Lab Assignment - 3.5

Name : B.ROHITH
HT.No. : 2303A51882
Batch No. : 02

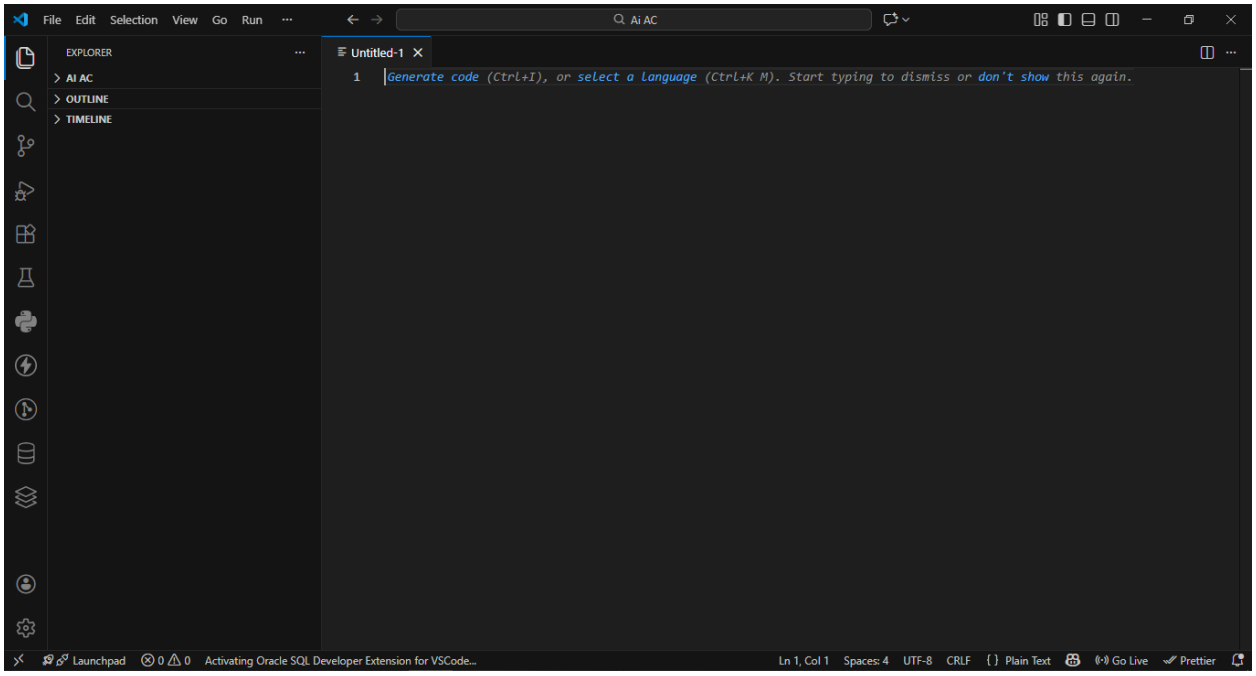
Question 1: Zero-Shot Prompting (Leap Year Check)

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

Task:

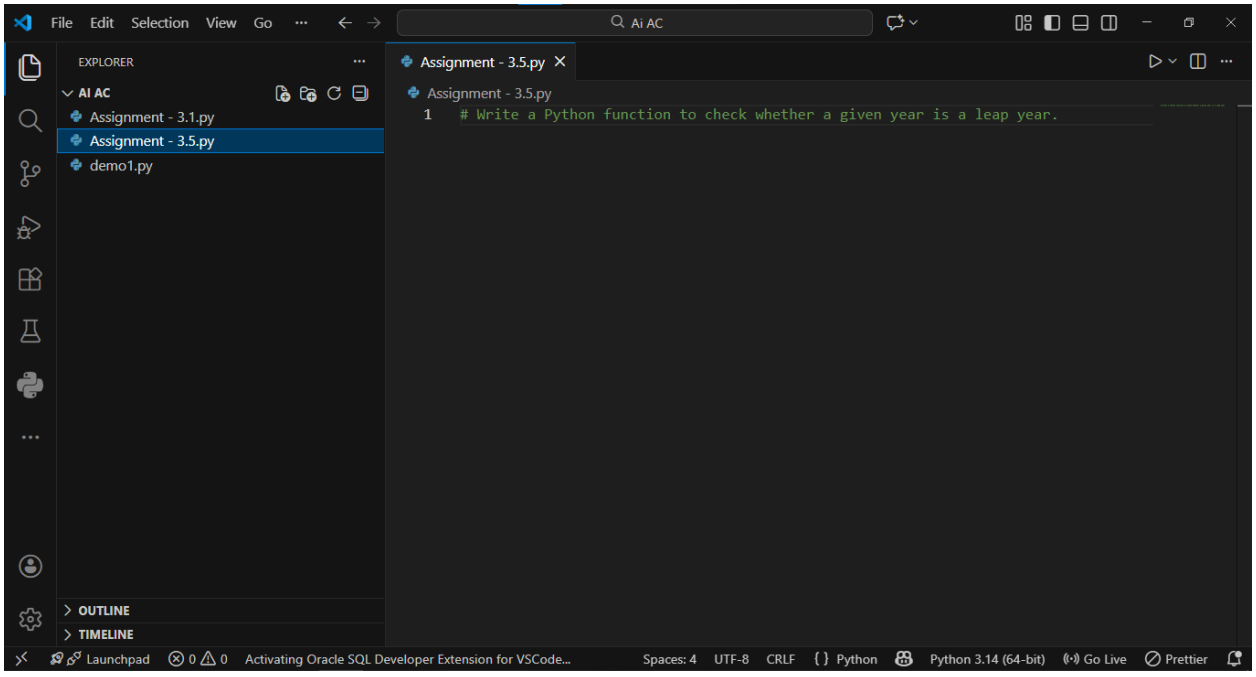
- Record the AI-generated code.
- Test with years like 1900, 2000, 2024.
- Identify logical flaws or missing conditions.

