# Lists & Data Abstraction

Name: _____ Period: ____ Date: _____

### Objectives

- Represent a list using a variable and access elements by index (AAP-1.C).
- Use list operations: access, append, insert, remove, length (AAP-2.N).
- Explain how data abstraction with lists manages complexity (AAP-1.D).

**Directions:** On the AP Exam Reference Sheet, lists are **1-indexed** — the first element is `aList[1]`. App Lab (JavaScript) arrays are **0-indexed**. Watch this carefully!

## Section 1: Creating and Accessing Lists

Use this list for the questions below: `scores ← [90, 85, 70, 100]`

**1.** What is `scores[1]`? What is `scores[4]`? (Remember: pseudocode is 1-indexed.)

Answer: _____

**2.** What is `LENGTH(scores)`?

Answer: _____

**3.** In App Lab (0-indexed), the same list is `var scores = [90, 85, 70, 100];`. What is `scores[0]` in App Lab, and which pseudocode index refers to the *same* element?

_____

## Section 2: List Operations

Start each part from the original list `nums ← [3, 6, 9]`. Give the resulting list.

**4.** `APPEND(nums, 12)` Result: _____

**5.** `INSERT(nums, 1, 0)` (insert value 0 at index 1) Result: _____

**6.** `REMOVE(nums, 2)` (remove the element at index 2) Result: _____

**7.** After `APPEND(nums, 12)` only, what does `LENGTH(nums)` return?

Answer: _____

 CC BY-NC 4.0

## Section 3: Tracing List Code

**8.** Trace this pseudocode. What is displayed?

```
aList ← [4, 8, 15, 16]
DISPLAY(aList[2])
DISPLAY(aList[LENGTH(aList)])
```

Answer: _____

## Section 4: Data Abstraction

**9.** What does it mean to say a list is a **data abstraction**?

---

**10.** Give one reason using a list (instead of many separate variables like `score1`, `score2`, `score3`, ...) makes a program easier to develop and maintain.

---

**11.** A list "allows multiple related items to be treated as a single value." Explain what that means for a list of 100 test scores.

---


CC BY-NC 4.0