

A Computer Services Educational Bulletin - Version 1.1  
 © Copyright 1995 All rights reserved.

## UNIX Editor (vi)

### Summary

*At Temple University, the UNIX<sup>TM</sup><sup>1</sup> operating system is run on Astro, a multi-user computer system. UNIX also runs on various other workstations throughout the university. This bulletin is designed to introduce you to the UNIX screen editor, vi. It is meant to get you started using vi and to familiarize you with cursor motions and simple editing. Many of the more useful vi commands that can make you more effective at using vi are presented. There is also a vi command summary at the end of this document. For information about the UNIX operating system, see E-300.*

### Table of Contents

<b>Part I. Introduction</b>	
What is vi and how does it work?.....	<b>3</b>
<b>Part II. Communicating with vi</b>	
What do I need to know to begin communicating with vi?.....	<b>4</b>
<b>A. Naming a File</b> .....	<b>4</b>
<b>B. Entering vi</b> .....	<b>4</b>
<b>C. Modes of vi</b> .....	<b>5</b>
<b>D. Control Commands</b> .....	<b>5</b>
<b>Part III. Getting Started</b>	
I want to create a file. How do I start?.....	<b>6</b>

<sup>1</sup>UNIX is a registered trademark of UNIX System Laboratories, Inc.

# Table of Contents, Continued

<b>Part IV. Moving Around on the <i>vi</i> Screen</b>	
What are the basic cursor movement commands?.....	<b>8</b>
<b>A. The <i>vi</i> Screen</b> .....	<b>8</b>
<b>B. <i>vi</i> Screen Cursor Movement Commands</b> .....	<b>8</b>
<b>C. Basic <i>vi</i> Word Cursor Movement Commands</b> .....	<b>9</b>
<b>D. Scrolling the <i>vi</i> Screen</b> .....	<b>9</b>
<b>Part V. Editing with <i>vi</i></b>	
What are the basic editing commands?.....	<b>10</b>
<b>A. Adding Text</b> .....	<b>10</b>
<b>B. Deleting Text</b> .....	<b>10</b>
<b>C. The "undo" Command</b> .....	<b>11</b>
<b>D. Changing Text</b> .....	<b>11</b>
<b>Part VI. Advanced Editing with <i>vi</i></b>	
What are some of the more advanced editing commands?.....	<b>12</b>
<b>A. Searching for Text</b> .....	<b>12</b>
<b>B. Copying and Moving Text</b> .....	<b>13</b>
<b>C. Joining Lines</b> .....	<b>14</b>
<b>Part VI. Summary of <i>vi</i> Commands Covered in this Bulletin</b> .....	<b>15</b>
<b>Part VII. Appendix</b>	
Practice exercises for the novice.....	<b>17</b>

# Introduction

## What is *vi* and how does it work?

- **What is *vi*?...**

The UNIX text editor is called *vi*. With *vi*, you can modify characters and lines by typing at the current cursor position. The cursor is the little blinking line or box that shows where the next character you type will be displayed. Because *vi* allows you to display and work on one screen of text at a time, it is referred to as a screen editor. With a screen editor, you move the cursor around the screen until you find the part you want to modify, then you add or change the text as desired. You can use the editor to add new text; delete or modify existing text; and move text from one part of a file to another.

- ***vi* works with a copy of the original file...**

It is important to remember that, when using *vi*, you are editing a copy of the original file, not the original file itself. When you are finished making changes, you have two versions of the file and must decide whether to keep the original file or save the edited copy.

- **Where to get more information on learning how to use the UNIX system...**

### For more information

This document offers a brief introduction to the UNIX screen editor, *vi*. To become more familiar with the UNIX operating system, refer to the following Computer Services documents:

- *UNIX Quick Reference Guide*
- *Astro Account Guide*
- E-300, *UNIX Operating System*

In addition to these documents, a summary of *vi* commands can be found in the file `/usr/local/info`.

To keep posted on the current status of the Astro UNIX system, refer to the Usenet newsgroups *temple.unix* and *temple.announcements*. You can read these newsgroups using the tin program. For information on tin, refer to the *Using tin on the Astro UNIX system to participate in Internet discussion groups* Quick Reference Guide.

Quick Reference Guides are available free from the following locations:

<b>Main Campus</b>	<b>Ambler Campus</b>	<b>Health Sciences Center</b>
Help Desk	Computer Services	Computer Services
Compute Bldg.	Library Bldg.	General Services Bldg.
Ground Floor	Room 29	Third Floor

In addition to Quick Reference Guides, faculty and staff members can also each obtain one free copy of any Educational Bulletin (E-Bulletin) or seminar handout from the locations listed above. Students, on the other hand, can purchase copies of E-Bulletins and seminar handouts from the following locations. The cost is five cents per page:

<b>Main Campus</b>	<b>Ambler Campus</b>	<b>Health Sciences Center</b>
Copy Center	Copy Center	Computer Services
Conwell Hall	Bright Hall	General Services Bldg.
6th Floor	Room 104	Third Floor

# Communicating with *vi*

## What do I need to know to begin communicating with *vi* ?

- *Naming files...*

### Naming a file

You use the *vi* text editor to create files that you can store on the computer and retrieve for use at a later time. Most files have a two-part name: the name and the extension. The name and extension are separated by a period; together they make up the full filename. You should choose a filename that will help you to remember what you put in the file.

The file extension should indicate the function of the file. There are several recommended file extensions in UNIX. For example, the ".f" file extension allows the FORTRAN compiler to recognize a file. (See Table 1.) Use the extension that is appropriate for the type of file being created.

Common UNIX Recommended File Extensions	
Extension	Type of File
.c	C source file
.f	FORTRAN file
.o	Object file
.p	Pascal source file
.s	Scheme source file
.tex	TeX and LaTeX source/manuscript file

Table 1.

