

AI-Assisted Coding

Assignment-3.5

Name: M.AKHIL REDDY

2303a52315

Batch:45

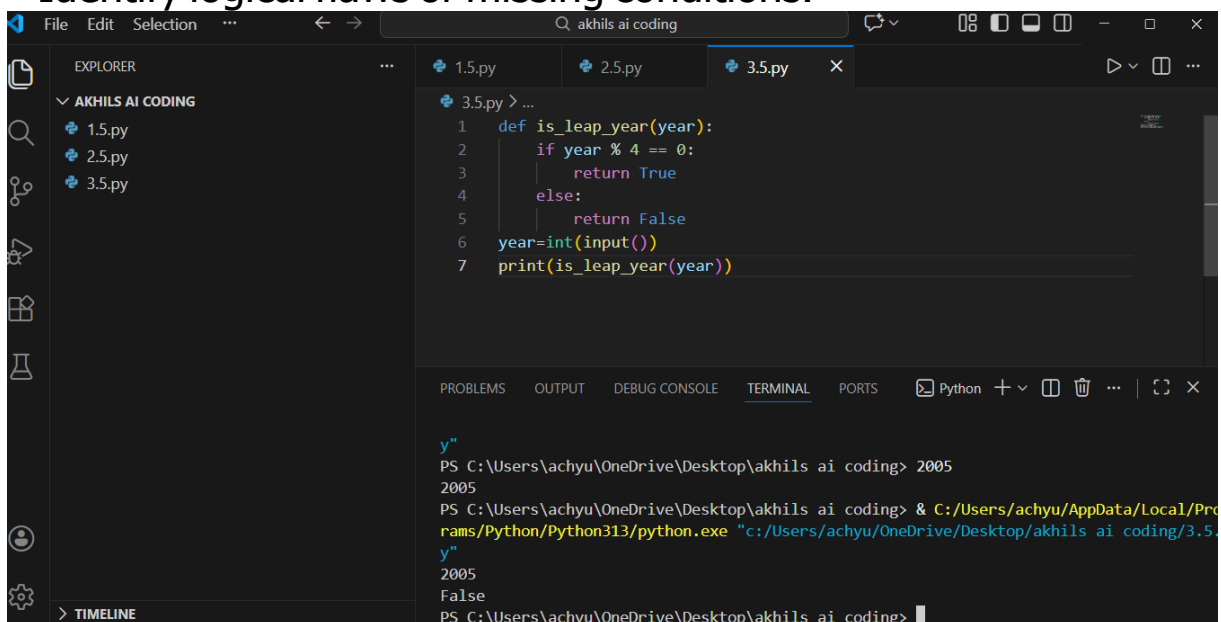
Question 1: Zero-Shot Prompting (Leap Year Check)

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 Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a folder named 'AKHILS AI CODING' containing three files: '1.5.py', '2.5.py', and '3.5.py'. The code editor is open to '3.5.py' and contains the following Python code:

```
1 def is_leap_year(year):
2     if year % 4 == 0:
3         return True
4     else:
5         return False
6 year=int(input())
7 print(is_leap_year(year))
```

Below the code editor is a terminal window. It shows the command prompt running the script for the year 2005. The output is 'True'.

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> 2005
True
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
2005
True
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

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

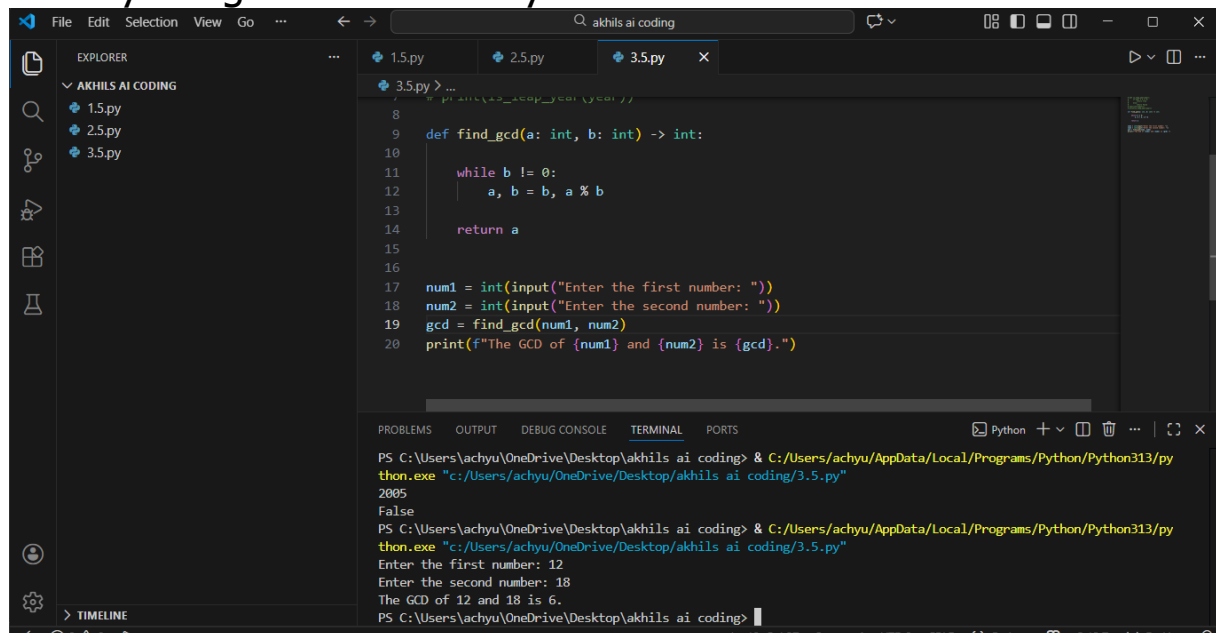
Write a one-shot prompt with one example to generate a Python function that finds the Greatest Common Divisor (GCD) of two numbers.

