**AI ASSISTED CODING**

**LAB TEST – 2**

Name: Perala Akshaya

HT NO.: 2403A51264

Batch No.: 11

**TASK – I**

**SUBTASK – I.1**

 Top-3 frequent words

Context:

Text analytics in real estate listings platform.

Your Task:

Top-3 words by frequency; tie-break lexicographically.

Data & Edge Cases:

Lowercase + split by spaces.

AI Assistance Expectation:

Ask AI for Counter and sort keys.

Constraints & Notes:

Correct tie-breaking.

Sample Input

to be or not to be that is the question

Sample Output

[('to', 2), ('be', 2), ('is', 1)]

Acceptance Criteria: Tie-breaking lexicographically

**Prompt:**

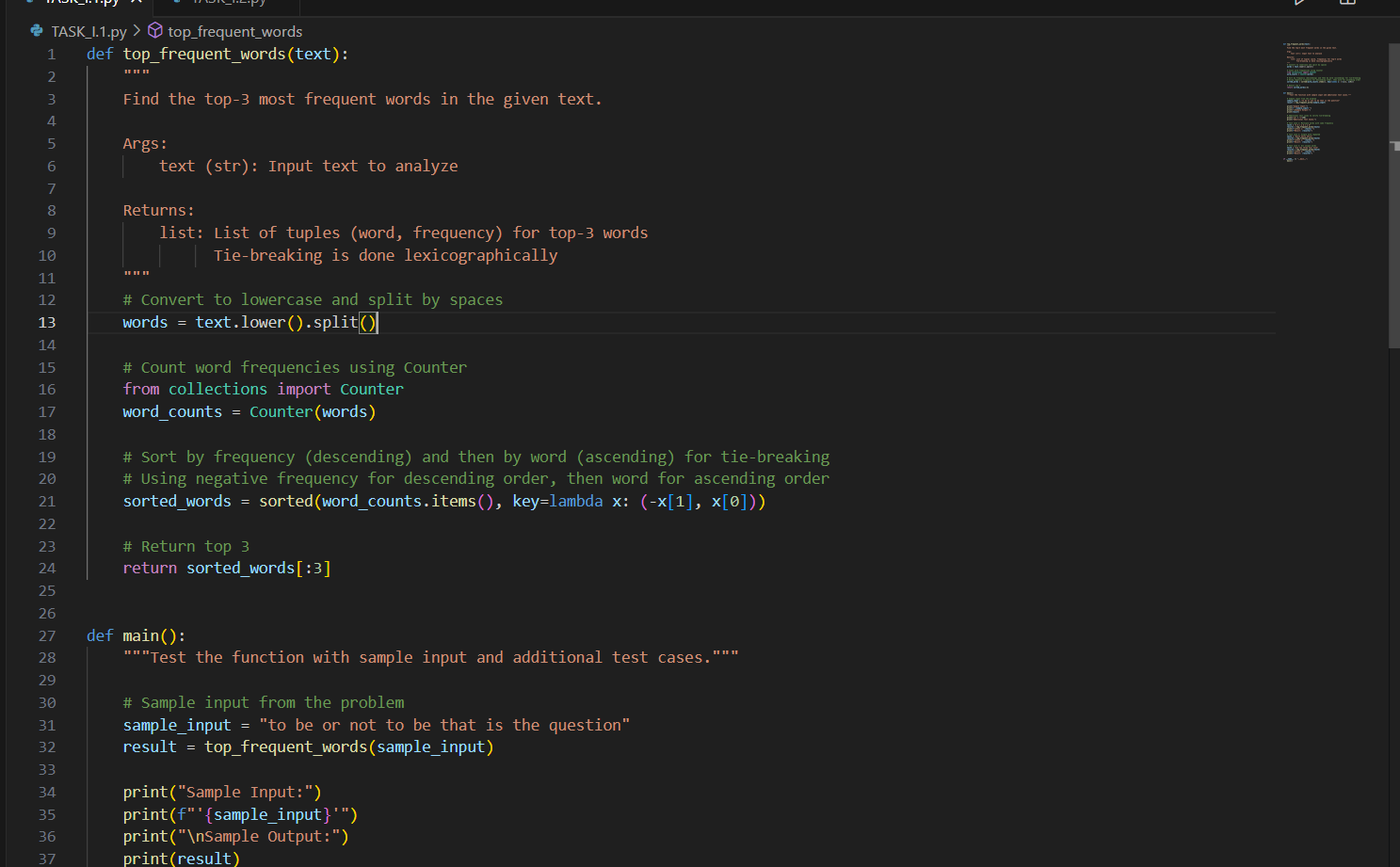
Given a lowercase string of words separated by spaces, find the top 3 most frequent words. If there is a tie, order them alphabetically. Return a list of (word, count) tuples.

