1 Notes

method notes
prompted chatgpt4 and copilot (Q4 2025 on think deeper settings) on LList w/cursor, RSA encode and decode methods

prompt1: Given this project code, find and document each method's weakest possible precondition as well as post-conditions following the standard Python docstring format. Next, convert the post-conditions to executable format.

prompt2: Given this project code, find and document each method's weakest possible precondition as well as postconditions following the standard Python docstring format. Next, convert the postconditions to executable format. For example, the weakest possible precondition in code is one that, when executed, will not throw any in-built exceptions. For example, NaN is not an exception in a Java, so that behavior is acceptable. When creating executable post conditions, use asserts or ifs to make sure that control is directed away from the standard path if need be.

used each LLM to grade the other's response in addition to manual review. Prompt: Given this changed code, is each pre and post condition correct?

First, AI generated code was run through a unit test suite to ensure correctness

results:
prompt1 chatgpt LList -
Correctly IDed: initpre, len post, iter pre and post (executables wrong), add post, insertathead post, insertattail post, removeitemathead pre and post (executable wrong), removeitematcursor pre and post, removeitemattail pre and post (executable wrong), itemathead pre and post, itemattail pre and post, cursorToStart post, cursorForward post (executable wrong)

Incorrectly IDed: initpost (got some, not all), len pre (claimed needed init first), add pre (doesn't require other item to be a LList, just iteratable), insertathead pre, insertaftercursor pre and post (both added extra tests, executable correct), insertattail pre (not weakest), itematcursor pre and post (not weakest, test correct), cursortostart pre (not weakest), cursorForward pre (not weakest)

prompt1 copilot LList -
didn't correct add executable postconds, seems like it added unit tests.
Correctly IDed: initpre, add post, insertathead pre and post (note: added some less than useful tests), insertaftercursor post, insertattail pre and post (post added some fluff tests), removeitemathead pre and post, removeitematcursor pre and post (fluff tests), removeitemattail pre and post, itemathead pre and post, itematcursor pre and post, itemattail pre and post, cursorToStart pre and post,

Incorrectly IDed: initpost (all given not weakest), len pre (claimed none) and post (added list not modified), iter pre (claimed none) and post (added extra test for not modified, correctly got yield node items), add pre (to many), insertaftercursor pre (far too many tests), takeaways: copilot added many tests that were not needed, kind of like asserting 1 == 1 in every method. The remove at 'x' tests chatgpt wrote were making sure that the old did not equal the new head but it could (if values matched). Copilot wrote odd tests when asked for executable post conditions, it created methods that were supposed to be called in the appropriate calling method, similar to unit tests. When given the Copilot generated code to ChatGPT, ChatGPT calls errors in insertAfterCursor postconditions, removeItemAtCursor, removeItemAtTail, post conditions (all pre conditions are correct). When given the ChatGPT code to Copilot, Copilot calls errors in add post, insertAfterCursor, removeitematcursor post.