AIAC ASSIGNMENT – 3.5

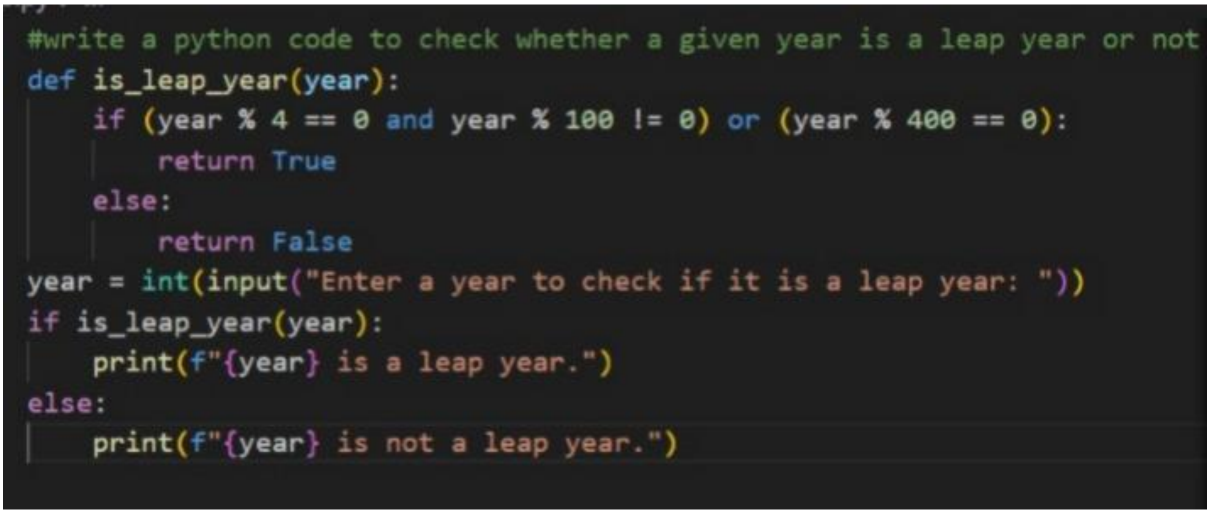
Question 1: Zero-Shot Prompting (Leap Year Check)

Write a zero-shot prompt to generate a Python function that checks whether a given year is a leap year.

Task:

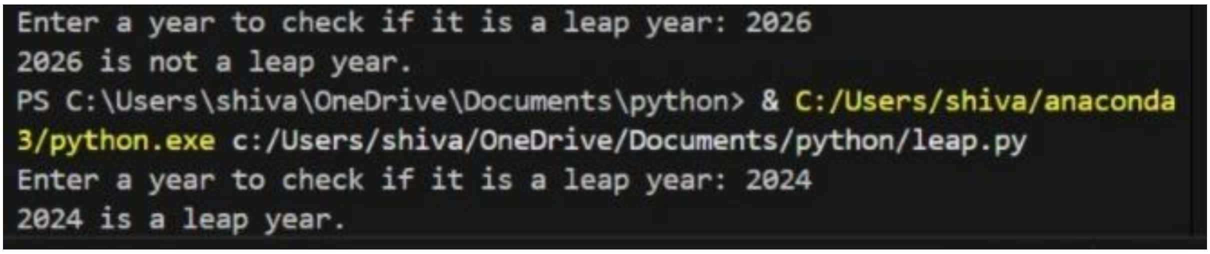
- Record the AI-generated code.
- Test with years like 1900, 2000, 2024.
- Identify logical flaws or missing conditions.

Code:



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

Output:



```
Enter a year to check if it is a leap year: 2026
2026 is not a leap year.
PS C:\Users\shiva\OneDrive\Documents\python> & C:/Users/shiva/anaconda
3/python.exe c:/Users/shiva/OneDrive/Documents/python/leap.py
Enter a year to check if it is a leap year: 2024
2024 is a leap year.
```

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.