There are some general guidelines to follow when naming files:

1. The filename can be any combination of 255 or fewer characters; no spaces are allowed in the name.
2. It is best to restrict characters in your filenames to letters, numbers, underscores, and periods.
3. The UNIX operating system is case sensitive. "Friends" and "friends" would refer to two different files, since the "F" in the first filename is uppercase, and "f" in the second filename is lowercase.

- *Command to begin a vi session...*

### Entering *vi*

To begin your UNIX editing session, type the "*vi*" command at the UNIX shell prompt (%), followed by the name of the file you wish to edit:

*vi filename*

and press the **Return** key (also labeled **Enter** on some keyboards). When you issue the command to begin your editing session, you stop communicating with the UNIX shell and begin communicating with the *vi* text editor.

• *There are two different modes within vi:*

- 1) *command mode*
- 2) *insert mode*

### **Modes of vi**

After you type the "vi" command, you enter command mode. The commands used in command mode are the ones you use to alter your text and are not printed on the screen. You do not need to press the **Return** key after you type them, and they are case sensitive.

The vi command "i" puts you into insert mode. Here, whatever you type becomes part of the file you are editing. To begin a new line of text, press the **Return** key. To return to command mode, press the **Escape** key (sometimes labeled **Esc**).

• *Control commands are preceded by a colon(:)...*

### **Control Commands**

There are certain commands in vi that are preceded by a colon (:). These are control commands that have to do with reading and writing files and other functions of vi.

The control commands used to end your editing session include those shown in Table 2.

• *Control commands:*

- 1) *:w*
- 2) *:wq*
- 3) *:q*
- 4) *:q!*

<b>Control Commands</b>	
<b>Command</b>	<b>Command Function</b>
<code>:w</code>	writes (saves) the file without quitting <i>vi</i>
<code>:w filename</code>	writes (saves) the file into a file named <i>filename</i>
<code>:wq</code>	writes (saves) the file and quits <i>vi</i> . (The original version of the file is replaced with the edited version. If no original file exists, this command will create one.)
<code>:q</code>	quits <i>vi</i> (You are returned to the UNIX shell. This command won't quit if changes have been made to the file.)
<code>:q!</code>	quits <i>vi</i> without writing (saving) the file. (You are returned to the UNIX shell.)

**Table 2.**

# Getting Started

I want to create a file. How do I start?

*The exercises below illustrate how to create a new file called "friends" and insert text into it using the vi text editor. It is assumed that you are already logged in to your UNIX account.*

## Creating a file

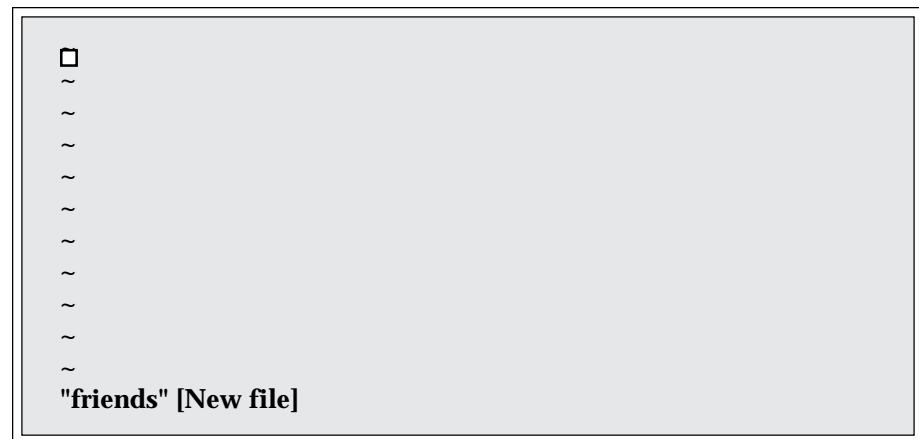
- **Command to start vi...**

*Step 1.* At the shell prompt (%) type:

**vi friends**

and press the **Return** key.

The filename will appear in the message line at the bottom of the screen. The "~" in the first column means that the file does not have enough lines to fill the screen. The cursor is depicted by a box (□) and is placed in the upper left corner when vi starts. (See Figure 1.)



**Figure 1.**

- **Command to insert text...**

*Step 2.* To begin inserting text in your file, type the vi command "i":

**i**

and do not press the **Return** key. The "i" will not appear on the screen; it indicates to vi that you want to insert text.

*Step 3.* Type:

**Mary is my friend.**

and press the **Return** key to move to the next line. If you make a typing mistake, use the **Backspace** key to move the cursor to the incorrect letter, and then, retype correctly.

*Step 4.* Type the second, third, and fourth lines of text as shown below, pressing the **Return** key after the second and third lines. Do not press the **Return** key after you type the last line. Doing so will create a blank line following the last line of text.

**Sue is my new friend.**

**Pat is an old friend.**

**Betty is my best friend.**

• **Command to exit the vi editor...**

*Step 5.* Press the **Esc** key once to signal *vi* that you have stopped entering text. Your screen will not change. Whatever you type after you press the **Esc** key will be interpreted as a *vi* command, not as part of your file.

• **Command to save...**

*Step 6.* To save your file and return to the UNIX shell, type:

**:wq**

and press the **Return** key. Be sure to type the colon (:) before the "wq." This command will appear at the bottom of the screen. After you have typed the command correctly, the message line displays the filename, number of lines, and number of characters at the bottom of the screen. You should then see the shell prompt (%).

# Moving Around on the vi Screen

## What are the basic cursor movement commands?

To edit a file with vi, you must first know how to move the cursor around on the screen.

