AI Assistant Coding

B.Bhargava Chary

Batch-24

2303A51747

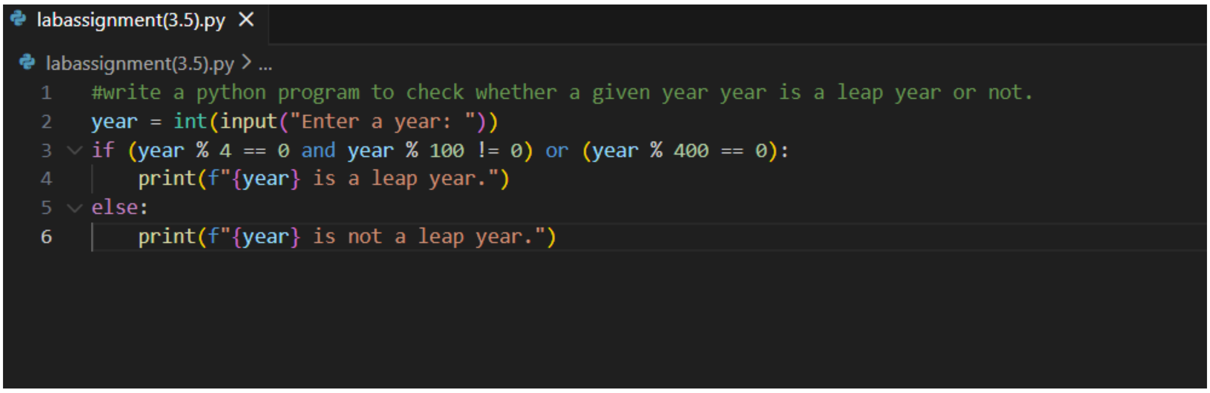
Question 1: Zero-Shot Prompting (Leap Year Check)

Write a zero-shot prompt to generate a Python function that checks

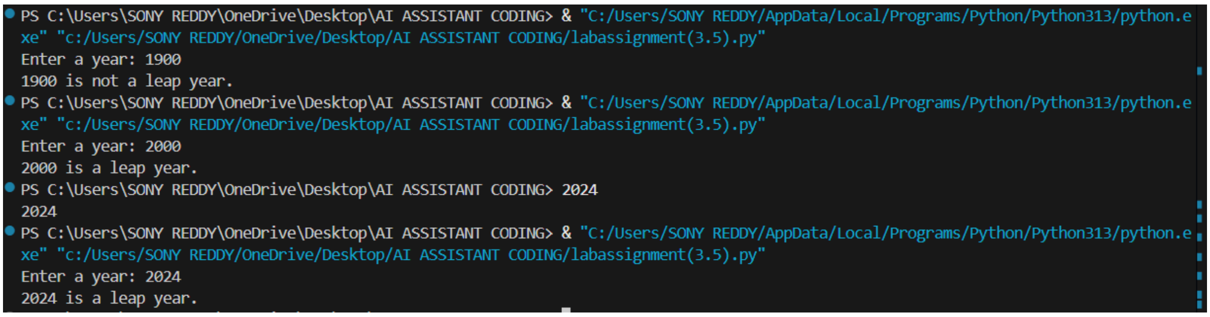
whether a given year is a leap year.

Task:

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



```python
#write a python program to check whether a given year year is a leap year or not.
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```



```
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a year: 1900
1900 is not a leap year.
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a year: 2000
2000 is a leap year.
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> 2024
2024
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a year: 2024
2024 is 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.