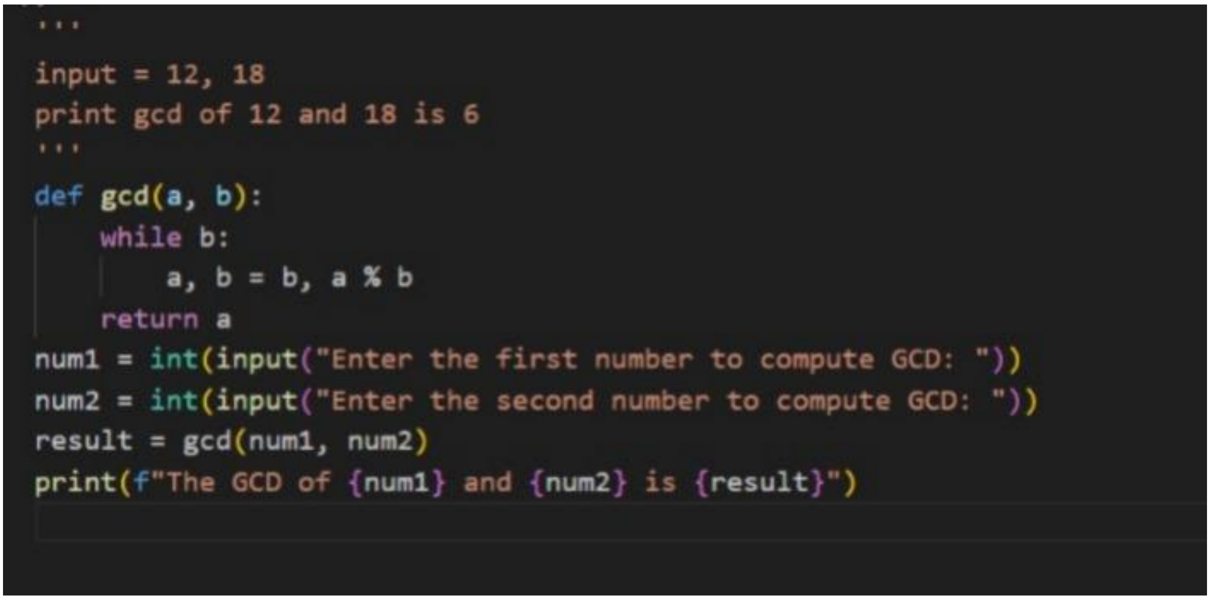
Example:

Input: 12, 18 → Output: 6

Task:

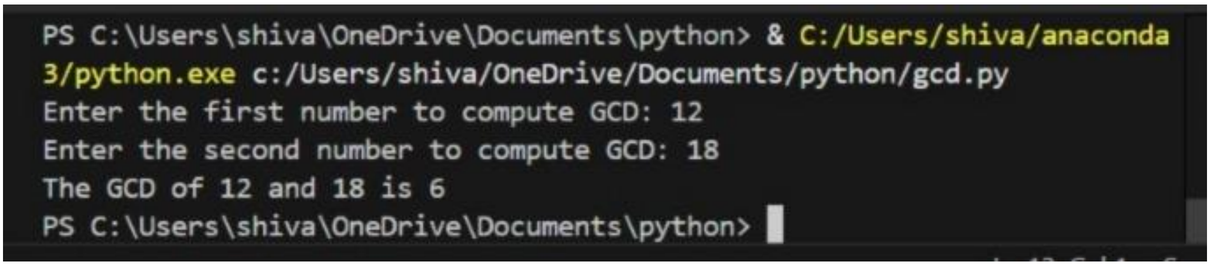
- Compare with a zero-shot solution.
- Analyse algorithm efficiency.