Copilot VSCode Workspace :

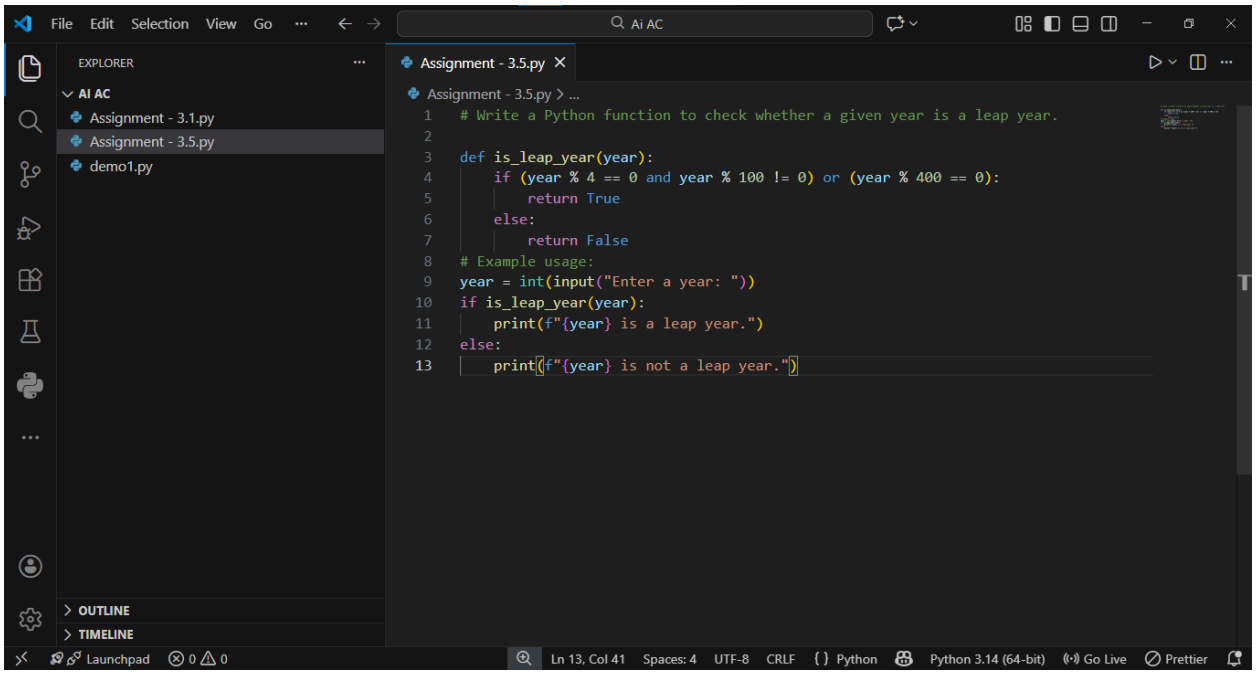


Prompt Used (Zero-Shot):

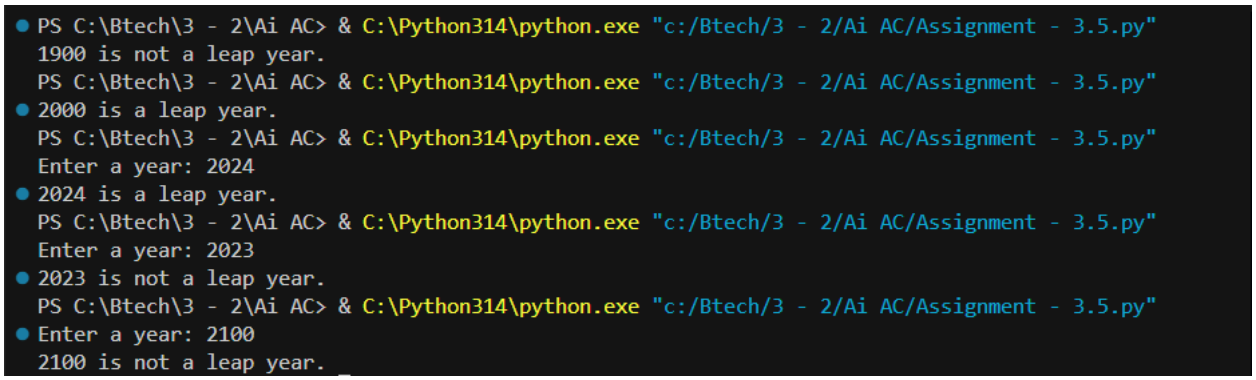
Write a Python function to check whether a given year is a leap year.