Example:

Input: 12, 18 → Output: 6

Task:

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



The screenshot shows a Visual Studio Code editor with a file explorer on the left containing three files: 1.5.py, 2.5.py, and 3.5.py. The main editor window displays the code for 3.5.py, which defines a function `find_gcd` and uses it to calculate the GCD of user inputs. The terminal at the bottom shows the command to run the script and its output, including the GCD of 12 and 18.

```
8
9 def find_gcd(a: int, b: int) -> int:
10
11     while b != 0:
12         a, b = b, a % b
13
14     return a
15
16
17 num1 = int(input("Enter the first number: "))
18 num2 = int(input("Enter the second number: "))
19 gcd = find_gcd(num1, num2)
20 print(f"The GCD of {num1} and {num2} is {gcd}.")
```

Terminal Output:

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
2005
False
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
Enter the first number: 12
Enter the second number: 18
The GCD of 12 and 18 is 6.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Question 3: Few-Shot Prompting (LCM Calculation)

Write a few-shot prompt with multiple examples to generate a Python

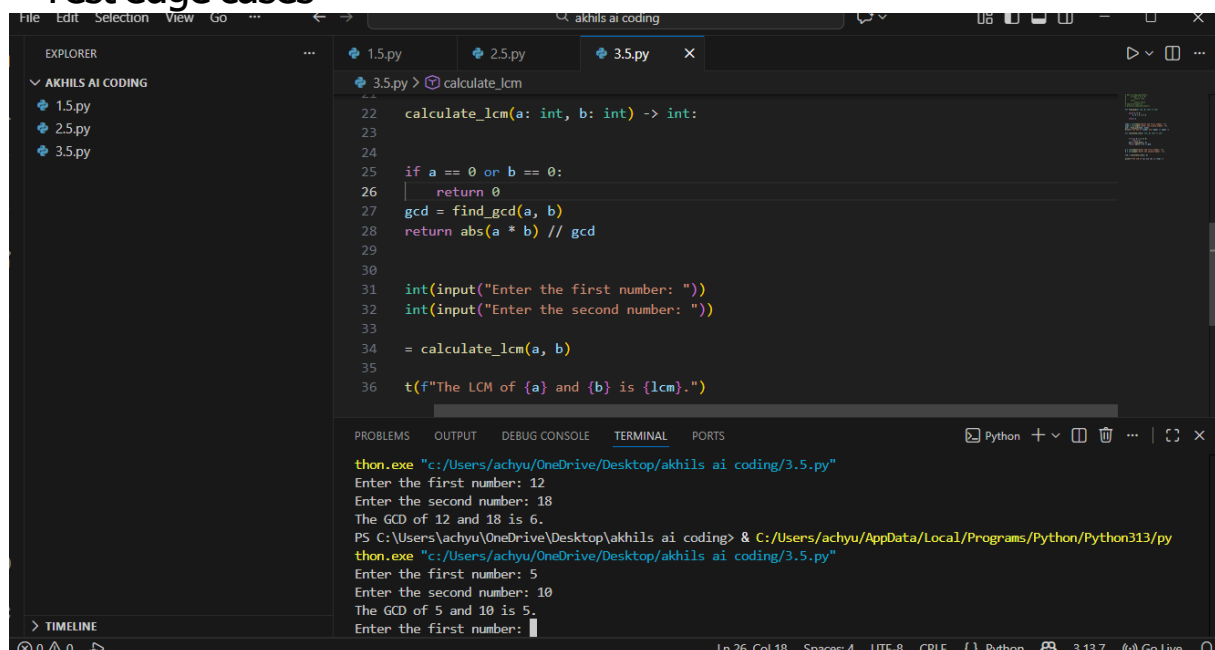
function that computes the Least Common Multiple (LCM).

