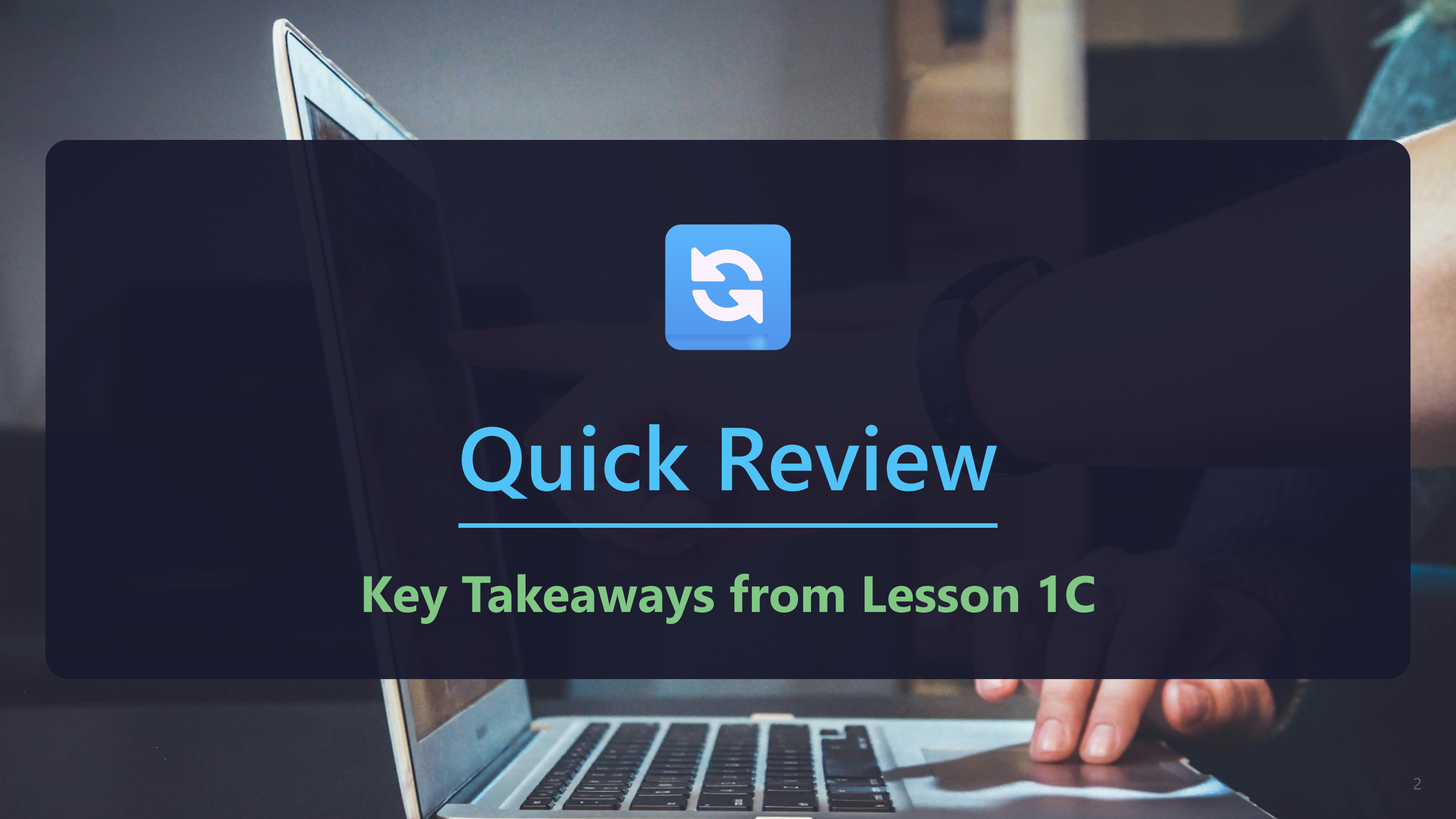
Practice Set

Lesson 1C: Finding & Organizing

Review + Practice Problems



Quick Review

Key Takeaways from Lesson 1C

Review Question 1

What's the difference between index() and count()?

Think about it before advancing...

🔄 Review Question 2

What's the difference between sort() and sorted()?

Which modifies the original? Which returns a new list?

