ASSINGMENT-3.5

U.VIGNESH

2303A51964

BATCH-24

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.

2303A51964-3.5.py > ...

```python
#write a python function to check if it is a leap year or not by user input check.
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
#the input from user is only a positive number
try:
    year_input = int(input("Enter a leap year: "))
    if year_input < 0:
        print("Please enter a positive number.")
    else:
        if is_leap_year(year_input):
            print(f"{year_input} is a leap year.")
        else:
            print(f"{year_input} is not a leap year.")
except ValueError:
    print("Invalid input. Please enter a valid year.")

^
^

```

OUTPUT:

```
PROBLEMS 6  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13
.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/2303A51964-3.5.py"
Enter a leap year: 2023
2023 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.

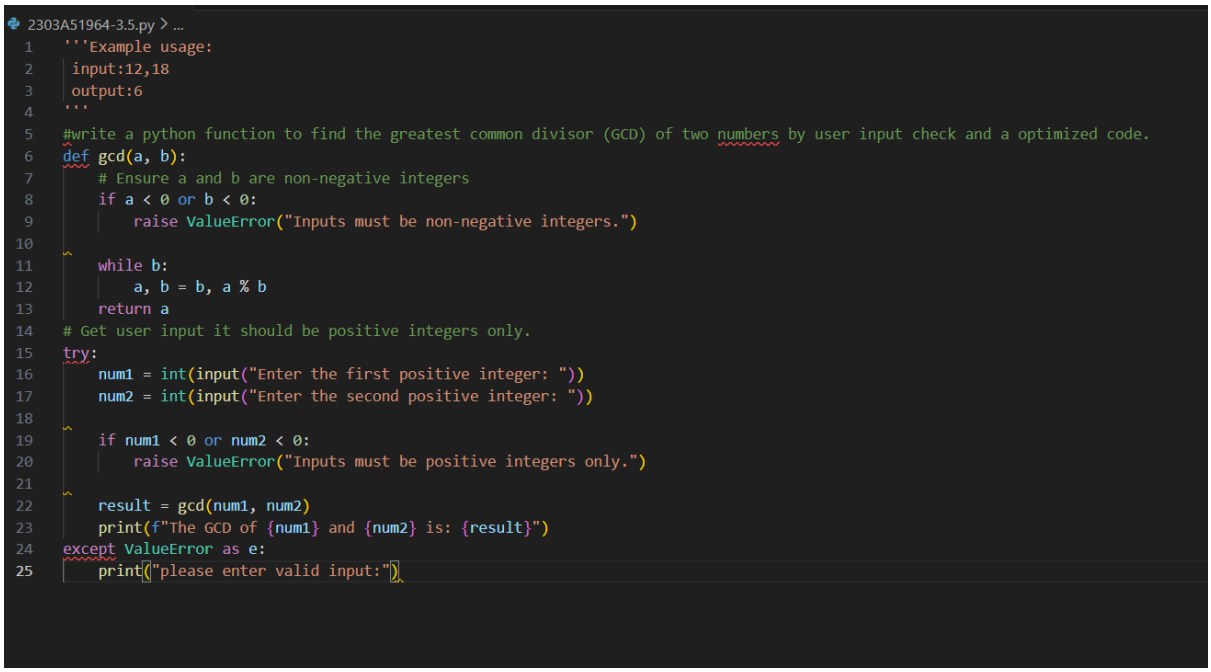
Zeroshot:

```
2303A51964-3.5.py > ...
1   #write a commented python code to find gcd of two numbers.
2   def gcd(a, b):
3       # Function to return gcd of a and b
4       while b:
5           a, b = b, a % b  # Update a and b
6       return a  # When b becomes 0, a contains the gcd
7   #taking user input the input should be a positive integers if the input is invalid it should print please enter a valid positive integer
8   try:
9       num1 = int(input("Enter first positive integer: "))
10      num2 = int(input("Enter second positive integer: "))
11      if num1 <= 0 or num2 <= 0:
12          raise ValueError("Input must be positive integers.")
13      print(f"The GCD of {num1} and {num2} is {gcd(num1, num2)}")
14  except ValueError:
15      print("Please enter a valid positive integer.")
16  
17
```

Output:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Deskto
p/AI ASSESTENT CODING/2303A51964-3.5.py"
Enter first positive integer: 6
Enter second positive integer: 3
The GCD of 6 and 3 is 3
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING>
```