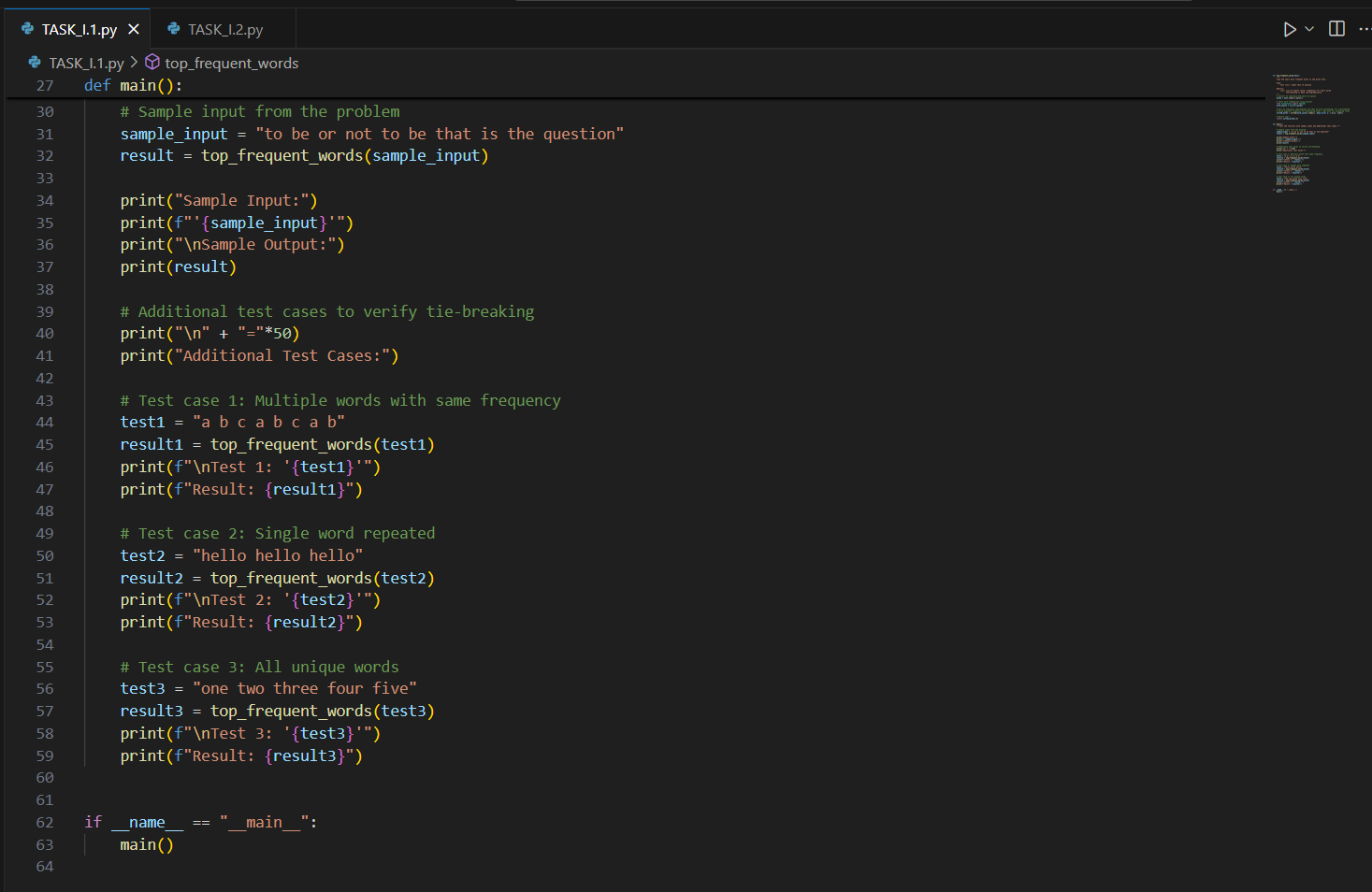
Example:

