**AI Assisted Coding - Lab Test 2**

**Htno : 2503A52L16**

**Subgroup D**

**D.1 — [S15D1] TDD: increment version suffix**

**Prompt :**

"Write a Python function bump\_version(name) that adds or increments a \_vNN version suffix before the file extension. Keep the extension, preserve zero-padding (at least 2 digits), and handle cases with or without an existing suffix (e.g., report\_v1.csv → report\_v02.csv, summary.csv → summary\_v01.csv)."

**Code logic :**

def bump\_version(name: str) -> str:

"""

Logic:

1. Split filename into base and extension.

2. Check if base ends with '\_vNN' using regex.

3. If yes → increment number, keep padding (min 2 digits).

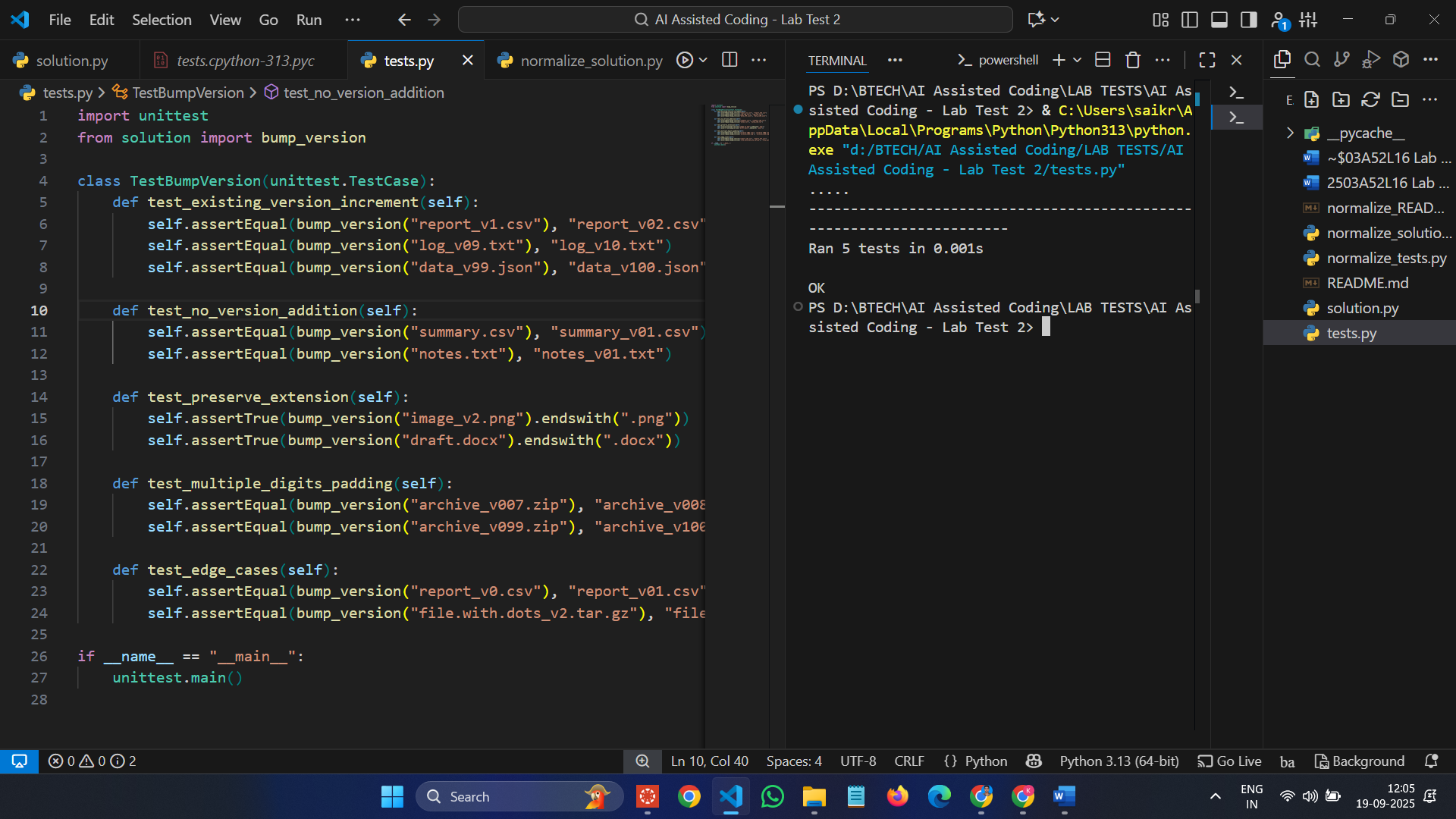
4. If no → add '\_v01'.

