

AI-Assisted Coding

Assignment-3.5

Name: MD .Abdul Layeeq

Batch:45

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 a VS Code interface with the following details:

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

```
def is_leap_year(year: int) -> bool:
    """
    Args:
        year (int): The year to check.

    Returns:
        bool: True if the year is a leap year, False otherwise.

    """
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False

    # Example usage:
year = int(input("Enter a year: "))
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```
- Terminal:** Shows command-line interaction with Python 3.14.2:

```
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 year: 1900
1900 is not a leap year.
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 year: 2000
2000 is a leap year.
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 year: 2024
2024 is a leap year.
PS D:\AI assited coding>
```
- Right-hand pane:** An AI assistant window titled "LEAP YEAR FUNCTION IN PYTHON WITH AI ASSISTED CODING" provides the following information:
 - Write a Python function named `is_leap_year` that accepts an integer representing a year. The function should return True if the year is a leap year according to the Gregorian calendar rules (divisible by 4, but not by 100 unless also divisible by 400), and False otherwise.
 - Include type hinting and a docstring.

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 the Visual Studio Code interface with the following details:

- Explorer View:** Shows files: assignment_3.5.py, find_gcd, main.py, and idea.
- Editor View:** Displays the code for `assignment_3.5.py` containing a `find_gcd` function implementation using Euclid's algorithm.
- Terminal View:** Shows the output of running the script with inputs 12 and 18, resulting in an output of 6.
- Chat View:** A sidebar titled "LEAP YEAR FUNCTION IN PYTHON WITH AI ASSISTED CODING" contains a prompt asking to write a Python function named `find_gcd` that calculates the GCD of two integers. It includes examples and tasks related to the function.

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

```
assignment_3.5.py
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 the VS Code interface with the following details:

- EXPLORER:** Shows files: assignment_3.5.py (selected), main.py, and idea.
- CODE EDITOR:** Displays the code for assignment_3.5.py:

```
assignment_3.5.py
def decimal_to_binary(decimal: int) -> str:
    """
    Returns:
        str: The binary representation as a string.
    """
    if decimal == 0:
        return "0"
    binary = ""
    while decimal > 0:
        binary = str(decimal % 2) + binary
        decimal //= 2
    return binary

# Example usage:
decimal_num = int(input("Enter a decimal number: "))
binary_rep = decimal_to_binary(decimal_num)
print(f"The binary representation of {decimal_num} is {binary_rep}.")
```
- CHAT:** A sidebar titled "LEAP YEAR FUNCTION IN PYTHON WITH AI ASSISTED CODING" contains the following text:

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.

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

Task: Ensure the function handles the integer 0 correctly and returns a clean string without Python's default prefix.
- TERMINAL:** Shows a terminal session:

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

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 Structure:** The left sidebar shows files: `assignment_3.5.py`, `idea`, and `main.py`.
- Code Editor:** The main area displays the `assignment_3.5.py` file content. The code defines a function `check_harshad_number` that takes an integer `num` and returns `True` if it's a Harshad number (divisible by its digit sum) and `False` otherwise. It handles negative numbers and prints usage instructions for the script.
- AI Assistant Panel:** On the right, there's a panel titled "AI assisted coding" which provides context and examples. It includes:
 - A definition of `check_harshad_number` with a docstring.
 - Explanations for Harshad numbers: "an integer that is divisible by its digit sum" and "Harshad numbers are typically positive integers".
 - Example usages: "Enter a number: 18" followed by "18 is a Harshad number.", "Enter a number: 21" followed by "21 is a Harshad number.", and "Enter a number: 19" followed by "19 is not a Harshad number."
- Terminal:** At the bottom, the terminal shows command-line interactions for running the script with inputs 18, 21, and 19, corresponding to the outputs shown in the AI panel.