Input: "to be or not to be that is the question"

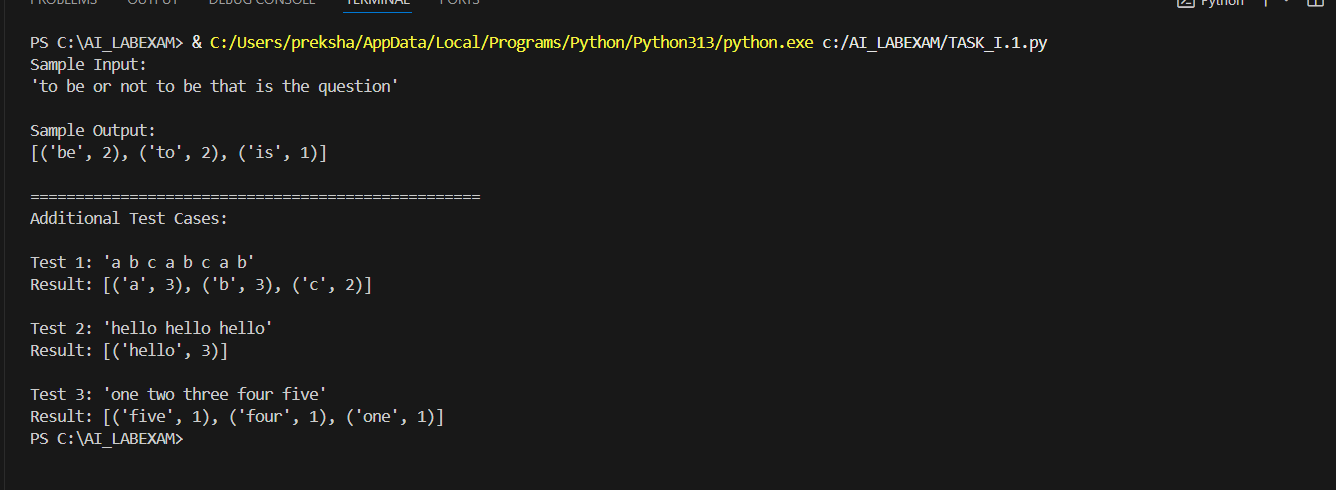
Output: [('to', 2), ('be', 2), ('is', 1)]

**CODE:**

****

****

**OUTPUT:**

****

**SUBTASK – I.2**

 Implement LRUCache (capacity 2)

Context:

LRU cache for real estate listings platform service.

 Your Task:

Implement capacity=2 LRU with get/put.

 Data & Edge Cases:

Operations sequence provided.

 AI Assistance Expectation:

OrderedDict approach.

 Constraints & Notes:

Deterministic behavior.

Sample Input

ops=[("put",1,1),("put",2,2),("get",1),("put",3,3),("get",2),("get",3)]

Sample Output

[None, None, 1, None, -1, 3]

Acceptance Criteria: Correct eviction

**Prompt:**

Implement an LRU cache with capacity 2 that supports get(key) and put(key, value) operations.

get returns the value or -1 if not found, and marks the key as recently used.

put adds or updates the key, evicting the least recently used if full.