5. Return new base + extension.

"""

Pass

**Output :**



**Explanation :**

The function checks if the filename already has a version like \_vNN. If yes, it increases the number while keeping at least 2 digits. If no version is found, it adds \_v01 before the extension. The file extension always stays the same.

**D.2 — [S15D2] Generate docstrings and usage examples**

**Prompt :**

"Write a Python function normalize(scores) that scales a list of numbers to [0,1]. If the list is empty, return []. If all values are equal, return zeros of the same length to avoid divide-by-zero."

**Code logic :**

def normalize(scores):

"""

Logic:

1. If scores is empty → return []

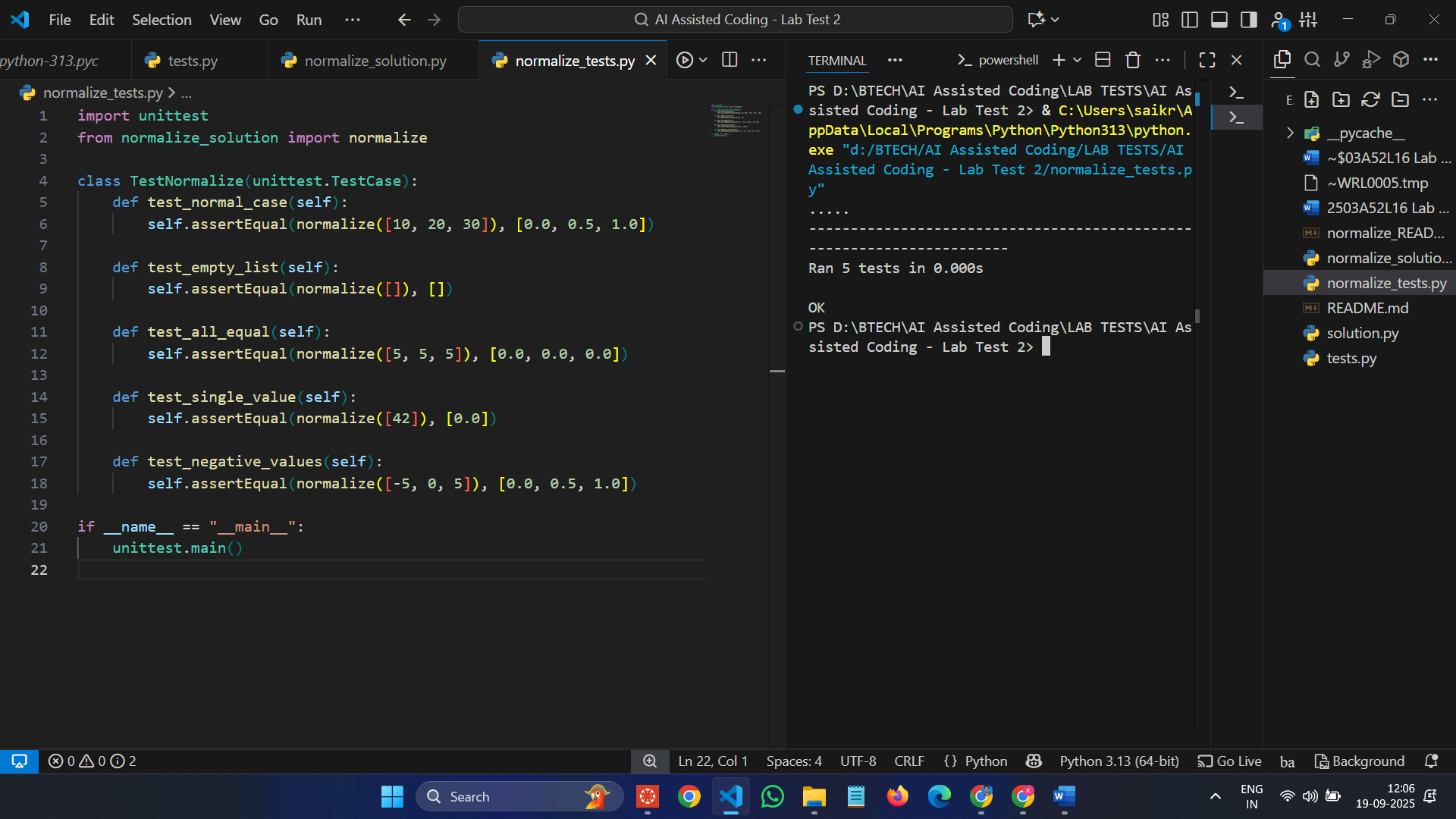
2. If all values equal → return zeros

3. Otherwise → (x - min) / (max - min) for each score

"""

pass

**Output :**



**Explanation :**

The function normalizes a list of numbers to the [0,1] range for comparability. It returns an empty list if the input is empty, zeros if all values are equal, and otherwise scales each number using (x - min) / (max - min).