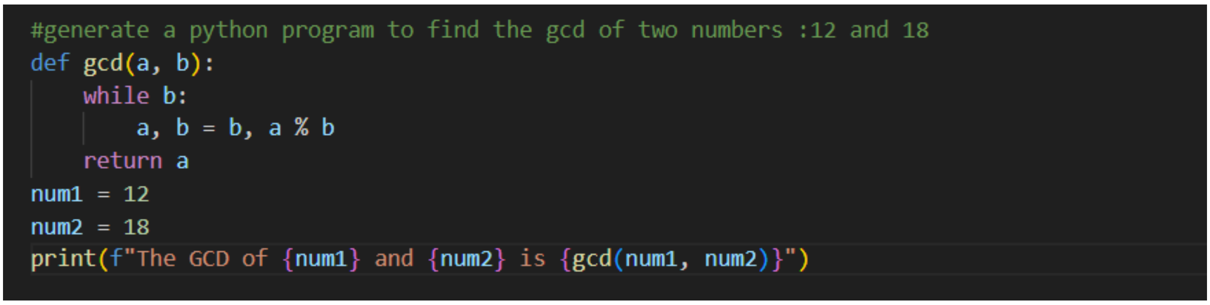
Example:

Input: 12, 18 → Output: 6

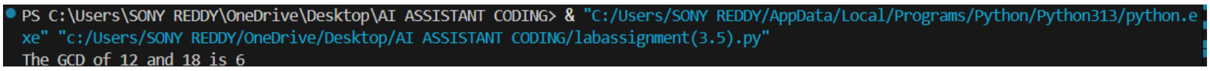
Task:

- Compare with a zero-shot solution.

- Analyze algorithm efficiency.



```python
#generate a python program to find the gcd of two numbers :12 and 18
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
num1 = 12
num2 = 18
print(f"The GCD of {num1} and {num2} is {gcd(num1, num2)}")
```



```
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
The GCD of 12 and 18 is 6
```

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

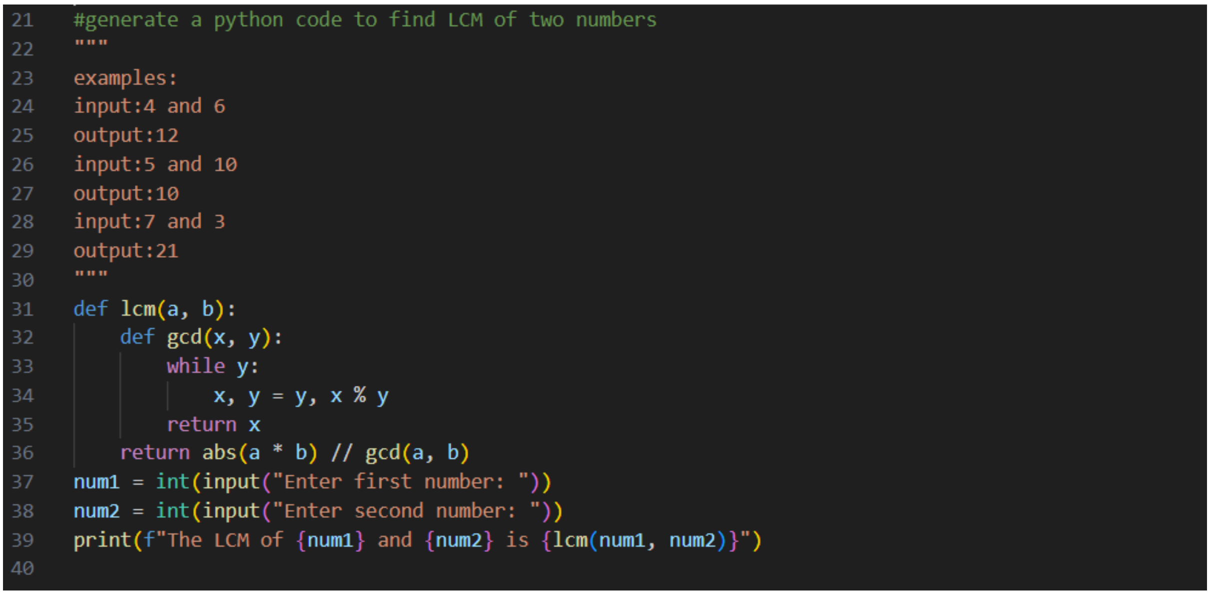
- Input: 5, 10 → Output: 10

- Input: 7, 3 → Output: 21

Task:

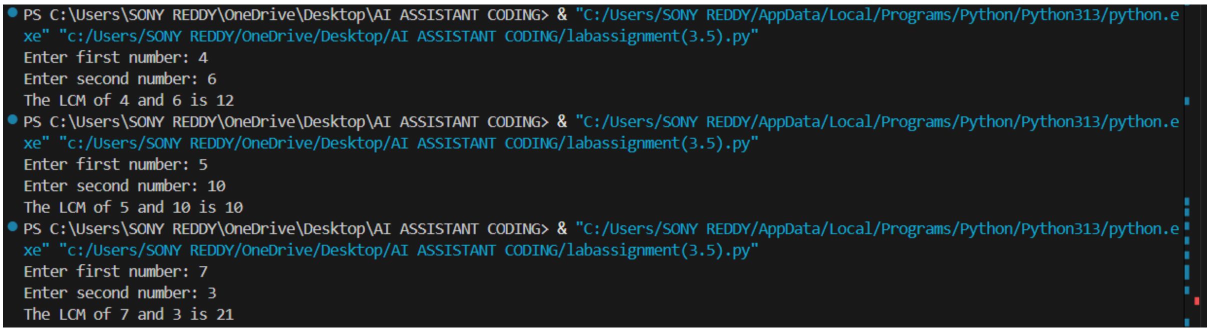
- Examine how examples guide formula selection.

- Test edge cases.



```python
#generate a python code to find LCM of two numbers
"""
examples:
input:4 and 6
output:12
input:5 and 10
output:10
input:7 and 3
output:21
"""
def lcm(a, b):
    def gcd(x, y):
        while y:
            x, y = y, x % y
        return x
    return abs(a * b) // gcd(a, b)
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print(f"The LCM of {num1} and {num2} is {lcm(num1, num2)}")

```



```
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter first number: 4
Enter second number: 6
The LCM of 4 and 6 is 12
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter first number: 5
Enter second number: 10
The LCM of 5 and 10 is 10
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter first number: 7
Enter second number: 3
The LCM of 7 and 3 is 21
```

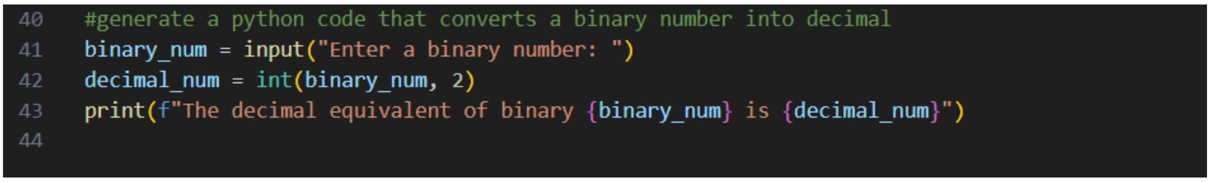
Question 4: Zero-Shot Prompting (Binary to Decimal Conversion)

Write a zero-shot prompt to generate a Python function that converts a

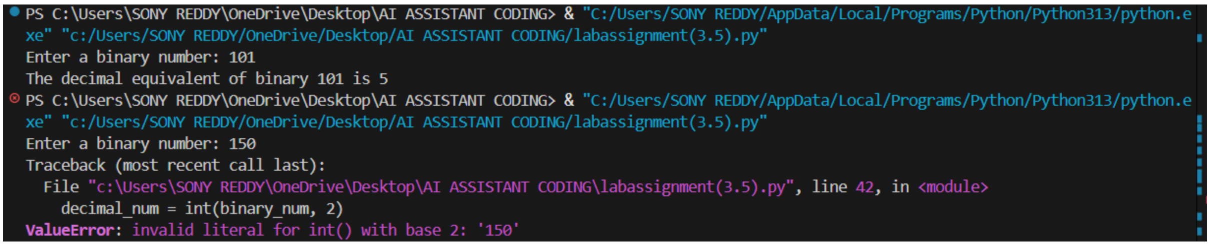
binary number to decimal.

