sed '2,$ d' file.txt

 'd' parameter basically tells [sed] to delete all the records from display output

last line is represented by $ symbol

sed -n '$ p' test

'$ p' basically prints (p for print) the last line in standard output screen. '-n' switch takes [sed] to silent mode so that [sed] does not print anything

sed –n '4 p' test

print the 4th line

sed '1 d' file.txt

delete the first line

sed –i '5,7 d' file.txt

delete line 5 to line 7 from the file file.txt

sed –i '96,100 d' file.txt

remove all the lines from 96 to 100 like below

tt=`wc -l file.txt | cut -f1 -d' '`;sed –i "`expr $tt - 4`,$tt d" test

first one (before the semi-colon) calculates the total number of lines present in the file and stores it in a variable called “tt”. The second command (after the semi-colon), uses the variable and works in the exact way as shows in the previous example.

echo "unix" | rev

to reverse a string

**AWK**

Awk is one of the most powerful tools in Unix used for processing the rows and columns in a file.Awk has built in string functions and associative arrays. Awk supports most of the operators, conditional blocks, and loops available in C language

awk 'BEGIN {start\_action} {action} END {stop\_action}' filename

awk '{print $1}' input\_file

print the first column in each row

To print the 4th and 6th columns in a file use awk '{print $4,$5}' input\_file

awk 'BEGIN {sum=0} {sum=sum+$5} END {print sum}' input\_file

sum of the values in the 5th column. This value is printed in the End block.

http://www.folkstalk.com/2011/12/good-examples-of-awk-command-in-unix.html