

↔ A recipe for solving toy problems ↔

* 📖 Really read the problem.

- Read it slowly.
- **Read it** again.
- Think to yourself, "what **assumptions** are hidden in this problem?"
 - There are **ALWAYS** assumptions.
 - Ask at least one question about those assumptions.
 - If you are working alone (recording a video), just ask yourself, out loud.
- Identify **input and outputs**.

* ✎ Outline your approach

- Decide on your **core data structure** or data type.
- What are the **start and end states** of that data?
- How does that data "mutate" from start state to end state?
 - Outline your state-changing logic in English **pseudo-code**.
 - Re-read it, edit it, make it clear and precise.
- **Outline your structure** with one or more empty functions (aka **stub functions**).
 - Give those stubs clear names.

* 💻 Implement carefully, step-by-step.

- **Fill in** the main function body.
- **Stub out** helper functions if need be.
- Each function should ideally have a **single purpose**.
 - It should do only one, small thing.
 - That one thing is expressed clearly by the function's name.
- **Test** as you go.
 - Each component (or "unit") -- for now, this means each function -- has a clear contract.
 - Write an **assertion**
 - Assert what behavior should be true about that component.