### The vi screen

To open and edit the file "friends," at the shell prompt (%) type:

- **Command to open a file...** `vi friends`

and press the **Return** key. You should see a screen that looks like that shown in Figure 2. (The cursor is represented by the small rectangle (□)).

```
□ Mary is my friend.  
Sue is my new friend.  
Pat is an old friend.  
Betty is my best friend.  
~  
~  
~  
~  
~  
~  
~  
"friends" 4 lines, 88 characters
```

Figure 2.

- **Screen cursor movement commands...**

### vi screen cursor movement commands

The vi screen motion commands are summarized in the table below. (The "n" in the table refers to the number of spaces or lines the cursor will be moved.)

vi Screen Cursor Movement Commands	
Command	Command Function
h	moves cursor one space to left
nh	moves cursor n spaces to left
j	moves cursor down 1 line
nj	moves cursor down n lines
k	moves cursor up 1 line
nk	moves cursor up n lines
l	moves cursor one space to right
nl	moves cursor n spaces to right
0 (zero)	moves cursor to beginning of line
\$	moves cursor to end of line

Table 3.

The **Spacebar** can also be used to move right. Some versions of *vi* will let you use the arrow keys on your keyboard. However, usually the arrow keys will be disabled. You cannot use any cursor movement commands in insert mode. (Note: Astro allows this with a vt100 terminal.)

### Basic *vi* word cursor movement commands

Words are recognized by *vi* as letters and numbers separated by blanks, tabs, or punctuation marks. The basic commands for moving by word(s) in a file are shown in the table below.

- **Basic word cursor movement commands...**

Basic <i>vi</i> Word Cursor Movement Commands	
Command	Command Function
w	moves cursor forward one word
nw	moves cursor forward <i>n</i> words
b	moves cursor backward one word
nb	moves cursor backward <i>n</i> words
e	move cursor to end of word

Table 4.

- **Commands to scroll the screen...**

### Scrolling the *vi* screen

Because the practice file, "*friends*" is relatively small, it can be displayed on a single screen. When files are large, you can only look at them one screen at a time. You, therefore, need to be able to scroll through your file. Table 5 explains the most common commands for scrolling to different parts of your files. The "^" in the table represents the **Control** key (sometimes labeled **Ctrl**). Commands prefaced by the "^" symbol are invoked by first holding down the **Ctrl** key and then simultaneously typing the appropriate key once.

Basic Commands for Scrolling	
Command	Command Function
^u	scrolls a half screen "up" or backward
^d	scrolls a half screen "down" or forward
^b	scrolls one screen "up" or backward
^f	scrolls one screen "down" or forward
^g	tells you which line of the file you are on and the number of lines in the file
nG	moves the cursor to line number <i>n</i> of the file

Table 5.

# Editing with vi

## What are the basic editing commands?

• **Command to append text...**

### Adding text

To add text, first position the cursor and type the "a" command. This puts you in vi's append mode. Everything you type will be appended to the text after the character indicated by the cursor position. When you are finished adding text, press the **Esc** key. The cursor will then move back to the last character you entered to indicate that you are no longer in insert mode.

• **Command to insert text...**

You can also insert text by using the vi "i" command. The "i" command works like the "a" command, but it inserts, instead of appending. This means that the characters you type will be placed before the current character position. Like the "a" command, "i" requires an **Esc** to finish inserting. The cursor will then move back to the last character you entered, to indicate that you are no longer in insert mode.

Hint: For people who like to know what mode they are in, put "set showmode" in the .exrc file. To modify the .exrc file, first copy it into your home directory from the /usr/local/sample directory.

Basic vi Commands for Adding Text	
Command	Command Function
a	appends after cursor
i	inserts before cursor

Table 6.

• **Commands to delete text...**

### Deleting text

There are two commands that delete text in vi: "x" and "d" (see Table 7). To delete one character, you use the "x" command. The "x" command deletes the character at the current cursor position, and moves the rest of the line left into the void created by the deleted character. The "x" command can be preceded by a number to indicate how many characters you want to delete.

The "d" command is always followed by other characters that specify what will be deleted. For example, if it is followed by a "w," a word will be deleted. If it is followed by a "\$," the current position to the end of the line will be deleted.

The "dd" command is used to delete the current line. When a line is deleted, subsequent lines move upward into the void created by the deleted line. The "dd" command can be preceded by a number to indicate the number of lines to delete.

Basic Commands for Deleting Text	
Command	Command Function
x	deletes character
dd	deletes line
d(+word cursor motion)	deletes what is indicated by movement command

Table 7.

• **The undo command cancels the last change made to your file...**

### The "undo" command

The vi command "u" (for undo) will cancel the last change made to your file. To undo the previous command, type:

**u**

The "u" command can only undo commands that can change the file, such as "x," "dd," "a," and even "u" itself. It cannot undo commands which interact with disk files, such as "w" and "vi."

• **Commands to change text...**

### Changing text

With *vi* you can change, or replace, a character or characters in a file. The "c" command is used to change text. It is followed by a cursor motion, and whatever lies in the motion can be changed. When the "c" command is invoked, it puts you in input mode so you can type your changes. You must press the **Esc** key to get back into command mode. For example, to change (or replace) an entire word, first place the cursor on the first letter of the word, and then type:

**cw**

(for "change word"). A dollar sign (\$) will appear at the last character affected by the change. Type the new word and then press the **Esc** key.

The "cc" command is a special case of the "c" command, and is used to change the entire line.

The "r" command is used to replace a character on which the cursor is positioned. It is used by first placing the cursor on the desired character, typing the command, and then typing the correct letter.

