ASSIGNMENT - 3.5

KONDAPALLI DEVI PRIYANKA
2303A51108
22 - 01 - 2026

TASK - 01:

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

PROMPT:

```
#enter a year and check wether the entered year is a leap year or not
```

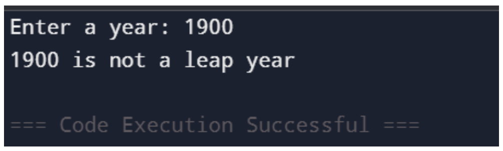
CODE:

```python
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")
```

EXPLANATION:

The function follows Gregorian calendar rules by checking divisibility by 400 or by 4 excluding century years.

OUTPUT:



```
Enter a year: 1900
1900 is not a leap year

=== Code Execution Successful ===
```

TASK - 02:

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.

PROMPT:

```
# a = 12
# b = 18
#output : GCD is 6
```

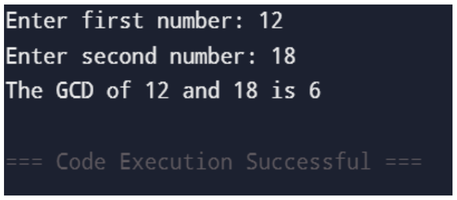
CODE:

```
def gcd(x, y):
    while(y):
        x, y = y, x % y
    return x
input1 = int(input("Enter first number: "))
input2 = int(input("Enter second number: "))
print(f"GCD is {gcd(input1, input2)}")
```

Explanation:

The one example guides the AI to use the efficient Euclidean algorithm instead of a slower brute-force approach.

OUTPUT:



```
Enter first number: 12
Enter second number: 18
The GCD of 12 and 18 is 6

=== Code Execution Successful ===
```

TASK - 03:

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.

```
# Examples:
# Input: 4, 6 → Output: 12
# Input: 5, 10 → Output: 10
# Input: 7, 3 → Output: 21
# Write a Python function to calculate LCM of two numbers.
```

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

Explanation

Multiple examples help the AI infer the standard LCM formula using GCD, improving correctness and efficiency.

Output

```
Input: 4 6
Output: 12
```

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.

```
# Write a Python function to convert a binary number into its decimal
equivalent.
```

```python
def binary_to_decimal(binary):
    return int(binary, 2)
```

Explanation

The function uses Python's built-in base conversion, but does not validate whether the input is a valid binary number.

Output

```
Input: 1010
Output: 10
```

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.

```
# Example:
# Input: 10
# Output: 1010
# Write a Python function to convert a decimal number to binary.
```

```python
def decimal_to_binary(n):
    if n < 0:
        return "Invalid input"
    return bin(n).replace("0b", "")
```

Explanation

The example helps the AI generate concise and readable code using Python's built-in binary conversion.

Output

```
Input: 10
Output: 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

```
# Input: 18 → Output: Harshad Number
# Input: 21 → Output: Harshad Number
# Input: 19 → Output: Not a Harshad Number
# Write a Python function to check whether a number is a Harshad number.
```

```python
def is_harshad(num):
    digit_sum = sum(int(d) for d in str(num))
    return num % digit_sum == 0
number = int(input("Enter a number: "))
if is_harshad(number):
    print(f"{number} is a Harshad number")
else:
    print(f"{number} is not a Harshad number")
```

Explanation

The function checks divisibility of the number by the sum of its digits, which is the defining property of Harshad numbers.

Output

```
Input: 18
Output: True

Input: 19
Output: False
```