Assignment 4.2

Task 01:- Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime

```python
# ============ PROMPT 1: Check if a number is PRIME ==========
# Write a Python function to determine whether a given number
# Example: Input: [1, 2, 3, 4] => Check each for prime

Tabnine | Edit | Test | Explain | Document
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

print("=" * 50)
print("PROMPT 1: Prime Number Checker")
print("=" * 50)
numbers = [1, 2, 3, 4, 5, 10, 13, 20]
for num in numbers:
    print(f"Input: {num} => Is Prime: {is_prime(num)}")

```

```
C:\Users\kunda>cd "c:\Users\kunda\OneDrive\Pictures - Copy\Dokumen\AI coding" && python "4.2 AI LAB.py"

Prime Number Checker:
Prime Number Checker:
------------------------------
1 is prime: False
2 is prime: True
3 is prime: True
4 is prime: False
5 is prime: True
10 is prime: False
13 is prime: True
20 is prime: False
29 is prime: True
100 is prime: False
```

Task 02 :- 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.

```python
# ============ PROMPT 2: Sum of elements in a list ============
# Using the above example, write a Python function that calculates the sum of elements ir
# Example: Input: [1, 2, 3, 4] => Output: 10

Tabnine | Edit | Test | Explain | Document
def sum_of_list(lst):
    total = 0
    for num in lst:
        total += num
    return total

print("\n" + "=" * 50)
print("PROMPT 2: Sum of Elements in a List")
print("=" * 50)
test_lists = [[1, 2, 3, 4], [5, 10, 15], [100, 200, 300], [-1, 2, 3]]
for lst in test_lists:
    result = sum_of_list(lst)
    print(f"Input: {lst} => Output: {result}")
```

```
==================================================
PROMPT 2: Sum of Elements in a List
==================================================
Input: [1, 2, 3, 4] => Output: 10
Input: [5, 10, 15] => Output: 30
Input: [100, 200, 300] => Output: 600
Input: [-1, 2, 3] => Output: 4
```

Task3 :- Give 2 – 3 examples to create a function that extracts digits from an alphanumeric string.

```python
# ============ PROMPT 3: Extract only digits from alphanumeric string
# Examples:
# Input: "abc123" => Output: "123"
# Input: "a1b2c3" => Output: "123"
# Input: "2025AI" => Output: "2025"

Tabnine | Edit | Test | Explain | Document
def extract_digits(s):
    result = ""
    for char in s:
        if char.isdigit():
            result += char
    return result

print("\n" + "=" * 50)
print("PROMPT 3: Extract Digits from Alphanumeric String")
print("=" * 50)
test_strings = ["abc123", "a1b2c3", "2025AI", "hello123world456"]
for string in test_strings:
    result = extract_digits(string)
    print(f'Input: "{string}" => Output: "{result}"')
```

```
==================================================
PROMPT 3: Extract Digits from Alphanumeric String
==================================================
Input: "abc123" => Output: "123"
Input: "a1b2c3" => Output: "123"
Input: "2025AI" => Output: "2025"
Input: "hello123world456" => Output: "123456"
```

Task 4 :- Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.

```
# ============ PROMPT 4: Count vowels in a string ============
# Write a Python function that counts the number of vowels in a string.
# Examples:
# Input: "hello" => Output: 2
# Input: "education" => Output: 5
# Input: "sky" => Output: 0

Tabnine | Edit | Test | Explain | Document
def count_vowels(s):
    vowels = "aeiouAEIOU"
    count = 0
    for char in s:
        if char in vowels:
            count += 1
    return count

print("\n" + "=" * 50)
print("PROMPT 4: Count Vowels in a String")
print("=" * 50)
test_vowel_strings = ["hello", "education", "sky", "Python", "AI"]
for string in test_vowel_strings:
    result = count_vowels(string)
    print(f'Input: "{string}" => Output: {result}')
```

```
==================================================
PROMPT 4: Count Vowels in a String
==================================================
Input: "hello" => Output: 2
Input: "education" => Output: 5
Input: "sky" => Output: 0
Input: "Python" => Output: 1
Input: "AI" => Output: 2
```

Explanation :- In zero-shot prompting, the model generates the function using its general understanding of programming and vowels. Since no examples are given, the model decides on its own how to handle cases like uppercase letters.

In few-shot prompting, the examples clearly show what the expected input and output look like. From these examples, the model learns important rules such as:

- Uppercase vowels should be counted
- Strings without vowels should return 0

Because of this guidance, the model produces a more accurate and user-aligned function. Examples act like hints, reducing ambiguity and improving correctness.

Task 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.

```python
# ============ PROMPT 5: Find minimum of three numbers (without using min()) ======
# Examples:
# Input: 3, 7, 5 => Output: 3
# Input: 10, 2, 8 => Output: 2
# Input: -1, -5, 0 => Output: -5

Tabnine | Edit | Test | Explain | Document
def find_minimum(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c

print("\n" + "=" * 50)
print("PROMPT 5: Find Minimum of Three Numbers (without min())")
print("=" * 50)
test_triplets = [(3, 7, 5), (10, 2, 8), (-1, -5, 0), (100, 50, 75)]
for triplet in test_triplets:
    result = find_minimum(triplet[0], triplet[1], triplet[2])
    print(f"Input: {triplet[0]}, {triplet[1]}, {triplet[2]} => Output: {result}")

print("\n" + "=" * 50)
print("ALL PROMPTS COMPLETED!")
print("=" * 50)
```

```
==================================================
PROMPT 5: Find Minimum of Three Numbers (without min())
==================================================
Input: 3, 7, 5 => Output: 3
Input: 10, 2, 8 => Output: 2
Input: -1, -5, 0 => Output: -5
Input: 100, 50, 75 => Output: 50
```