Code:



```
'''
input = 12, 18
print gcd of 12 and 18 is 6
'''
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
num1 = int(input("Enter the first number to compute GCD: "))
num2 = int(input("Enter the second number to compute GCD: "))
result = gcd(num1, num2)
print(f"The GCD of {num1} and {num2} is {result}")
```

Output:



```
PS C:\Users\shiva\OneDrive\Documents\python> & C:/Users/shiva/anaconda
3/python.exe c:/Users/shiva/OneDrive/Documents/python/gcd.py
Enter the first number to compute GCD: 12
Enter the second number to compute GCD: 18
The GCD of 12 and 18 is 6
PS C:\Users\shiva\OneDrive\Documents\python>
```

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).

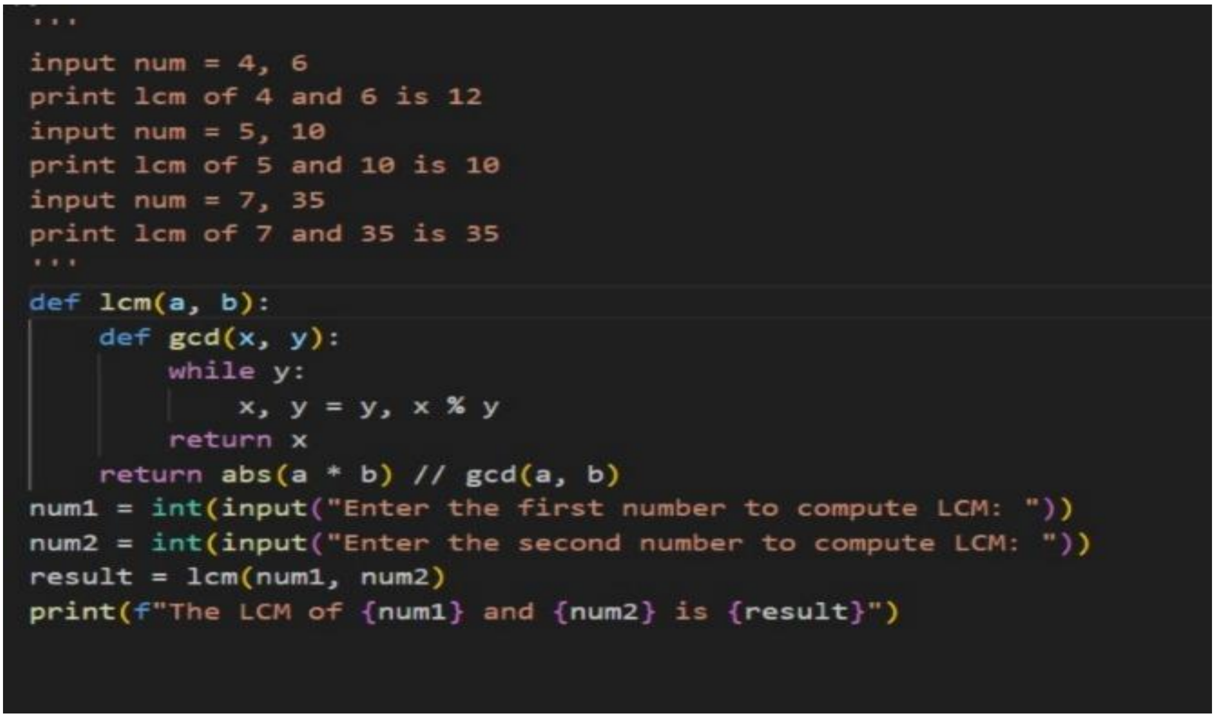
Examples:

- Input: 4, 6 → Output: 12
- Input: 5, 10 → Output: 10
- Input: 7, 3 → Output: 21

Task:

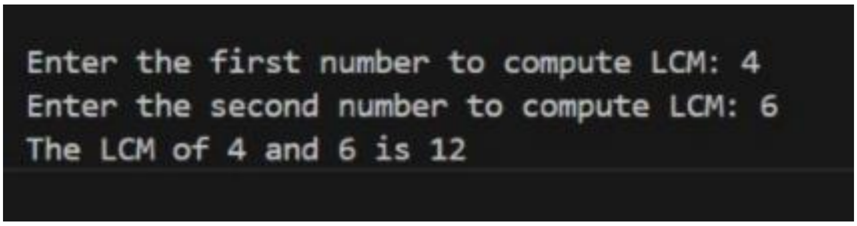
- Examine how examples guide formula selection.
- Test edge cases.

