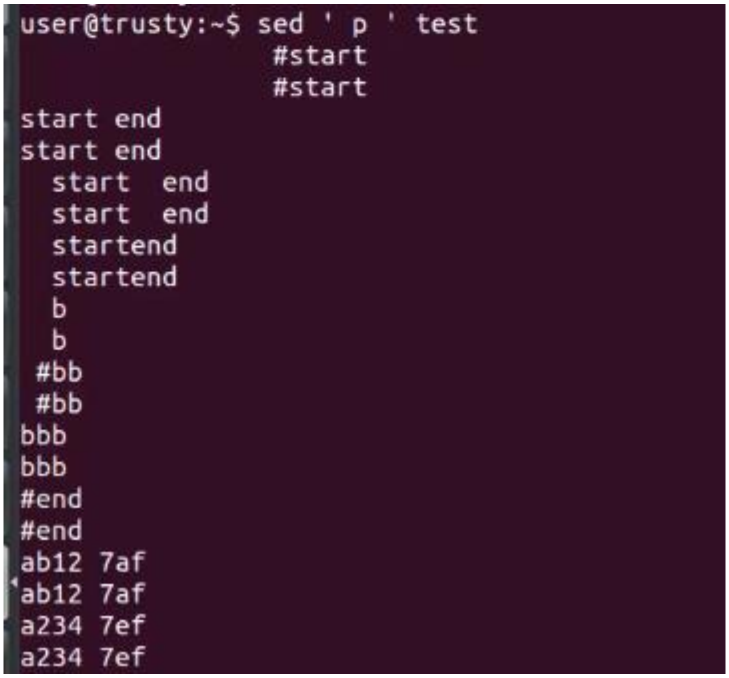
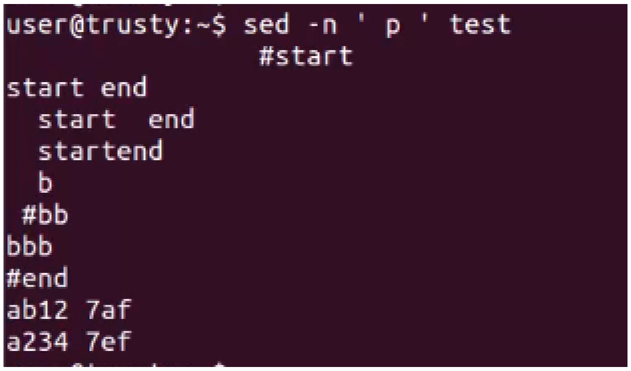
SED BASICS

SED –P PRINTS EVERY LINE TWICE AS IT USES STD OUTPUT



```
user@trusty:~$ sed ' p ' test
                #start
                #start
start end
start end
  start  end
  start  end
  startend
  startend
  b
  b
 #bb
 #bb
bbb
bbb
#end
#end
ab12 7af
ab12 7af
a234 7ef
a234 7ef
```

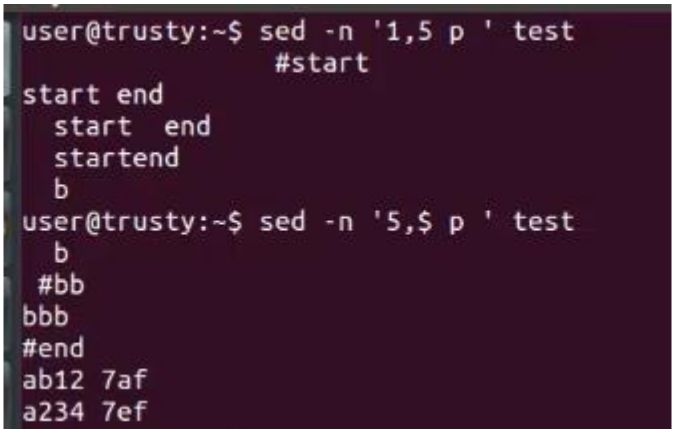
TO AVOID THESE WE CAN USE –n FLAG WITH THE ABOVE COMMAND WHICH SUPPRESS STD OPT



```
user@trusty:~$ sed -n ' p ' test
                #start
start end
  start  end
  startend
  b
 #bb
bbb
#end
ab12 7af
a234 7ef
```