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 shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a folder named 'AKHILS AI CODING' containing three files: '1.5.py', '2.5.py', and '3.5.py'. The code editor is open to '3.5.py', which contains the following Python code:

```
22 calculate_lcm(a: int, b: int) -> int:
23
24
25 if a == 0 or b == 0:
26     return 0
27 gcd = find_gcd(a, b)
28 return abs(a * b) // gcd
29
30
31 int(input("Enter the first number: "))
32 int(input("Enter the second number: "))
33
34 = calculate_lcm(a, b)
35
36 t(f"The LCM of {a} and {b} is {lcm}.")
```

The terminal at the bottom shows the execution of the script using 'thon.exe'. The output is as follows:

```
thon.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
Enter the first number: 12
Enter the second number: 18
The GCD of 12 and 18 is 6.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/py
thon.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
Enter the first number: 5
Enter the second number: 10
The GCD of 5 and 10 is 5.
Enter the first number: 
```

Question 5: One-Shot Prompting (Decimal to Binary Conversion)

Write a one-shot prompt with an example to generate a Python function

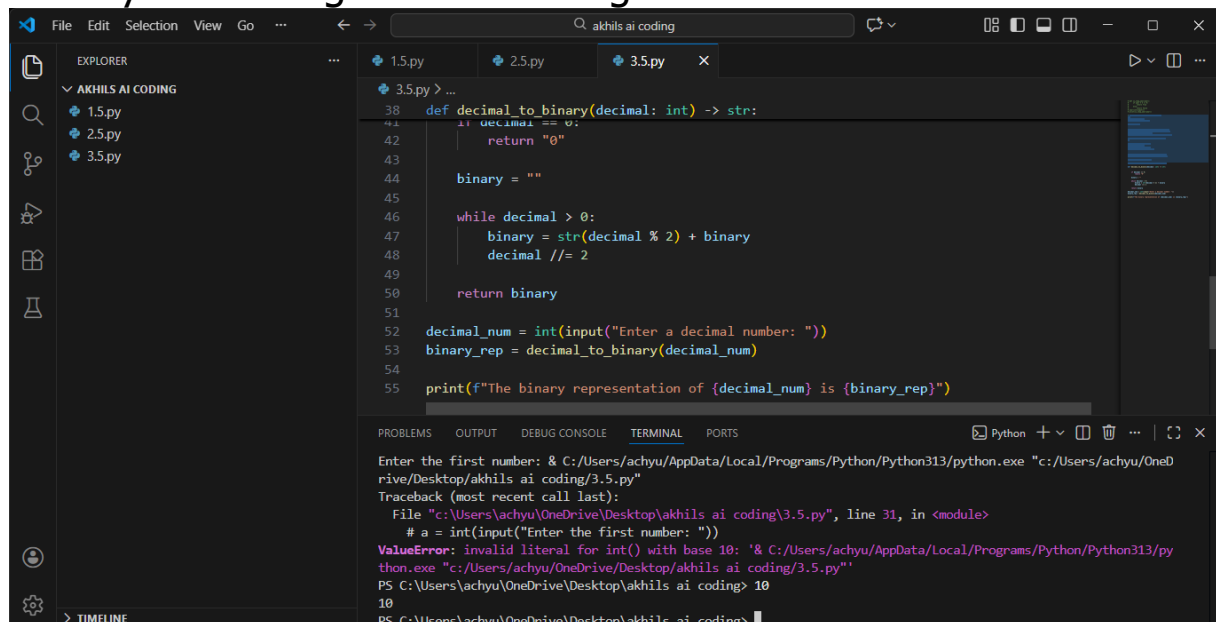
that converts a decimal number to binary.

Example:

Input: 10 → Output: 1010

Task:

- Compare clarity with zero-shot output.
- Analyze handling of zero and negative numbers.



