Solutions

2.1

- The offset added to base R2 → address R2 + 8 ✓.

2.2

- lw / sw (load/store) only ✓.

2.3

- `addi R1, R1, -1` then `bne R1, R0, loop` ✓.

3.1

- Init: `addi R2, R0, 200`; `add R5, R0, R0`; `addi R4, R0, 5` ✓.
- Body: `lw R3, 0(R2)`; `add R5, R5, R3`; `addi R2, R2, 4`; `addi R4, R4, -1`; `bne R4, R0, loop` ✓.
- Counter R4, sum R5; 5 iterations; R2 ends at 200 + 20 ✓.

3.2

- while → `addi R1, R1, -1` + `bne` (register + immediate) ✓.
- `a[i]*2` → `lw R3, 0(R2)`, `add R3, R3, R3`, `sw R3, 0(R2)` (base+offset, register, base+offset) ✓.
- if-equal → `beq R1, R2, L` (register, register) ✓.
- Register-only arithmetic: one decode shape, no memory wait in the execute stage – pipelines stay full ✓.