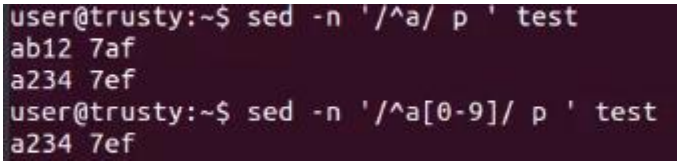
TO SEARCH 1 TO 5 LINES

TO SEE 5 TO LAST LINE USE DOLLAR

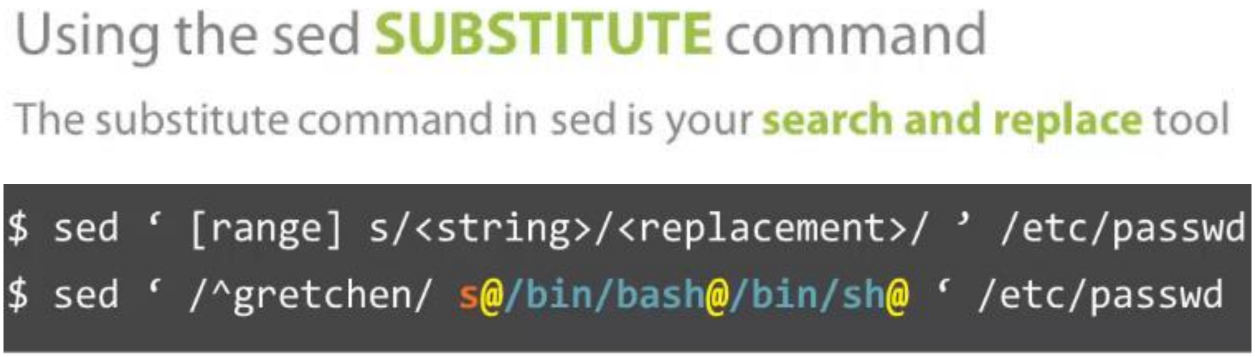


```
user@trusty:~$ sed -n '1,5 p ' test
                #start
start end
  start  end
  startend
  b
user@trusty:~$ sed -n '5,$ p ' test
  b
 #bb
bbb
#end
ab12 7af
a234 7ef
```

TO USE REGEX PUT IT INSIDE FORWARD SLASHES



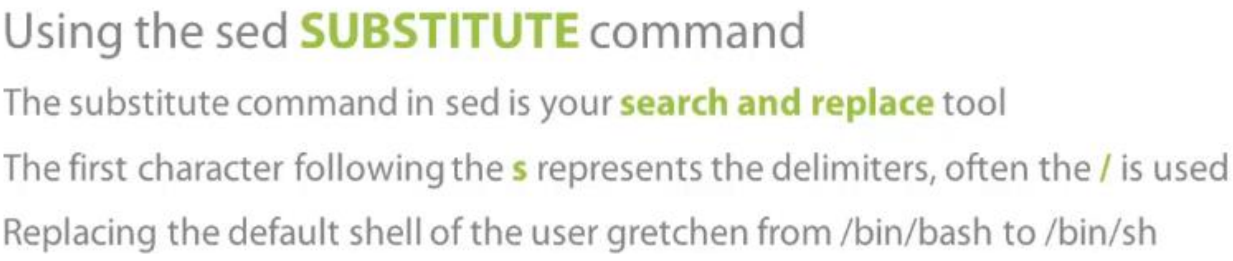
```
user@trusty:~$ sed -n '/^a/ p ' test
ab12 7af
a234 7ef
user@trusty:~$ sed -n '/^a[0-9]/ p ' test
a234 7ef
```



Using the sed **SUBSTITUTE** command

The substitute command in sed is your **search and replace** tool

```
$ sed ‘ [range] s/<string>/<replacement>/ ’ /etc/passwd
$ sed ‘ /^gretchen/ s@/bin/bash@/bin/sh@ ‘ /etc/passwd
```