Code:



```
'''
input num = 4, 6
print lcm of 4 and 6 is 12
input num = 5, 10
print lcm of 5 and 10 is 10
input num = 7, 35
print lcm of 7 and 35 is 35
'''
def lcm(a, b):
    def gcd(x, y):
        while y:
            x, y = y, x % y
        return x
    return abs(a * b) // gcd(a, b)
num1 = int(input("Enter the first number to compute LCM: "))
num2 = int(input("Enter the second number to compute LCM: "))
result = lcm(num1, num2)
print(f"The LCM of {num1} and {num2} is {result}")
```

Output:



```
Enter the first number to compute LCM: 4
Enter the second number to compute LCM: 6
The LCM of 4 and 6 is 12
```

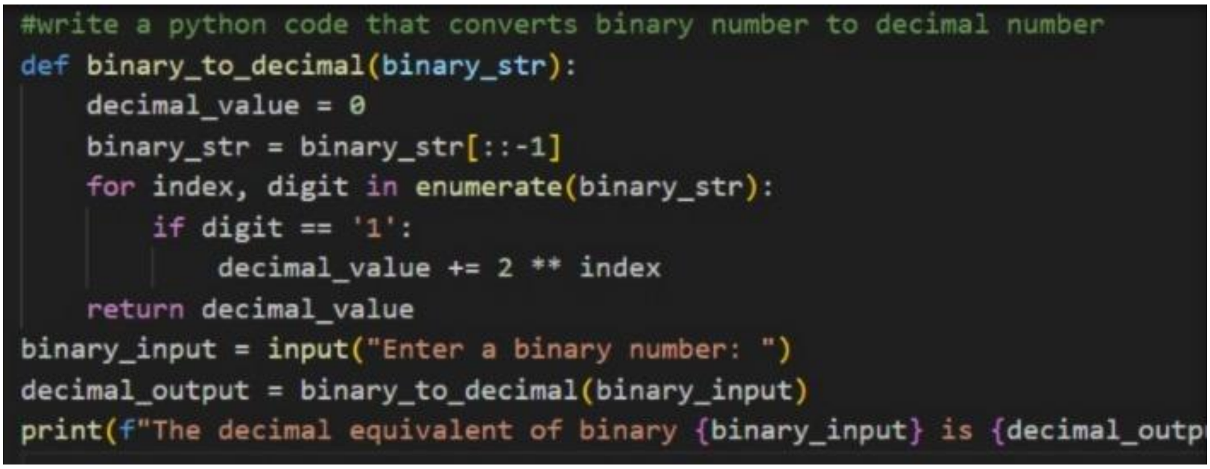
Question 4: Zero-Shot Prompting (Binary to Decimal Conversion)

Write a zero-shot prompt to generate a Python function that converts a binary number to decimal.

Task:

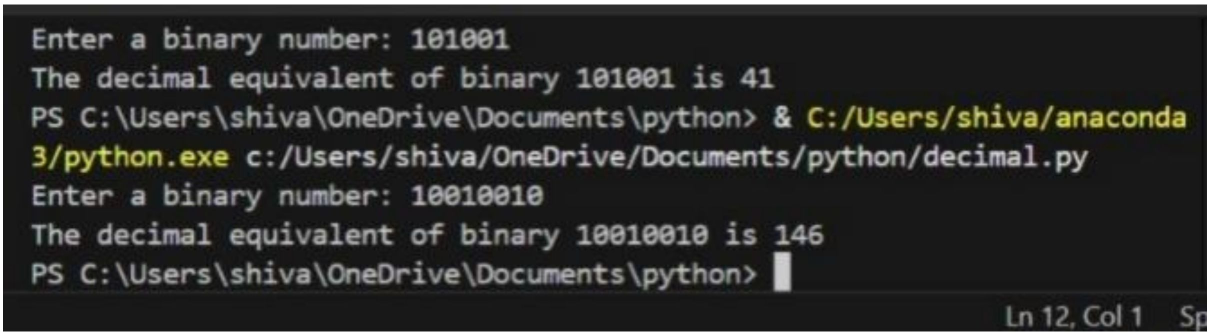
- Test with valid and invalid binary inputs.
- Identify missing validation logic.

