Generally, **led** sets the current line to the last line affected by a command; however, the exact effect on the current line depends on the command and is specified in Table 3 below.

Table 3. **led** Commands

| Command | Description of **led**'s Actions |
|---|---|
| $x$ **a** ↵ | In input mode, appends text into the buffer after line $x$. The current address is set to the last line entered. |
| $x$ **i** ↵ | In input mode, inserts text into the buffer before line $x$. The current address is set to the last line entered. |
| $x$ **v** ↵ | Pastes text from the clipboard into the buffer below line $x$. The current address is set to the last line entered. |
| $x$ **u** ↵ | Pastes text from the clipboard into the buffer above line $x$. The current address is set to the last line entered. |
| $x, y$ **d** ↵ | Deletes the line range $x$ through $y$ from the buffer. If there is a line after the deleted line range, then the current address is set to that line. Otherwise the current address is set to the line before the deleted line range. |
| $x, y$ **x** ↵ | Cuts the line range $x$ through $y$ from the buffer into the clipboard. If there is a line after the cut line range, then the current address is set to that line. Otherwise the current address is set to the line before the cut line range. |
| $x, y$ **r** ↵ | Replaces the line range $x$ through $y$ with input text. Equivalent to the command line $x, y$**d** followed by the command line $x$**i** |
| $x, y$ **j** ↵ | Joins the line range $x$ through $y$ together on one line at address $x$, such that each line in turn is appended to line $x$, separated by a single space. Line x becomes the current line, |
| $x, y$ **p** ↵ | Prints the line range $x$ through $y$ without affecting the current line address. |
| $x, y$ **c** ↵ | Prompts for and reads the text to be changed, and then prompts for and reads the replacement text. Searches each line in the line range for an occurrence of the specified string and changes all matched strings to the replacement text. |
| $x$ **−** ↵ | Moves the current line up by $x$ lines provided that there are $x$ lines above the current line; otherwise, prints the message **top of file reached** and sets the current line to first line in the buffer. If omitted, $x = 1$. |
| $x$ **+** ↵ | Moves the current line down by $x$ lines provided that there are $x$ lines below the current line; otherwise, prints the message **end of file reached** and sets the current line to last line in the buffer. If omitted, $x = 1$. |
| $x$ **g** ↵ | Goes to the specified line $x$, meaning that it sets the current line to $x$ and prints it. Prints the message **invalid range** if $x$ is invalid. If omitted, $x =$ the current line address. |
| **w** ↵ | Writes out entire **buffer** to its associated file. If the **buffer** is not associated with a user named file, it prompts for and reads the name of the associated file. |
| **q** ↵ | Quits **led**. Before quitting, however, it gives the user a last chance to save the buffer. If the user takes the chance, it simulates the **w** command. |
| ***** ↵ | same as $1, \$$ **p** ↵ |
| ↵ | same as 1**+** ↵ |