Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

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

2.1

a normal day: 10 Feb → 3 Mar (in a non-leap year)

across a month or year end: 20 Dec → 10 Jan

across 29 February in a leap year: 15 Feb 2028 → 7 Mar 2028

2.2

it checks an internal detail, so a harmless refactor breaks it

check behaviour: `member.loans_count() == 1`, or that a 6th borrow is refused

2.3

`fine()`: unit

LoanService with a database: integration

browser script: end-to-end

2.4

whether the tests check the results (a test with no assertion still counts)

whether the boundary and error cases are tested

3.1

(a) loans held: 0–4 (allowed) and 5 or more (refused); fine: none or unpaid

(b) (4, no fine) → true; (5, no fine) → false; (0, unpaid fine) → false; (−1, …) → error: invalid input

(c) first change the boundary tests to 5 → true and 6 → false, and see them fail; then change the constant until all are green

3.2

(a) $1 + 2 + 8 + 20 = 31$ min; $30 \times 31 = \mathbf{930}$ min a day

(b) run unit tests first and stop on the first failure; run end-to-end tests in parallel, or only before a release and nightly

(c) otherwise broken code reaches `main`, and everyone builds on it

3.3

(a) $0.995^{200} \approx \mathbf{0.37}$

(b) runs fail for no reason most of the time, so people re-run or ignore red results, and real bugs slip through

(c) $0.999^{200} \approx \mathbf{0.82}$

exercises.lol exercises.com.cn

(d) waiting a fixed time instead of waiting for the page to be ready (or shared test data, or the real clock)

(e) quarantine flaky tests out of the merge check and fix them by cause; move most checks down to unit and integration tests, keeping a few end-to-end tests for the main journeys