Solutions

Each answer shows the steps, then the **result**.

2.1

- any one: RISC has **few simple fixed-length** instructions, CISC has **many complex variable-length** ones; in RISC only load/store access memory

2.2

- it **overlaps the stages** of consecutive instructions so more than one is processed at once (one finishes per cycle when full)

2.3

- **Single Instruction, Multiple Data**
- example: a **GPU** / graphics card / CPU vector unit

2.4

- **software** that behaves like a separate (virtual) computer, running its own operating system on a host machine

3.1

- any two: **fixed-length** instructions (each stage takes equal time); **simple** instructions (each ~1 cycle); register-to-register operations (only load/store touch memory), so stages are predictable

3.2

- while one instruction is being executed, the **next is being decoded and another fetched**
- so instructions overlap and throughput rises to ~one per cycle
- a **data** hazard (or control/branch hazard) can stall it

3.3

- **SISD, SIMD, MISD, MIMD**

3.4

- any one: **thousands of processors**; each with its **own (distributed) memory**; they communicate by **message passing**; it is MIMD

3.5

- any **four**: a small set of **simple** instructions; **fixed-length** instructions; only **load/store** instructions access memory; **many general-purpose registers**; a **hardwired** control unit; designed for **pipelining** (about one instruction per cycle)