

CS218: Data Structures (Fall 2024)

Assignment # 02

Console Based Notepad Application

Create a notepad that allows the user to write text (character by character) on the console. The user should be able to control and track the movement of the cursor, navigate to any part of the console, and perform insertion or deletion of text. Internally, the notepad will use a two-dimensional linked list to store characters, allowing for flexible text manipulation. Additionally, the assignment incorporates the concept of stacks to manage undo and redo actions, enabling efficient tracking and reversal of text modifications.

Functionality of Console based Notepad Application:

Insertion:

- Input text character by character without using `getline()` or similar function (String input is not allowed). Capture input character by character and special keys (arrow keys, Enter, backspace).
- Inserting text will not overwrite existing characters but shift them forward, similar to a typical text editor.
- Inserting text can be done at the start, middle, and end of each line, as well as inserting lines between lines.
- If a word being typed reaches the end of the current line and cannot fully fit within the remaining space, the entire word should automatically move to the next line, ensuring that no part of the word is split between lines.
- Pressing Enter moves the cursor to the next line. If Enter is pressed mid-line, the remaining text after the cursor is moved to the new line.
- Enter pressed multiple times will produce empty lines.
- Ensure that only alphabetic characters (A-Z, a-z) can be entered. Non-alphabetic characters will be ignored.

Deletion:

- Characters are deleted one by one using the backspace key, with the remaining text shifting left to fill the gap.
- Deletion can be performed at the start, middle, and end of each line, as well as deleting lines between lines.
- A penalty will be applied if the text contains trailing spaces at the end of lines. This will result in a 20% deduction of marks.

Undo/Redo:

- The undo/redo functionality tracks the last actions performed by the user.
- Each action corresponds to a single word.
- The user can undo or redo up to the last 5 words of their actions.
- If more than 5 words have been inserted, deleted, or modified, only the most recent 5 words will be undoable or redoable.
- Actions beyond the limit of 5 words will not be affected by undo/redo operations.

Window Size:

- Implement a visible window size to limit how much of the text is visible at one time. The cursor should move freely within the text, but not beyond the text boundaries.
- **Main Text Area (60%):** This part of the window will take up 60% of the total screen space for writing and editing text. It provides the main area for text manipulation.
- **Suggestions Area (20%):** Positioned at the bottom of the window, this area will occupy 20% of the screen to show potential word suggestions based on user input. Though suggestions will be implemented in Assignment 3, this frame space must be included in this assignment to facilitate future development.
- **Search Area (20%):** Located on the right-hand side of the screen, this area will occupy 20% of the window for searching within the text. Like the suggestions area, the search functionality will be implemented in Assignment 3, but this frame space is included now to enable seamless integration later.

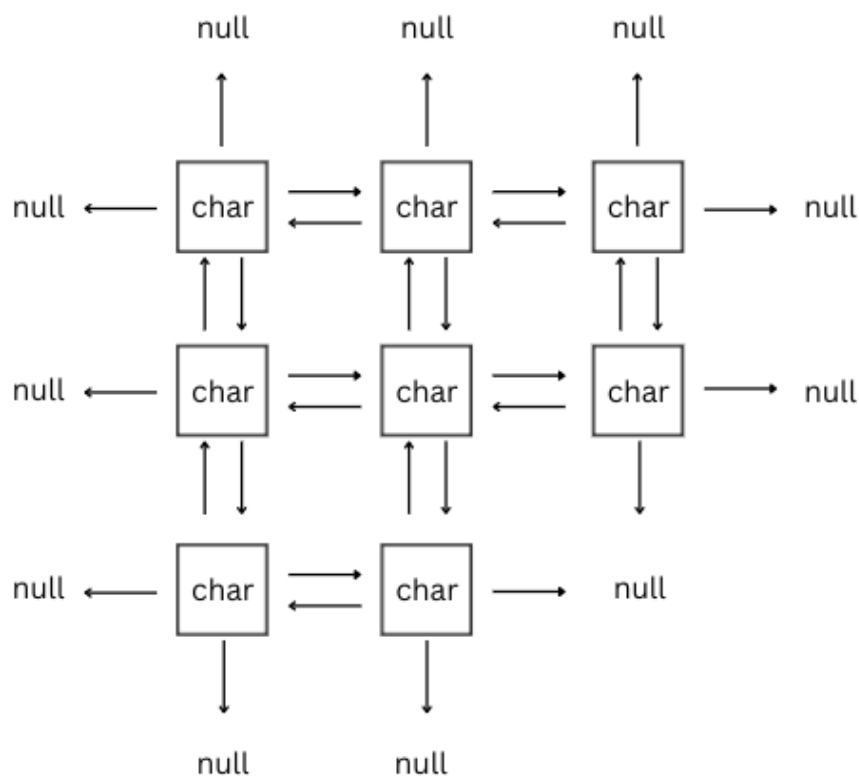
Other Functionalities:

- If the user chooses to save or load a file that does not exist, the program will create it automatically.
- Upon starting the program, display a menu with options to create a new file, load an existing file, save the current file, or exit the program (with an option to save before quitting).
- Upon closing the program, ensure that all dynamically allocated memory (nodes of the 2D linked list) is freed to prevent memory leaks.

