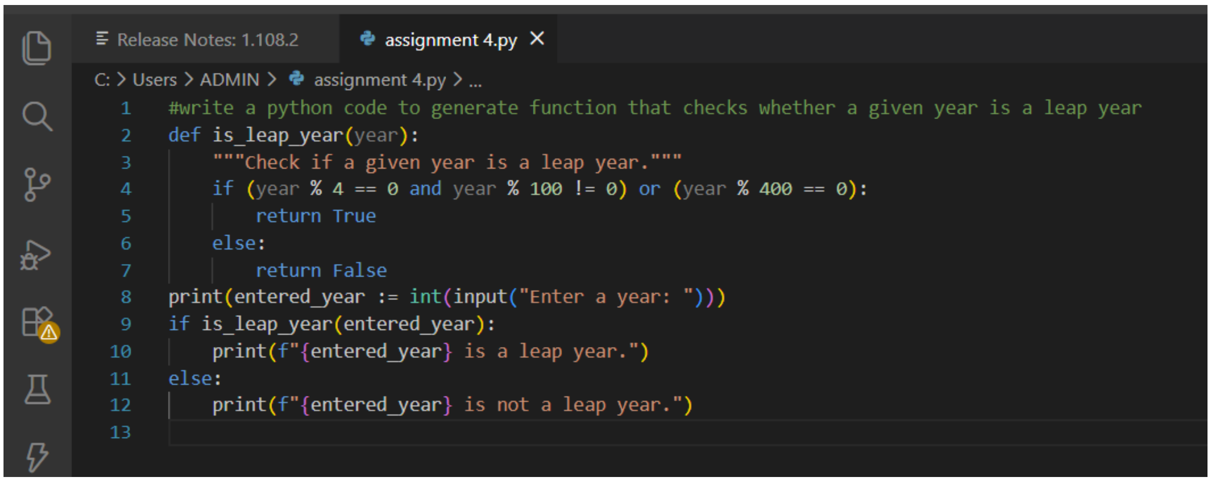
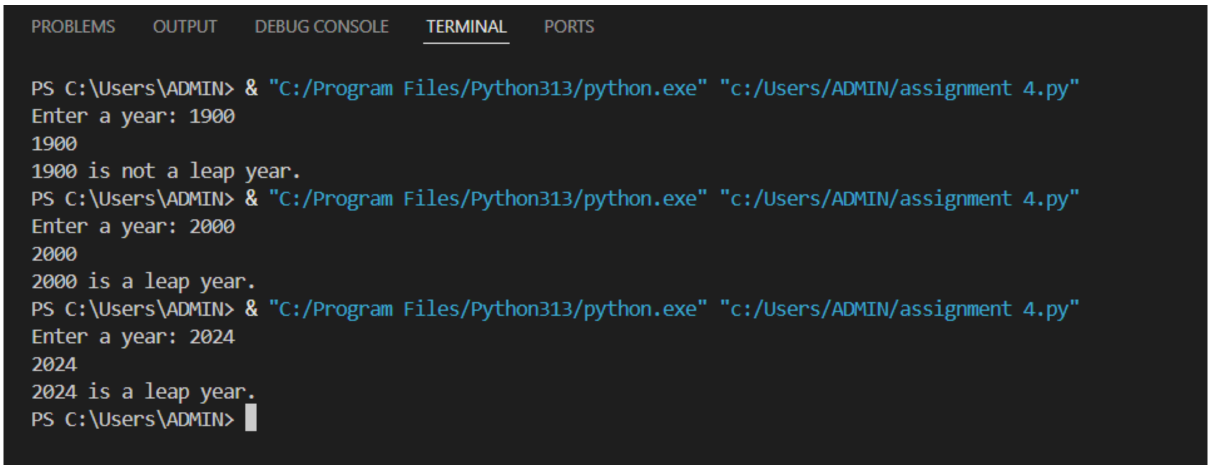
ASSIGNMNET 3.5

HEMAVATHI N
2303A51965
Batch 24

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.
Week2 -
Task:
• Record the AI-generated code.
• Test with years like 1900, 2000, 2024.
• Identify logical flaws or missing conditions.



