203A51266

AI-ASSISTED CODING

LAB EXAM-2

SET – C

C.1 — [S18C1] Debug de-duplication (case-insensitive)

Context:

Contact records in the real estate listings platform CRM contain duplicates differing only by case.

Your Task:

Return the first occurrence of each email (case-insensitive) while preserving original order.

Data & Edge Cases:

Input: list of emails. Normalize keys using lowercase; output retains original casing.

AI Assistance Expectation:

Use AI to spot bug (reinitializing `seen` in loop) and correct to stable algorithm.

Constraints & Notes:

Add unit tests covering mixed-case duplicates.

Sample Input

['A@x.com', 'a@x.com', 'B@y.com']

Sample Output

['A@x.com', 'B@y.com']

PROMPT:

Write Python code to remove duplicate email addresses from a list, treating them as duplicates regardless of case.

Requirements:

- Preserve the original casing and order of the first occurrence
- Ignore duplicates that differ only by case (e.g., 'B@Y.com' and 'b@y.com' are duplicates)

Constraints:

- Normalize comparison using lowercase
- Output must retain original casing and order

Sample Input: ['a@x.com', 'B@Y.com', 'b@y.com']
Expected Output: ['a@x.com', 'B@Y.com']

Include unit tests for mixed-case duplicates.

Bonus: Spot and fix the bug where seen is reinitialized inside the loop.

Code:

```python
def deduplicate_emails(emails):
    seen = set()
    result = []
    for email in emails:
        key = email.lower()
        if key not in seen:
            seen.add(key)
            result.append(email)
    return result
def test_deduplicate_emails():
    assert deduplicate_emails(['A@x.com', 'a@x.com', 'B@y.com']) == ['A@x.com', 'B@y.com']
    assert deduplicate_emails(['a@x.com', 'A@x.com', 'A@X.com']) == ['a@x.com']
    assert deduplicate_emails(['a@x.com', 'b@y.com', 'B@y.com', 'c@z.com']) == ['a@x.com', 'b@y.com', 'c@z.com']
    assert deduplicate_emails([]) == []
    assert deduplicate_emails(['A@x.com']) == ['A@x.com']

if __name__ == "__main__":
    test_deduplicate_emails()
    print("All tests passed.")
    print(deduplicate_emails(['A@x.com', 'a@x.com', 'B@y.com']))
    print(deduplicate_emails(['a@x.com', 'A@x.com', 'A@X.com']))
    print(deduplicate_emails(['a@x.com', 'b@y.com', 'B@y.com', 'c@z.com']))
```

Output:

```
PS C:\Users\DEll\Desktop\lab 7>  c:; cd 'c:\Users\DEll\Desktop\lab 7'; & 'c:\Users\DEll\AppData\Local\Programs\Python\Python313\python.exe' '
\Users\DEll\.vscode\extensions\ms-python.debugpy-2025.13.2025091201-win32-x64\bundled\libs\debugpy\launcher' '62215' '--' 'C:\Users\DEll\Deskt
p\lab 7\ai.lab.py'
All tests passed.
['A@x.com', 'B@y.com']
['a@x.com']
['a@x.com', 'b@y.com', 'c@z.com']
PS C:\Users\DEll\Desktop\lab 7>  c:; cd 'c:\Users\DEll\Desktop\lab 7'; & 'c:\Users\DEll\AppData\Local\Programs\Python\Python313\python.exe' '
\Users\DEll\.vscode\extensions\ms-python.debugpy-2025.13.2025091201-win32-x64\bundled\libs\debugpy\launcher' '62236' '--' 'c:\Users\DEll\Deskt
p\lab 7\ai.lab.2.py'
```

C.2 — [S18C2] TDD: slugify titles

Context:

Titles in the real estate listings platform CMS must become SEO slugs.

Your Task:

TDD for slugify(text): lowercase, remove non-alnum except hyphen, spaces->hyphen, collapse and trim hyphens.

Data & Edge Cases:

Include punctuation and multiple spaces.

AI Assistance Expectation:

Have AI propose parametric tests then implement regex solution.

Constraints & Notes:

Return correct slugs.

Sample Input

['Hello World!', 'AI & You', 'Set18-C2']

Sample Output

['hello-world', 'ai-you', 'set18-C2']

Prompt:

Write Python code using Test-Driven Development (TDD) for a slugify(text) function.

Requirements:

- Convert text to lowercase
- Replace spaces with hyphens

- Remove all non-alphanumeric characters except hyphens
- Collapse multiple hyphens into one
- Trim leading and trailing hyphens

Include parametric tests using pytest for edge cases like punctuation, multiple spaces, and empty strings.

Sample Input: ["Hello World!", "AI & You", "Set18-C2"]
Expected Output: ["hello-world", "ai-you", "set18-c2"]

Return only the slug string. All tests must pass.

Code:

ai.lab.2.py > deduplicate_emails

```python
import re
import pytest

def slugify(text):
    text = text.lower()
    text = re.sub(r'[^a-z0-9\s-]', '', text)
    text = re.sub(r'\s+', '-', text)
    text = re.sub(r'-{2,}', '-', text)
    text = text.strip('-')
    return text

@pytest.mark.parametrize("input_text,expected_slug", [
    ("Hello World!", "hello-world"),
    ("AI & You", "ai-you"),

])
def test_slugify(input_text, expected_slug):
    result = slugify(input_text)
    print(f"Input: '{input_text}' => Slug: '{result}'")
    assert result == expected_slug
    assert slugify("  Multiple   Spaces  ") == "multiple-spaces"
    assert slugify("Special #$&* Characters!") == "special-characters"

def deduplicate_emails(emails):
    seen = set()
    result = []
    for email in emails:
        key = email.lower()
        if key not in seen:
            seen.add(key)
            result.append(email)
    return result
```

Output:

```
PS C:\Users\DEll\Desktop\lab 7> & C:/Users/DEll/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DEll/Desktop/lab
7/ai.lab.2.py"
Input: 'Hello World!' => Slug: 'hello-world'
Input: 'AI & You' => Slug: 'ai-you'
```