THE EDITOR

OVERVIEW

This C program simulates a basic text editor with undo functionality using a singly linked list. It reads a sequence of commands and arguments from an input text file and performs various text editing operations, including adding, deleting, moving, updating, and undoing actions.

Please note that there is an issue in the program logic, and it is mentioned that it does not provide the desired output due to errors.

- ## STRUCTURES:

1. NODE STRUCTURE

- Represents a node in the linked list.
- Contains two fields:
  - **data**: An integer representing the data stored in the node.
  - **next**: A pointer to the next node in the linked list.

2. EDITOR STRUCTURE

- Represents the text editor.
- Contains the following fields:
  - **head**: A pointer to the head of the linked list, representing the text content.
  - **cursor**: A pointer to the current cursor position in the linked list.
  - **undoCount**: An integer representing the number of available undo states.
  - **undoStack[5]**: An array of pointers to **EditorState** structures, which stores previous editor states for undo functionality.

3. EDITORSTATE STRUCTURE

- Represents an editor state for undo functionality.
- Contains two fields:
  - **head**: A pointer to the head of the linked list representing the text content.
  - **cursor**: A pointer to the cursor position within the linked list.

- ## FUNCTIONS:

1. CREATEEDITOR

- Creates and initializes an **Editor** structure.
- Allocates memory for the editor and sets default values.
- Returns a pointer to the created editor.

2. PUSHEDITORSTATE

- Pushes the current editor state onto the undo stack.
- If the undo stack is full (contains 5 states), it removes the oldest state to make room for the new state.
- Creates a new **EditorState**, copies the current editor's state into it, and adds it to the undo stack.

3. POPEDITORSTATE

- Pops the previous editor state from the undo stack and restores the editor to that state.
- Frees the memory of the popped state.

4. TEXT EDITING FUNCTIONS

- Several functions are provided for various text editing operations such as adding, deleting, moving the cursor, updating, and more. These functions include **addNumber**, **delete**, **addIndexNumber**, **deleteIndex**, **moveForward**, **moveBackward**, **addImmediateNumber**, **deleteImmediate**, **updateIndexNumber**, **updateImmediateNumber**, and **shiftIndex**.

5. PRINT

- Prints the current content of the linked list (text) in square brackets.
- Indicates the cursor position within the printed text.

6. UNDO

- Restores the previous editor state from the undo stack, effectively undoing the last action.

7. MAIN FUNCTION

- The **main** function is the entry point of the program.
- It creates an editor using **createEditor**.
- Reads input commands and arguments from an input file ("Test01.txt") to perform text editing operations.
- Utilizes the various text editing functions and undo functionality.
- Outputs the current state of the text using the **print** function.
- Reads and executes commands until the end of the input file.
- Closes the input file and returns 0 to indicate successful execution.

ERROR NOTE:

The program contains an error in its logic, and it does not produce the desired output. Further analysis and debugging are required to identify and correct the errors in the program.