🔄 Review Question 3

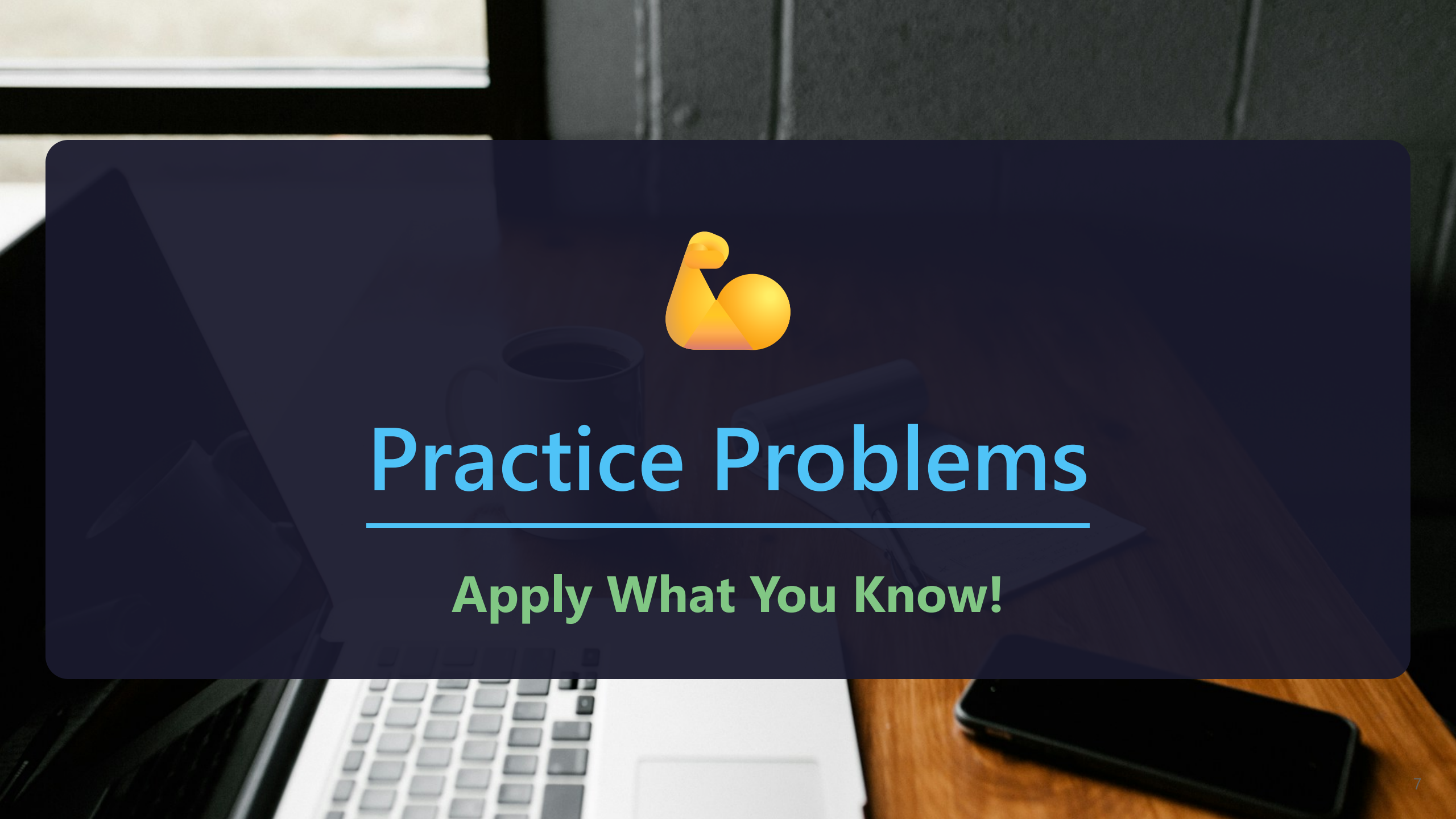
What does sort() return?

What's wrong with: result = my_list.sort()

🔄 Review Question 4

How do you reverse a list WITHOUT modifying the original?

What syntax creates a reversed copy?



Practice Problems

Apply What You Know!

🔮 Problem 1: Code Prediction

Trace this code and predict ALL outputs:

```
scores = [88, 92, 75, 95, 88, 82]

print(scores.count(88))
print(scores.count(100))
print(scores.index(88))
```

What prints for each line?

🔮 Problem 2: Code Prediction

What does this print?

```
nums = [5, 2, 8, 1, 9]
result = nums.sort()

print(result)
print(nums)
```

What prints for result? What prints for nums?

📋 Problem 3: Multiple Choice

Which code safely finds the position of "Python" in a list?

A) position = languages.index("Python")

B) if "Python" in languages: position = languages.index("Python")

C) position = languages.count("Python")

D) position = languages.find("Python")

🔮 Problem 4: Code Prediction

What does this code print?

```
original = [30, 10, 50, 20, 40]
sorted_copy = sorted(original)
original.reverse()

print(original)
print(sorted_copy)
```

Think about which operations create copies vs modify!

🐛 Problem 5: Spot the Bug

This code is supposed to show the highest score, but has a bug:

```
def get_highest(scores):
    sorted_scores = scores.sort(reverse=True)
    return sorted_scores[0]

player_scores = [85, 92, 78, 95, 88]
highest = get_highest(player_scores)
print(f"Highest: {highest}")
```

What's the bug? How would you fix it?

💻 Problem 6: Code Writing

Write a function that finds all positions of a value in a list:

```
def find_all_positions(items, target):
    """
    Return a list of ALL indexes where target appears.

    Example:
        scores = [88, 92, 88, 75, 88]
        find_all_positions(scores, 88)  # Returns [0, 2, 4]
        find_all_positions(scores, 99)  # Returns []
    """
```

Hint: index() only finds the FIRST one. You need a loop!

📝 Problem 7: Explain the Difference

What's the difference between these two approaches?

Approach A:

```
names = ["Zara", "Alice", "Beth"]
names.sort()
names.reverse()
print(names)
```

Approach B:

```
names = ["Zara", "Alice", "Beth"]
names.sort(reverse=True)
print(names)
```

Do they produce the same result? Is one better?

Complete Your Answers

Good luck!

Remember: Does this modify or return?