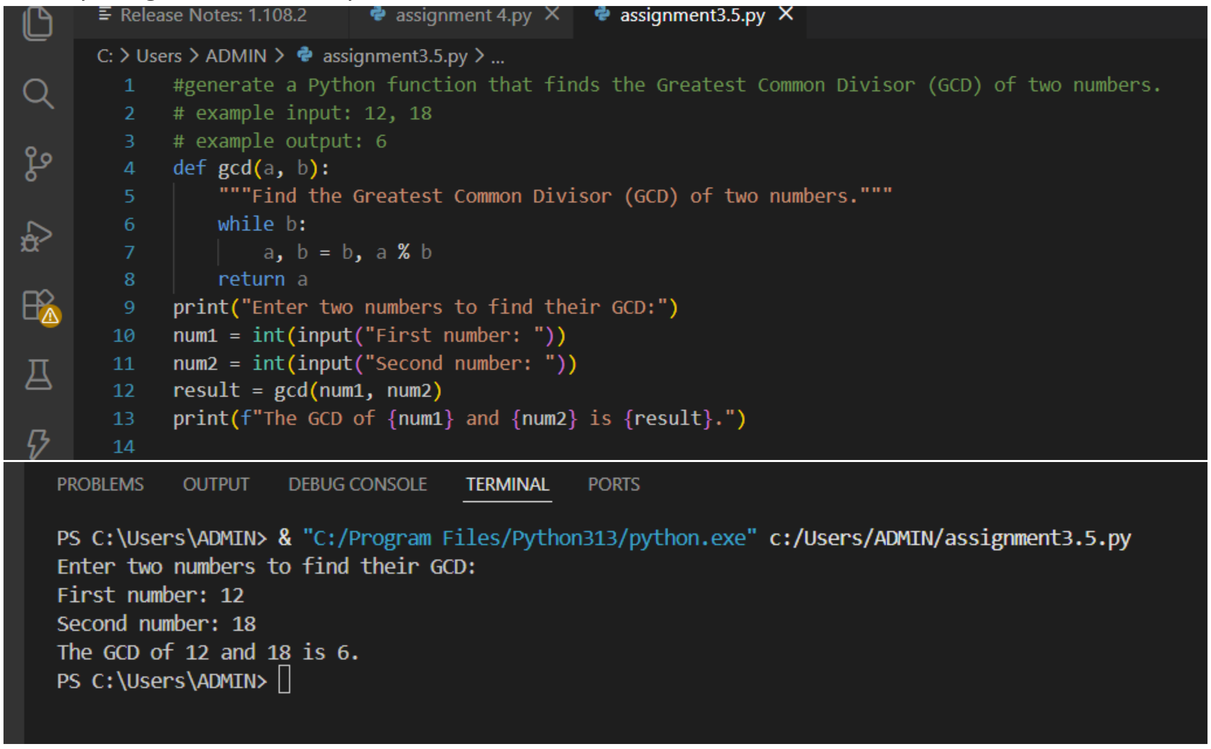
```python
#write a python code to generate function that checks whether a given year is a leap year
def is_leap_year(year):
    """Check if a given year is a leap year."""
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
print(entered_year := int(input("Enter a year: ")))
if is_leap_year(entered_year):
    print(f"{entered_year} is a leap year.")
else:
    print(f"{entered_year} is not a leap year.")
```



```
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" "c:/Users/ADMIN/assignment 4.py"
Enter a year: 1900
1900
1900 is not a leap year.
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" "c:/Users/ADMIN/assignment 4.py"
Enter a year: 2000
2000
2000 is a leap year.
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" "c:/Users/ADMIN/assignment 4.py"
Enter a year: 2024
2024
2024 is a leap year.
PS C:\Users\ADMIN> 
```

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.

