

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 code editor interface with the following details:

- File Explorer:** Shows a folder named "AKHILS AI CODING" containing files 1.5.py, 2.5.py, and 3.5.py.
- Code Editor:** The 3.5.py file is open, displaying the following Python code:

```
def is_leap_year(year):
    if year % 4 == 0:
        return True
    else:
        return False
year=int(input())
print(is_leap_year(year))
```
- Terminal:** The terminal window shows the output of running the script:

```
y"
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> 2005
2005
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>
```

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

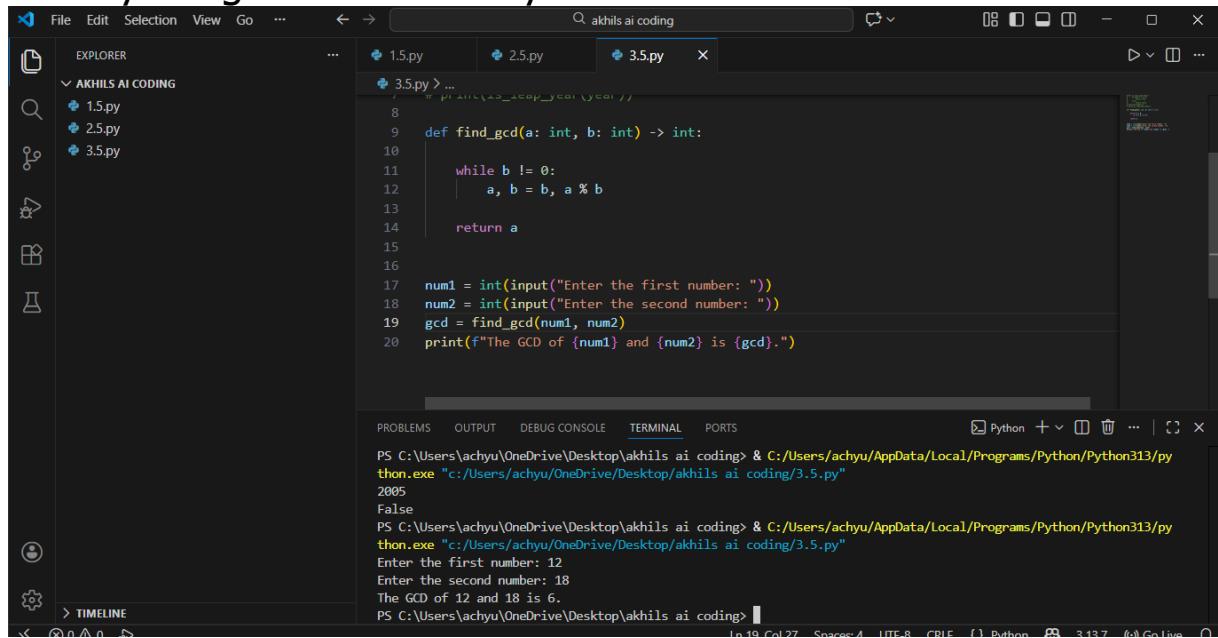
Write a one-shot prompt with one example to generate a Python function that finds the Greatest Common Divisor (GCD) of two numbers.

Example:

Input: 12, 18 → Output: 6

Task:

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



A screenshot of a code editor (Visual Studio Code) showing a Python file named 3.5.py. The code implements the Euclidean algorithm to find the GCD of two integers, num1 and num2. It prompts the user for input and prints the result. The terminal below shows the execution of the script and its output.

```
def find_gcd(a: int, b: int) -> int:
    while b != 0:
        a, b = b, a % b
    return a

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}.")
```

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 Python code editor with the following code:

```
1.5.py 2.5.py 3.5.py <-- active

3.5.py > calculate_lcm
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 window below shows the execution of the script and its output:

```
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: 1
```

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

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

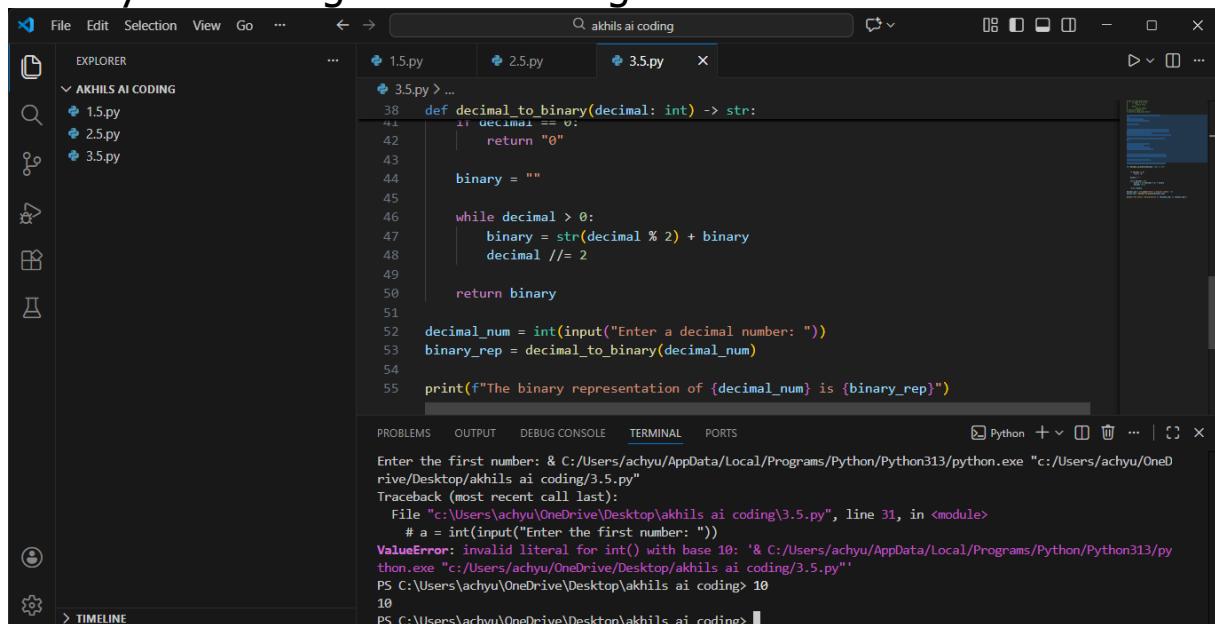
that converts a decimal number to binary.

Example:

Input: 10 → Output: 1010

Task:

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



```
EXPLORER          1.5.py    2.5.py    3.5.py    ...
AKHILS AI CODING
1.5.py
2.5.py
3.5.py

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

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
Python +  ...  |  ...  x
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 "<stdin>", line 1, in <module>
    decimal_num = int(input("Enter a decimal 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

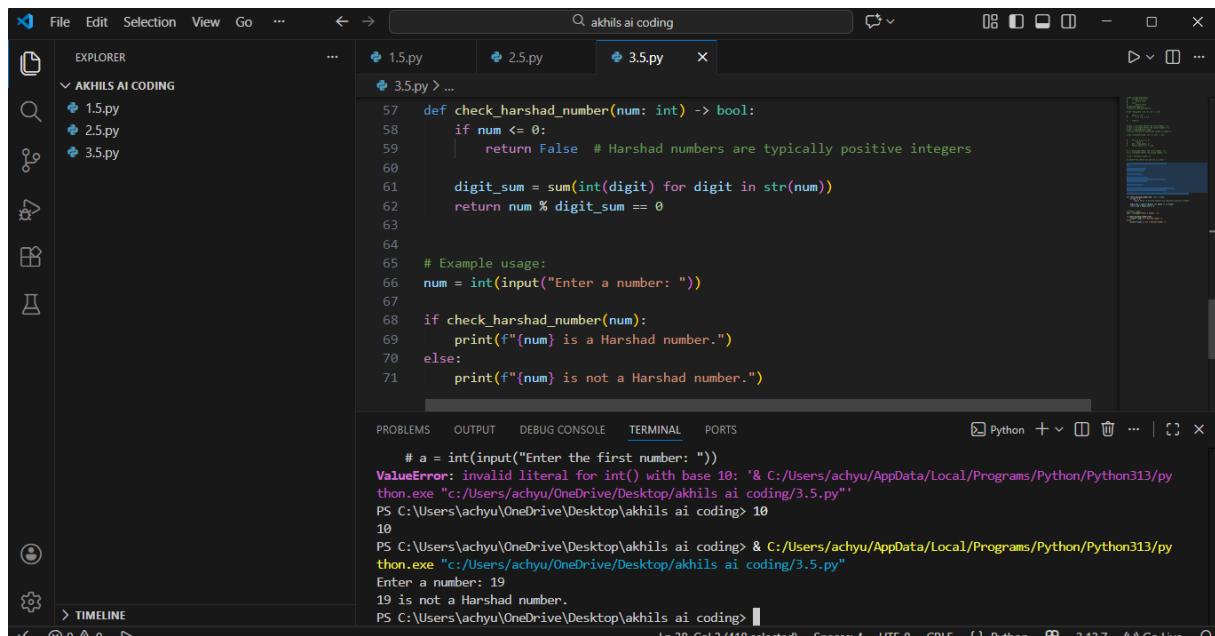
whether a number is a Harshad (Niven) number.

Examples:

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

Task:

- Test boundary conditions.
- Evaluate robustness



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

- File Explorer:** Shows a folder named "AKHILS AI CODING" containing files 1.5.py, 2.5.py, and 3.5.py.
- Code Editor:** The file 3.5.py is open, displaying the following Python code:

```
def check_harshad_number(num: int) -> bool:
    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:** The terminal tab shows the following session:

```
# 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
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>
```