Code:



```
#write a python code that converts binary number to decimal number
def binary_to_decimal(binary_str):
    decimal_value = 0
    binary_str = binary_str[::-1]
    for index, digit in enumerate(binary_str):
        if digit == '1':
            decimal_value += 2 ** index
    return decimal_value
binary_input = input("Enter a binary number: ")
decimal_output = binary_to_decimal(binary_input)
print(f"The decimal equivalent of binary {binary_input} is {decimal_outp
```

Output:



```
Enter a binary number: 101001
The decimal equivalent of binary 101001 is 41
PS C:\Users\shiva\OneDrive\Documents\python> & C:/Users/shiva/anaconda
3/python.exe c:/Users/shiva/OneDrive/Documents/python/decimal.py
Enter a binary number: 10010010
The decimal equivalent of binary 10010010 is 146
PS C:\Users\shiva\OneDrive\Documents\python>
```

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.

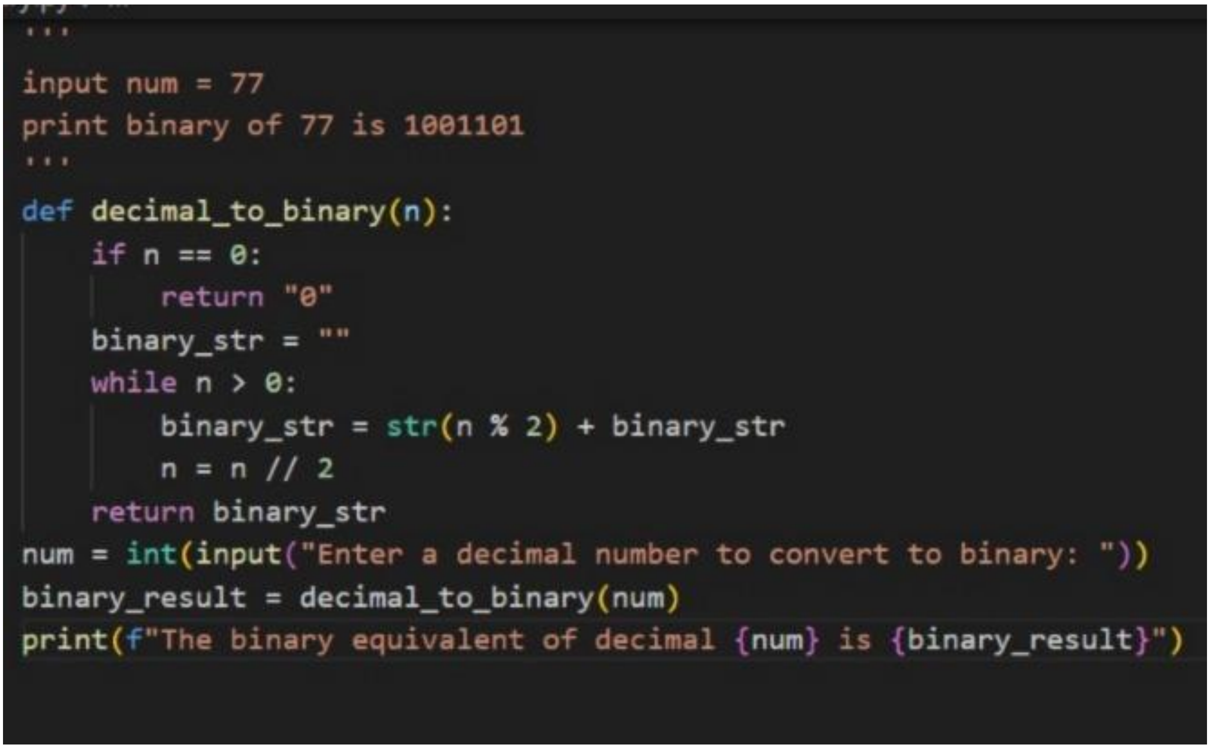
Example:

Input: 10 → Output: 1010

Task:

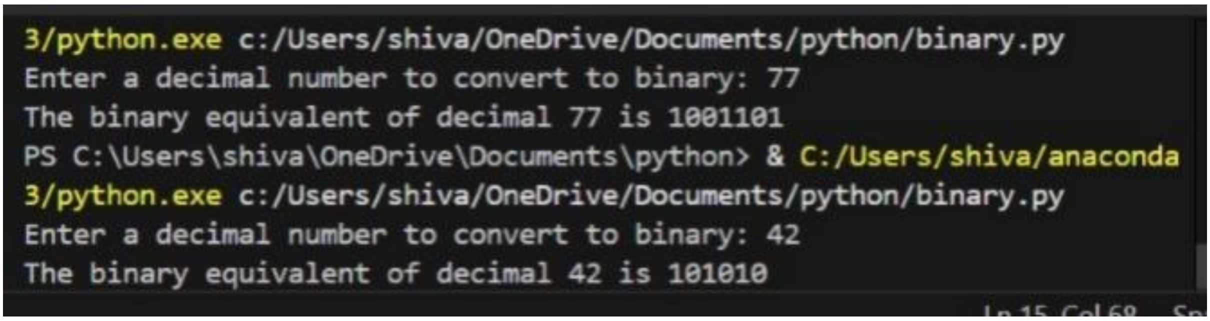
- Compare clarity with zero-shot output.
- Analyse handling of zero and negative numbers.

Code:



```
'''
input num = 77
print binary of 77 is 1001101
'''
def decimal_to_binary(n):
    if n == 0:
        return "0"
    binary_str = ""
    while n > 0:
        binary_str = str(n % 2) + binary_str
        n = n // 2
    return binary_str
num = int(input("Enter a decimal number to convert to binary: "))
binary_result = decimal_to_binary(num)
print(f"The binary equivalent of decimal {num} is {binary_result}")
```

Output:



```
3/python.exe c:/Users/shiva/OneDrive/Documents/python/binary.py
Enter a decimal number to convert to binary: 77
The binary equivalent of decimal 77 is 1001101
PS C:\Users\shiva\OneDrive\Documents\python> & C:/Users/shiva/anaconda
3/python.exe c:/Users/shiva/OneDrive/Documents/python/binary.py
Enter a decimal number to convert to binary: 42
The binary equivalent of decimal 42 is 101010
```

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:

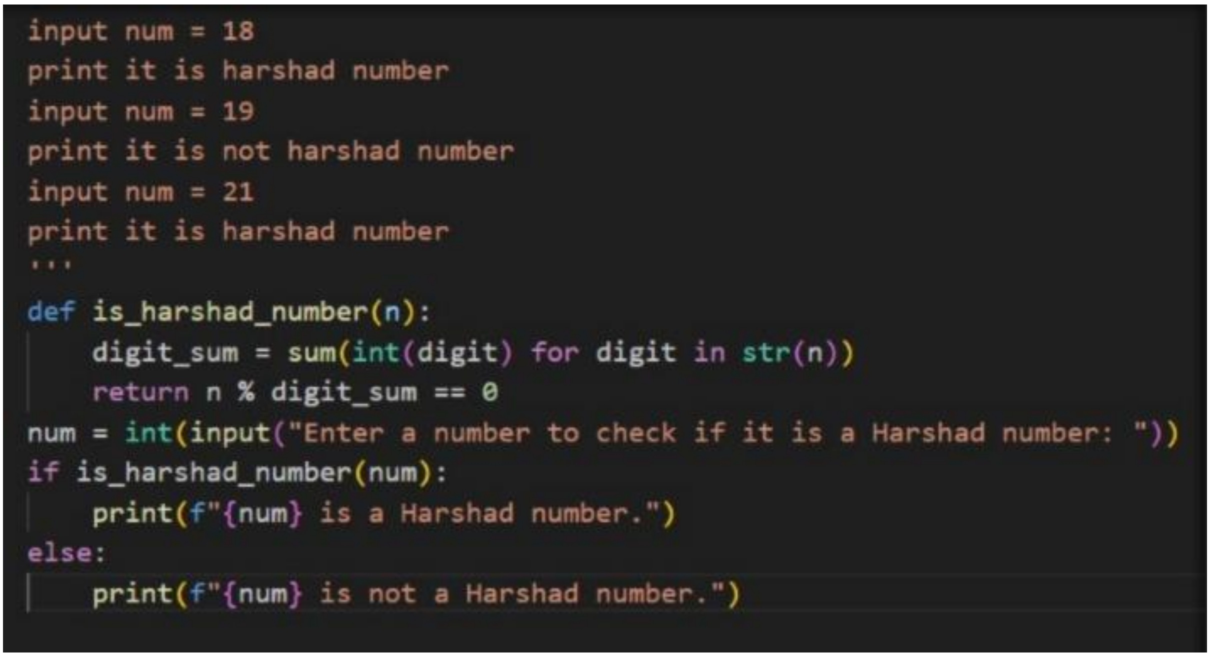
- Input: 18 → Output: Harshad Number

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

Task:

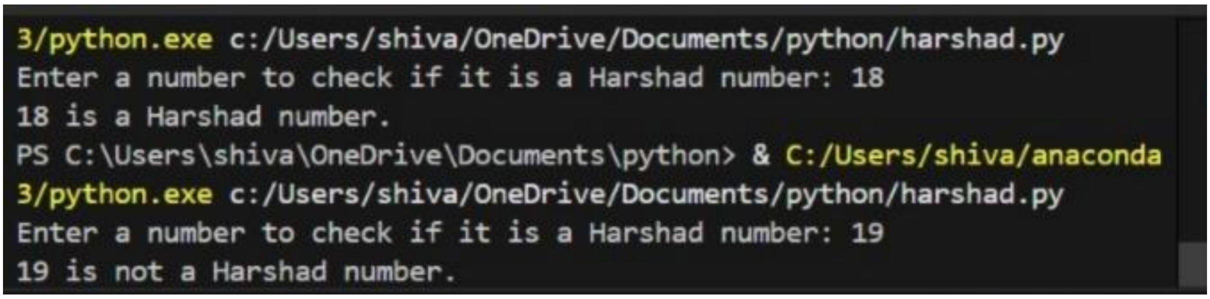
- Test boundary conditions.
- Evaluate robustness

Code:



```
input num = 18
print it is harshad number
input num = 19
print it is not harshad number
input num = 21
print it is harshad number
'''
def is_harshad_number(n):
    digit_sum = sum(int(digit) for digit in str(n))
    return n % digit_sum == 0
num = int(input("Enter a number to check if it is a Harshad number: "))
if is_harshad_number(num):
    print(f"{num} is a Harshad number.")
else:
    print(f"{num} is not a Harshad number.")
```

Output:



```
3/python.exe c:/Users/shiva/OneDrive/Documents/python/harshad.py
Enter a number to check if it is a Harshad number: 18
18 is a Harshad number.
PS C:\Users\shiva\OneDrive\Documents\python> & C:/Users/shiva/anaconda
3/python.exe c:/Users/shiva/OneDrive/Documents/python/harshad.py
Enter a number to check if it is a Harshad number: 19
19 is not a Harshad number.
```