Copilot Suggested Code :



Input / Output test cases screenshot :



Observation & Logical Flaws :

- Fails for **century years**
- Missing condition:

  Years divisible by **100** are **not leap years** unless divisible by **400.**

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:

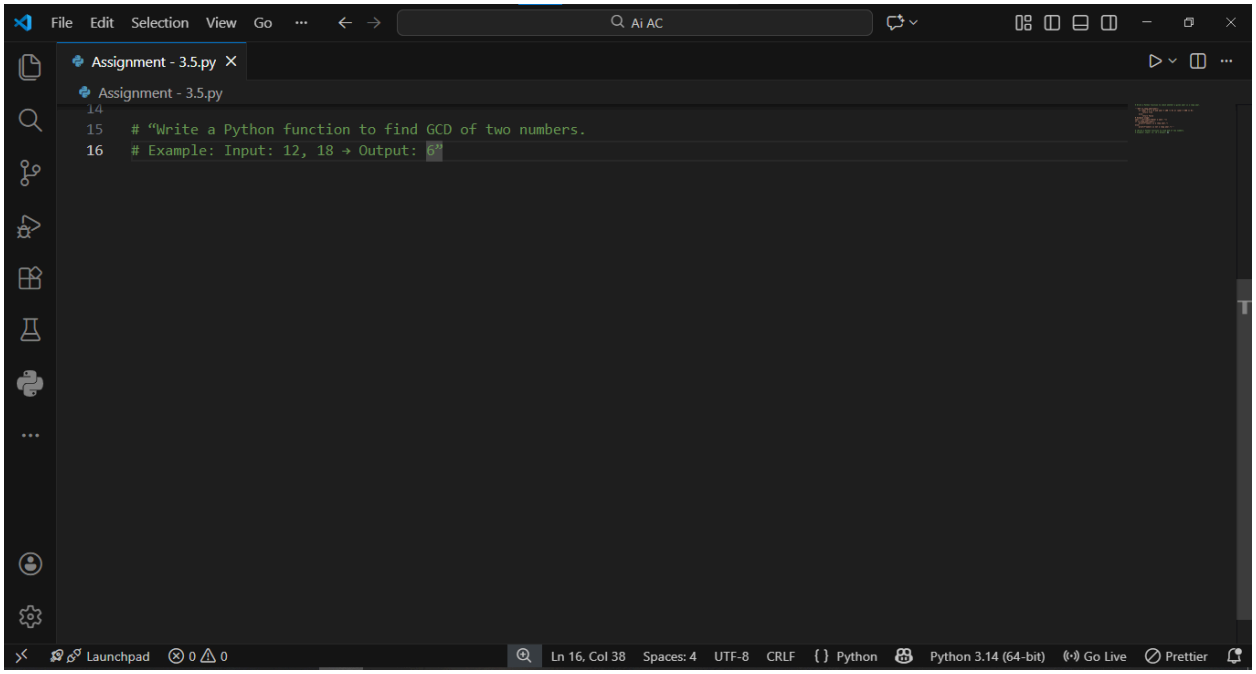
• Compare with a zero-shot solution.

• Analyze algorithm efficiency.

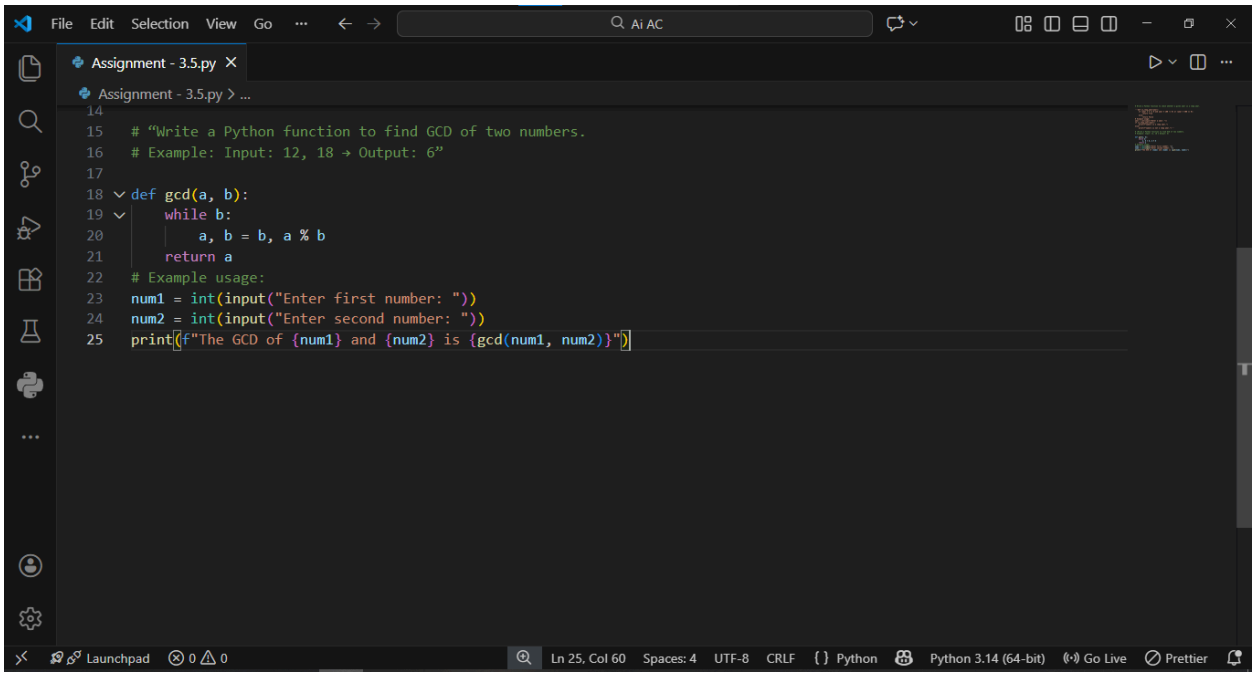
Prompt Used (One-Shot):

Write a Python function to find GCD of two numbers.

