

# Answer Keys & Grading Notes

Objective-aligned exemplars, full-credit models, and the mistakes to watch for. **Do not distribute to students.**

**True-up step:** the objective sections below are built from the AP learning objectives. Paste your real Code.org prompts into chat and I'll fill exact question-by-question keys. The *AI-Proof Worked Answers* section matches the handwritten assignment exactly.

## A · Recursion tracing

Given <code>fact(n){ if(n&lt;=1) return 1; return n*fact(n-1);}</code>	Value
<code>fact(1)</code>	1 (base case)
<code>fact(3)</code>	$3*2*1 = 6$
<code>fact(4)</code>	24
missing base case	infinite recursion → <code>StackOverflowError</code>

**Common mistakes:** No base case (infinite recursion); base case that's never reached; multiplying in the wrong order; forgetting the return value bubbles up.

## B · Recursive search/sort concepts

**Full-credit exemplar:** Binary search recursively halves a SORTED array (compare middle, recurse on one half). Merge sort splits, sorts halves recursively, merges. Full credit names the base case (empty/size-1) and the halving step.

**Common mistakes:** Binary search on unsorted data; no base case named; thinking recursion is 'just a loop' without the call stack idea.

## AI-Proof Worked Answers — Recursion — Handwritten FRQ

---

Grade the handwritten sheet against these. Item numbers match the assignment.

Item	Correct answer
1) mystery(4)	$4 \times 3 \times 2 \times 1 = 24$ (mystery(1)=1 is the base case)
2) recursive sum	if( $i \geq a.length$ ) return 0; return $a[i] + \text{sum}(a, i+1)$ ; (base case: $i$ past the end)
3) binary search sorted	each step discards half based on order — only valid if the array is already sorted