Task:

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



```python
#generate a python code that converts a binary number into decimal
binary_num = input("Enter a binary number: ")
decimal_num = int(binary_num, 2)
print(f"The decimal equivalent of binary {binary_num} is {decimal_num}")
```



```
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a binary number: 101
The decimal equivalent of binary 101 is 5
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a binary number: 150
Traceback (most recent call last):
  File "c:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING\labassignment(3.5).py", line 42, in <module>
    decimal_num = int(binary_num, 2)
ValueError: invalid literal for int() with base 2: '150'
```

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

Write a one-shot prompt with an example to generate a Python function

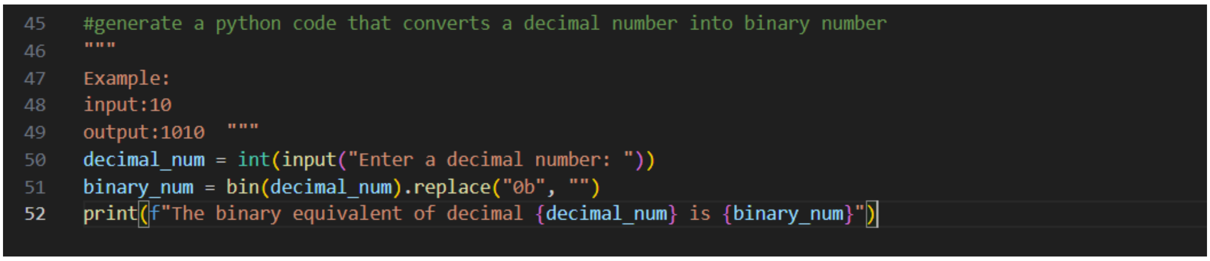
that converts a decimal number to binary.

Example:

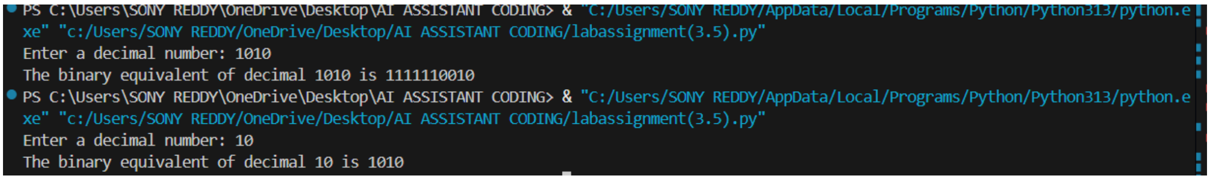
Input: 10 → Output: 1010

Task:

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



```python
#generate a python code that converts a decimal number into binary number
"""
Example:
input:10
output:1010  """
decimal_num = int(input("Enter a decimal number: "))
binary_num = bin(decimal_num).replace("0b", "")
print(f"The binary equivalent of decimal {decimal_num} is {binary_num}")
```



```
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a decimal number: 1010
The binary equivalent of decimal 1010 is 1111110010
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a decimal number: 10
The binary equivalent of decimal 10 is 1010
```