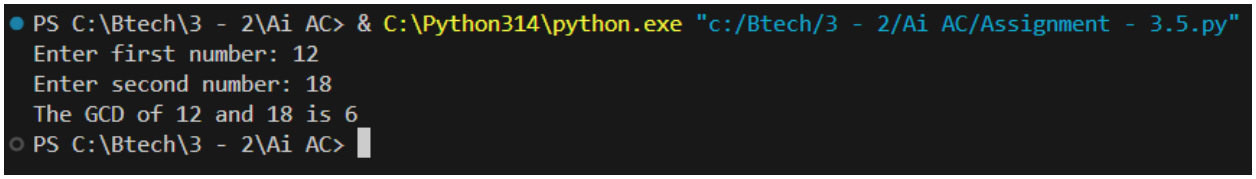
Example: Input: 12, 18 → Output: 6



Copilot Suggested Code :



Input / Output test cases screenshot :



Comparison with Zero-Shot :

Aspect	Zero-Shot	One-Shot
Clarity	Medium	High
Algorithm	Trial-based	Euclidean
Efficiency	Low	High

Efficiency Analysis :

- Time Complexity: **O(log min(a, b))**
- Very efficient for large numbers

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.

Prompt Used (Few-Shot Prompt):

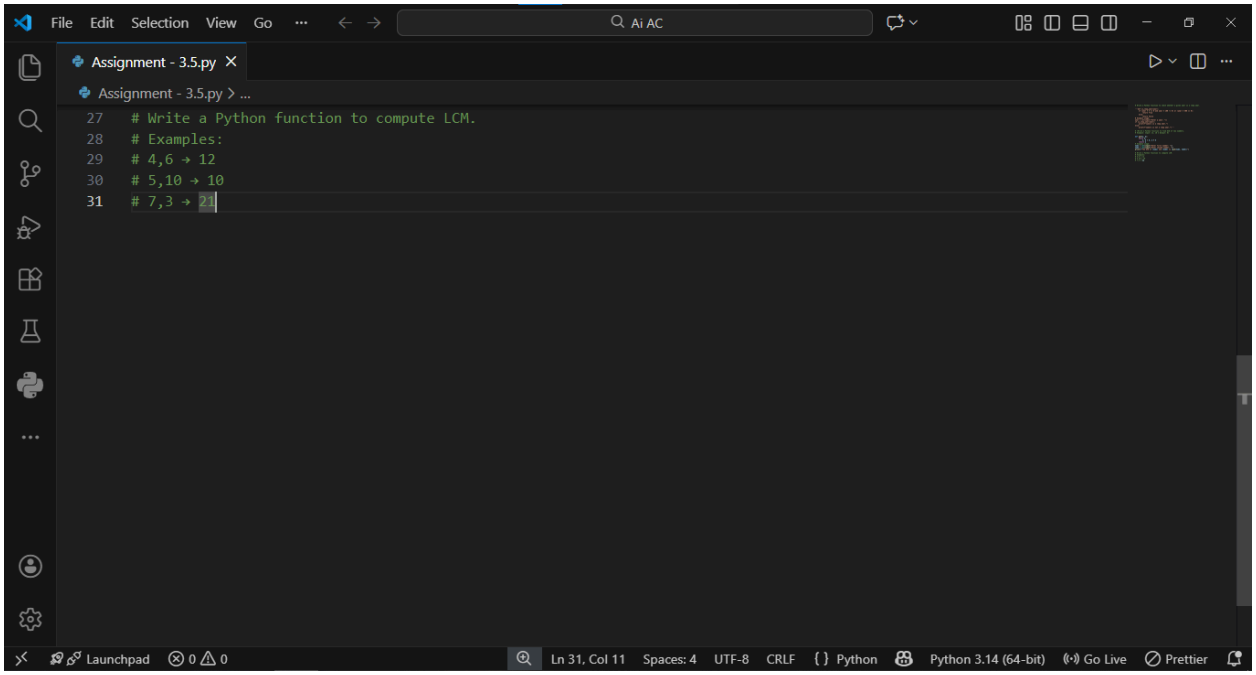
Write a Python function to compute LCM.

Examples:

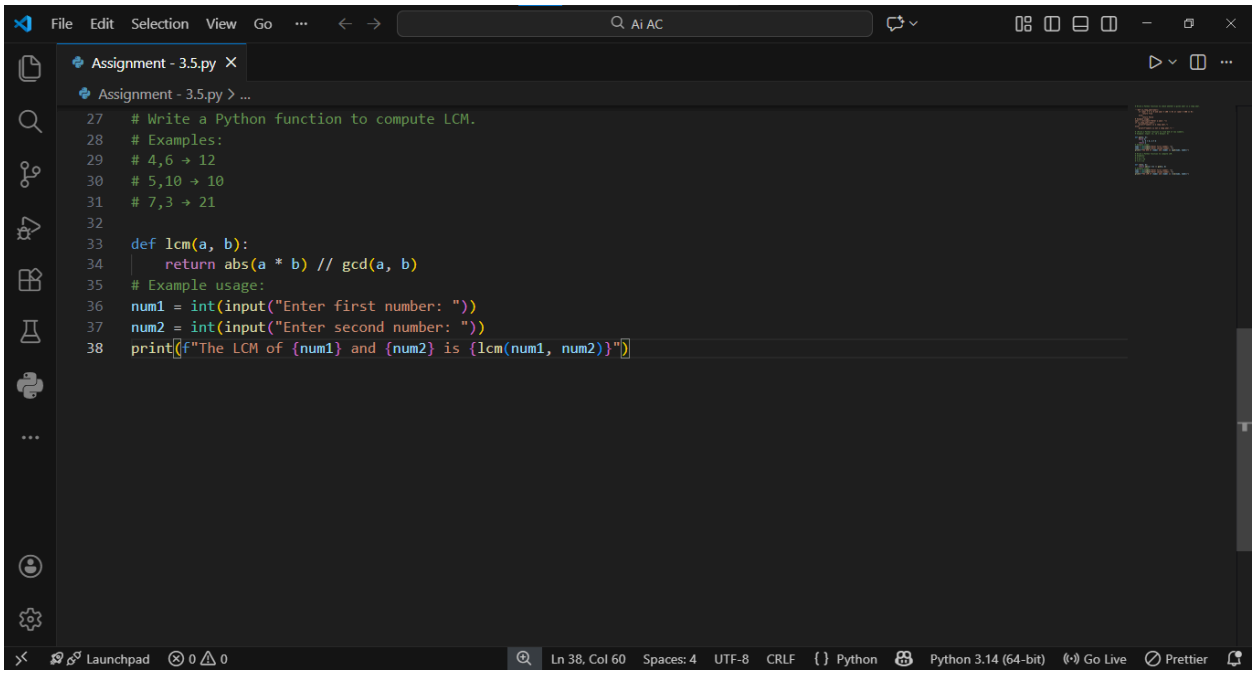
4,6 → 12

5,10 → 10

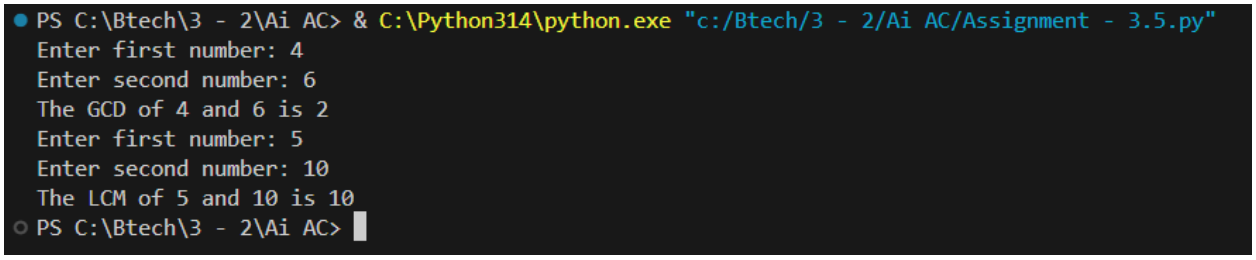
7,3 → 21



Copilot Suggested Code :



Input / Output test cases screenshot :



How Examples Guided the Model :

- Correctly inferred LCM = (a × b) / GCD
- Used efficient GCD internally

Edge Case Testing :

```
print(lcm(0, 5))   # Output: 0

print(lcm(1, 1))   # Output: 1
```

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

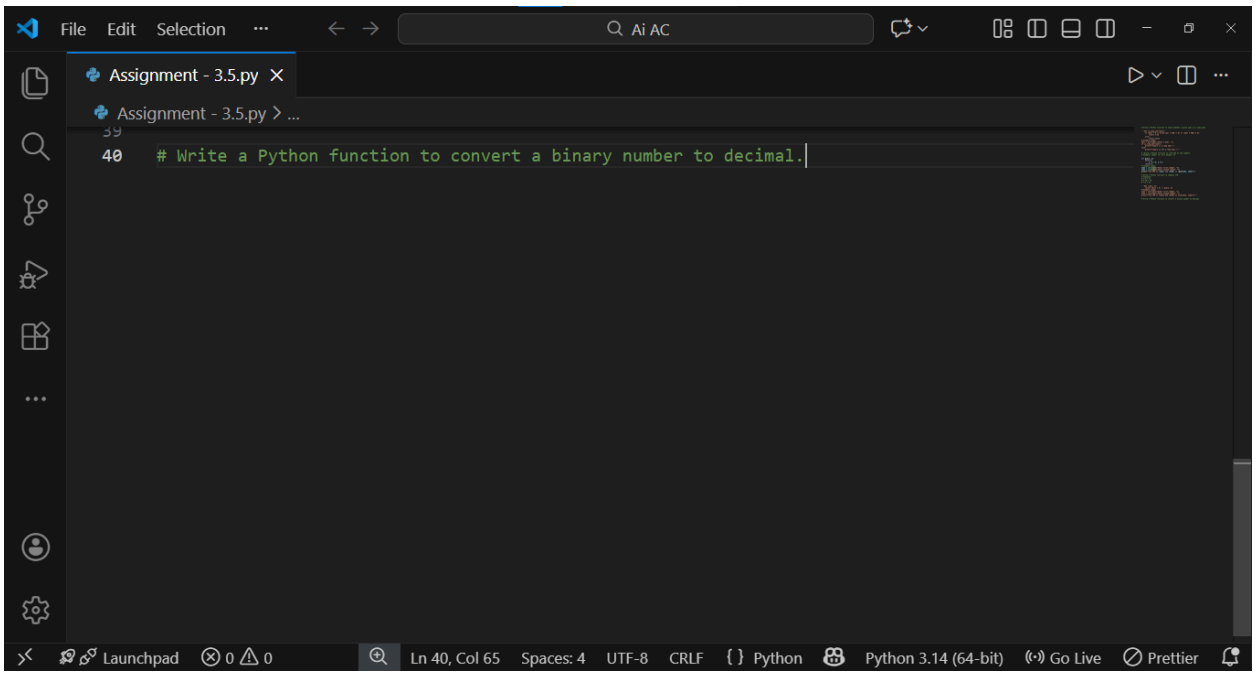
Write a zero-shot prompt to generate a Python function that converts a binary number to decimal.