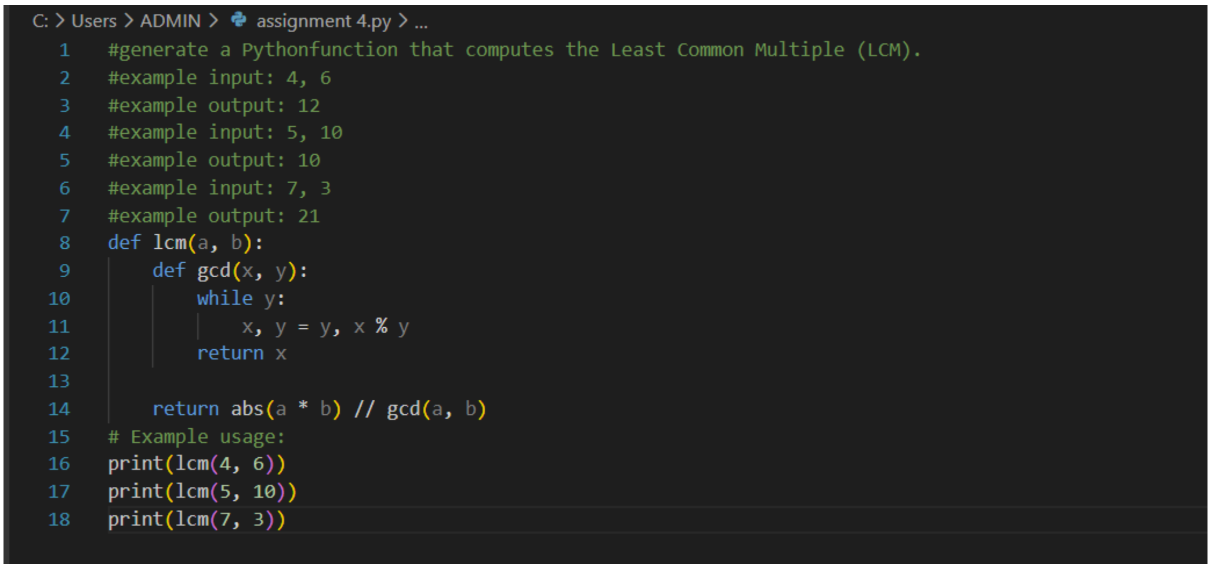
• Analyze algorithm efficiency.



```python
#generate a Python function that finds the Greatest Common Divisor (GCD) of two numbers.
# example input: 12, 18
# example output: 6
def gcd(a, b):
    """Find the Greatest Common Divisor (GCD) of two numbers."""
    while b:
        a, b = b, a % b
    return a
print("Enter two numbers to find their GCD:")
num1 = int(input("First number: "))
num2 = int(input("Second number: "))
result = gcd(num1, num2)
print(f"The GCD of {num1} and {num2} is {result}.")
```

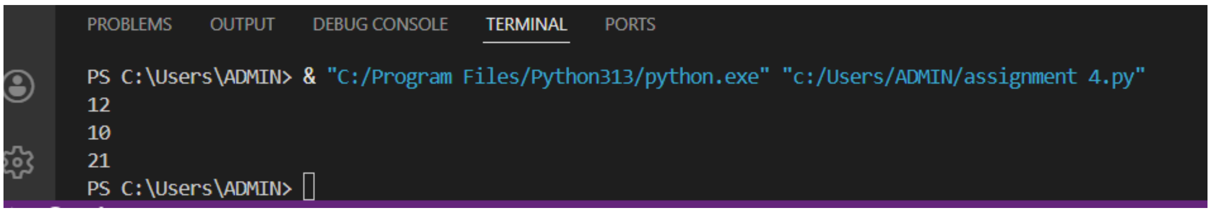
```
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
Enter two numbers to find their GCD:
First number: 12
Second number: 18
The GCD of 12 and 18 is 6.
PS C:\Users\ADMIN>
```

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.