Oneshot:

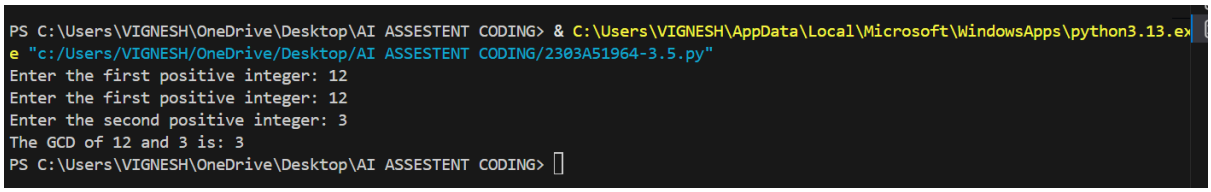


```
2303A51964-3.5.py > ...
'''Example usage:
input:12,18
output:6
'''
#write a python function to find the greatest common divisor (GCD) of two numbers by user input check and a optimized code.
def gcd(a, b):
    # Ensure a and b are non-negative integers
    if a < 0 or b < 0:
        raise ValueError("Inputs must be non-negative integers.")

    while b:
        a, b = b, a % b
    return a
# Get user input it should be positive integers only.
try:
    num1 = int(input("Enter the first positive integer: "))
    num2 = int(input("Enter the second positive integer: "))

    if num1 < 0 or num2 < 0:
        raise ValueError("Inputs must be positive integers only.")

    result = gcd(num1, num2)
    print(f"The GCD of {num1} and {num2} is: {result}")
except ValueError as e:
    print("please enter valid input:")
```



```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.ex
e "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/2303A51964-3.5.py"
Enter the first positive integer: 12
Enter the first positive integer: 12
Enter the second positive integer: 3
The GCD of 12 and 3 is: 3
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING>
```

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.

```python
#write a commented python function that calculates the least common multiple (LCM) of two positive integers use an efficient approach.
def gcd(a, b):
    """Calculate the Greatest Common Divisor (GCD) of two integers using the Euclidean algorithm."""
    while b:
        a, b = b, a % b
    return a
def lcm(x, y):
    """Calculate the Least Common Multiple (LCM) of two positive integers."""
    # LCM can be calculated using the formula: LCM(a, b) = abs(a*b) // GCD(a, b)
    return abs(x * y) // gcd(x, y)
#take a valid positive integer input from the user and print the lcm of the two numbers
try:
    num1 = int(input("Enter the first positive integer: "))
    num2 = int(input("Enter the second positive integer: "))
    if num1 <= 0 or num2 <= 0:
        print("Both numbers must be positive integers.")
    else:
        result = lcm(num1, num2)
        print(f"The Least Common Multiple of {num1} and {num2} is: {result}")
except ValueError as e:
    print("invalid input:")
```

Output:

```
 ASSESTENT CODING/2303A51964-3.5.py"
Enter the first positive integer: -33
Enter the second positive integer: 33
Both numbers must be positive integers.
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Deskto
p/AI ASSESTENT CODING/2303A51964-3.5.py"
Enter the first positive integer: dd
invalid input:
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING>
```

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.

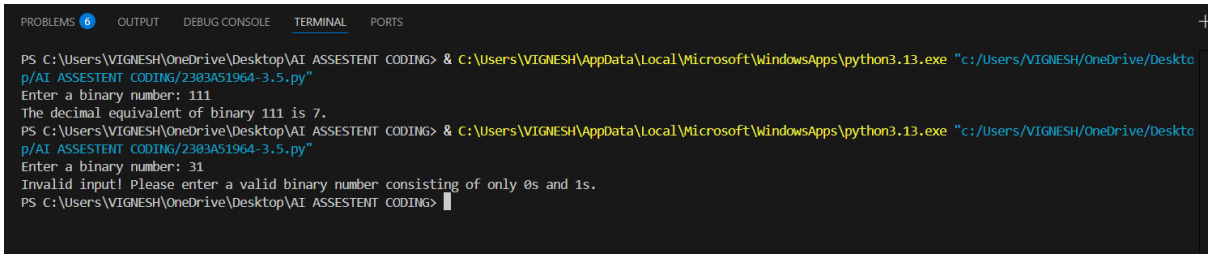
Task:

• Test with valid and invalid binary inputs.

