2303A51193

Batch:04

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
Course Coordinator Name	Dr. Rishabh Mittal		
Instructor(s) Name	Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju		
CourseCode	23CS002PC304	**Course Title**	AI Assisted Coding
Year/Sem	III/II	**Regulation**	R23
Date and Day of Assignment	**Week2 –**	**Time(s)**	23CSBTB01 To 23CSBTB52
Duration	2 Hours	**Applicable to Batches**	All batches
Assignment Number: 4.1(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	***Expected Time to complete***
1	Question 1: Zero-Shot Prompting (Leap Year Check)	Week2 -

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.

Prompt:

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

```python
def is_leap_year(year):
    """
    Function to check whether a given year is a leap year.
    """
    if year % 400 == 0:
        return True
    elif year % 100 == 0:
        return False
    elif year % 4 == 0:
        return True
    else:
        return False


# Test cases
years = [1900, 2000, 2024]

for y in years:
    if is_leap_year(y):
        print(f"{y} is a Leap Year")
    else:
        print(f"{y} is NOT a Leap Year")
```

Output:

```
PS C:\Users\madha\OneDrive\Desktop\full_stack_2026> & C:/Users/madha/AppData/Local/Microsoft/Wi
1900 is NOT a Leap Year
2000 is a Leap Year
2024 is a Leap Year
PS C:\Users\madha\OneDrive\Desktop\full_stack_2026>
```

Explanation

- This is a zero-shot prompt because no examples are given.
- The AI must rely only on its general knowledge of leap year rules.

- Leap year logic:
  - Divisible by 4 ✓
  - Not divisible by 100 ✗
  - Unless divisible by 400 ✓

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.

Prompt

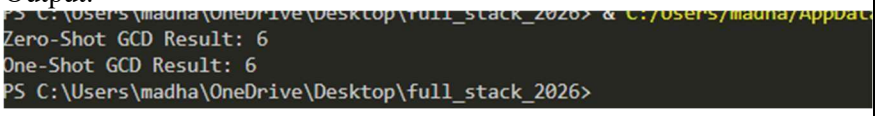
Write a Python function to find the Greatest Common Divisor (GCD) of two numbers.

Code:

```python
def gcd_zero_shot(a, b):
    gcd = 1
    for i in range(1, min(a, b) + 1):
        if a % i == 0 and b % i == 0:
            gcd = i
    return gcd
def gcd_one_shot(a, b):
    while b != 0:
        a, b = b, a % b
    return a
# Test input
x = 12
y = 18

print("Zero-Shot GCD Result:", gcd_zero_shot(x, y))
print("One-Shot GCD Result:", gcd_one_shot(x, y))
```

Output:



Explanation

- This is one-shot prompting because one example is provided.
- The example helps the AI infer: o Expected input format o Correct output

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.

Prompt

Write a Python function to compute the Least Common Multiple (LCM) of two numbers.

Code:

```python
def gcd(x, y):
    while y != 0:
        x, y = y, x % y
    return x
def lcm(a, b):
    return (a * b) // gcd(a, b)
# Test cases from the prompt
test_cases = [(4, 6), (5, 10), (7, 3)]

print("LCM Results:")
for a, b in test_cases:
    print(f"LCM of {a} and {b} is {lcm(a, b)}")
# Edge case testing
print("\nEdge Case Testing:")
edge_cases = [(0, 5), (5, 0), (1, 1), (9, 9)]

for a, b in edge_cases:
    print(f"LCM of {a} and {b} is {lcm(a, b)}")
```

Output:

```
LCM Results:
LCM of 4 and 6 is 12
LCM of 5 and 10 is 10
LCM of 7 and 3 is 21

Edge Case Testing:
LCM of 0 and 5 is 0
LCM of 5 and 0 is 0
LCM of 1 and 1 is 1
LCM of 9 and 9 is 9
```

Explanation

- This is few-shot prompting because multiple examples are given.
- The AI clearly identifies the pattern:
    - LCM(a, b) = (a × b) // GCD(a, b)

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 converts a binary number to its decimal equivalent.

Code:

```
def binary_to_decimal(binary):
    return int(binary, 2)
test_inputs = ["1010", "1111", "1001", "1021", "abc", ""]

print("Binary to Decimal Conversion Results:")
for b in test_inputs:
    try:
        result = binary_to_decimal(b)
        print(f"Binary: {b} → Decimal: {result}")
    except ValueError:
        print(f"Binary: {b} → Invalid binary input")
```

Output:

```
Binary to Decimal Conversion Results:
Binary: 1010 → Decimal: 10
Binary: 1111 → Decimal: 15
Binary: 1001 → Decimal: 9
Binary: 1021 → Invalid binary input
Binary: abc → Invalid binary input
Binary:  → Invalid binary input
```

Explanation

- Zero-shot prompt → no examples.
- AI usually:
  - Uses int(binary, 2)
  - Or manual base-2 calculation

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.

Prompt

Write a Python function that converts a decimal number to binary.

Code:

```
def decimal_to_binary(n):
    if n == 0:
        return "0"

    is_negative = n < 0
    n = abs(n)

    binary = ""
    while n > 0:
        binary = str(n % 2) + binary
        n = n // 2

    if is_negative:
        binary = "-" + binary

    return binary
test_values = [10, 0, -10, 7, 1]

print("Decimal to Binary Conversion Results:")
for val in test_values:
    print(f"Decimal: {val} → Binary: {decimal_to_binary(val)}")
```

Output:

```
Decimal to Binary Conversion Results:
Decimal: 10 → Binary: 1010
Decimal: 0 → Binary: 0
Decimal: -10 → Binary: -1010
Decimal: 7 → Binary: 111
Decimal: 1 → Binary: 1
```

Explanation

- One example clarifies:

	o Output format (no 0b prefix) □ AI often uses: o bin(n)[2:] o Or repeated division method	
	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 Write a Python function to check whether a number is a Harshad (Niven) number. Code:	

```python
def is_harshad(n):
    if n <= 0:
        return "Not a Harshad Number"

    digit_sum = sum(int(d) for d in str(n))

    if digit_sum == 0:
        return "Not a Harshad Number"

    if n % digit_sum == 0:
        return "Harshad Number"
    else:
        return "Not a Harshad Number"
test_cases = [18, 21, 19]

print("Harshad Number Test Results:")
for num in test_cases:
    print(f"Input: {num} → Output: {is_harshad(num)}")
print("\nBoundary / Edge Case Testing:")
edge_cases = [1, 0, -18, 10, 100, 11]

for num in edge_cases:
    print(f"Input: {num} → Output: {is_harshad(num)}")
```

Output:

```
Harshad Number Test Results:
Input: 18 → Output: Harshad Number
Input: 21 → Output: Harshad Number
Input: 19 → Output: Not a Harshad Number

Boundary / Edge Case Testing:
Input: 1 → Output: Harshad Number
Input: 0 → Output: Not a Harshad Number
Input: -18 → Output: Not a Harshad Number
Input: 10 → Output: Harshad Number
Input: 100 → Output: Harshad Number
Input: 11 → Output: Not a Harshad Number
```

Explanation

- Few-shot prompting helps AI understand:
  - Harshad number definition
  - Output wording