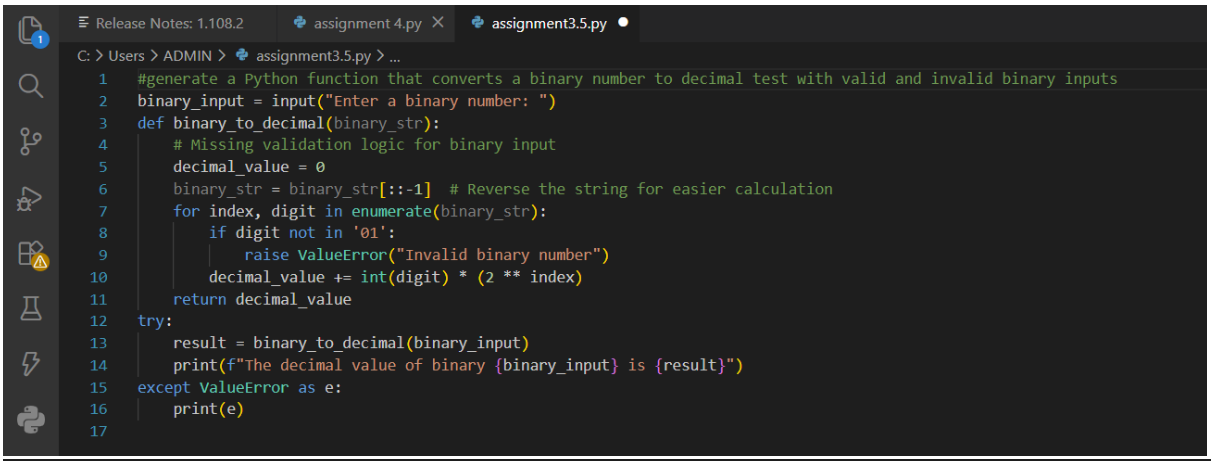
```python
#generate a Pythonfunction that computes the Least Common Multiple (LCM).
#example input: 4, 6
#example output: 12
#example input: 5, 10
#example output: 10
#example input: 7, 3
#example output: 21
def lcm(a, b):
    def gcd(x, y):
        while y:
            x, y = y, x % y
        return x

    return abs(a * b) // gcd(a, b)
# Example usage:
print(lcm(4, 6))
print(lcm(5, 10))
print(lcm(7, 3))
```

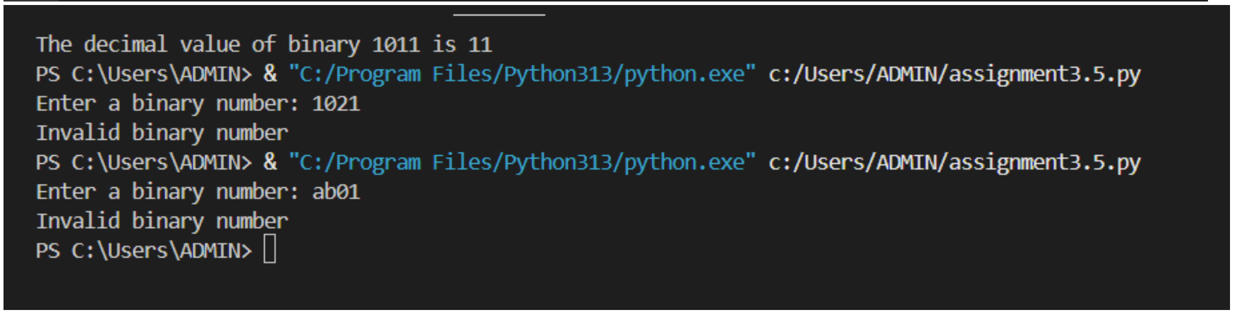


```
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" "c:/Users/ADMIN/assignment 4.py"
12
10
21
PS C:\Users\ADMIN> 
```

Question 4: Zero-Shot Prompting (Binary to Decimal Conversion)
Write a zero-shot prompt to generate a Python function that converts a binary number to decimal.
Task:
• Test with valid and invalid binary inputs.
• Identify missing validation logic.