Task:

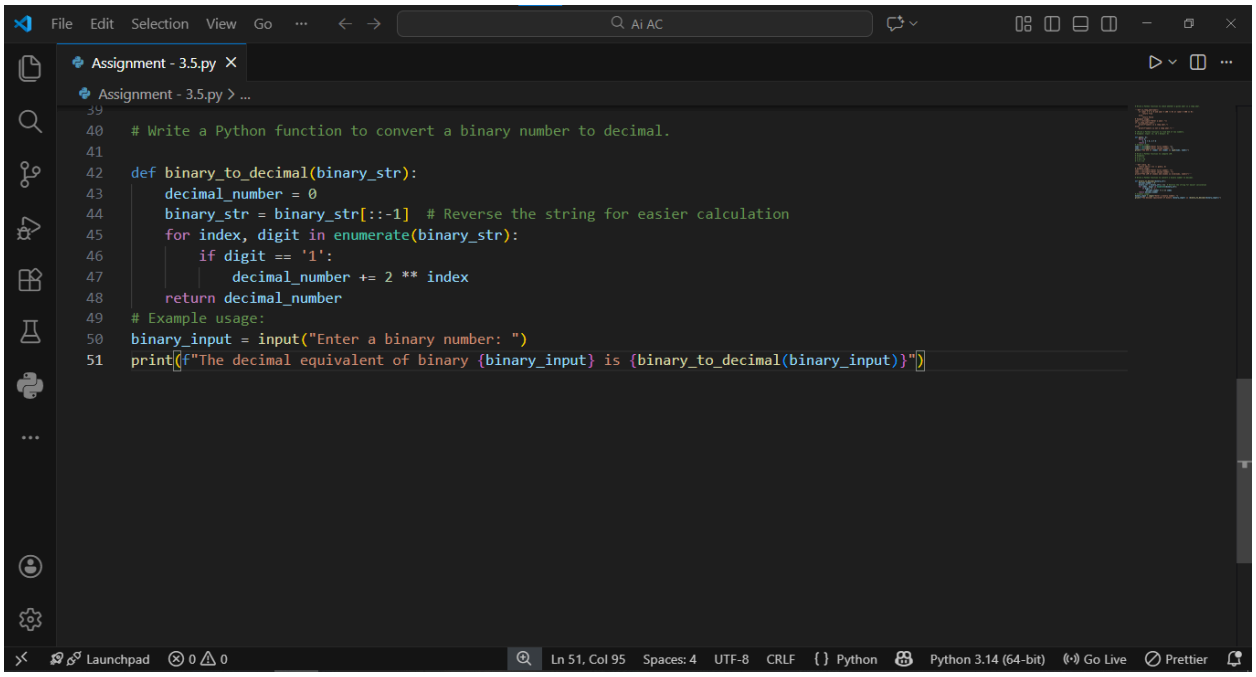
- Test with valid and invalid binary inputs.
- Identify missing validation logic.

Prompt Used (Zero-Shot):

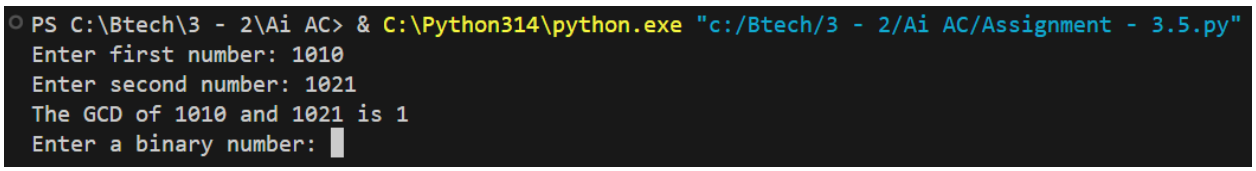
Write a Python function to convert a binary number to decimal.



Copilot Suggested Code :



Input / Output test cases screenshot :



Testing :

```
print(binary_to_decimal("1010"))   # Output: 10

print(binary_to_decimal("1021"))   # Error
```

Missing Validation :

- No check for invalid binary digits

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:

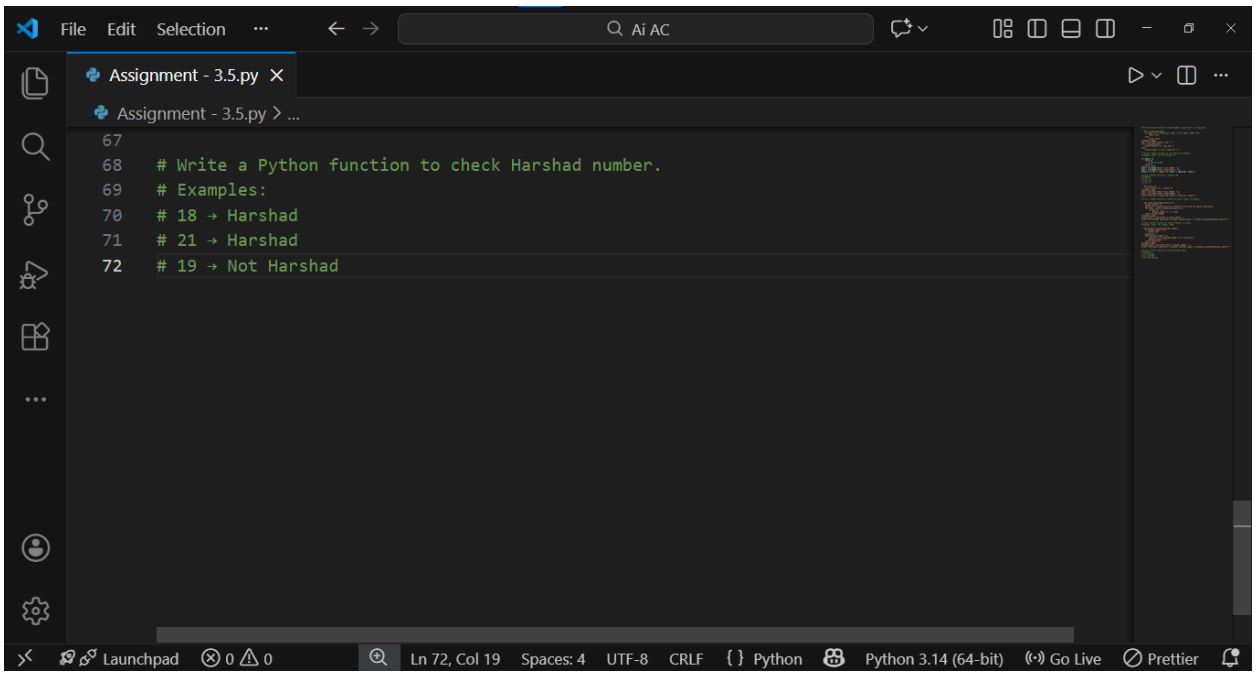
• Compare clarity with zero-shot output.

• Analyze handling of zero and negative numbers.

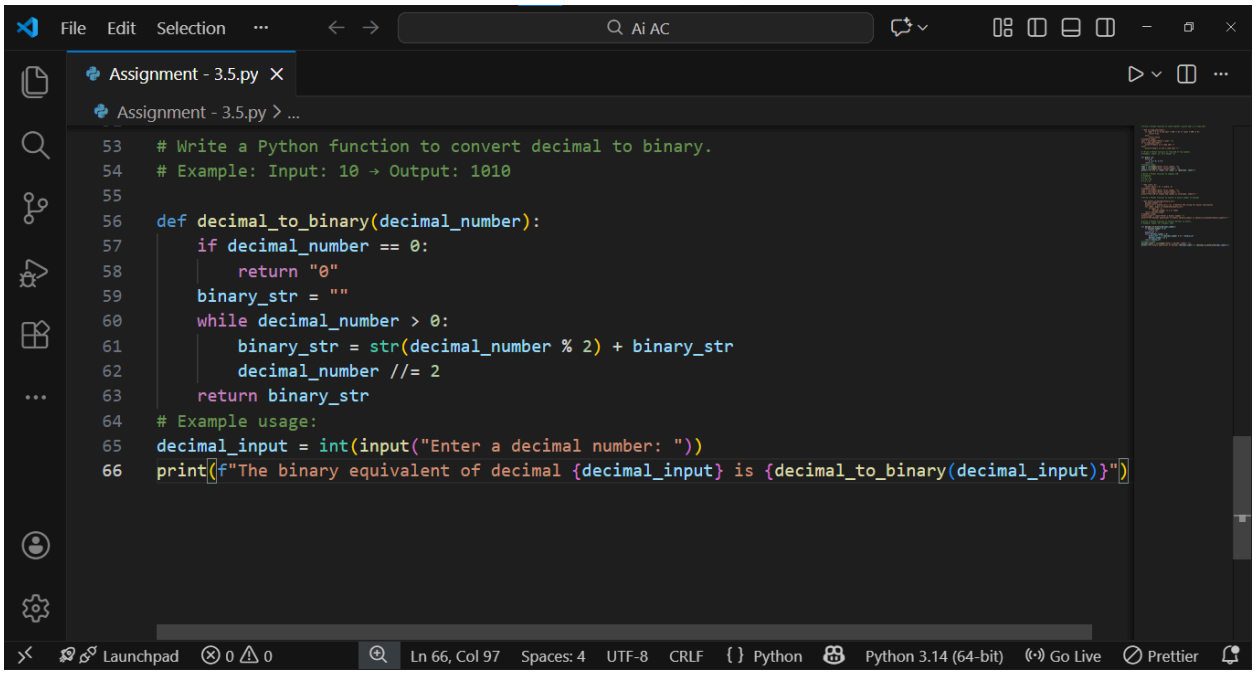
Prompt Used (One-Shot):

Write a Python function to convert decimal to binary.

Example: Input: 10 → Output: 1010



Copilot Suggested Code :



Input / Output test cases screenshot :

Edge Case Analysis :`

```
print(decimal_to_binary(0))     # Output: 0

print(decimal_to_binary(-5))    # Output: b101
(incorrect)
```

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

Prompt Used (Few-Shot Prompt):

