

## Data Structures (CS 1520) Homework #4      Due: Oct. 15 (Saturday) at 11:59 PM

**Objective:** To reuse your cursor-based list using doubly-linked nodes from HW #3 to implement a text-editor program. NOTE: If you did not complete HW #3 fully, then you can complete HW #3 as part of this assignment and get credit for both homeworks.

You are to write a simple text-editor program that utilizes your `CursorBasedList` class. Each node in the list will hold a single string representing a single line of the text file. When done editing, the data contained in the nodes is written to a text file. NOTE: You should NOT need to modify your `CursorBasedList` class only create a list object and use its methods. (**STRONG HINT:** You might want to start with the `cursorBasedListTester.py` program from HW#3 as a rough starting point for your text-editor program.)

Your text-editor program should start by allowing the user to either:

- enter a filename of a text (.txt) file to edit. Your program should then insert each line from the text file into a cursor-based list. (i.e., one line of the file per node in the list -- see diagram on next page), or
- create a new text (.txt) file to edit. Your program should create an empty cursor-based list.

Then repeatedly present a menu with text-editor options that allows the user to:

- navigate and display the first line, i.e., the first line should be the current line
- navigate and display the last line, i.e., the last line should be the current line
- navigate and display the next line, i.e., the next line should become the current line. If there is no next line, tell the user and don't change the current line
- navigate and display the previous line
- insert a new line before the current line
- insert a new line after the current line
- delete the current line and have the line following it become the current line. If there is no following line, the current line should be the last line.
- replace the current line with a new line
- save the current list back to a text file

**Warning:** When you load a text file into your list nodes, you can leave the '\n' characters on the end of each line of text. However, remember to add a '\n' character to end of inserted lines or replacement lines.

**Implement AND fully test your text-editor program.** Part of your grade will be determined by how **robust** your text-editor runs (i.e., does not crash) and how **user-friendly/intuitive** your program is to use. You are required to submit a brief User's manual on how to use your text-editor.

**For extra credit,** your program may also do one or more of the following to provide additional text-editor functionality:

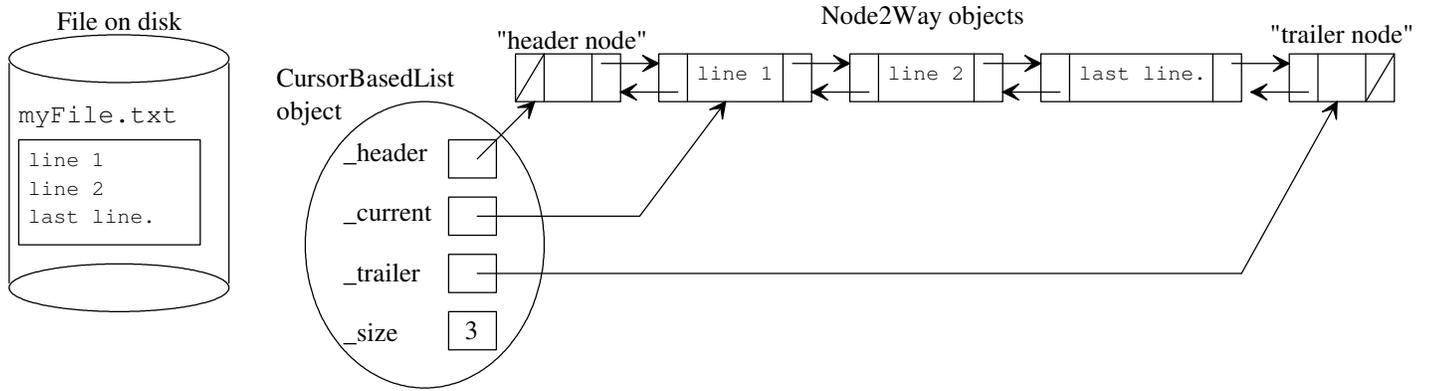
- Find word and Find next occurrence
- Replace a specified word/string on the current line by another word/string
- Copy and Paste a line, etc.

Be sure to include these additional features in your User's manual.

**Submit all necessary files (userManual.doc, cursor\_based\_list.py, node.py, node2way.py, etc.) with your text-editor program file(s) as a single zipped file (called hw4.zip) electronically at**

[https://www.cs.uni.edu/~schafer/submit/which\\_course.cgi](https://www.cs.uni.edu/~schafer/submit/which_course.cgi)

To help visualize loading the text file into the list:



Suggestions on the text-editor interface:

- You might consider always displaying the current line after each edit, and possibly surrounding lines with the current line clearly indicated.
- Use meaningful user-commands, i.e., 'N'/'n' to advance the cursor to the Next line, 'F'/'f' to navigate to the First line, etc. I don't want to remember or look up a number for a command.
- Accept both upper and lower case menu options.
- Print the menu after every edit or allow the user to display the menu again if needed.