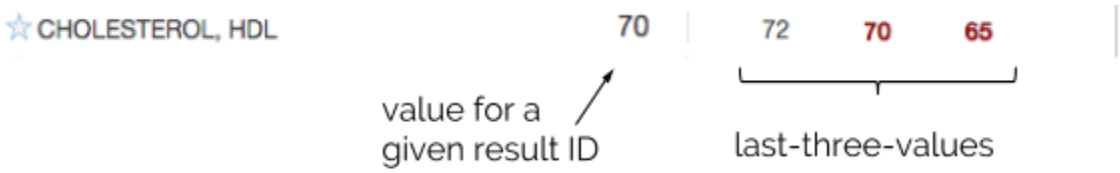
Elated Doctor, MD has been seeing some of her patients for a very long time. They've been getting regular checkups for decades, and every checkup includes new lab results.

When Dr. Doctor looks at new labs, she always wants to know: is this patient trending better or worse? Otherwise, she has no way to know whether that new medication or diet regime is helping, hurting, or having no effect for this patient.

Good thing that electronic data can be flexibly presented! The data might have come in as a series of separate labs, but we can do something smarter. Every time Dr. Doctor looks at, say, HDL (Cholesterol) results, we can show her that patient's previous three HDL results too, so she can answer that question with zero extra effort. Something like:



Your task is to build a simple system to support this need. **For any result ID, we need to efficiently find the 3 previous lab results (called last-three-values) for the same lab name and same patient.** There are a lot of results in the system and each patient can accumulate thousands of results!

Lab reports will also come in on a regular basis, and they may not always be chronologically new reports: a new set of formerly paper labs might make it into the system from ages ago, or a new lab center might add an interface to our system and send us their historical records. So you'll **also need to be able to update your system to account for the new data**.

You have two data files: `initial_lab_results.json` contains thousands of lab results for many patients. Each lab result has a lab name, a result ID (which uniquely identifies a particular result), a patient ID, a value, and a date. `new_lab_results.json` contains JSON in the same format, to help you simulate a few lab results that will come in "later." We have also provided a few files with which to back tests of your program: `sample_expectations_rd1.json` and `sample_expectations_rd2.json`. The rd1 file includes some example results you should see after ingesting `initial_lab_results.json`. The rd2 file updates those expectations to account for ingesting `new_lab_results.json`.

Feel free to write the system in **whatever language you are most comfortable in** (but keep in mind that a human will read your solution, so please keep it clean/accessible). However, **external services are out of scope (e.g., using a database or search server or redis wouldn't be allowed).** You can assume that you will have enough memory to store your data structure completely in memory.

Please include a test runner in your program which verifies that it correctly produces the expected results from the two sample_expectations files. **Also include instructions** on how to run your program (both the test runner and a version for arbitrary input besides `initial_lab_results.json`).

We expect this assignment to take ~3 hours. Please take this into consideration when you decide on your approach.