

AI-Assisted Coding

Assignment-3.5

Name: A.Sai Shrehan

Batch:45

Roll no:2303A52299

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.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files: assignment_3.5.py, assignment_3.5.py (intellisense), _idea, and main.py.
- Code Editor:** Displays the Python code for a leap year check. The code includes a docstring and logic for determining leap years based on the Gregorian calendar rules.
- Terminal:** Shows command-line interactions where the AI-generated code is run for years 1900, 2000, and 2024, with the output indicating whether each is a leap year.
- IntelliSense/Chat Panel:** A large panel on the right provides AI-generated documentation for the function, including its purpose, arguments, return value, and a detailed explanation of the leap year logic.

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

The screenshot shows a code editor interface with the following details:

- File Structure:** Explorer shows files: assignment_3.5.py, assignment_3.5.ipynb, and main.py.
- Code Editor:** The file assignment_3.5.py contains the following code:

```
def find_gcd(a: int, b: int) -> int:
    """
    Calculates the Greatest Common Divisor (GCD) of two integers using Euclid's algorithm.

    Args:
        a (int): The first integer.
        b (int): The second integer.

    Returns:
        int: The GCD of a and b.
    """
    while b != 0:
        a, b = b, a % b
    return a

# Example usage:
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
gcd = find_gcd(num1, num2)
print(f"The GCD of {num1} and {num2} is {gcd}.")
```

- Terminal:** Shows command-line interactions for testing the function.
- AI Chat:** A sidebar titled "LEAP YEAR FUNCTION IN PYTHON WITH AI ASSISTED CODING" provides instructions and examples related to generating a leap year function.
- Bottom Status Bar:** Shows file path (D:/AI assisted coding/assignment_3.5.py), line (Ln 30, Col 1), and encoding (UTF-8).

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

The screenshot shows the PyCharm IDE interface with the "AI ASSISTED CODING" feature open. The left sidebar shows files: assignment_3.5.py, main.py, and .idea. The main editor window displays the code for assignment_3.5.py, which defines a function calculate_lcm(a: int, b: int) -> int. The code uses the Euclidean algorithm to find the GCD and then calculates the LCM. A tooltip provides documentation for the function. The right panel shows a "CHAT" window with a message from the AI suggesting a formula and a reference to existing code. The bottom status bar indicates the code is in Python 3.14.2 and UTF-8 encoding.

```
def calculate_lcm(a: int, b: int) -> int:
    """
    The LCM is the smallest positive integer that is a multiple of both a and b.

    Args:
        a (int): The first integer.
        b (int): The second integer.

    Returns:
        int: The LCM of a and b.
    """
    if a == 0 or b == 0:
        return 0 # LCM is undefined for zero, but per examples, assume positive integers
    gcd = find_gcd(a, b)
    return abs(a * b) // gcd

a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
lcm = calculate_lcm(a, b)
print(f"The LCM of {a} and {b} is {lcm}.")
```

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.

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files `assignment_3.5.py`, `main.py`, and `.idea`.
- Code Editor:** Displays the `assignment_3.5.py` file content:

```
def decimal_to_binary(decimal: int) -> str:
    """
    Returns:
        str: The binary representation as a string.

    Args:
        decimal (int): The decimal number to convert.

    Returns:
        str: The binary representation as a string.

    Examples:
        decimal = 10
        binary_rep = decimal_to_binary(decimal)
        print(f"The binary representation of {decimal} is {binary_rep}.")
```
- Terminal:** Shows command-line interactions:

```
PS D:\AI assited coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assited coding/assignment_3.5.py"
Enter the second number: 3
The LCM of 7 and 3 is 21.
PS D:\AI assited coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assited coding/assignment_3.5.py"
Enter a binary number: 1010
The decimal equivalent of 1010 is 10.
PS D:\AI assited coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assited coding/assignment_3.5.py"
Enter a decimal number: 10
The binary representation of 10 is 1010.
PS D:\AI assited coding>
```
- CHAT Panel:** Contains a task description:

```
Write a Python function named decimal_to_binary that accepts an integer and returns its binary representation as a string. Use the example below to understand the expected output format.
```

 and examples:

```
Example: Input: decimal_to_binary(10)
Output: "1010"
```

 and notes:

```
Task: Ensure the function handles the integer 0 correctly and returns a clean string without Python's default prefix."
```
- Bottom Status Bar:** Shows file information: `Ln 100, Col 70`, `Spaces:4`, `UTF-8`, `CRLF`, `() Python`, `3.14.2`, `Go Live`.

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

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files: assignment_3.5.py, assignment_3.5.py > ..., and main.py.
- Code Editor:** Displays the content of assignment_3.5.py. The code defines a function `check_harshad_number` that takes an integer `num` and returns `True` if it is a Harshad number (divisible by its sum of digits) and `False` otherwise. It includes a docstring and example usage.
- Terminal:** Shows command-line interactions with the code. It prompts for input, runs the function for inputs 18 and 21, and prints the results.
- CHAT Panel:** A sidebar titled "AI ASSISTED CODING" contains a conversation log. The user asks for a Harshad number check function and receives a template response. The template includes a docstring, variable descriptions, and a note about Harshad numbers being typically positive integers.