2303A51214
BATCH:-4

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.

PROMPT:-Write a Python function to check whether a given year is a leap year.

CODE:

```
def is_leap_year(year):
    if year % 4 == 0:
        return True
    else:
        return False
```

TESTING:

```
print(is_leap_year(1900))  Expected: False
print(is_leap_year(2000))  Expected: True
print(is_leap_year(2024))  Expected: True
```

Observed Output

```
True
True
True
```

Missing conditions:

- Years divisible by **100 are NOT leap years**
- Except years divisible by **400**

Correct Logic:

```
def is_leap_year(year):
    if year % 400 == 0:
        return True
    if year % 100 == 0:
        return False
    if year % 4 == 0:
        return True
    return False
```

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.

One-Shot Prompt:
Write a Python function to find the GCD of two numbers.

Example:
Input: 12, 18
Output: 6

CODE:

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a
```

Zero-Shot Comparison:

```
def gcd(a, b):
    for i in range(1, min(a, b) + 1):
        if a % i == 0 and b % i == 0:
            g = i
    return g
```

Algorithm Efficiency

Method	**Time Complexity**
Loop method	O(min(a, b))
Euclidean Algorithm	O(log n)

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

.
Few-Shot Prompt:
Write a Python function to calculate LCM.

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

AI-Generated Code:

```
def lcm(a, b):
    def gcd(x, y):
        while y:
            x, y = y, x % y
        return x
    return (a * b) // gcd(a, b)
```

How Examples Guided Formula Selection

AI selected **LCM = (a × b) / GCD**
Few-shot examples prevent brute-force solutions

Edge Case Testing

```
print(lcm(0, 5))   # 0
print(lcm(1, 7))   # 7
```

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.

Zero-Shot Prompt:
Write a Python function to convert a binary number to decimal.

CODE:

```
def binary_to_decimal(b):
    return int(b, 2)
```

Testing:

```
print(binary_to_decimal("1010"))  # 10
print(binary_to_decimal("1021"))  # Error
```

Missing Validation:_
Improved Version:

```
def binary_to_decimal(b):
    if not all(ch in '01' for ch in b):
        return "Invalid binary number"
    return int(b, 2)
```

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.

One-Shot Prompting – Decimal to Binary:
Write a Python function to convert decimal to binary.

Example:
Input: 10
Output: 1010

CODE:

```
def decimal_to_binary(n):
    return bin(n)[2:]
```

Edge Case Testing:

```
print(decimal_to_binary(0))   # 0
print(decimal_to_binary(-5))  # b101
```

Improved Handling:-

```
def decimal_to_binary(n):
    if n < 0:
        return "Negative numbers not supported"
    return bin(n)[2:]
```

Analysis

One-shot improves clarity
Zero-shot often ignores negatives and zero

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

Few-Shot Prompt:
Write a Python function to check whether a number is a Harshad number.

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

Code:

```
def is_harshad(n):
    s = sum(int(d) for d in str(n))
    if n % s == 0:
        return "Harshad Number"
    else:
        return "Not a Harshad Number"
```

Boundary Testing:

```
print(is_harshad(1))   # Harshad
print(is_harshad(0))   # Error (division by zero)
```

Robust Version:

```
def is_harshad(n):
    if n <= 0:
        return "Invalid input"
    s = sum(int(d) for d in str(n))
    return "Harshad Number" if n % s == 0 else "Not a Harshad Number"
```

Overall Conclusion:

Prompt Type	**Quality**
Zero-Shot	Simple but often incomplete
Oneshot	Better logic, clearer intent
Few-Shot	Most accurate and robust