Solutions

2.1

- opcode 1, rd 3, rs 1, rt 2, shift 0, funct 1 ✓.

2.2

- Largest 32 767; smallest −32 768 ✓.

2.3

- `lw` and `sw` ✓.

3.1

- `addi R5, R2, 60`: opcode 2, rt 5, rs 2, pad 0, immediate $60 = 111100_2$ ✓.
- Opcode 3, rt 6, rs 3, imm 12 decodes to `lw R6, 12(R3)` ✓.
- The opcode alone tells them apart (2 versus 3) ✓.
- R-type reuses one opcode for many operations – the funct field completes the selection ✓.

3.2

- $40\,000 > 32\,767$: outside the 16-bit signed immediate ✓.
- E.g. `addi R7, R0, 20000` (opcode 2, rt 7, rs 0, imm 20000) then `add R7, R7, R7` (opcode 1, rd 7, rs 7, rt 7, shift 0, funct 1) ✓.
- Largest positive `lw` offset: 32 764 – the greatest multiple of 4 inside the signed 16-bit range; the field's full unsigned span covers $2^{16} = 64$ KiB of addresses around the base ✓.
- R0 is hardwired: every read yields zero and writes to it vanish ✓.