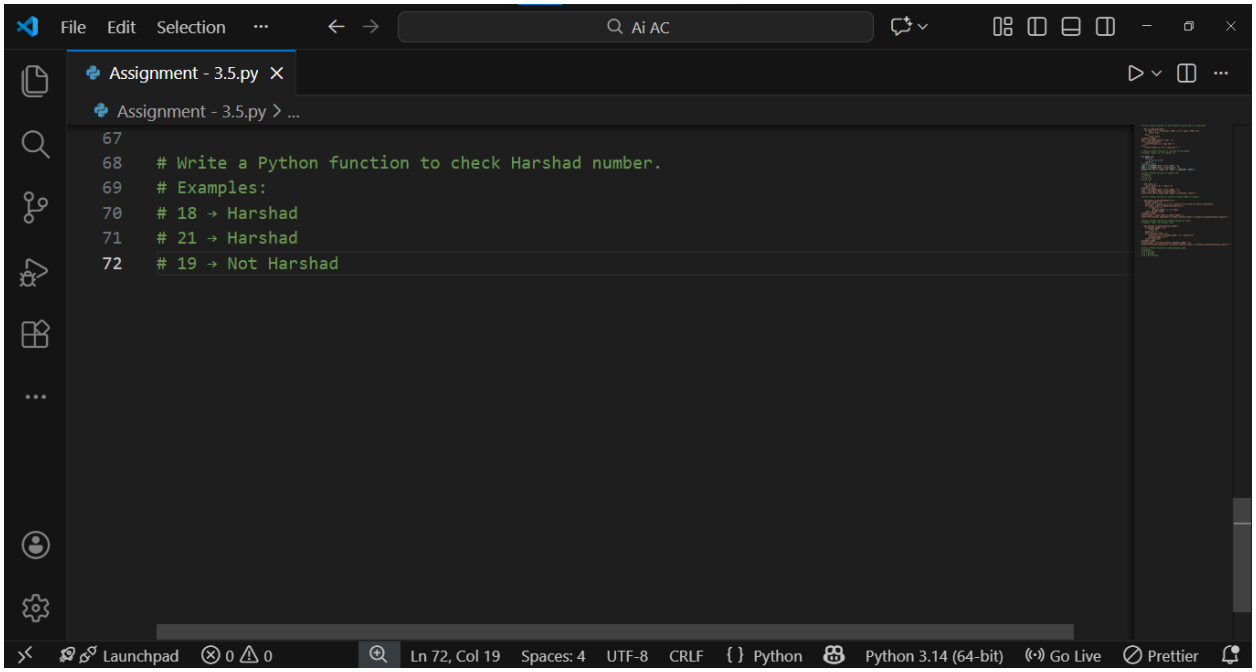
Write a Python function to check Harshad number.

Examples:

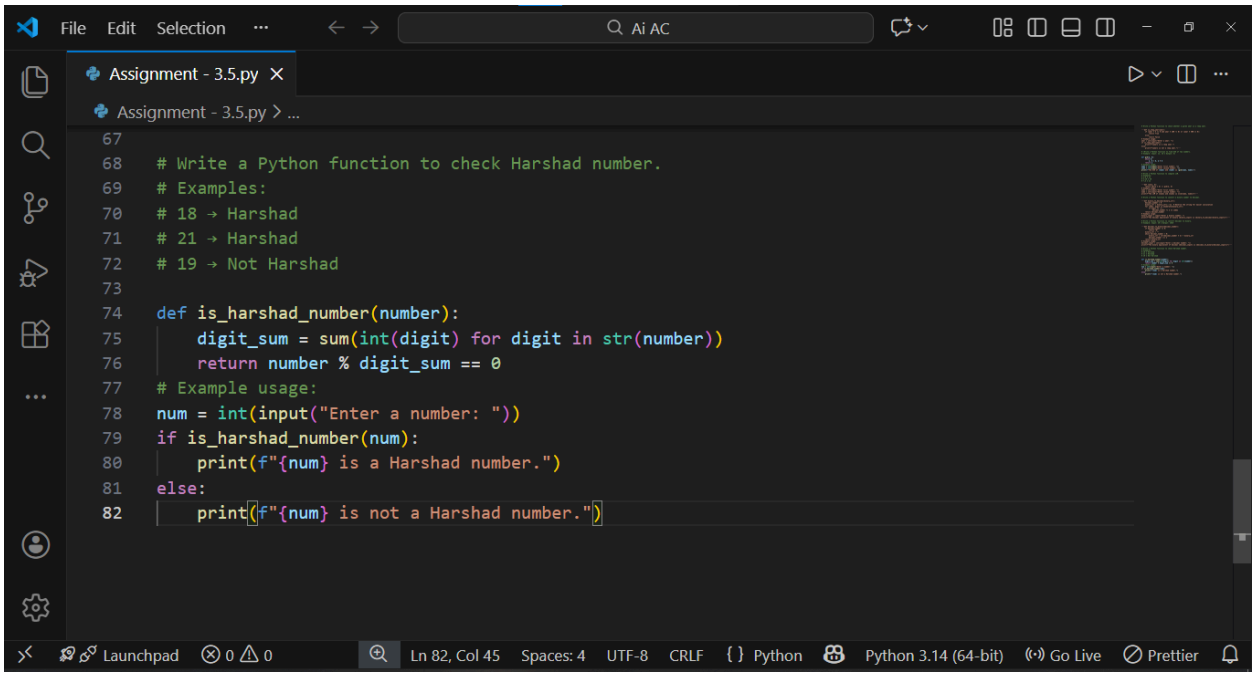
18 → Harshad

21 → Harshad

19 → Not Harshad



Copilot Suggested Code :



Input / Output test cases screenshot :

Testing :

```
print(is_harshad(18))  # True

print(is_harshad(21))  # True

print(is_harshad(19))  # False
```

Boundary Conditions :

- Fails for `n = 0` (division by zero)

Robust Version :

```
def is_harshad(n):
    if n <= 0:
        return False
    digit_sum = sum(int(d) for d in str(n))
    return n % digit_sum == 0
```