The screenshot shows a Visual Studio Code editor with a file explorer on the left containing three Python files: 1.5.py, 2.5.py, and 3.5.py. The main editor window displays the code for 3.5.py, which defines a function `decimal_to_binary` and uses it to convert an input decimal number to binary. The code is as follows:

```
38 def decimal_to_binary(decimal: int) -> str:
39     if decimal == 0:
40         return "0"
41
42     binary = ""
43
44     while decimal > 0:
45         binary = str(decimal % 2) + binary
46         decimal //= 2
47
48     return binary
49
50
51 decimal_num = int(input("Enter a decimal number: "))
52 binary_rep = decimal_to_binary(decimal_num)
53
54 print(f"The binary representation of {decimal_num} is {binary_rep}")
```

The terminal at the bottom shows the execution of the script. It prompts the user to enter a decimal number, and the user enters 10. The output is "The binary representation of 10 is 1010".

```
Enter the first number: & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
Traceback (most recent call last):
  File "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py", line 31, in <module>
    # a = int(input("Enter the first number: "))
ValueError: invalid literal for int() with base 10: '& C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"'
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> 10
10
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Question 6: Few-Shot Prompting (Harshad Number Check)

Write a few-shot prompt to generate a Python function that checks

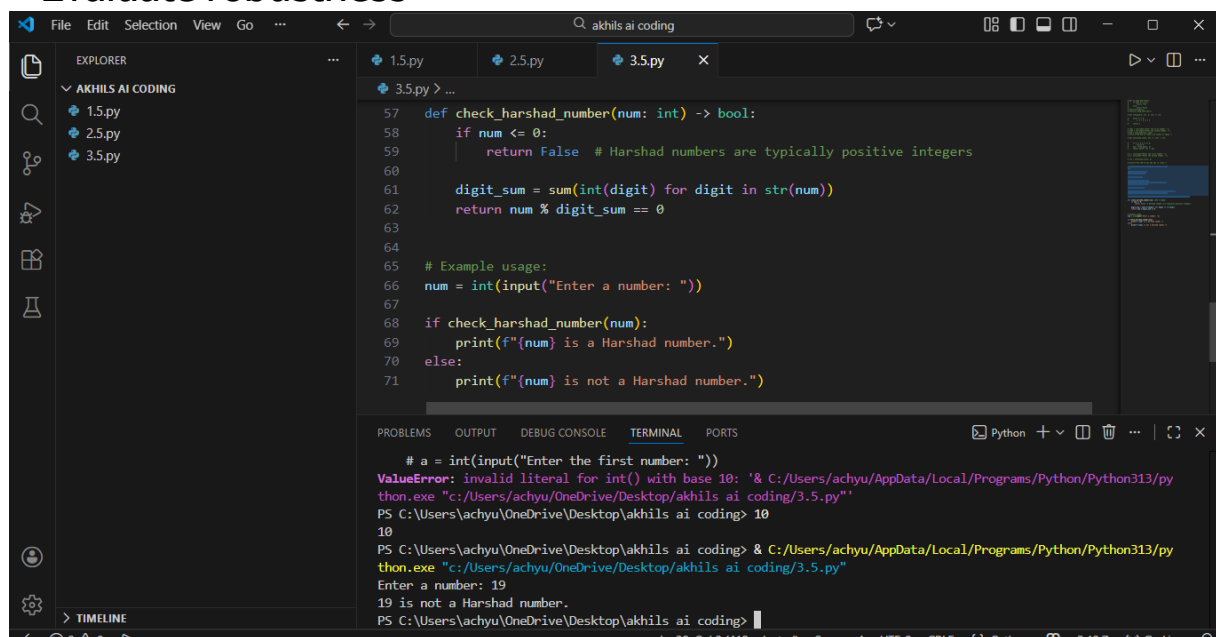
whether a number is a Harshad (Niven) number.

Examples:

- Input: 18 → Output: Harshad Number
- Input: 21 → Output: Harshad Number
- Input: 19 → Output: Not a Harshad Number

Task:

- Test boundary conditions.
- Evaluate robustness



```
File Edit Selection View Go ... akhils ai coding
EXPLORER
AKHILS AI CODING
1.5.py
2.5.py
3.5.py
3.5.py > ...
57 def check_harshad_number(num: int) -> bool:
58     if num <= 0:
59         return False # Harshad numbers are typically positive integers
60
61     digit_sum = sum(int(digit) for digit in str(num))
62     return num % digit_sum == 0
63
64
65 # Example usage:
66 num = int(input("Enter a number: "))
67
68 if check_harshad_number(num):
69     print(f"{num} is a Harshad number.")
70 else:
71     print(f"{num} is not a Harshad number.")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + -
# a = int(input("Enter the first number: "))
ValueError: invalid literal for int() with base 10: '& C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"'
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> 10
10
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
Enter a number: 19
19 is not a Harshad number.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```