Basic Commands for Changing Text	
Command	Command Function
c(+word cursor motion)	changes what is indicated by movement command
cc	changes line
r	replaces single character

**Table 8.**

# Advanced Editing with vi

## What are some of the more advanced editing commands?

- **Command for a forward search...**

### Searching for text

With *vi*, you can search for text in your file with the "/" command, followed by the text (string) for which you want to search. (Searches are case-sensitive.) This command is used to search forward through a file, beginning at the current cursor position, until it finds an occurrence of the text you have specified. It then positions the cursor at that location. To continue the search, you can use the "n" command. This command will repeat the last search and position the cursor on the next occurrence of the specified text.

If you do not begin your search at the beginning of the file, when the search reaches the bottom of the file it will go to the beginning and continue until it reaches the original starting point of the search. If no occurrence of the specified text is found, a message will be displayed to inform you that none was found.

- **Command for a backward search...**

To perform a backward search, use the "?" command, followed by the text (string) for which you want to search. This command works just like the "/" command, except that it searches backward in the file, beginning at the current cursor position. As with the forward search, the "n" command can be used to repeat the last search.

Unlike most *vi* commands, both the "/" and "?" commands require the user to press the **Return** key (indicated by <CR> in the table below) after typing the search string.

Commands for Searching for Text	
Command	Command Function
/string <CR>	searches forward for string
?string <CR>	searches backward for string
n	repeats last search (forward or backward)
N	repeats last search in reverse direction of that initially designated

Table 9.

## Copying and moving text

You can use *vi* to move lines from one place to another in your file and to copy (duplicate) lines from one part of your file to another. Rearranging text involves three steps:

### • *Three steps for copying and moving text...*

1. Yank or copy the material.
2. Move the cursor to where the material is to go.
3. Put the yanked or copied material there.

### • *Command for copying (yanking) text...*

The command "y" (for yank) copies the characters starting at the cursor into a storage area called the buffer. Yank allows a numeric prefix to specify the number of objects to be yanked, and a suffix after "y" defines the objects to be yanked. For example, "yw" is used to yank a word, and "3yw" is used to yank three words. The "Y" command (uppercase Y) or the "yy" command can be used to yank the entire current line.

### • *Command for moving (putting) text...*

To move yanked text, place the cursor where you wish to place the material, and then use the "p" command to put the text there. The "p" command (lowercase p) puts the yanked text to the right of the cursor. If an entire line was yanked, the text is placed below the current line. The "P" command (uppercase P) puts the yanked text to the left of the cursor. If an entire line was yanked, the text is placed above the current line.

Commands for Copying and Moving Text	
Command	Command Function
y	yanks object and saves in temporary buffer
yx	yanks object/s (x) and saves in temporary buffer
nyx	yanks n number of objects (x) and saves in temporary buffer
Y	yanks line and saves in temporary buffer
nY	yanks n number of lines and saves in temporary buffer
p	puts saved buffer after cursor
P	puts saved buffer before cursor

Figure 4.

In addition to its editing buffer, *vi* maintains several other temporary storage areas called working buffers. If you do not specify which buffer to use, *vi* automatically saves the material you last yanked ("y"), deleted ("d"), or changed ("c") in its unnamed buffer. To save to a specific buffer, you must precede your command with a double quote (") and the name of the buffer you wish to use. There are 26 named buffers, named a, b, c, d,...z. This gives you 26 storage areas. For example, you would type:

**"a3Y**

to yank three lines into a buffer named "a." To paste any one of the 26 storage areas, you would type:

**"<buffer>p**

where <buffer> is the name of the buffer.

- *Command for joining lines of text...*

### **Joining lines**

The "J" command (uppercase J) is used to join the line on which the cursor is positioned with the line immediately following it. To join two short lines into one long one, move the cursor anywhere on the first of the two lines you want to join. Press "J" to move the second line just after the first one. The cursor will remain on the first line.

# Summary of *vi* Commands Covered in this Bulletin

## • *Command to start vi...*

### Command for using *vi* from UNIX (page 4)

Command	Command Function
<i>vi</i> ( <i>filename</i> )	edit file or create a new file

## • *Control commands...*

### *vi* control commands (page 5)

Command	Command Function
:w	writes (saves) the file without quitting <i>vi</i>
:wq	writes (saves) the file and quits <i>vi</i> (The original version of the file is replaced with the edited version. If no original file exists, this command will create one.)
:q	quits <i>vi</i> (You are returned to the UNIX shell. This command won't quit if changes have been made to the file.)
:q!	quits <i>vi</i> without writing (saving) the file (You are returned to the UNIX shell.)

## • *Cursor movement commands...*

### *vi* screen cursor movement commands (page 8)

Command	Command Function
h	moves cursor one space to left
<i>nh</i>	moves cursor <i>n</i> spaces to left
j	moves cursor down 1 line
<i>nj</i>	moves cursor down <i>n</i> lines
k	moves cursor up 1 line
<i>nk</i>	moves cursor up <i>n</i> lines
l	moves cursor one space to right
<i>nl</i>	moves cursor <i>n</i> spaces to right
0 (zero)	moves cursor to beginning of line
\$	moves cursor to end of line

## • *Word commands...*

### Basic *vi* word cursor movement commands (page 9)

Command	Command Function
w	moves cursor forward one word
<i>nw</i>	moves cursor forward <i>n</i> words
b	moves cursor backward one word
<i>nb</i>	moves cursor backward <i>n</i> words
e	move cursor to end of word

• *Commands for scrolling...*    **Basic vi commands for scrolling (page 9)**

Command	Command Function
^c	scrolls a half screen "up" or backward
^d	scrolls a half screen "down" or forward
^b	scrolls one screen "up" or backward
^f	scrolls one screen "down" or forward
^g	tells you which line of the file you are on and the number of lines in the file
nG	moves the cursor to line number <i>n</i> of the file

