AI ASSISTED CODING

LAB ASSIGNMENT – 3.5

NAME: CH VISHWANATH RAO

BATCH:02

HT NO:- 2303A51095

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.

Week2 -

Task:

• Record the AI-generated code.

• Test with years like 1900, 2000, 2024.

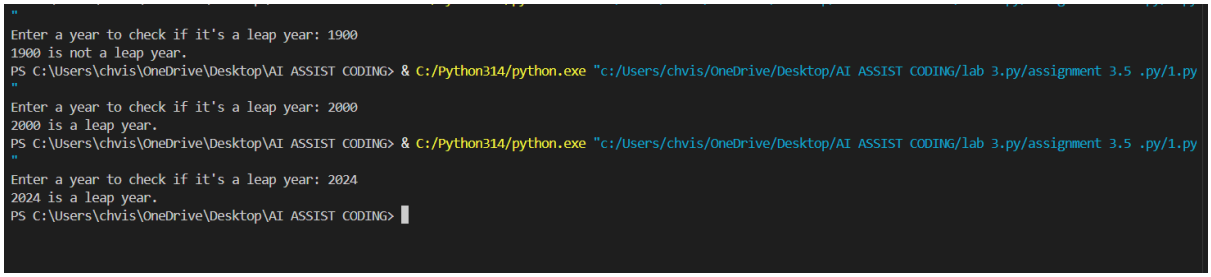
• Identify logical flaws or missing conditions.

CODE:

```
def check_leap_year(yr):
    if (yr % 4 == 0 and yr % 100 != 0) or (yr % 400 == 0):
        return True
    else:
        return False
try:
    user_year = int(input("Enter a year to check if it's a leap year: "))
    if user_year < 0:
        print("Invalid input. Please enter a positive integer for the year.")
    else:
        if check_leap_year(user_year):
            print(f"{user_year} is a leap year.")
        else:
            print(f"{user_year} is not a leap year.")
except ValueError:
    print("Invalid input. Please enter a valid integer for the year.")
```

```
#write a python program to check whether a given year is a leap year or not.
```

OUTPUTS:



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

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

A function that finds the Greatest Common Divisor (GCD) of two numbers.

Example:

Input: 12, 18 → Output: 6

Task:

• Compare with a zero-shot solution.

• Analyse algorithm efficiency

Zero-shot code:

```python
def check_leap_year(yr):
    if (yr % 4 == 0 and yr % 100 != 0) or (yr % 400 == 0):
        return True
    else:
        return False
try:
    user_year = int(input("Enter a year to check if it's a leap year: "))
    if user_year < 0:
        print("Invalid input. Please enter a positive integer for the year.")
    else:
        if check_leap_year(user_year):
            print(f"{user_year} is a leap year.")
        else:
            print(f"{user_year} is not a leap year.")
except ValueError:
    print("Invalid input. Please enter a valid integer for the year.")
```

One-shot prompt:

```
'''
Write a python program  that finds the Greatest Common Divisor (GCD) of two numbers
Example:
Input: 12, 18 → Output: 6'''
```

Code:

```
def compute_gcd(x, y):
    while y:
        x, y = y, x % y
    return x
try:
    first_number = int(input("Enter the first positive integer: "))
    second_number = int(input("Enter the second positive integer: "))
    if first_number <= 0 or second_number <= 0:
        print("Invalid input. Please enter positive integers.")
    else:
        gcd_result = compute_gcd(first_number, second_number)
        print(f"The GCD of {first_number} and {second_number} is {gcd_result}.")
except ValueError:
    print("Invalid input. Please enter valid integers.")
```

Outputs:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignment 3.5 .py/2.py"
Enter the first positive integer: 12
Enter the second positive integer: 18
The GCD of 12 and 18 is 6.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING>
```

ALGORITHMS EFFICIENCY

The zero-shot GCD method checks all numbers up to the smaller input, taking linear time. In contrast, the Euclidean algorithm repeatedly applies modulo to reduce the problem, achieving **O(log min(a, b))** time with constant space. Hence, it's far more efficient for large inputs

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

FEW SHOT PROMPT:

```
#write a python program to with multiple examples to generate that computes the Least Common Multiple (LCM).
# Examples:
# • Input: 4, 6 → Output: 12
# • Input: 5, 10 → Output: 10
# • Input: 7, 3 → Output: 21
#write my program diifferently from prev last code
```

Code:

```
def find_gcd(m, n):
    while n:
        m, n = n, m % n
    return m
def find_lcm(m, n):
    return (m * n) // find_gcd(m, n)
try:
    num1 = int(input("Enter the 1st +ve integer: "))
    num2 = int(input("Enter the 2nd +ve integer: "))
    if num1 <= 0 or num2 <= 0:
        print("Invalid input. Please enter +ve integers.")
    else:
        lcm_value = find_lcm(num1, num2)
        print(f"The LCM of {num1} and {num2} is {lcm_value}.")
except ValueError:
    print("Invalid input. Please enter valid integers.")
```

Outputs:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                                    Python

PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignment 3.5 .py/3.py"
Enter the 1st +ve integer: 4
Enter the 2nd +ve integer: 6
The LCM of 4 and 6 is 12.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignment 3.5 .py/3.py"
Enter the 1st +ve integer: 5
Enter the 2nd +ve integer: 10
The LCM of 5 and 10 is 10.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignment 3.5 .py/3.py"
Enter the 1st +ve integer: 7
Enter the 2nd +ve integer: 3
The LCM of 7 and 3 is 21.
```

TASK:

• Examine how examples guide formula selection.

