

2.2 PROGRAMMING TECHNIQUES · 2.2.1(B)

Recursion

Base case, the call stack, tracing & recursion vs iteration. Spec 2.2.1(b).

01 The two parts

Base case	Solved directly; the stop condition .
General	Calls itself on a smaller input, → base case.
Must	Reach the base case, or it recurses forever.

02 The call stack

Down	Each call pushed (paused, data saved).
Up	From base case, calls popped & return.
Risk	Too deep → stack overflow .

03 Tracing — read bottom-up

factorial(4)	$4 \times \text{fact}(3)$
factorial(3)	$3 \times \text{fact}(2) = 6$
factorial(2)	$2 \times \text{fact}(1) = 2$
factorial(1)	base → 1 → answer 24

04 Recursion vs iteration

Recursion	
+	Elegant for recursive problems.
Recursion	
–	Stack memory; risk overflow; slower.
Iteration	
+	Fixed memory; usually faster.
Iteration	
–	Clumsier for recursive problems.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on recursion questions.

01

Every recursion needs a **reachable base case**.

02

The general case must move **towards** the base case (get smaller).

03

Trace **bottom-up**: resolve values as the calls **return**.

04

Each call is **pushed**; results are **popped** in reverse.

05

No base case → **stack overflow** (infinite recursion).

06

Compare fairly: recursion = elegant; iteration = less memory, faster.