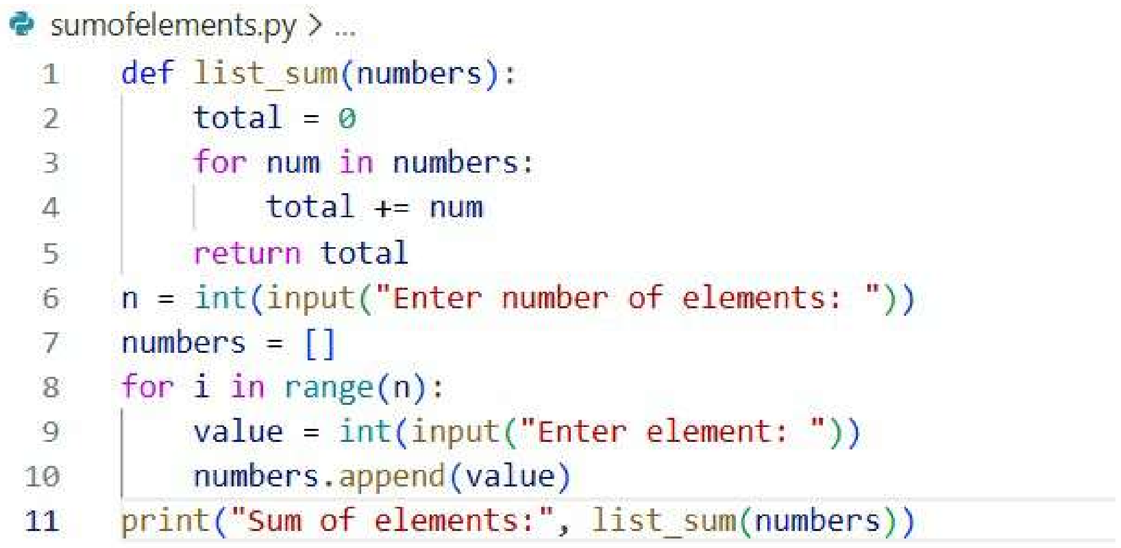
```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True
n = int(input("Enter a number: "))
if is_prime(n):
    print("Prime number")
else:
    print("Not a prime number")
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe c:/AIAC
Enter a number: 5
Prime number
PS C:\AIAC> & C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python
.py
Enter a number: 8
Not a prime number
```

Explanation:
The function checks whether a given number is prime by first eliminating numbers less than or equal to 1, since such numbers cannot be prime. It then tests whether the number is divisible by any integer starting from 2 up to the square root of the number. Checking only up to the square root is sufficient because any factor greater than the

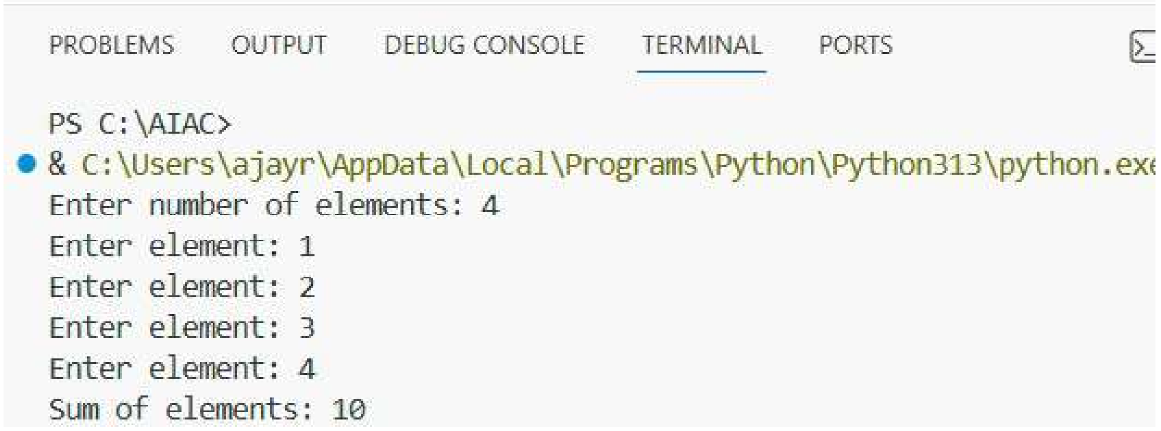
square root would have a corresponding factor smaller than it.