• *Commands for adding text...*    **Basic vi commands for adding text (page 10)**

Command	Command Function
a	appends after cursor
i	inserts before cursor

• *Commands for deleting text...*    **Basic vi commands for deleting text (page 10)**

Command	Command Function
x	deletes character
dd	deletes line
d(+cursor motion)	deletes what is indicated by movement command

• *Commands for changing text...*    **Basic vi commands for changing text (page 11)**

Command	Command Function
c(+cursor motion)	changes what is indicated by movement command
cc	changes line
r	replaces text

• *Commands for searching for text...*    **Commands for searching for text (page 12)**

Command	Command Function
/string <CR>	searches forward for string
?string <CR>	searches backward for string
n	repeats last search (forward or backward)
N	repeats last search in reverse direction of that initially designated

• *Commands for copying and moving text...*    **Commands for copying and moving text (page 13)**

Command	Command Function
y	yanks (copies) object and saves in temporary buffer
yx	yanks (copies) object ( <i>x</i> ) and saves in temporary buffer
nyx	yanks (copies) <i>n</i> number of objects ( <i>x</i> ) and saves in temporary buffer
Y	yanks (copies) line and saves in temporary buffer
nY	yanks (copies) <i>n</i> number of lines and saves in temporary buffer
p	puts saved buffer after cursor
P	puts saved buffer before cursor

# Appendix

## Practice exercises for the novice

*This section is designed to be used by those who are new to vi. In the following exercises, it is assumed that you are logged in to your UNIX account, and that you have created a file called "friends," as described on pages 6-7. Remember that the UNIX operating system is case sensitive.*

### • Opening a file...

#### Opening a file

*Step 1.* To open and edit your file, type the following at the shell prompt (%):

```
vi friends
```

and press the *Return* key.

### • Practice exercises for basic screen cursor movement commands...

#### Basic vi screen cursor movement commands

Use the following exercises to practice the basic cursor movements shown in Table 3 on page 8. Notice that the commands do not show on the screen. This is because you are working in the command mode of vi. If you try to move past the beginning or end of the file, vi will beep. Before beginning these exercises, your cursor should be resting on the first letter of the first word in line 1, "Mary."

*Step 1.* Press "j" (lowercase j) to move down one line. The cursor should rest on the first letter of the first word in line 2, "Sue."

*Step 2.* Press "j" (lowercase j) to move down one line. The cursor should rest on the first letter of the first word in line 3, "Pat."

*Step 3.* Press "l" (lowercase l) to move one space to the right. The cursor should rest on the second letter of the first word in line 3, "Pat."

*Step 4.* Press "k" (lowercase k) to move up one line. The cursor should rest on the second letter of the first word in line 2, "Sue."

*Step 5.* Press "l" (lowercase l) to move one space to the right. The cursor should rest on the third letter of the first word in line 2, "Sue."

*Step 6.* Press "k" (lowercase k) to move up one line. The cursor should rest on the third letter of the first word in line 1, "Mary."

*Step 7.* Press "h" (lowercase h) to move one space to the left. The cursor should rest on the second letter of the first word in line 1, "Mary."

*Step 8.* Press "3j" (3, lowercase j) to move down 3 lines. The cursor should rest on the second letter of the first word in line 4, "Betty."

*Step 9.* Press "3k" (3, lowercase k) to move up 3 lines. The cursor should rest on the second letter of the first word in line 1, "Mary."

*Step 10.* Press "**2l**" (2, lowercase l) to move 2 spaces to the right. The cursor should rest on the fourth letter of the first word in line 1, "Mary."

*Step 11.* Press "**3h**" (3, lowercase h) to move 3 spaces to the left. The cursor should rest on the first letter of the first word in line 1, "Mary."

*Step 12.* Press "**\$**" (Shift-4) to move to the end of the line. The cursor should rest on the period "." at the end of line 1.

*Step 13.* Press "**0**" (zero) to move to the beginning of the line. The cursor should rest on the first letter at the beginning of line 1, "Mary."

• **Practice exercises for basic word cursor movement commands...**

### **Basic vi word cursor movement commands**

Use the following exercises to practice the basic cursor movements shown in Table 4 on page 9. Notice that the commands do not show on the screen. This is because you are working in the command mode of *vi*. If you try to move past the beginning or end of the file, *vi* will beep. Before beginning these exercises, your cursor should be resting on the first letter of the first word in line 1, "Mary," as shown in Figure 2 on page 8.

*Step 1.* Press "**w**" (lowercase w) to move forward one word. The cursor should rest on the first letter of the second word in line 1, "is."

*Step 2.* Press "**e**" (lowercase e) to move to the end of the word. The cursor should rest on the last letter of the second word in line 1, "is."

*Step 3.* Press "**e**" (lowercase e) to move to the end of the next word. The cursor should rest on the last letter of the third word in line 1, "my."

*Step 4.* Press "**w**" (lowercase w) to move forward one word. The cursor should rest on the first letter of the fourth word in line 1, "friend."

*Step 5.* Press "**b**" (lowercase b) to move back to the previous word. The cursor should rest on the first letter of the third word in line 1, "my."

*Step 6.* Press "**2b**" (2, lowercase b) to move back 2 words. The cursor should rest on the first letter of the first word in line 1, "Mary."

*Step 7.* Press "**3w**" (3, lowercase w) to move forward 3 words. The cursor should rest on the first letter of the fourth word in line 1, "friend."

*Step 8.* Press "**3b**" (3, lowercase b) to move back 3 words. The cursor should rest on the first letter of the first word in line 1, "Mary."

