MathWorks

PROBLEM STATEMENT

This exercise involves designing the underlying buffer manager class found within many text editors. Write a text buffer class that performs most basic functionalities found in a class such as an std::string, but in addition maintains state, so that the user can undo and redo. The text buffer class is initialized with some text. Then modification operations (insert/erase/replace) are performed on it. At any point, we can call undo/redo to revert the underlying buffer to its previous/next state.

DETAILS

ID	Statement
1.	The class shall store the text typed by the user
2.	The class shall expose the following interface: • Insert a substring at a position • Append a substring • Erase *n* characters at a given position • Erase trailing *n* characters • Replace all occurrences of a substring by another • Undo • Redo • Load from TXT file • Save to TXT file
3.	Other considerations: • Production code shall be object-oriented, written in C++ • Use of STL containers is allowed and recommended • Take into account the memory consumption of the buffer in the presence of a sequence of operations. Note that each operation could potentially only slightly modify the contents of the buffer.

No graphical interface is required.

DELIVERABLES

- Source code, in one or more files
- Unit tests
- Makefile or script to build and run
- Any setup or usage instructions