# **Lean pre-coding sheet — 73/89**

**Component:** fixtures/annex\_b/part\_0/expected\_result.json (Part 0 **expected outputs**)

## **1) Goal & success**

**Goal:** Provide the **expected outcome snapshot** for the Part 0 run so engines can compare their computed Result against fixed fields (gates, allocations, label); also carry the placeholder for the **canonical hash** to be filled after the first certified run.

**Success:** The engine’s Result matches the expected gates/allocations/label exactly; once certified, expected\_canonical\_hash equals the canonical SHA-256 of the Result artifact bytes (UTF-8, sorted keys, LF, UTC).

## **2) Scope**

**In scope:** Minimal expected fields needed to assert correctness for Part 0 (e.g., gate pass/fail, national support %, seats by option, final label; optional executive/IRV summaries).

**Out of scope:** Full report rendering; presentation rounding (handled in Doc 7 with **one-decimal** rule).

## **3) Inputs → outputs**

**Inputs:** The computed **Result** from the engine run over Part 0 fixtures (Registry, BallotTally, ParameterSet, optional Manifest).

**Outputs:** JSON file with expected{ ... } fields and expected\_canonical\_hash (null until certified).

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

## **5) Variables (only ones used here)**

**None directly.** Values shown are **outcomes**; VM-VARs live in the ParameterSet and influence the produced Result that this fixture checks. (Defaults for small canonical tests are noted in Part 0.)

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

N/A (fixture only).

## **7) Algorithm outline (how it’s consumed)**

Run pipeline to produce Result.

Compare Result.gates (with raw values) and **label** against expected.

If the test includes seats/power, compare total\_seats\_by\_party (or equivalent per test).

When the test pack is **certified**, compute canonical SHA-256 of Result (sorted keys, LF, UTC) and write it into expected\_canonical\_hash.

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

Used **after** BUILD\_RESULT in acceptance tests; does **not** affect computation, only validation.

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

Comparison should assume **stable ordering** (Units by ID; Options by order\_index) and canonical JSON; percentages in reports are **one decimal**, but expected values here should be based on **exact internal math** (no double rounding).

For approval ballots, majority/support expectations rely on the **approval-rate** denominator (approvals\_for\_change / valid\_ballots).

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

If a gate value or label diverges: flag test **Fail** with the mismatched field path(s).

If canonical hash comparison is enabled and differs: suspect non-canonical serialization or nondeterminism; re-check **sorted keys / LF / UTC** and ordering.

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

Expected **gates**, **seats/power**, and **label** match engine output for Part 0 baselines (e.g., PR 1–2–3–4; WTA winner; convergence case).

After certification, expected\_canonical\_hash equals the engine’s canonical Result hash on all OS/arch (cross-OS determinism).