

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 displays a code editor with a Python function `is_leap_year` and its execution output. The function is defined as follows:

```
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.")
```

The terminal output shows the function being tested with the years 1900, 2000, and 2024:

```
PS D:\AI assisted coding> python.exe "d:/AI assisted coding/assignment_3.5.py"
Enter a year: 1900
1900 is not a leap year.
PS D:\AI assisted coding> python.exe "d:/AI assisted coding/assignment_3.5.py"
Enter a year: 2000
2000 is a leap year.
PS D:\AI assisted coding> python.exe "d:/AI assisted coding/assignment_3.5.py"
Enter a year: 2024
2024 is a leap year.
PS D:\AI assisted coding>
```

On the right side, a chat window titled "LEAP YEAR FUNCTION IN PYTHON WITH..." provides a natural language explanation of the function's logic, including the Gregorian calendar rules for leap years.

Question 2: One-Shot Prompting (GCD of Two Numbers)

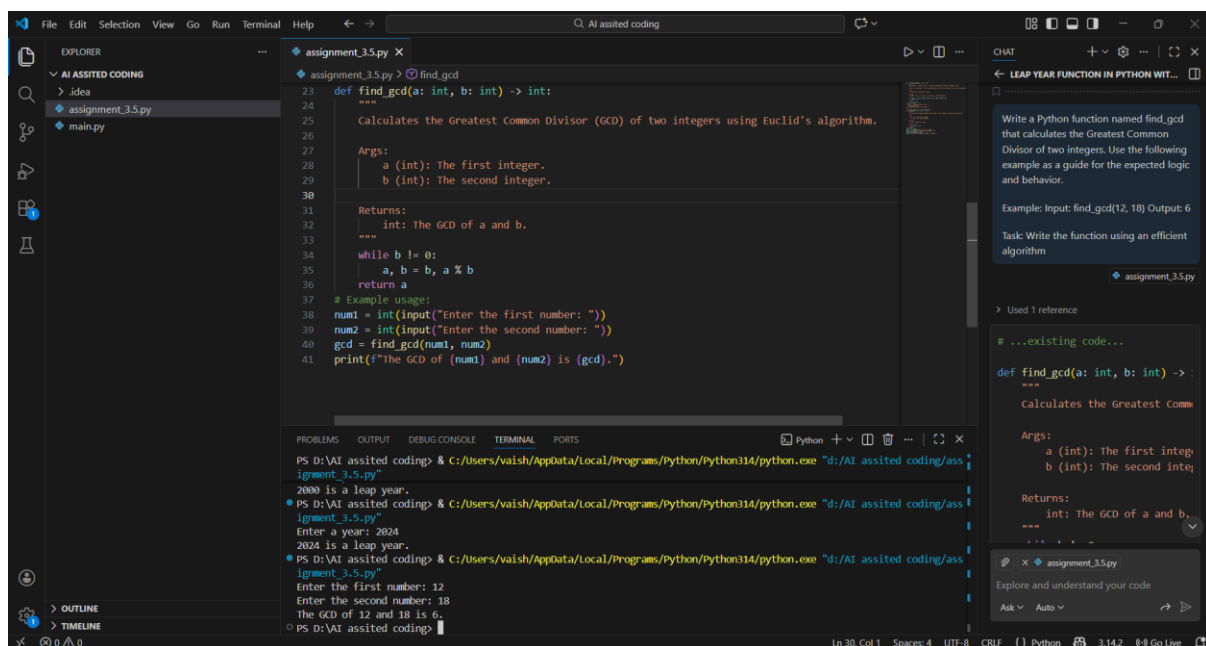
Write a one-shot prompt with one example to generate a Python function that finds the Greatest Common Divisor (GCD) of two numbers.

Example:

Input: 12, 18 → Output: 6

Task:

- Compare with a zero-shot solution.
- Analyze algorithm efficiency



The screenshot shows a code editor with a Python file named `assignment_3.5.py`. The code defines a function `find_gcd` that uses Euclid's algorithm to find the GCD of two integers. It includes docstrings for arguments and returns, and a main block that takes user input and prints the result. The terminal shows the execution of the script, which prompts for two numbers (12 and 18) and outputs their GCD (6). The right sidebar shows a chat window with a prompt to write a Python function for GCD, and the chat response displays the code from the editor.

```
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 output:

```
PS D:\AI assisted coding> C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/assignment_3.5.py"
2000 is a leap year.
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 year: 2024
2024 is a leap year.
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 first number: 12
Enter the second number: 18
The GCD of 12 and 18 is 6.
PS D:\AI assisted coding>
```

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 displays the Visual Studio Code interface. The Explorer pane on the left shows a project named 'AI ASSISTED CODING' with files 'assignment_3.5.py' and 'main.py'. The main editor window shows the code for 'assignment_3.5.py'. The code defines a function 'calculate_lcm(a: int, b: int) -> int' with a docstring explaining LCM and includes a few-shot prompt with three examples. The main code takes user input for two numbers and prints the LCM. The output pane at the bottom shows the execution results for three test cases: (4, 6) resulting in 12, (5, 10) resulting in 10, and (7, 3) resulting in 21. The right sidebar shows a chat window with a prompt about the LCM function and a response that confirms the function's correctness.