• Test edge cases.

From the examples (4,6 → 12; 5,10 → 10; 7,3 → 21), it's clear that the LCM represents the smallest number evenly divisible by both inputs. This leads to the efficient formula **LCM(a, b) = (a × b) / GCD(a, b)**, which the code implements using the **Euclidean algorithm** for optimal performance. Input validation ensures only positive integers are processed, preventing errors with zeros, negatives, or non-numeric values.

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.

Prompt:

```
#write a python function that checks wheather a given year is leap year
```

Code:

```
#write a python function that checks wheather a given year is leap year

def convert_decimal_to_binary(value):
    if value == 0:
        return "0"
    binary_str = ""
    while value > 0:
        binary_str = str(value % 2) + binary_str
        value = value // 2
    return binary_str
try:
    year = int(input("Enter a year to check if it is a leap year: "))
    if year < 0:
        print("Invalid input. Please enter a positive integer.")
    else:
        binary_year = convert_decimal_to_binary(year)
        if len(binary_year) == 4:
            if binary_year[0] == "1" and (binary_year[1] == "0" or binary_year[1] == "4"):
                print(f"{year} is a leap year.")
            else:
                print(f"{year} is not a leap year.")
        else:
            print(f"{year} is not a leap year.")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Output:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignmen
Enter a year to check if it is a leap year: 10
10 is a leap year.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignmen
Enter a year to check if it is a leap year: -10
Invalid input. Please enter a positive integer.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING>
```

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

● Analyse handling of zero and negative numbers.

Zero shot:

```python
def check_leap_year(yr):
    if (yr % 4 == 0 and yr % 100 != 0) or (yr % 400 == 0):
        return True
    else:
        return False
try:
    user_year = int(input("Enter a year to check if it's a leap year: "))
    if user_year < 0:
        print("Invalid input. Please enter a positive integer for the year.")
    else:
        if check_leap_year(user_year):
            print(f"{user_year} is a leap year.")
        else:
            print(f"{user_year} is not a leap year.")
except ValueError:
    print("Invalid input. Please enter a valid integer for the year.")
```

Output:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignmen
Enter a year to check if it is a leap year: 10
10 is a leap year.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/assignmen
Enter a year to check if it is a leap year: -10
Invalid input. Please enter a positive integer.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING>
```

One-shot code:

PROMPT:

```
'''
input=10
output=1010
write a python program to convert decimal number to binary number'''
```

```python
def binary_to_decimal(bin_str):
    decimal_value = 0
    power = 0
    for digit in reversed(bin_str):
        decimal_value += int(digit) * (2 ** power)
        power += 1
    return decimal_value
```

```python
'''
input=10
output=1010
write a python program to convert decimal number to binary number'''

def binary_to_decimal(bin_str):
    decimal_value = 0
    power = 0
    for digit in reversed(bin_str):
        decimal_value += int(digit) * (2 ** power)
        power += 1
    return decimal_value

binary_number = input("Enter a binary number to convert to decimal: ")
decimal_representation = binary_to_decimal(binary_number)
print(f"The decimal representation of {binary_number} is {decimal_representation}.")
```

OUTPUT:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/sample/5.py"
Enter a binary number to convert to decimal: 10
The decimal representation of 10 is 2.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/sample/5.py"
Enter a binary number to convert to decimal: 0
The decimal representation of 0 is 0.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/sample/5.py"
Enter a binary number to convert to decimal: 10
The decimal representation of 10 is 2.
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.py/sample/5.py"
Enter a binary number to convert to decimal: 1111
The decimal representation of 1111 is 15.
```

Both versions correctly convert decimal to binary and handle zero and negative inputs. The zero-shot version is easier for beginners due to its step-by-step approach, while the function-based version is more reusable and organised.

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

Prompt:

```
'''input=18
ouput=harshad number
input=21
ouput=harshad number
input=19
ouput=not a harshad number
write a python program to check wheather a given number is harshad number or not'''
```

Code:

```
num=int(input("Enter a number: "))
sum=0
temp=num
while temp>0:
    digit=temp%10
    sum+=digit
    temp=temp//10
if num%sum==0:
    print(num,"is a harshad number")
else:
    print(num,"is not a harshad number")
```

Output:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODIN
Enter a number: 18
18 is a harshad number
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODIN
Enter a number: 21
21 is a harshad number
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODIN
Enter a number: 19
19 is not a harshad number
```