

AI-Assisted Coding

ASSIGNMENT - 3.1

M.Goutham.

2303A52010

batch-31

Question 1: Zero-Shot Prompting (Palindrome Number Program)

Write a zero-shot prompt (without providing any examples) to generate a Python function that checks whether a given number is a palindrome prompt:

```
#write a python function to check whether a number is palindrome or not.
```

code:

```
def is_palindrome(number):
    """
    Args:
        number: int : the number to be checked
    Returns: bool : True if the number is palindrome, False otherwise
    """
    str_num = str(number)

    if str_num == str_num[::-1]:
        return True
    else:
        return False
print(is_palindrome(12321)) # Output: True
print(is_palindrome(12345)) # Output: False
```

output:

```
● PS C:\Users\DELL\OneDrive\Desktop\VS Code\AIAC_LAB> python
True
False
```

Explanation:

- A zero-shot prompt was written without giving examples.
- AI generated a palindrome-checking function in Python.
- The function reverses the number and compares it with the original.
- Answer returns boolean type answer True or False

Question 2: One-Shot Prompting (Factorial Calculation)

Write a one-shot prompt by providing one input-output example and ask the AI to generate a Python function to compute the factorial of a given number.

Example:

Input: 5 → Output: 120

prompt:

```
#write a python function to give factorial of a number.
#example: 5! = 120
```

code:

```
def factorial(n):
    """args:
    n: int : the number to calculate factorial for
    returns: int : factorial of the number
    """
    if n < 0:
        return "Factorial is not defined for negative numbers"
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result
print(factorial(5)) # Output: 120
print(factorial(4)) # Output: 24
```

output:

```
PS C:\Users\DELL\OneDrive\Desktop\VS Code\AIAC_LAB> python
120
24
```

Explanation:

- Better clarity than zero-shot due to example-guided generation
- Improved correctness with input validation
- More readable and maintainable code

Question 3: Few-Shot Prompting (Armstrong Number Check)

Write a few-shot prompt by providing multiple input-output examples to guide the AI in generating a Python function to check whether a given number is an Armstrong number.

Examples:

- Input: 153 → Output: Armstrong Number
- Input: 370 → Output: Armstrong Number
- Input: 123 → Output: Not an Armstrong Number

prompt:

```
"""
write a python function to check whether the given number is armstrong number or not
input: 153
output: armstrong number
input: 370
output: armstrong number
input: 123
output: not an armstrong number
"""
```

code:

```
def is_armstrong(number):
    """args:
    number: int : the number to be checked
    returns: bool : True if the number is armstrong, False otherwise
    """
    try:
        num_str = str(number)
        num_digits = len(num_str)
        sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)

        if sum_of_powers == number:
            return "Armstrong Number"
        else:
            return "Not an Armstrong Number"
    except ValueError:
        return "Invalid input. Please enter a valid integer."

print(is_armstrong(153)) # Output: Armstrong Number
print(is_armstrong(370)) # Output: Armstrong Number
print(is_armstrong(123)) # Output: Not an Armstrong Number
```

output:

```
● PS C:\Users\DELL\OneDrive\Desktop\VS Code\AIAC_LAB> python
Armstrong Number
Armstrong Number
Not an Armstrong Number
```

Explanation:

- Better clarity than zero-shot due to example-guided generation
- Improved correctness with input validation
- Guides correct power calculation based on digit count
- More readable and maintainable code

Question 4:

Context-Managed Prompting (Optimized Number

Classification)

Design a context-managed prompt with clear instructions and constraints to generate an optimized Python program that classifies a number as prime, composite, or neither.

prompt:

```
#write a python function to check whether the number is prime or composite or neither.
```

code:

```
def prime_or_composite(number):
    """args:
    number: int : the number to be checked
    returns: str : "Prime" if the number is prime, "Composite" if composite, "Neither" otherwise
    """
    if number <= 1:
        return "Neither"
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return "Composite"
    return "Prime"

print(prime_or_composite(7))      # Output: Prime
print(prime_or_composite(10))     # Output: Composite
print(prime_or_composite(1))      # Output: Neither
```

output:

```
● PS C:\Users\DELL\OneDrive\Desktop\VS Code\AIAC_LAB> python
Prime
Composite
Neither
```

Explanation:

- A zero-shot prompt was written without giving examples.
- AI generated a prime or composite or neither-checking function in Python.
- The function reverses the number and compares it with the original.
- Answer returns whether the number is prime or composite or neither

Question 5: Zero-Shot Prompting (Perfect Number Check)

Write a zero-shot prompt (without providing any examples) to generate a Python function that checks whether a given number is a perfect number.

prompt:

```
#write a python function to check whether the given number is perfect number or not
```

code:

```
def is_perfect_number(number):
    """args:
    number: int : the number to be checked
    returns: bool : True if the number is perfect, False otherwise
    """
    if number <= 1:
        return False

    divisors_sum = 0
    for i in range(1, number):
        if number % i == 0:
            divisors_sum += i

    return divisors_sum == number

print(is_perfect_number(6))    # Output: True
print(is_perfect_number(28))    # Output: True
```

output:

```
● PS C:\Users\DELL\OneDrive\Desktop\VS Code\AIAC_LAB> python
True
True
```

Explanation:

- A zero-shot prompt was written without giving examples.
- AI generated a perfect number checking function in Python.
- The function reverses the number and compares it with the original.
- Answer returns boolean type answer True or False

Question 6: Few-Shot Prompting (Even or Odd Classification with Validation)

Write a few-shot prompt by providing multiple input-output examples to guide the AI in generating a Python program that determines whether a given number is even or odd, including proper

input validation.

Examples:

- Input: 8 → Output: Even
- Input: 15 → Output: Odd
- Input: 0 → Output: Even

prompt:

```
"""
write a python function to check whether the given number is Even or Odd
input: 8
output: Even
input: 15
output: Odd
input: 0
output: Even
"""
```

code:

```
def even_or_odd(number):
    """
    args:
        number: int : the number to be checked
    returns: str : "Even" if the number is even, "Odd" otherwise
    """

    if number % 2 == 0:
        return "Even"
    else:
        return "Odd"
print(even_or_odd(8))    # Output: Even
print(even_or_odd(15))  # Output: Odd
print(even_or_odd(0))   # Output: Even
```

output:

```
● PS C:\Users\DELL\OneDrive\Desktop\VS Code\AIAC_LAB> python
Even
Odd
Even
```

Explanation:

- Generated a python function to check whether the number
- Few shot prompt helps us giving the code in high accuracy
- Better clarity than zero-shot due to example-guided generation
- Answer either Even or Odd