• *Practice exercises for adding text...*

### Adding text

Use the following exercises to practice appending and inserting text in the file "friends." Before beginning these exercises, your cursor should be resting on the second letter of the first word in line 1, "Mary." If you just completed the exercises on page 18, press "l" (lowercase l) to position the cursor before beginning the following exercises. Your screen should look like that shown below in Figure 3.

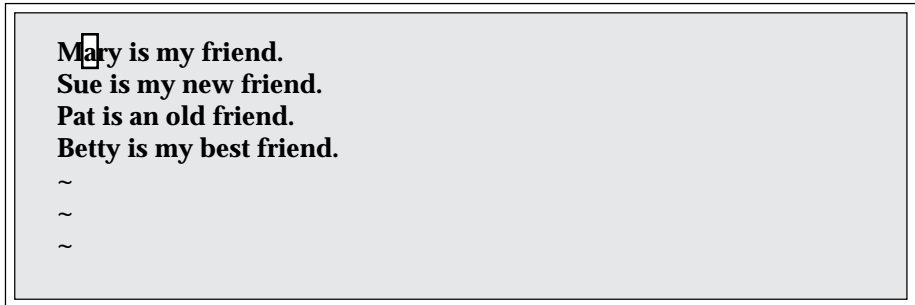


Figure 3.

Step 1. Type the following:

**amilk**

Your screen should have changed as shown in Figure 4.

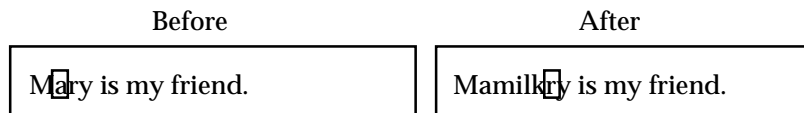


Figure 4.

Step 2. Press the **Esc** key. The cursor moves back to the last character you entered, "k," and you are no longer in insert mode. (See Figure 5.)

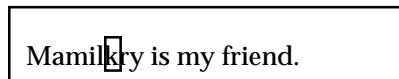


Figure 5.

Step 3. Type the following:

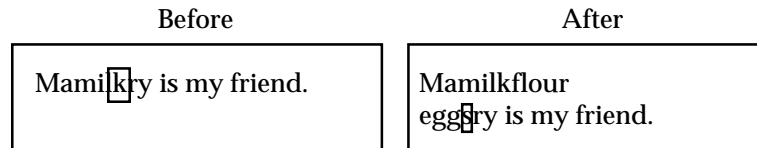
**afLOUR**

and press the **Return** key. Then type:

**eggs**

and press the **Esc** key.

The appending started between the "k" and the "r" in the first line, causing the remainder of the line to be carried to the next line when the **Return** key was pressed. (See Figure 6 on the next page.)



**Figure 6.**

*Step 4.* Press "**3j**" (3, lowercase j) to move down 3 lines. The cursor should rest on the fourth letter of the first word in line 5, "Betty."

*Step 5.* Type the following:

**isugar**

to insert the letters "sugar" to the left of the cursor position. Then press the **Esc** key to exit insert mode and move the cursor back to the last character inserted. (See Figure 7.)

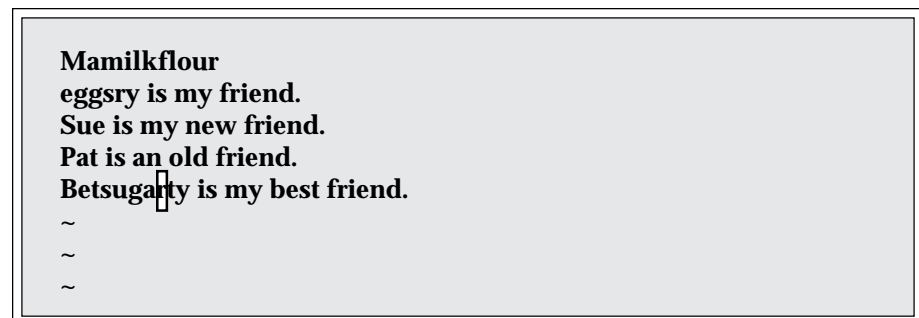


**Figure 7.**

• *Practice exercises for basic vi text deletion commands...*

### Deleting text

Use the following exercises to practice deleting text in the file "*friends*," as created in previous exercises. Before beginning these exercises, your cursor should be resting on the eighth character in the first word in line 5, "Betsugarty." (See Figure 8.)



**Figure 8.**

*Step 1.* Press "**x**" (lowercase x) to delete the letter "r." The rest of the line moves into the void created by the deletion, and the cursor should rest on the eighth character of the first word in line 5, "Betsugaty."

*Step 2.* Press "**x**" (lowercase x) to delete the letter "t." The rest of the line moves into the void created by the deletion, and the cursor should rest on the eighth character of the first word in line 5, "Betsugay."

*Step 3.* Press "**d\$**" (lowercase d, Shift-4) to delete "y is my best friend." The cursor should rest on the seventh character of the word in line 5, "Betsuga."

*Step 4.* Press "xxxxxxx" (lowercase x, 7 times) to delete "Betsuga." When "x" is used at the end of a line, it deletes the character at the current cursor position and then moves the cursor left to what is now the new end of line 5.

