Solution 26: Python Word Frequency

Part A — Review: dictionary traversal + membership

1) Traverse keys (what does it print?)

(One possible order.)

```
freq = {"cat": 3, "dog": 1}

for word in freq:
    print(word)
```

Output:

```
cat
dog
```

2) Traverse key-value pairs (what does it print?)

(One possible order.)

```
freq = {"cat": 3, "dog": 1}

for word, count in freq.items():
    print(word, count)
```

Output:

```
cat 3
dog 1
```

3) `in` checks keys only (what does it print?)

```
freq = {"cat": 3, "dog": 1}

print("cat" in freq)
print(3 in freq)
print("bird" in freq)
```

Output:

```
True
False
False
```

4) Build word frequency (fill in the blanks)

```
words = ["cat", "dog", "cat", "cat"]

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Expected output:

```
{'cat': 3, 'dog': 1}
```

5) Build word frequency (what is the final dictionary?)

```
words = ["A", "B", "A", "C", "B", "B"]

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Output:

```
{'A': 2, 'B': 3, 'C': 1}
```

Part B — Counting patterns (write code + fix bugs)

6) Write code: build `freq`

```
words = ["red", "blue", "red", "green", "blue", "red"]

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Expected output:

```
{'red': 3, 'blue': 2, 'green': 1}
```

7) Fill in the missing line (increase frequency)

```
words = ["hi", "hi", "bye", "hi"]
freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Expected output:

```
{'hi': 3, 'bye': 1}
```

8) Fix the bug (starting count)

Corrected code:

```
words = ["A", "A", "B"]
freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Output:

```
{'A': 2, 'B': 1}
```

9) Total number of words (write code)

```
freq = {"to": 2, "be": 2, "or": 1, "not": 1}

total = 0
for word, count in freq.items():
    total = total + count

print(total)
```

Output:

```
6
```

10) Safe lookup (write code)

```
freq = {"cat": 3, "dog": 1}
target = "bird"

if target in freq:
    print(freq[target])
else:
    print(0)
```

Expected output:

```
0
```

Part C — From text to words (`split`) + counting

11) `split()` (what does it print?)

```
text = "to be or not to be"
words = text.split()
print(words)
```

Output:

```
['to', 'be', 'or', 'not', 'to', 'be']
```

12) Split + count (what is the final dictionary?)

```
text = "to be or not to be"
words = text.split()

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Output:

```
{'to': 2, 'be': 2, 'or': 1, 'not': 1}
```

13) Case-insensitive counting (fill in the blanks)

```
words = ["To", "to", "TO", "be"]

freq = {}

for w in words:
    w = w.lower()
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Expected output:

```
{'to': 3, 'be': 1}
```

14) Remove punctuation (fill in the blanks)

```
text = "hi, hi. bye, hi."
text = text.replace(",", "")  # remove commas
text = text.replace(".", "")  # remove periods

words = text.split()
print(words)
```

Expected output:

```
['hi', 'hi', 'bye', 'hi']
```

Part D — Most frequent word

15) Find the most frequent word (fill in the blanks)

```
freq = {"to": 2, "be": 2, "or": 1, "not": 1}

best_word = ""
best_count = 0

for word, count in freq.items():
    if count > best_count:
        best_word = word
        best_count = count

print(best_word, best_count)
```

Expected output (one possible answer):

```
to 2
```

16) Write code: most frequent word from a list

(One correct solution.)

```
words = ["sun", "moon", "sun", "star", "moon", "sun"]

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

best_word = ""
best_count = 0

for word, count in freq.items():
    if count > best_count:
        best_word = word
        best_count = count

print(best_word, best_count)
```

Example output:

```
sun 3
```