```
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}."
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\AI assisted coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/ass
gment_3.5.py"
Enter the first number: 4
Enter the second number: 6
The LCM of 4 and 6 is 12.
PS D:\AI assisted coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/ass
gment_3.5.py"
Enter the first number: 5
Enter the second number: 10
The LCM of 5 and 10 is 10.
PS D:\AI assisted coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/ass
gment_3.5.py"
Enter the first number: 7
Enter the second number: 3
The LCM of 7 and 3 is 21.
```

CHAT

```
LEAP YEAR FUNCTION IN PYTHON WIT...
correctly.
> Used 1 reference
# ...existing code...
def calculate_lcm(a: int, b: int)
    """
    Calculates the Least Common
    The LCM is the smallest posi
    Args:
        a (int): The first integ
        b (int): The second inte
    Returns:
        int: The LCM of a and b.
    """
    if a == 0 or b == 0:
        return 0 # LCM is undef
    gcd = find_gcd(a, b)
    return abs(a * b) // gcd
# ...existing code...
```

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 displays an IDE environment with three main panels:

- Editor:** Contains the file `assignment_3.5.py` with the following Python code:

```
79 def decimal_to_binary(decimal: int) -> str:
80     """
81     Returns:
82         str: The binary representation as a string.
83     """
84     if decimal == 0:
85         return "0"
86     binary = ""
87     while decimal > 0:
88         binary = str(decimal % 2) + binary
89         decimal //= 2
90     return binary
91
92 # Example usage:
93 decimal_num = int(input("Enter a decimal number: "))
94 binary_rep = decimal_to_binary(decimal_num)
95 print(f"The binary representation of {decimal_num} is {binary_rep}.")
96
```
- Terminal:** Shows the execution of the script:

```
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 binary number: 1010
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>
```
- Chat:** Displays a one-shot prompt for the function:

```
<- LEAP YEAR FUNCTION IN PYTHON WITH...
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."
```

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 an IDE with a Python file named `assignment_3.5.py`. The code defines a function `check_harshad_number` that takes an integer `num` and returns a boolean. The function checks if the number is divisible by the sum of its digits. Comments explain that Harshad numbers are typically positive integers. Example usage is provided with inputs 18, 21, and 19. The terminal shows the execution of the program, which prompts the user to enter a number and prints the result. The chat window on the right shows a few-shot prompt for the function.

```
def check_harshad_number(num: int) -> bool:
    """
    Args:
        num (int): The integer to check.

    Returns:
        bool: True if the number is a Harshad number, False otherwise.
    """
    if num <= 0:
        return False  # Harshad numbers are typically positive integers
    digit_sum = sum(int(digit) for digit in str(num))
    return num % digit_sum == 0

# Example usage:
num = int(input("Enter a number: "))
if check_harshad_number(num):
    print(f"{num} is a Harshad number.")
else:
    print(f"{num} is not a Harshad number.")
```

Terminal Output:

```
PS D:\AI assisted coding> python assignment_3.5.py
Enter a number: 18
18 is a Harshad number.
PS D:\AI assisted coding> python assignment_3.5.py
Enter a number: 21
21 is a Harshad number.
PS D:\AI assisted coding> python assignment_3.5.py
Enter a number: 19
19 is not a Harshad number.
PS D:\AI assisted coding>
```

Chat Prompt:

```
<- LEAP YEAR FUNCTION IN PYTHON WIT...
> Used 1 reference

def check_harshad_number(num: int) -> bool:
    """
    Args:
        num (int): The integer to check.

    Returns:
        bool: True if the number is a Harshad number, False otherwise.
    """
    if num <= 0:
        return False  # Harshad numbers are typically positive integers
    digit_sum = sum(int(digit) for digit in str(num))
    return num % digit_sum == 0

# Example usage:
num = int(input("Enter a number: "))
if check_harshad_number(num):
    print(f"{num} is a Harshad number.")
else:
    print(f"{num} is not a Harshad number.")
```