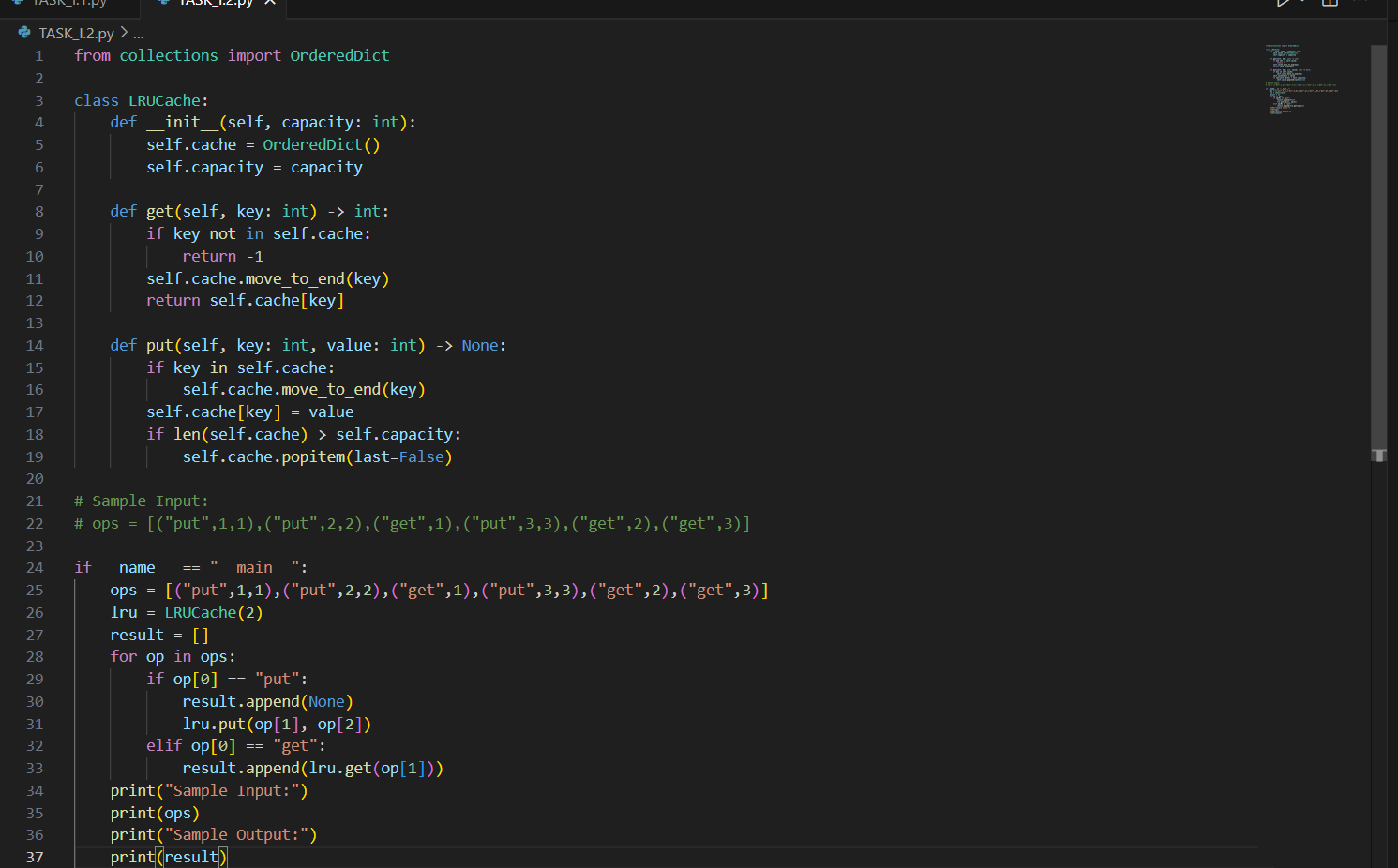
Return a list of results for a sequence of operations (None for put, value or -1 for get).

Example:

Input: [("put",1,1),("put",2,2),("get",1),("put",3,3),("get",2),("get",3)]

Output: [None, None, 1, None, -1, 3]

**CODE:**



**OUTPUT:**