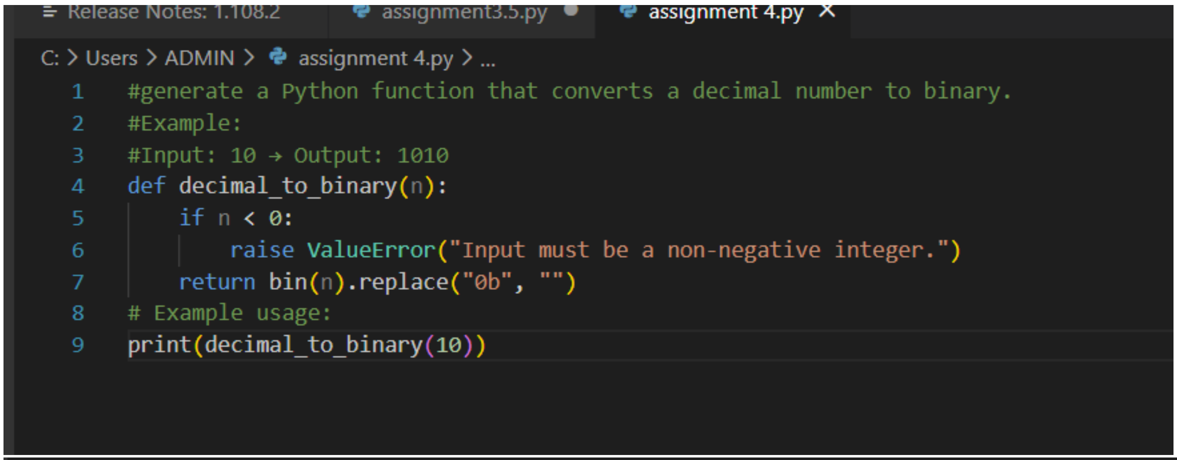


```python
#generate a Python function that converts a binary number to decimal test with valid and invalid binary inputs
binary_input = input("Enter a binary number: ")
def binary_to_decimal(binary_str):
    # Missing validation logic for binary input
    decimal_value = 0
    binary_str = binary_str[::-1]  # Reverse the string for easier calculation
    for index, digit in enumerate(binary_str):
        if digit not in '01':
            raise ValueError("Invalid binary number")
        decimal_value += int(digit) * (2 ** index)
    return decimal_value
try:
    result = binary_to_decimal(binary_input)
    print(f"The decimal value of binary {binary_input} is {result}")
except ValueError as e:
    print(e)
```

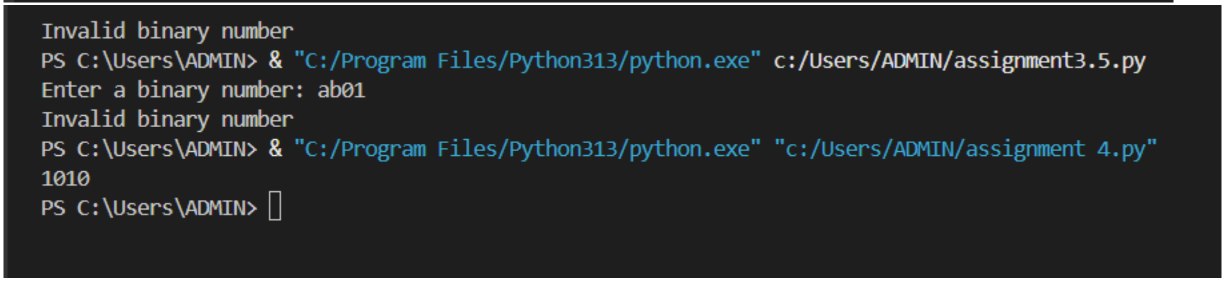


```
The decimal value of binary 1011 is 11
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
Enter a binary number: 1021
Invalid binary number
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
Enter a binary number: ab01
Invalid binary number
PS C:\Users\ADMIN> 
```

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.

