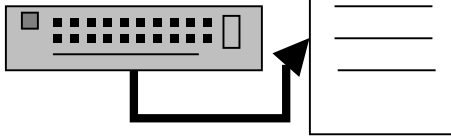


# The *vi* editor

## Insert mode



All keys except `ESC` insert text.

`esc`

## Move mode

modify

`c` `w`      `c` `c`  
`i`              `o`  
`a`              `R`  
`I`              `C`  
`O`              `n` `c` `c`

`/` *search text* `rtn`

`?` *search text* `rtn`

find

`n`, `N` Re-search

`.` Repeat previous command      `u` Undo

delete

`x`   `X`      `d` `w`  
`D`      `d` `d`

`n` `y` `y` Copy *n* lines to buffer

`p`, `P` Paste below or above

`J` Join lines

`Ctrl-g` show line number

`Z` `Z` same as `:` `w` `q` `rtn`

move

`w`   `W`   `b`   `Ctrl-f`  
`B`   `e`   `E`   `Ctrl-b`

`:` Go to command mode

`→` `←` `↓` `↑`      `n` `G`

## Command mode `ESC` cancels

`n,md` == delete lines *n-m*

`n,ms` /<target text>/<new text>/g

== substitute text (g=all occurrences) on lines from *n* to *m*

`w` *file*, `w!` , `q` , `q!` , `wq`, write *file*, quit (!=force)

`rtn`

# UNIX basics

To get a command line interface on Irix, use "Open UNIX Shell" from the Toolbox.

## Commands you should know:

**ls** == list files ( use *ls -l* for long format)

**grep** *string file* == find a character string within a file.

**mkdir** *directory* == make a new directory (use **rmdir** to remove it)

**cd** == change your working directory (use *cd ..* to go up one directory, *cd ../..* to go up two directories.)

**pwd** == print working directory

**more** *file* == view a file page by page

**vi** *file* == edit a file. (see vi page)

# UNIX basics

**rm** *file* == delete *file*

**mv** *file directory/newname* == rename (move)  
*file*

Redirecting input/output:

**sort** < *inputfile* > *outputfile*

Running a command in the background:

**sort** < *inputfile* > *outputfile* & (or use ctrl-z)

Piping one command to another:

**sort** *inputfile* | *more*

Setting the X-windows display:

on the local machine: **xhost +**

on the remote machine:

**setenv DISPLAY** *localmachine.rpi.edu*

Getting more information on UNIX commands:

**man** *command* == read the manual

# UNIX basics: **telnet**, **ssh** and **ftp**

To connect the terminal window to a remote machine: Without encryption:

**telnet** *IP\_address*, then login using username and password

With encryption: **ssh** *username@IP\_address*  
then type password

IP addresses have the form *A.B.C.D* where *A*, *B*, *C* and *D* are either integers (0-255) or names (such as bioinf45.bio.rpi.edu).

To transfer files between machines:

**ftp** *IP\_address* , then login using username and password

FTP commands

**get file**, **put file** == download or upload

**mput**, **mget** == download, upload multiple files

**cd directory** == change directory remotely

**ls** == list files remotely

**bin**, **asc** == set file transfer to binary, ascii

**bye** == disconnect

# Pseudocode

<b>constant</b>	real (1.23), integer (4), character("4"), or Boolean
<b>variable</b>	a name, assignable to a constant.
<b>statement</b>	{<assignment> <conditional statement> <loop>} [<statement>]
<b>assignment</b>	$a \leftarrow b$ or $a = b$ , where $a$ is a variable and $b$ is a variable, constant, or a formula using <i>binary and unitary operators</i> .
<b>condition</b>	[NOT] $a$ <boolean operator> $b$ [{ <b>AND</b>   <b>OR</b> } <condition> Evaluates to <i>true (1)</i> or <i>false (0)</i> .
<b>conditional statement</b>	<b>if</b> (<boolean>) <b>then</b> <statement> [ <b>else</b> <statement>] <b>endif</b>
<b>loop</b>	{<for loop> <while loop>}
<b>for loop</b>	<b>for</b> ( $i$ <b>from</b> $a$ <b>to</b> $b$ ) <b>do</b> <statement> <b>enddo</b>
<b>while loop</b>	<b>while</b> (<condition>) <b>do</b> <statement> <b>enddo</b>
<b>array</b>	$a[i]$ A numbered set of variables.
<b>2D array</b>	$a[i,j]$ Range in $i$ : $a[i..k,j] = (a[i,j].a[i+1,j],\dots,a[k,j])$
<b>binary operators</b>	+ - * / ^ % (add,subtract,multiply,divide,modulus)
<b>unitary operators</b>	<b>sqrt()</b> , <b>log()</b> , <b>exp()</b> , <b>sin()</b> , <b>cos()</b> , <b>tan()</b> , <b>len()</b> ,etc. (built-in functions)
<b>Boolean operators</b>	> < == (greater than, less than, equal to)
<b>read</b>	<b>read file</b> <variables> , assigns variables from <i>file</i> .
<b>write</b>	<b>write file</b> <variables>, output to <i>file</i> . (Use * for standard input/output)
<b>return</b>	<b>return</b> $x$ When used in a function, sets the returned value of the function to $x$ .
<b>function</b>	$a \leftarrow F(x,y)$ { <statements> <b>return</b> $x$ }

A Pseudocode program for calculating the distance between two points, **a** and **b**:

```
a[1..3] = (1.1, 2.2, 3.3)
b[1..3] = (4.4, 5.5, 6.6)
sum = 0
for i from 1 to 3 do
    x <-- (a[i] - b[i])^2
    sum <-- sum + x
enddo
x = sqrt(sum)
write *, "The distance is ", x
```

A Pseudocode program using a function. Converts a character array (string) to an integer array:

```
// 1="A", 2="C",3="G",4="T"
base[1..4] = ("A","C","G","T")

read *, sequence
for i from 1 to len(sequence) do
    s[i] = getbase(sequence[i])
enddo

write *, "The sequence is ",
sequence[1..len(sequence)]

write *, "The integer sequence is ",
s[1..len(sequence)]

getbase(a) {
    i <-- 1
    while (NOT base[i] == a) do
        i <-- i + 1
        if (i > 4) return 0
    enddo
    return i
}
```