AI ASSISTED CODING ASSIGNMENT - 3.5

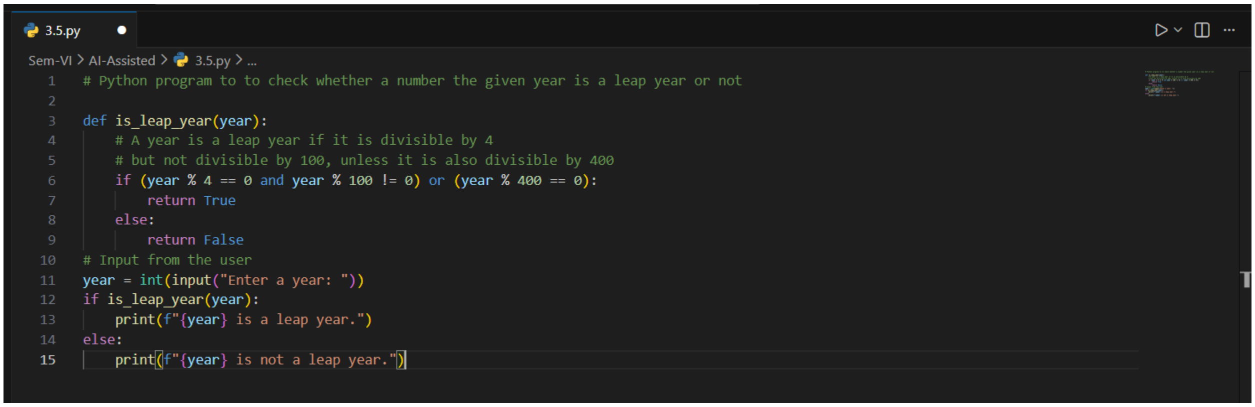
J.SRIJITH RAO
2303A51019
BATCH-01

Question 1: Zero-Shot Prompting (Leap Year Check)

Write a zero-shot prompt to generate a Python function that checks whether a given year is a leap year.

Task:

- Record the AI-generated code.
- Test with years like 1900, 2000, 2024.
- Identify logical flaws or missing conditions.



```python
# Python program to to check whether a number the given year is a leap year or not

def is_leap_year(year):
    # A year is a leap year if it is divisible by 4
    # but not divisible by 100, unless it is also divisible by 400
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
# Input from the user
year = int(input("Enter a year: "))
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Question 2: One-Shot Prompting (GCD of Two Numbers)

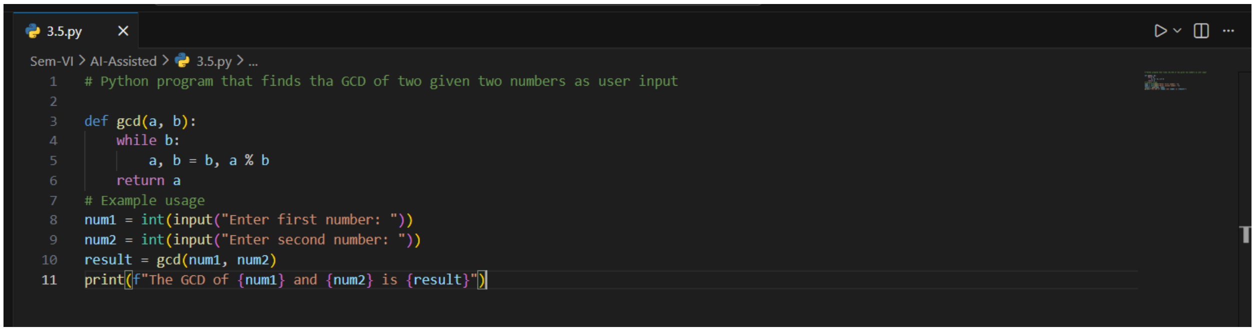
Write a one-shot prompt with one example to generate a Python function that finds the Greatest Common Divisor (GCD) of two numbers.

Example:

Input: 12, 18 → Output: 6

Task:

- Compare with a zero-shot solution.
- Analyze algorithm efficiency.



```python
# Python program that finds tha GCD of two given two numbers as user input