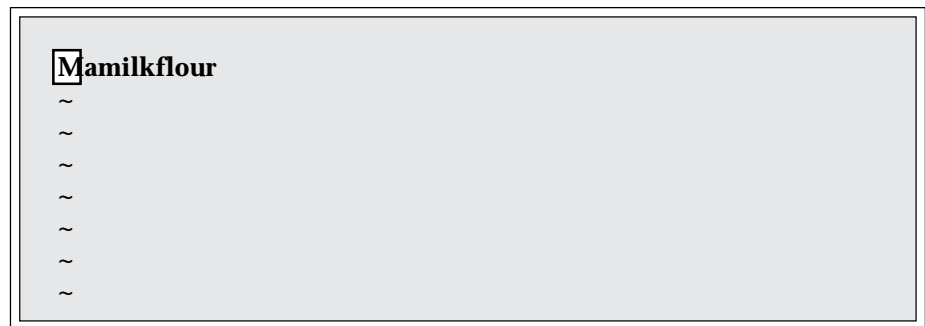
*Step 5.* Press "2x" (2, lowercase x). There is no change on the screen, and you will get a "beep" because you are trying to delete non-existent characters.

*Step 6.* Press "dd" (lowercase d, lowercase d) to delete line 5. Line 5 is replaced with a "~," and the cursor should rest on the first character of the first word in line 4, "Pat."

*Step 7.* Press "2k" (2, lowercase k) to move up 2 lines. The cursor should rest on the first character of the first word in line 2, "eggsry."

*Step 8.* Press "dw" (lowercase d, lowercase w) to delete the first word in line 2, "eggsry" The cursor should rest on the new first character of the first word in line 2, "is."

*Step 9.* Press "3dd" (3, lowercase d, lowercase d) to delete lines 2, 3, and 4. The cursor should rest on the first character of the first word in line 1, "Mamilkflour." Your file should now look like that shown below in Figure 9.



```
Mamilkflour
~
~
~
~
~
~
~
```

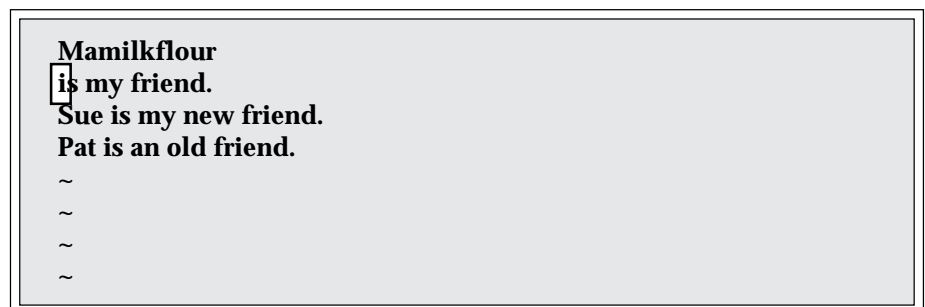
Figure 9.

• *Practice exercise for using the "undo" command...*

### The "undo" command

Use the following exercise to practice the "undo" command after completing the exercises above.

*Step 1.* Press "u" (lowercase u) to undo the last command (step 9 above). Lines 2, 3, and 4 are now undeleted. The cursor should rest on the first character of the first word in line 2, "is." (See Figure 10.)



```
Mamilkflour
is my friend.
Sue is my new friend.
Pat is an old friend.
~
~
~
~
```

Figure 10.

• *Practice exercises for basic commands to change text...*

### Changing text

Use the following exercises to practice changing text in the file "friends," as created in previous exercises. Before beginning these exercises, your cursor should be resting on the first letter of the first word in line 1, "Mamilkflour." If you just completed the exercises on page 21, press "k" to position the cursor before beginning the following exercises.

*Step 1.* Press "cw" (lowercase c, lowercase w) to change the word, "Mamilkflour." The cursor remains on the "M," and a "\$" is displayed at the last character affected by the change, "r." (See Figure 11.)

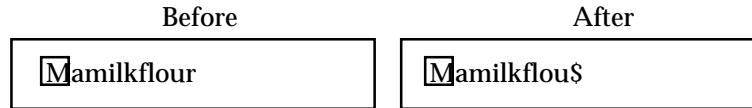


Figure 11.

*Step 2.* Type:

**Mary**

and press the **Esc** key. "Mamilkflour" is changed to "Mary," and you are returned to command mode. The cursor should rest on the last character entered, the "y" in "Mary."

*Step 3.* Press "0" (zero) to move to the beginning of line 1. The cursor should rest on the first character of the first word in line 1, "Mary."

*Step 4.* Press "3j" (3, lowercase j) to move down 3 lines. The cursor should rest on the first character of the first word in line 4, "Pat."

*Step 5.* Press "cc" (lowercase c, lowercase c) to change the line. Line 4 disappears.

*Step 6.* Type:

**Sam is my neighbor.**

and press the **Esc** key. "Pat is an old friend." is changed to "Sam is my neighbor." You are returned to command mode, and the cursor should rest on the last character entered, the period "." at the end of the sentence.

*Step 7.* Press "0" (zero) to move the cursor to the beginning of line 4. The cursor should rest on the first character of the first word in line 4, "Sam."

*Step 8.* Press "r" (lowercase r) to replace the letter on which the cursor is positioned, "S."

Then type:

**P**

and press the **Esc** key. The "S" is replaced with the letter "P." (See Figure 12.)



Figure 12.



Figure 13.

• *Practice exercises for searching for text...*

### Searching for text

Use the following exercises to practice searching for text in the file "friends," as created in previous exercises. Before beginning these exercises, your cursor should be resting on the first letter of the first word in line 4, "Pam." (See Figure 13.)

Step 1. Type:

**/neighbor**

and press the **Return** key to search forward for the word, "neighbor." The cursor should rest on the first letter of the last word in line 4, "neighbor."

Step 2. Type:

**?is**

and press the **Return** key to search backward for the word, "is." The cursor should rest on the first letter of the second word in line 4, "is."

Step 3. Type:

**n**

to continue searching backward for the word "is." The cursor should rest on the first letter of the second word in line 3, "is."

