CISC 361, Spring 2020

Name	James Villemarette, 702473365, jvill@udel.edu
Date	February 20, 2020

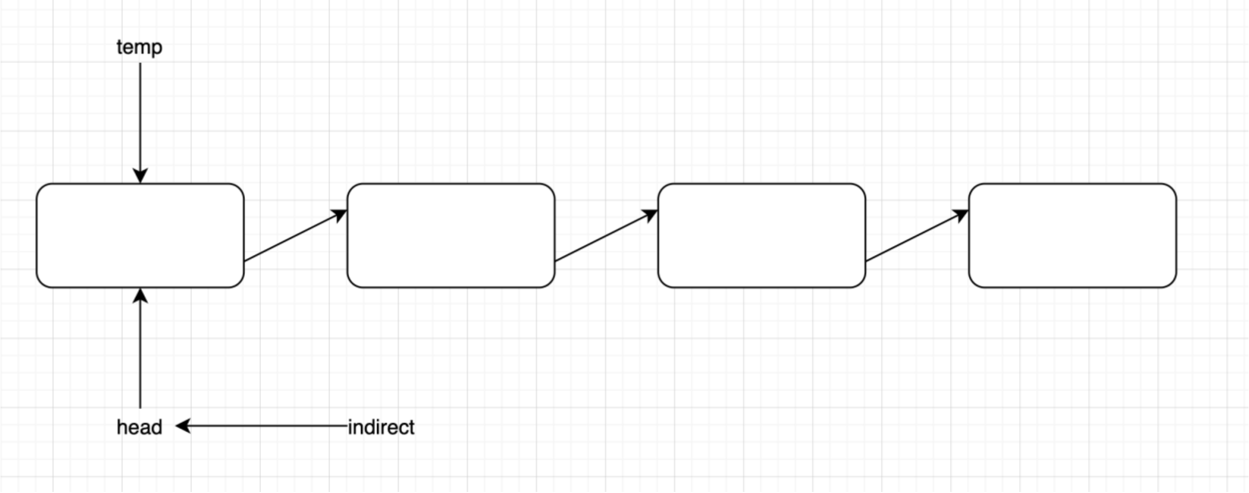
Programming Assignment #1 Report

I revamped the delete() function in tastylist.c file. First, I reformatted the entire file with proper tab-indentations and line breaks to remove the whitespace errors on compilation. Then, I kept the main(), count(), and display() functions the same, along with the different add functions. I copied the delete() function from the "PA 1 Linus.pdf" file and reformatted it.

I took Professor Shen's advice, and checked to see if the next node is null, as the user could try to delete a value that does not exist. I added return values to the delete() function so that it would keep the same behavior as was intended in the original tastylist.c file. I also freed the memory when the program exits to prevent a memory leak.

After testing with valgrind for memory leaks, I found no memory that was definitely or indirectly lost. Just starting the program and immediately exiting results in 4,200 bytes possibly lost, and 17,439 bytes still reachable. **Note:** Running my version of valgrind (which I had to hack in since I couldn't install it with *brew install valgrind* on macOS Catalina) on a four-line HelloWorld program results in the same number of bytes "possibly lost" and "still reachable". But overall, the program is shorter, more efficient, and more readable.

Here is the diagram Shen requested of the delete() function and how indirect works:


temp
head
indirect