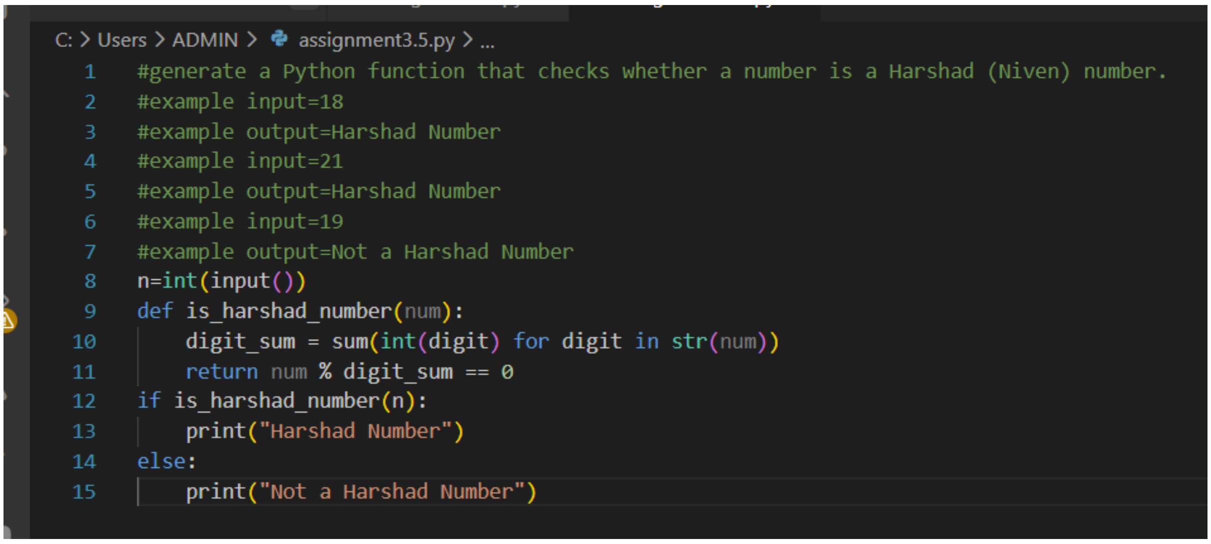


```python
#generate a Python function that converts a decimal number to binary.
#Example:
#Input: 10 → Output: 1010
def decimal_to_binary(n):
    if n < 0:
        raise ValueError("Input must be a non-negative integer.")
    return bin(n).replace("0b", "")
# Example usage:
print(decimal_to_binary(10))
```

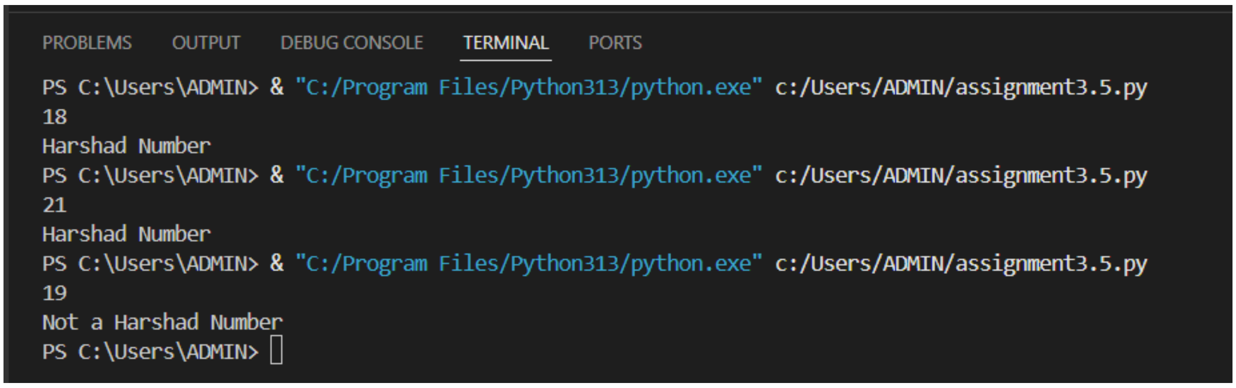


```
Invalid binary number
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
Enter a binary number: ab01
Invalid binary number
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" "c:/Users/ADMIN/assignment 4.py"
1010
PS C:\Users\ADMIN> 
```

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



```python
#generate a Python function that checks whether a number is a Harshad (Niven) number.
#example input=18
#example output=Harshad Number
#example input=21
#example output=Harshad Number
#example input=19
#example output=Not a Harshad Number
n=int(input())
def is_harshad_number(num):
    digit_sum = sum(int(digit) for digit in str(num))
    return num % digit_sum == 0
if is_harshad_number(n):
    print("Harshad Number")
else:
    print("Not a Harshad Number")
```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
18
Harshad Number
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
21
Harshad Number
PS C:\Users\ADMIN> & "C:/Program Files/Python313/python.exe" c:/Users/ADMIN/assignment3.5.py
19
Not a Harshad Number
PS C:\Users\ADMIN>
```