Step 4. Type:

**N**

to reverse the search for the word "is." The cursor should move forward and rest on the first letter of the second word in line 4, "is." Your screen should now look like that shown below in Figure 14.

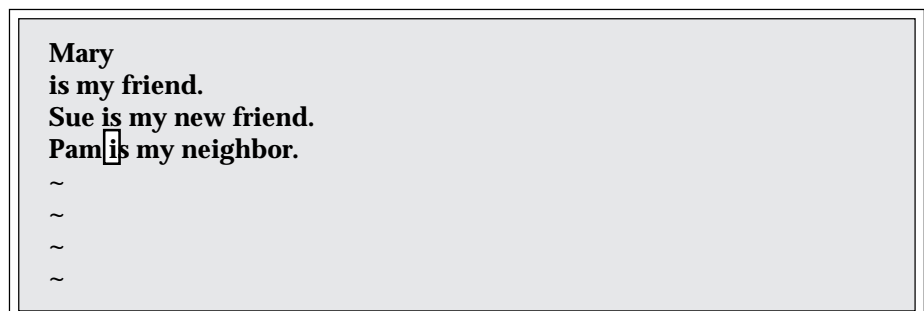


Figure 14.

• *Practice exercises for copying and moving text...*

### Copying and moving text

Use the following exercises to practice yanking text in the file "friends," as created in previous exercises. Before beginning these exercises, your cursor should be resting on the first letter of the first word in line 4, "Pam." (If you just finished the exercises on page 23, press "b" (lowercase b) to move your cursor back one word.) Your screen should look like that shown in Figure 15.

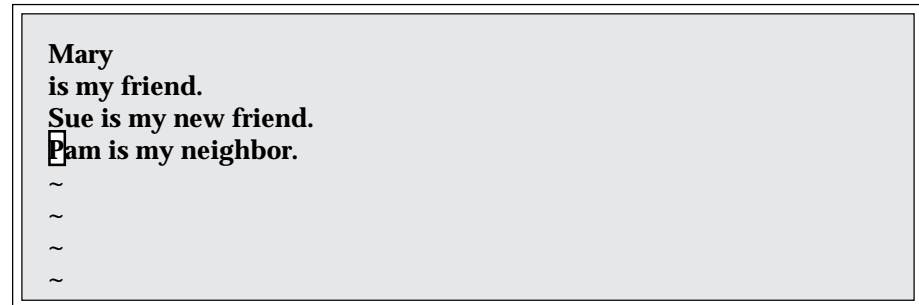


Figure 15.

*Step 1.* Press "yw" (lowercase y, lowercase w) to yank the word "Pam" from the file and save it in a temporary buffer. There is no change on the screen.

*Step 2.* Press "k" (lowercase k) to move the cursor up one line. The cursor should rest on the first letter of the first word in line 3, "Sue."

*Step 3.* Press "P" (uppercase P) to put the yanked word "Pam" to the left of the cursor in line 3. The cursor should rest on the space between the words "Pam" and "Sue" in line 3.

*Step 4.* Press "w" (lowercase w) to move the cursor to the beginning of the next word, "Sue."

*Step 5.* Press ""zy\$" (double quote, lowercase z, lowercase y, Shift-4) to yank line 3 from the file, starting at the cursor position, and save it in a temporary buffer called, "z." There is no change on the screen.

*Step 6.* Press "j" (lowercase j) to move the cursor down one line. The cursor should rest on the first letter of the second word in line 4, "is."

*Step 7.* Press ""zp" (double quote, lowercase z, lowercase p) to put the words "Sue is my friend." to the right of the cursor in line 4. The cursor should rest on the second letter of the word "iSue" in line 4.

*Step 8.* Press "u" (lowercase u) to undo the last command. The placed text is removed from line 4 and the cursor should rest on the first letter of the first word in line 4, "Pam."

*Step 9.* Press "Y" (uppercase Y) to yank line 4 from the file. There is no change on the screen.

*Step 10.* Press "2k" (2, lowercase k) to move the cursor up 2 lines. The cursor should rest on the first letter of the first word in line 2, "is."

*Step 11.* Press "p" (lowercase p) to place the text below line 2 and create a new line 3. The cursor should rest on the first letter of the first word in line 3, "Pam."

*Step 12.* Press "u" (lowercase u) to undo the last command. The newly placed text is removed from the file, and the cursor should rest on the first letter of line 2, "is."

*Step 13.* Press "P" (uppercase P) to place the text above line 2 and create a new line 2. The cursor should rest on the first letter of the first word in the new line 2, "Pam."

*Step 14.* Press "u" (lowercase u) to undo the last command. Line 2 is removed from the file and the cursor should rest on the first letter of the first word in line 1, "Mary."

Your screen should now look like that shown below in Figure 16.

```
Mary
is my friend.
Pam Sue is my new friend.
Pam is my neighbor.
~
~
~
~
```

Figure 16.

• *Practice exercise for joining lines of text...*

### Joining lines

Use the following exercise to practice joining lines in the file "friends," as created in previous exercises. Before beginning these exercises, your cursor should be resting anywhere in line 1.

*Step 1.* Press "J" (uppercase J) to join lines 1 and 2. The cursor will remain on line 1, which now reads, "Mary is my friend." Your file should look similar to that shown below in Figure 17.

```
Mary is my friend.
Pam Sue is my new friend.
Pam is my neighbor.
~
~
~
~
```

Figure 17.

• *Ending the exercises...*

### Ending the exercises

To quit vi without saving the file, type:

`:q!`

and press the **Return** key. You are then returned to the shell prompt (%), and the original version of the file "friends" remains.

For more practice with vi, you can again open the file and repeat the exercises, beginning on page 17. To exit from Astro, type "exit" at the shell prompt (%) and press the **Return** key.