# **Lean pre-coding sheet — 74/89**

**Component:** tests/vm\_tst\_core.rs (core engine tests)

## **1) Goal & success**

**Goal:** Lock the **baseline behaviors** of tabulation, allocation, gating denominators, and pipeline step order using the canonical **Part 0** fixtures and the **Doc 6A** core cases (VM-TST-001/002/003).

**Success:** Tests pass on Win/macOS/Linux with **identical outputs** given the same inputs (no net I/O). Approvals use the **approval-rate** denominator; PR/WTA allocations match the locked vectors.

## **2) Scope**

**In:** Pipeline **step order**, tabulation for **plurality/approval/score** (smoke), **Sainte-Laguë**, **WTA**, and **method convergence** case; gating denominator rule.

**Out:** IRV/Condorcet details (covered in vm\_tst\_ranked.rs), MMP specifics (in vm\_tst\_mmp.rs), cross-OS byte hash checks (in determinism.rs).

## **3) Inputs → outputs**

**Inputs:** Part 0 fixtures: division\_registry.json, ballots.json (or tally variant), parameter\_set.json; optional manifest.json for manifest-mode run.

**Outputs (asserted):** allocations by option, quorum/majority values & pass/fail, **approval-rate** support %, and final **label**.

## **4) Entities/Tables (minimal)**

## **5) Variables (used/assumed)**

Use **Doc 2** defaults unless a test overrides: VM-VAR-001 ballot type, 010 allocation, 012 PR threshold, 020/022/023 gates, 030/031 aggregation, tie policy 050, RNG 052 (not used in core cases).

## **6) Functions (test signatures only)**

rust

CopyEdit

#[test] fn vm\_tst\_001\_pr\_baseline\_sainte\_lague();

#[test] fn vm\_tst\_002\_wta\_winner\_take\_all\_m1();

#[test] fn vm\_tst\_003\_method\_convergence\_lr\_vs\_ha();

#[test] fn vm\_tst\_004\_gate\_denominator\_approval\_rate();

#[test] fn vm\_tst\_005\_pipeline\_order\_and\_stop\_rules();

(Names mirror Doc 6A and step-order rules.)

## **7) Test logic (bullet outline)**

**VM-TST-001 (PR baseline):** One national unit, m=10, approvals A=10,B=20,C=30,D=40; expect seats **1/2/3/4** (A/B/C/D). Label **Decisive**.

**VM-TST-002 (WTA):** ballot\_type=plurality, allocation\_method=winner\_take\_all, m=1; winner **D**, 100% power. Enforce **m=1** constraint.

**VM-TST-003 (convergence):** m=7, shares 34/33/33 for A/B/C. Run three methods: LR, Sainte-Laguë, D’Hondt → same seat vector (locked case).

**VM-TST-004 (approval gate denominator):** With ballot\_type=approval, assert **support %** computed as approvals\_for\_change / valid\_ballots (not approvals share). Cross-check against panel value.

**VM-TST-005 (pipeline order/stop):** Force **VALIDATE** failure → ensure pipeline skips 3–8, still packages **Invalid** Result/RunRecord with reasons. Force **gates fail** path → **Invalid**, **skip Frontier**.

## **8) State flow (very short)**

Tests drive CLI/library to execute **LOAD→…→BUILD\_RUN\_RECORD** and assert **stop/continue** semantics per Doc 5.

## **9) Determinism & numeric rules**

Integers/rational comparisons; **round-half-even** only at defined points; **one-decimal** applies in reports, not in these assertions (use exact internal numbers). Stable orders: Units by ID; Options by order\_index.

## **10) Edge cases & failure policy**

If a seat sum ≠ Unit.magnitude (PR) or WTA ≠ 100%: fail with clear diff.

If approval panel uses the wrong denominator or double-rounds: fail and print the raw numerators/denominators.

## **11) Test checklist (must pass)**

All three **Doc 6A** allocation cases match expected vectors and labels.

Gate panel shows approval **support %** per fixed rule.

Pipeline stop/continue behavior matches Doc 5; **Invalid** path still packages artifacts.