def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
# Example usage
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
result = gcd(num1, num2)
print(f"The GCD of {num1} and {num2} is {result}")
```

Question 3: Few-Shot Prompting (LCM Calculation)

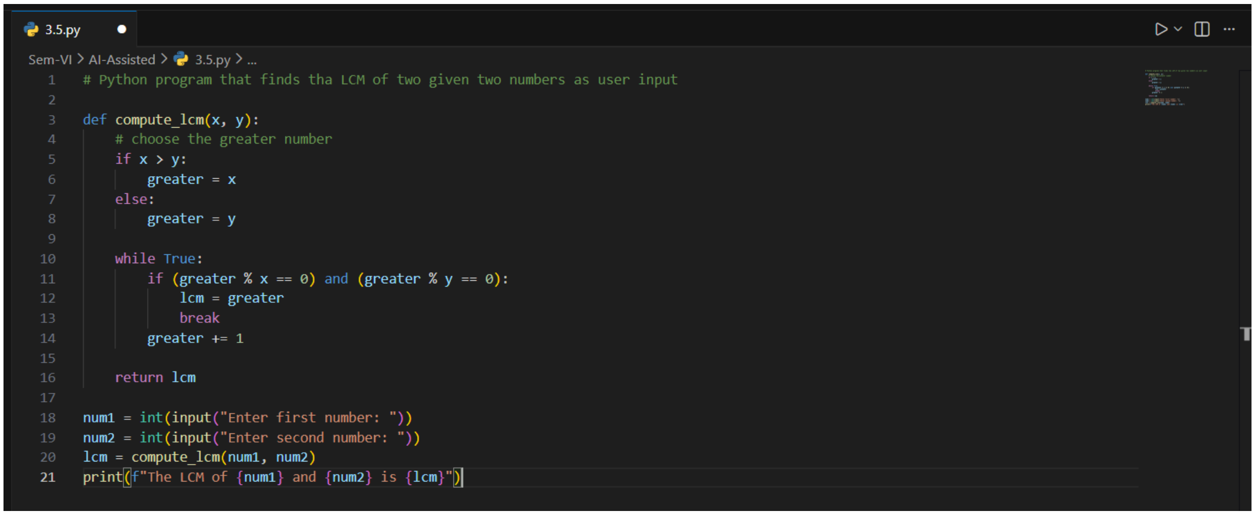
Write a few-shot prompt with multiple examples to generate a Python function that computes the Least Common Multiple (LCM).

Examples:

- Input: 4, 6 → Output: 12
- Input: 5, 10 → Output: 10
- Input: 7, 3 → Output: 21

Task:

- Examine how examples guide formula selection.
- Test edge cases.



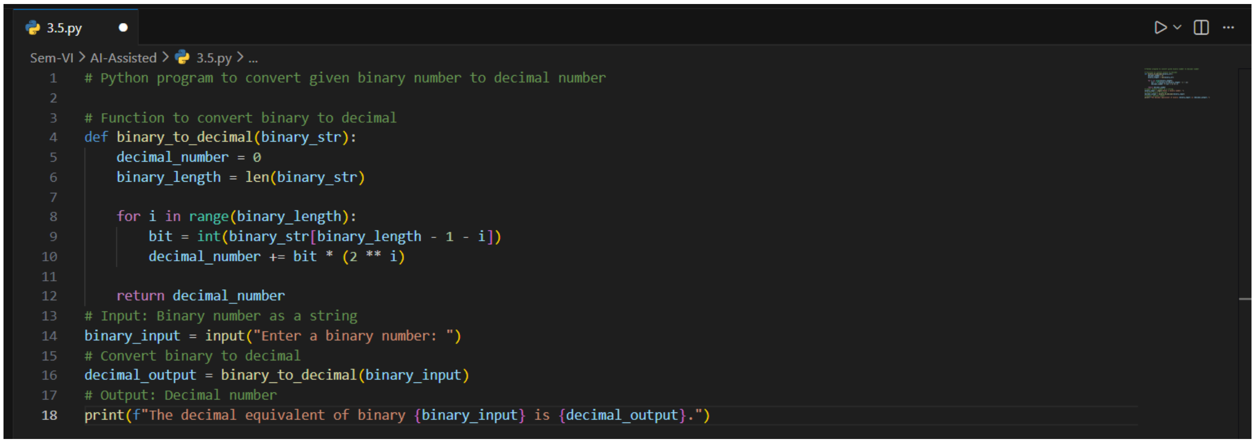
```python
# Python program that finds tha LCM of two given two numbers as user input

def compute_lcm(x, y):
    # choose the greater number
    if x > y:
        greater = x
    else:
        greater = y

    while True:
        if (greater % x == 0) and (greater % y == 0):
            lcm = greater
            break
        greater += 1

    return lcm

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
lcm = compute_lcm(num1, num2)
print(f"The LCM of {num1} and {num2} is {lcm}")
```

Question 4: Zero-Shot Prompting (Binary to Decimal Conversion)

Write a zero-shot prompt to generate a Python function that converts a binary number to decimal.

Task:

- Test with valid and invalid binary inputs.
- Identify missing validation logic.



