OBJECTIVE: To demonstrate the working of various commands of Vi Editor in Linux.

THEORY:

There are many ways to edit files in Unix and for me one of the best ways is using screen-oriented text editor **vi**. This editor enables you to edit lines in context with other lines in the file. The vi is generally considered the de facto standard in Unix editors because −

- It's usually available on all the flavors of Unix system.
- Its implementations are very similar across the board.
- It requires very few resources.
- It is more user friendly than any other editors like ed or ex.

You can use **vi** editor to edit an existing file or to create a new file from scratch. You can also use this editor to just read a text file.

Starting the vi Editor

Operation Modes

While working with vi editor you would come across following two modes −

- **Command mode** − This mode enables you to perform administrative tasks such as saving files, executing commands, moving the cursor, cutting (yanking) and pasting lines or words, and finding and replacing. In this mode, whatever you type is interpreted as a command.
- **Insert mode** − This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and finally it is put in the file.

Getting Out of vi

Command	Description
k	Moves the cursor up one line.
j	Moves the cursor down one line.
h	Moves the cursor to the left one character position.
l	Moves the cursor to the right one character position.

There are following two important points to be noted −

- The vi is case-sensitive, so you need to pay special attention to capitalization when using commands.
- Most commands in vi can be prefaced by the number of times you want the action to occur. For example, 2j moves cursor two lines down the cursor location.

There are many other ways to move within a file in vi. Remember that you must be in command mode (press Esc twice). Here are some more commands you can use to move around the file −

Control Commands

There are following useful command which you can use along with Control Key −

Editing Files

To edit the file, you need to be in the insert mode. There are many ways to enter insert mode from the command mode −

Command	Description
i	Inserts text before current cursor location.
I	Inserts text at beginning of current line.
a	Inserts text after current cursor location.
A	Inserts text at end of current line.
o	Creates a new line for text entry below cursor location.
O	Creates a new line for text entry above cursor location.

Deleting Characters

Here is the list of important commands which can be used to delete characters and lines in an

Change Commands

You also have the capability to change characters, words, or lines in vi without deleting them. Here are the relevant commands −

Copy and Past Commands

You can copy lines or words from one place and then you can past them at another place using following commands −

Command	Description
yy	Copies the current line.
yw	Copies the current word from the character the lowercase w cursor is on until the end of the word.
p	Puts the copied text after the cursor.
P	Puts the yanked text before the cursor.
dd	delete the line

Advanced Commands

There are some advanced commands that simplify day-to-day editing and allow for more efficient use of vi −

Word and Character Searching

The vi editor has two kinds of searches: string and character. For a string search, the / and ? commands are used. When you start these commands, the command just typed will be shown on the bottom line, where you type the particular string to look for.
These two commands differ only in the direction where the search takes place −

- The / command searches forwards (downwards) in the file.
- The ? command searches backwards (upwards) in the file.

Character	Description
^	Search at the beginning of the line. (Use at the beginning of a search expression.)

The character search searches within one line to find a character entered after the command. The The t and T commands search for a character on the current line only, but for t, the cursor moves to the position before the character, and T searches the line backwards to the position after the character.

Set Commands

You can change the look and feel of your vi screen using the following **:set** commands. To use

Command	Description
:set ic	Ignores case when searching
:set ai	Sets autoindent
:set noai	To unset autoindent.
:set nu	Displays lines with line numbers on the left side.

OUTCOME: We have successfully studied the Vi editor in Linux and also demonstrated various commands on it.

PRACTICAL NO. 5
Study of Nano–editor

OBJECTIVE: To demonstrate the working of various commands of Nano Editor in Linux.

THEORY: ***Nano*** is a text *editor* suited to working in UNIX. It is not as powerful as PC window-based editors, as it does not rely on the mouse, but still has many useful features. Most *nano commands* are invoked by holding down the Ctrl key.

Nano is a great command -
line text editor for a number of reasons,It's small, lightweight and is included in most distributions;It's easy to use and gives plenty of on-screen feedback;It includes many power -user features to compete with Vi(m) and Emacs.ano runs in text mode at the command line; this may prompt you to think:"Why would I want to learn a text mode editor? Kate, Gedit and FooEdit 2000 do everything I need to on my desktop." Well, firstly, all regular Linux users and administrators should be well -versed in a command line editor. If something goes wrong with the X Window System, for example, you'll end up at a CLI prompt and your skills will be essential for editing config files.Secondly, if you set up a Linux machine as a server, it almost certainly won't include X or any kind of graphical tools, so you'll be using a text mode editor frequently. Learning some core skills beyond simply
opening and saving files can make you work much more productively, so that's what we'll be looking at here.

nano index.php Open or create the file “index.php” with nano on command line.
Ctrl-o Y Enter Save changes.

Ctrl-r Alt-f	Open a new file with a new buffer within nano.
Alt->	Switch to the next file buffer in nano.
Alt-<	Switch to the previous file buffer in nano.
Ctrl-x	Quit nano.

Navigating through file contents in nano

Ctrl-a	Move to the beginning of the current line.
Ctrl-e	Move to the end of the current line.
Ctrl-v	Move down one page.
Ctrl-y	Move up one page.
Alt-	Go to the beginning of the file.
Alt-/	Go to the end of the file.
Alt-g	Go to a target line number.
Alt-]	Jump to matching open/close symbol.
Alt-a Alt-}	Select a block and indent the block.
Alt-a Alt-{	Select a block and outden the block.

Copy and Paste in nano

Alt-a	To select a block for copy or cut operation, do Alt-a again to unselect.
Alt-a Alt-^	Copy a highlighted block to the clipboard.
Alt-a Ctrl-k	Cut a highlighted block to the clipboard.
Ctrl-k	Cut from the current cursor position to the end of the current line.
Ctrl-u	Paste the contents from the clipboard at the current cursor position.

Search and Replace in nano

Ctrl-w	Search for a target string.
Alt-w	Repeat the last search.
Alt-r	Search and replace

OUTCOME: We have successfully studied the Nano editor in Linux and also demonstrated various commands on it.