2D LinkList:

Each node contains:

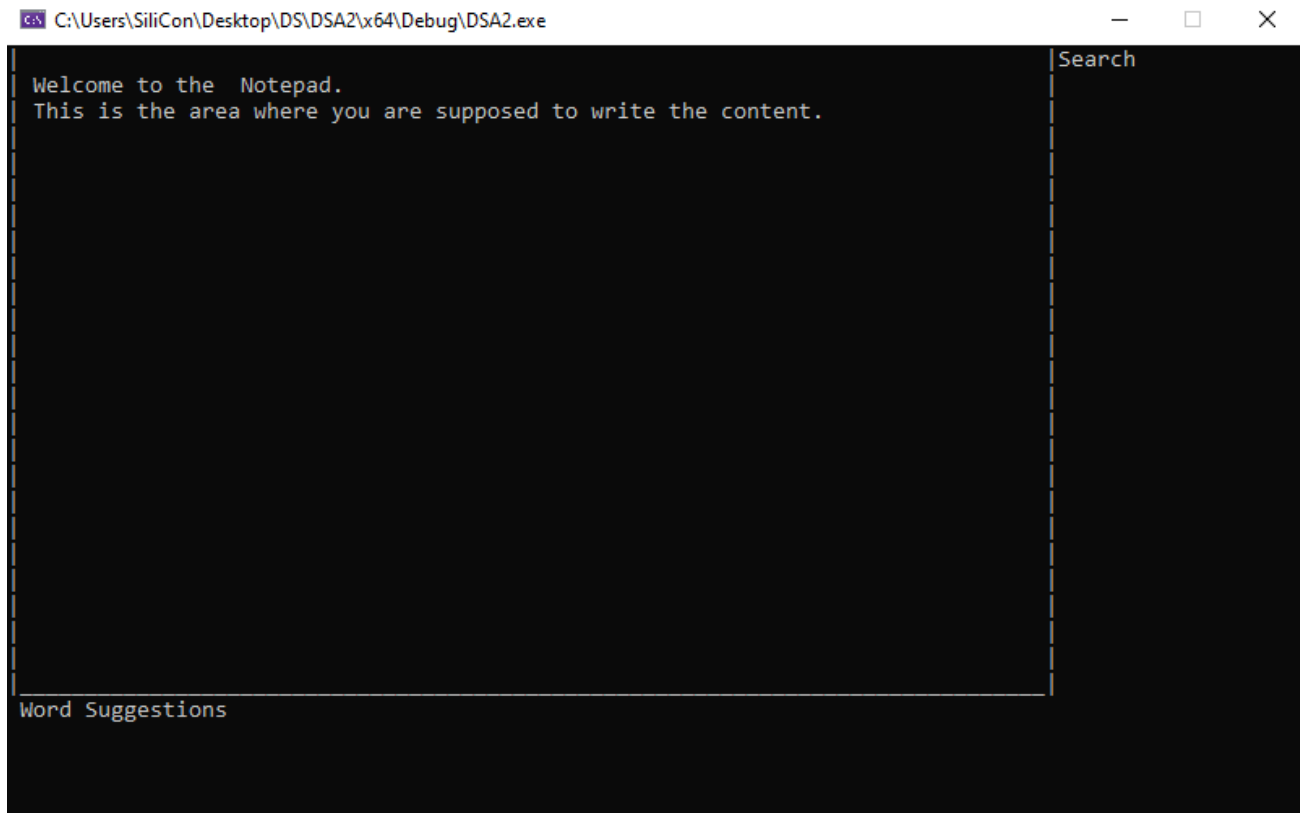
- A character.
- Four pointers (left, right, up, down) to neighboring characters/nodes. The linked list grows dynamically as text is added, and text can be inserted or deleted at any position.



NOTE:

- You are not allowed to declare any array of any kind for any reason. Whenever such functionality is desired, use linked lists.
- You can use any key for functionalities such as saving, undoing, redoing, and moving to the next line. Ensure that the key mappings are intuitive and clearly documented.
- Please note that no marks will be awarded for the undo and redo functionalities if the insertion and deletion features are not functioning correctly. It is essential to first ensure that text insertion and deletion operate as expected before implementing and testing undo/redo features.
- Please note that no test cases will be provided for this assignment. You are responsible for thoroughly testing your program by creating a comprehensive and well-structured `main()` function. Ensure that your `main()` function handles all edge cases, validates input, and demonstrates the full functionality of your notepad application, including insertion, deletion, undo/redo, and file operations.

For help Cursor movement sample code is also attached.



Submission Criteria & Guidelines:

1. **Submission:** You are required to use Visual Studio 2022. Combine all your work in one **.cpp** file (assignment2.cpp (including main function)) and zip the file. **DO NOT SUBMIT COMPLETE PROJECT**. Name the .zip file as ROLL_NUM_SECTION_02.zip (e.g. **23i-0001_A_02.zip**). Submit a zip file in the google classroom within the given deadline. Failure to submit according to the above format would result in **ZERO** marks. You MUST check your zip file [1) Is it virus free 2) Is required files present or not.]. The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.
 2. You must write **generic codes**.
 3. If the file is not generated in the required path no marks will be awarded.
 4. Use Classes. If the assignment is **classes less** no marks will be awarded.
 5. Files should be read once.
 6. Do not use STLs, String, arrays or any built-in functions or libraries.
 7. If the required output is generated, you will be awarded full marks. Failing to generate the correct output will result in zero marks (black box checking only). Whenever code ceases execution (for example segmentation fault or any other runtime error), zero marks will be given in that portion.
 8. If we are unable to compile because of **syntax error** no marks will be awarded.
 9. Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in this assignment. **Copying from the internet is the easiest way to get caught!**
 10. **Deadline:** Deadline to submit assignment is **05th October 2024 11:59 PM**. Correct and timely submission of assignments is the responsibility of every student; hence no relaxation will be given to anyone.
-