```python
# Python program to convert given binary number to decimal number

# Function to convert binary to decimal
def binary_to_decimal(binary_str):
    decimal_number = 0
    binary_length = len(binary_str)

    for i in range(binary_length):
        bit = int(binary_str[binary_length - 1 - i])
        decimal_number += bit * (2 ** i)

    return decimal_number
# Input: Binary number as a string
binary_input = input("Enter a binary number: ")
# Convert binary to decimal
decimal_output = binary_to_decimal(binary_input)
# Output: Decimal number
print(f"The decimal equivalent of binary {binary_input} is {decimal_output}.")
```

Question 5: One-Shot Prompting (Decimal to Binary Conversion)

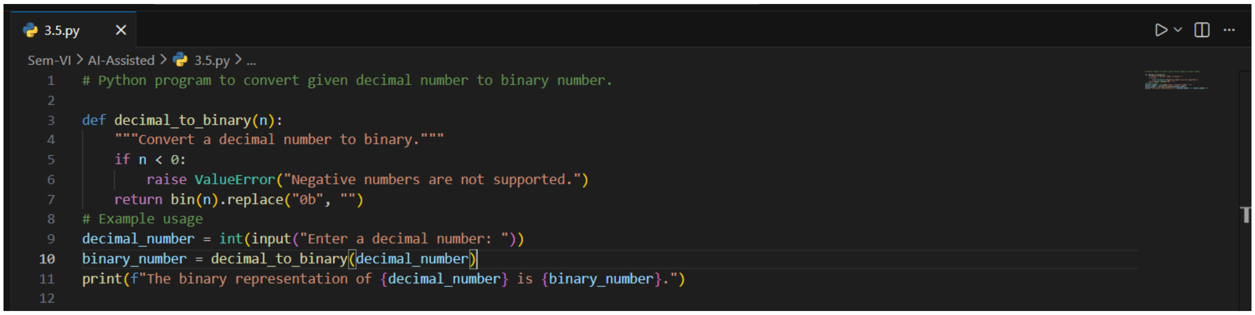
Write a one-shot prompt with an example to generate a Python function that converts a decimal number to binary.

Example:

Input: 10 → Output: 1010

Task:

- Compare clarity with zero-shot output.
- Analyze handling of zero and negative numbers.



```python
# Python program to convert given decimal number to binary number.

def decimal_to_binary(n):
    """Convert a decimal number to binary."""
    if n < 0:
        raise ValueError("Negative numbers are not supported.")
    return bin(n).replace("0b", "")
# Example usage
decimal_number = int(input("Enter a decimal number: "))
binary_number = decimal_to_binary(decimal_number)
print(f"The binary representation of {decimal_number} is {binary_number}.")
```

Question 6: Few-Shot Prompting (Harshad Number Check)

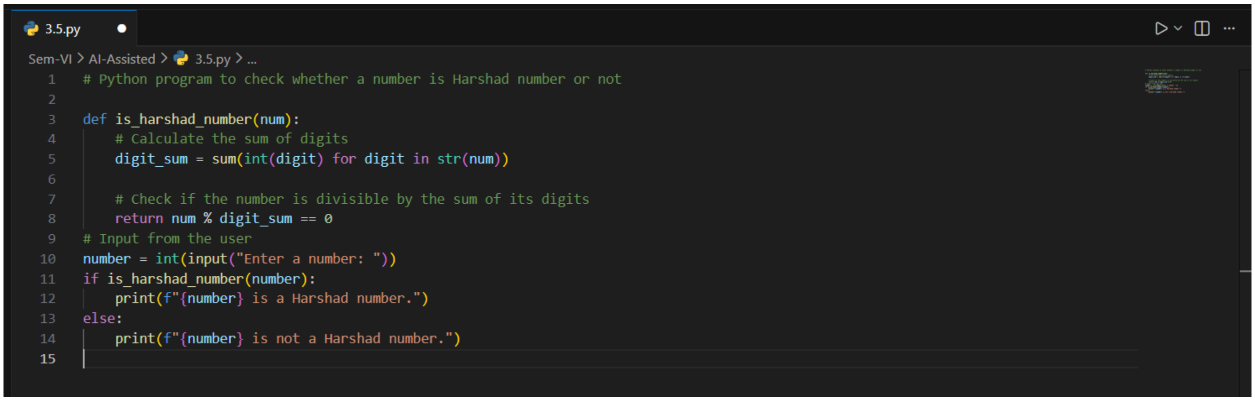
Write a few-shot prompt to generate a Python function that checks whether a number is a Harshad (Niven) number.

Examples:

- Input: 18 → Output: Harshad Number
- Input: 21 → Output: Harshad Number
- Input: 19 → Output: Not a Harshad Number

Task:

- Test boundary conditions.
- Evaluate robustness



```python
# Python program to check whether a number is Harshad number or not

def is_harshad_number(num):
    # Calculate the sum of digits
    digit_sum = sum(int(digit) for digit in str(num))

    # Check if the number is divisible by the sum of its digits
    return num % digit_sum == 0
# Input from the user
number = int(input("Enter a number: "))
if is_harshad_number(number):
    print(f"{number} is a Harshad number.")
else:
    print(f"{number} is not a Harshad number.")
```