

AI ASSISTED CODING

LAB TEST-02

NAME:SUNKARI NAGARJUNA REDDY

HTNO:2403A52064

BATCH:03

SUBGROUP-F

F.1 — [S14F1] Add type hints and fix None handling

Scenario (telecom network):

Context:

A helper in telecom network sometimes returns None, causing downstream type errors.

Your Task:

Annotate types and ensure the function always returns str or raises ValueError if not found.

Data & Edge Cases:

Given ids=['a','b'] and target='c' -> expect ValueError('not found').

AI Assistance Expectation:

Use AI to propose a typed signature and non-None return guarantees.

Constraints & Notes:

Include a negative test that expects ValueError.

Sample Input

ids=['a','b'], target='c'

Sample Output

ValueError('not found')

Acceptance Criteria: Typed; raises on miss; no implicit None

CODE:

```
1 from typing import List
2
3 def find_id(ids: List[str], target: str) -> str:
4     """
5     Search for target in ids list.
6     Returns the target string if found.
7     Raises ValueError if not found.
8     """
9     if target in ids:
10         return target
11     raise ValueError("not found")
12
13
14 # --- Dynamic Input ---
15 ids_input = input("Enter IDs separated by space: ").split()
16 target_input = input("Enter target ID: ")
17
18 try:
19     result = find_id(ids_input, target_input)
20     print("Result:", result)
21 except ValueError as e:
22     print("Error:", e)
```

Python Debug Console

```
libs\debugpy\launcher '50165' '-' 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff\from typing import List.py'
Enter IDs separated by space: a b c
Enter target ID: b
Result: b
```

OUTPUT:

```
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> & 'c:\Users\nagar\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\nagar\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50165' '-' 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff\from typing import List.py'
Enter IDs separated by space: a b c
Enter target ID: b
Result: b
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> ^C
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff>
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> c: cd 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff' & 'c:\Users\nagar\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\nagar\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64067' '-' 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff\from typing import List.py'
Enter IDs separated by space: a b
Enter target ID: z
Error: not found
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> ^C
```

OBSERVATION:

- 1.If the target is present-prints it.
- 2.If the target is absent-raises valueerror and prints the error.

F.2 — [S14F2] Optimize membership checks

Scenario (telecom network):

Context:

A streaming job in telecom network checks if IDs are in a large corpus.

Your Task:

Optimize membership checks by converting corpus to a set once, then mapping stream to
booleans.

Data & Edge Cases:

corpus=[1,2,3,4,5]; stream=[2,5,9] -> [True, True, False].

AI Assistance Expectation:

Ask AI to suggest complexity improvements and micro-bench ideas.

Constraints & Notes:

Return list[bool] aligned to stream order.

Sample Input

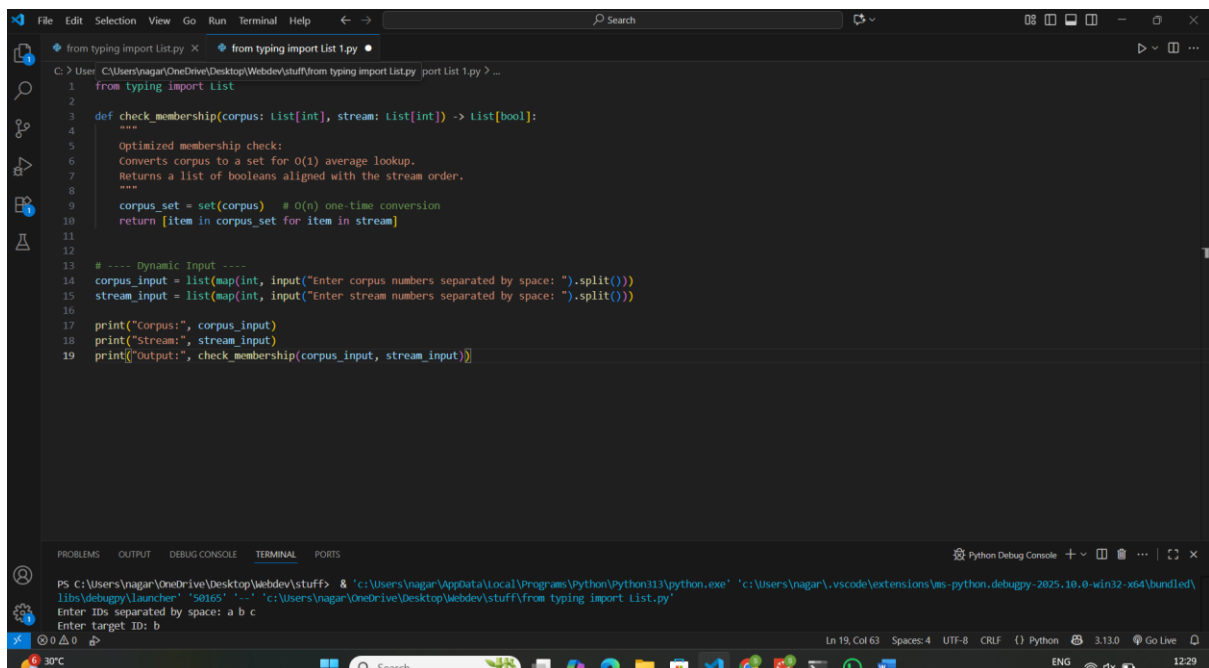
corpus=[1,2,3,4,5], stream=[2,5,9]

Sample Output

[True, True, False]

Acceptance Criteria: Uses set; correct Booleans

CODE:



```
1 from typing import List
2
3 def check_membership(corpus: List[int], stream: List[int]) -> List[bool]:
4     """
5     Optimized membership check:
6     Converts corpus to a set for O(1) average lookup.
7     Returns a list of booleans aligned with the stream order.
8     """
9     corpus_set = set(corpus) # O(n) one-time conversion
10    return [item in corpus_set for item in stream]
11
12
13 # ---- Dynamic Input ----
14 corpus_input = list(map(int, input("Enter corpus numbers separated by space: ").split()))
15 stream_input = list(map(int, input("Enter stream numbers separated by space: ").split()))
16
17 print("Corpus:", corpus_input)
18 print("Stream:", stream_input)
19 print("Output:", check_membership(corpus_input, stream_input))
```

OUTPUT:



```
Enter corpus numbers separated by space: 2 5 1
Enter stream numbers separated by space: 2 5
Corpus: [2, 5, 1]
Stream: [2, 5]
Output: [True, True]
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff>
```

OBSERVATION:

The solution uses a set, which makes membership checking very fast and independent of input size.

The output matches the stream order giving the correct sequence of boolean values.