• Identify missing validation logic

```python
#generate a commented python function that converts a binary number to decimal.
def binary_to_decimal(binary_str):
    """Convert a binary string to its decimal equivalent."""
    decimal = 0
    power = 0
    # Iterate through the binary string from right to left
    for digit in reversed(binary_str):
        if digit == '1':
            decimal += 2 ** power
        power += 1
    return decimal
#take user input and test with valid and invalid binary inputs.
if __name__ == "__main__":
    user_input = input("Enter a binary number: ")
    # Validate the input to ensure it is a binary number
    if all(char in '01' for char in user_input):
        decimal_value = binary_to_decimal(user_input)
        print(f"The decimal equivalent of binary {user_input} is {decimal_value}.")
    else:
        print("Invalid input! Please enter a valid binary number consisting of only 0s and 1s.")
```

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.

Oneshot:

```python
#Write a commented Python function named decimal_to_binary(n) that converts a decimal number into binary representation.
'''
Example:
Input: 10
Output: 1010

The function should handle zero correctly and include comments explaining the logic.
'''
def decimal_to_binary(n):    # Check if the input number is zero
    if n == 0:
        return "0"

    # Initialize an empty string to store the binary representation
    binary_representation = ""

    # Use a while loop to convert the decimal number to binary
    while n > 0:
        # Get the remainder when n is divided by 2 (this gives us the least significant bit)
        remainder = n % 2
        # Prepend the remainder to the binary representation string
        binary_representation = str(remainder) + binary_representation
        # Update n by performing integer division by 2
        n = n // 2

    # Return the final binary representation
    return binary_representation
#taking valid user input
try:
    num = int(input("Enter a decimal number: "))
    if num < 0:
        print("Please enter a non-negative integer.")
    else:
        # Call the function and print the result
        print(f"The binary representation of {num} is: {decimal_to_binary(num)}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Output:

```
PROBLEMS 12  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/2303A519
64-3.5.py"
Enter a decimal number: 13
The binary representation of 13 is: 1101
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/2303A519
64-3.5.py"
Enter a decimal number: rq
Invalid input. Please enter a valid integer.
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING>
```

Zeroshot:

```python
#Write a Python function to convert a decimal number into binary by taking user input.
def decimal_to_binary():
    try:
        decimal_number = int(input("Enter a decimal number: "))
        if decimal_number < 0:
            print("Please enter a non-negative integer.")
            return
        binary_number = bin(decimal_number).replace("0b", "")
        print(f"The binary representation of {decimal_number} is {binary_number}.")
    except ValueError:
        print("Invalid input. Please enter a valid integer.")
# Call the function
decimal_to_binary()
```

```
PROBLEMS 5  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/
64-3.5.py"
Enter a decimal number: 132
The binary representation of 132 is 10000100.
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING>
```

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
#Write a Python function named is_harshad(number) that checks whether a given number is a Harshad (Niven) number.
'''
Examples:
Input: 18 → Output: Harshad Number
Input: 21 → Output: Harshad Number
Input: 19 → Output: Not a Harshad Number
'''
def is_harshad(number):
    # Calculate the sum of the digits
    digit_sum = sum(int(digit) for digit in str(number))

    # Check if the number is divisible by the sum of its digits
    if number % digit_sum == 0:
        return "Harshad Number"
    else:
        return "Not a Harshad Number"
# taking a valid positive user input and Test boundary conditions.
try:
    user_input = int(input("Enter a positive integer: "))
    if user_input <= 0:
        print("Please enter a positive integer greater than zero.")
    else:
        result = is_harshad(user_input)
        print(f"Input: {user_input} → Output: {result}")
except ValueError:
    print("Invalid input. Please enter a valid positive integer.")

```

Output:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/2303A51964-3.5.py"
Enter a positive integer: 32
Input: 32 → Output: Not a Harshad Number
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTENT CODING/2303A51964-3.5.py"
Enter a positive integer: -q3
Invalid input. Please enter a valid positive integer.
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING>
```