

Commands

This section provides a list of the commands the program will support. Commands can take the following forms, with each part separated by one or more spaces. A command can be preceded by white space.

- <> An empty line; no non-blank characters.
- <n> Line number (n), a positive integer
- <c> Command (c), one of “a”, “d”, “i”, “l”, “p”, “q”, “r”, “s”, “w”, “/”, “?” • <c> <i> Command (c) and an integer (i) offset (can be negative) • <c> <text1> Command (c) and a text parameter (text1)
- <c> <text1> <text2> Command and two text parameters (text1, text2)

Below is a table summarizing the commands. The last line accessed by a command is called the “current line.” Some commands result in moving around in the file, causing the current line to change. How a command changes the current line is documented in the table.

c	Action	Usage	Semantics	Current Line
a	Add text	a	Add lines of text after current line. End input with an empty line (no characters on it). Can add lines to an empty file.	Last line added
d	Delete	d	Delete current line. An error if the file is empty.	Next line; last line if at file end
		d n	Delete current line and next n. An error if the file is empty. There is no error message if n goes beyond file end.	Next line; last line if at file end
		d -n	Delete previous n lines and current line. An error if the file is empty. There is no error message if n goes before file start.	Next line; first line if at file start
i	Insert	i	Insert lines of text before current line. End input with empty line (no characters on it). Can insert lines to an empty file.	Last line inserted
l	Load	l fname	Read file named “fname”. An error if the open fails. No check if the user failed to write out previous file.	Last line of file
p	Print	p	Print current line. No error if file is empty.	No change
		p n	Print current line and n more. There is no error message if n goes beyond file	Last line printed

			end. No error if file is empty.	
		p -n	Print n previous lines and current line. There is no error message if n goes before file start. No error if file is empty.	No change
q	Quit	q	Exit the program. No check if the user failed to write out previous file.	
r	Replace	r text1 text2	If “text1” is present in current line, replace it with “text2” (only once). No error if file is empty or if replace fails.	No change
		r text1	If “text1” is present in current line, replace it with “” (empty). No error if file is empty or if replace fails.	No change
s	Sort	s	Sort the lines in the file into ascending order (don’t add descending order). No error if file is empty.	Last line of file
w	Write	w fname	Write to file “fname”. An error if the open fails.	Last line of file
		w	Write to file (use name from load command). An error if the open fails.	Last line of file
/	Search forward	/ text1	Search from next line for “text1”, wrapping around to file head if needed. Only find first occurrence of “text1”. When cycle	If match, line of the match; If not, no change

			around to current line, stop without matching. No error if search fails.	
?	Search backward	? text1	Search from previous line for “text1”, wrapping around to file tail if needed. Only find first occurrence of “text1”. When cycle around to current line, stop without matching. No error if search fails.	If match, line of the match; If not, no change
<n>	Line number	<n>	Go to line <n>. It is an error to go beyond the end of file or before the first line in the file.	Line <n>; No change if error

<>	Print next	<>	Print next line. No printing if next line is end of file.	Line printed else no change
----	------------	----	---	-----------------------------------