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

Write a few-shot prompt to generate a Python function that checks

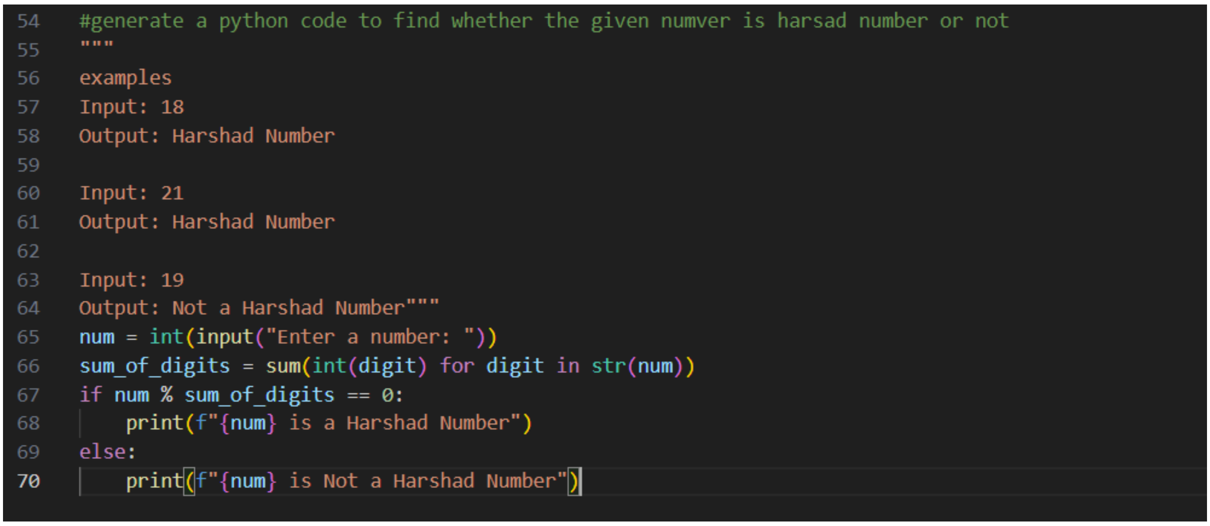
whether a number is a Harshad (Niven) number.

Examples:

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

Task:

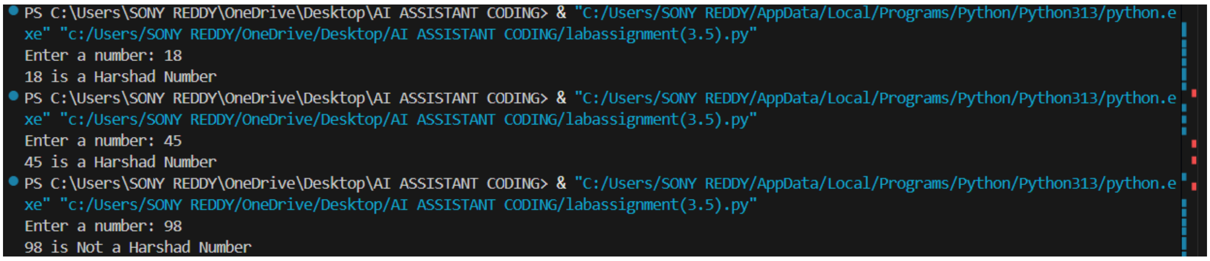
- Test boundary conditions.
- Evaluate robustness



```python
#generate a python code to find whether the given numver is harsad number or not
"""
examples
Input: 18
Output: Harshad Number

Input: 21
Output: Harshad Number

Input: 19
Output: Not a Harshad Number"""
num = int(input("Enter a number: "))
sum_of_digits = sum(int(digit) for digit in str(num))
if num % sum_of_digits == 0:
    print(f"{num} is a Harshad Number")
else:
    print(f"{num} is Not a Harshad Number")
```



```
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a number: 18
18 is a Harshad Number
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a number: 45
45 is a Harshad Number
PS C:\Users\SONY REDDY\OneDrive\Desktop\AI ASSISTANT CODING> & "C:/Users/SONY REDDY/AppData/Local/Programs/Python/Python313/python.e
xe" "c:/Users/SONY REDDY/OneDrive/Desktop/AI ASSISTANT CODING/labassignment(3.5).py"
Enter a number: 98
98 is Not a Harshad Number
```