Task Description-2
One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.
Expected Output:
A correct conversion function guided by the single example.
Prompt:
Write a Python function to calculate the sum of elements in a list.
CODE AND OUTPUT:



sumofelements.py > ...

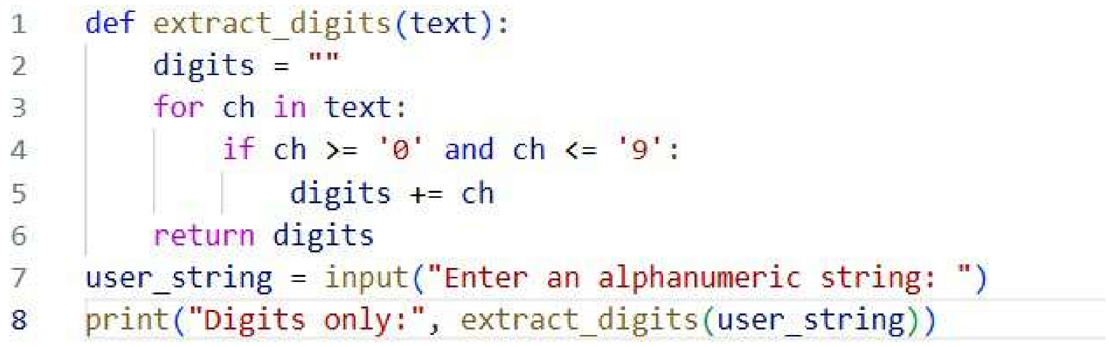
```python
def list_sum(numbers):
    total = 0
    for num in numbers:
        total += num
    return total
n = int(input("Enter number of elements: "))
numbers = []
for i in range(n):
    value = int(input("Enter element: "))
    numbers.append(value)
print("Sum of elements:", list_sum(numbers))
```



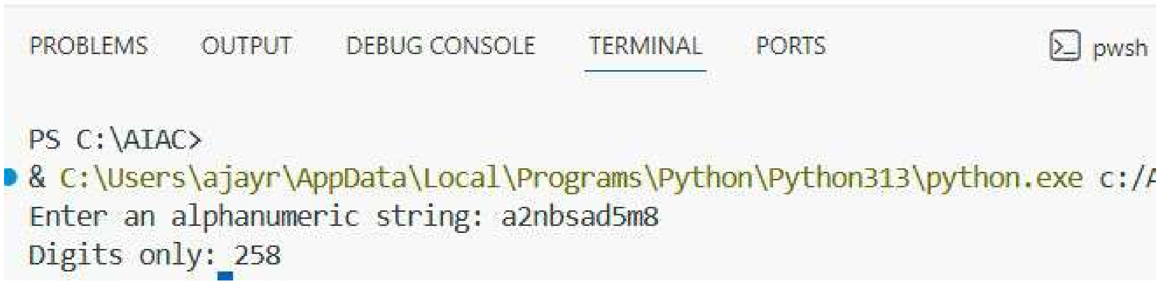
```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe
Enter number of elements: 4
Enter element: 1
Enter element: 2
Enter element: 3
Enter element: 4
Sum of elements: 10
```

Explanation:
The function calculates the sum of all elements in a list by initializing a variable to store the total value and then iterating through each number in the list. During each iteration, the current number is added to the total. After all elements have been processed, the function returns the final sum.

Task Description-3
Few-shot: Give 2–3 examples to create a function that extracts digits from an

alphanumeric string.
Expected Output:
Accurate function that returns only the digits from alphanumeric string.
Prompt:
Create a Python function that extracts only digits from an alphanumeric string.
CODE AND OUTPUT:


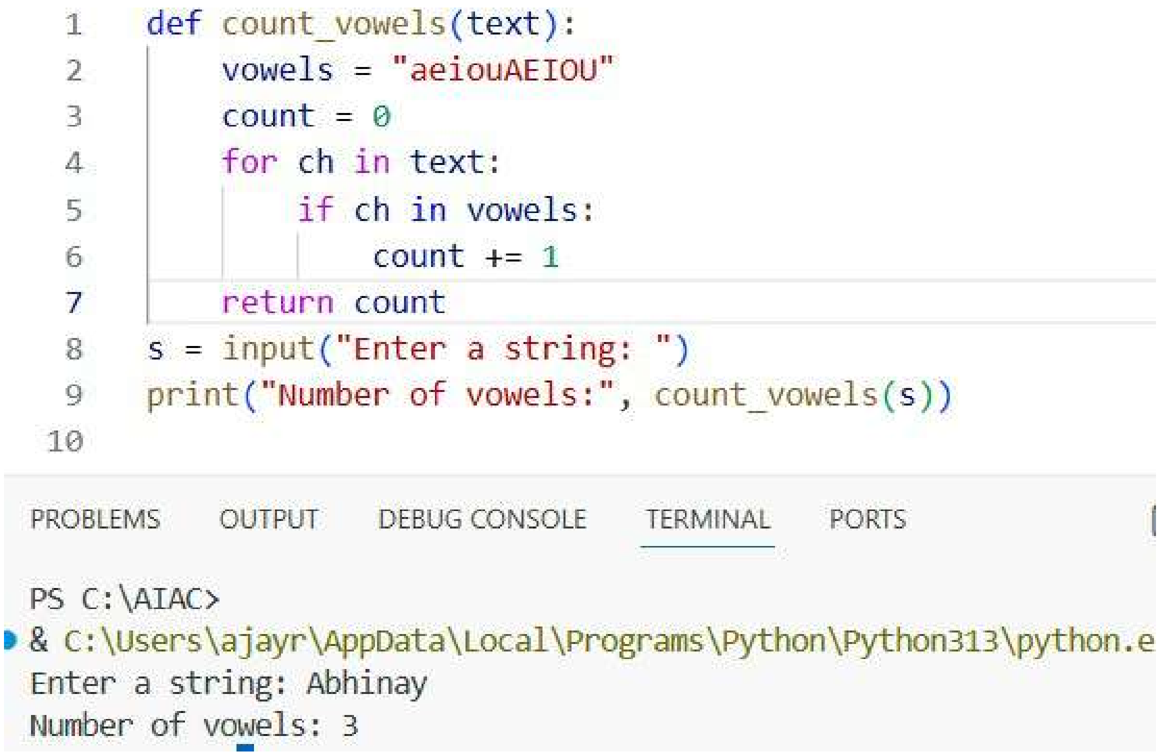
```python
def extract_digits(text):
    digits = ""
    for ch in text:
        if ch >= '0' and ch <= '9':
            digits += ch
    return digits
user_string = input("Enter an alphanumeric string: ")
print("Digits only:", extract_digits(user_string))
```



```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe c:/A
Enter an alphanumeric string: a2nbsad5m8
Digits only: 258
```

Explanation:
The function extracts digits from a user-defined alphanumeric string by checking each character to determine whether it falls between '0' and '9'. If the condition is satisfied, the character is added to a new string that stores only numeric values. The input is taken dynamically from the user, and the function returns the digits in the same order they appear in the original string. The multiple examples in the prompt guide the AI to correctly identify the required pattern, making this a clear demonstration of few-shot prompting.

Task Description-4
Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string
Expected Output:
Output comparison + student explanation on how examples helped the model.
Prompt:
Write a Python function to count the number of vowels in a string.
CODE AND OUTPUT:

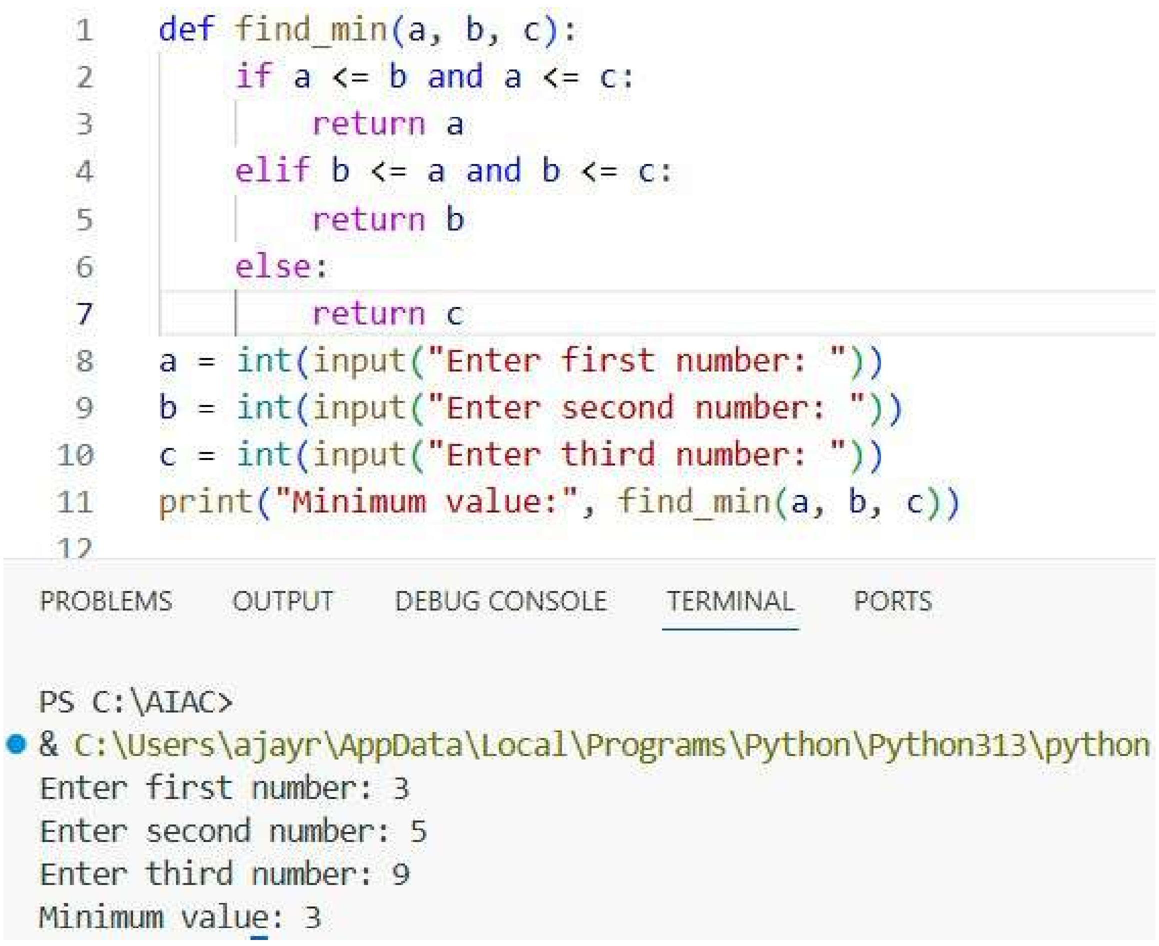


```python
def count_vowels(text):
    vowels = "aeiouAEIOU"
    count = 0
    for ch in text:
        if ch in vowels:
            count += 1
    return count
s = input("Enter a string: ")
print("Number of vowels:", count_vowels(s))
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.e
Enter a string: Abhinay
Number of vowels: 3
```

Explanation:
In zero-shot prompting, the AI generates the vowel-counting function using only the instruction, relying on its general understanding of the problem. The solution works correctly but is based on assumptions inferred by the model. In few-shot prompting, the inclusion of examples clearly shows how vowels should be counted, including uppercase letters. These examples help reduce ambiguity and guide the AI toward a more structured and confident solution. Overall, few-shot prompting improves clarity and reliability by demonstrating the expected input-output behavior.

Task Description-5
Use few-shot prompting with 3 sample inputs to generate a function that determines the minimum of three numbers without using the built-in min() function.
Expected Output:
A function that handles all cases with correct logic based on example patterns.
Prompt:
Write a Python function to determine the minimum of three numbers without using the built-in min() function.
Examples:
Input: (4, 9, 6) → Output: 4
Input: (12, 3, 7) → Output: 3
Input: (5, 5, 8) → Output: 5
CODE AND OUTPUT:



```python
def find_min(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
print("Minimum value:", find_min(a, b, c))
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python
Enter first number: 3
Enter second number: 5
Enter third number: 9
Minimum value: 3
```

Explanation:
The function determines the minimum of three user-defined numbers by comparing them using conditional statements rather than the built-in min() function. It first checks whether the first number is less than or equal to the other two. If this condition is false, it checks whether the second number is the smallest. The examples provided in the few-shot prompt help the AI understand the comparison pattern, including cases where numbers are equal, ensuring the function works correctly for all inputs.