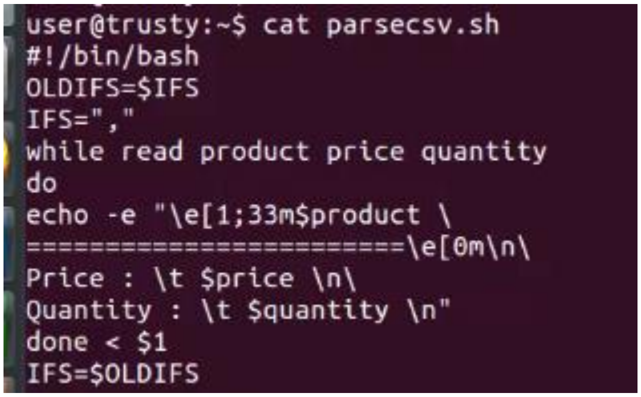
Using the sed **SUBSTITUTE** command

The substitute command in sed is your **search and replace** tool

The first character following the **s** represents the delimiters, often the **/** is used

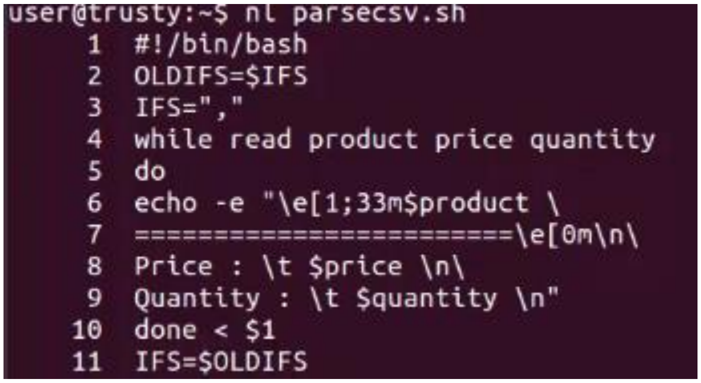
Replacing the default shell of the user gretchen from /bin/bash to /bin/sh

Now the task is to do some tabs within do done block in script below

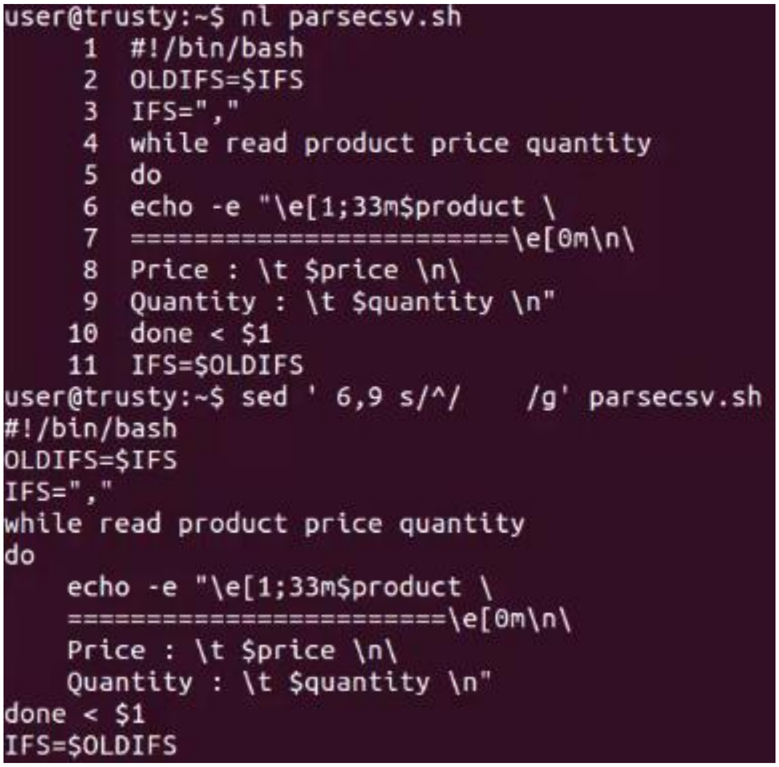


```
user@trusty:~$ cat parsecsv.sh
#!/bin/bash
OLDIFS=$IFS
IFS=","
while read product price quantity
do
echo -e "\e[1;33m$product \
=========================\e[0m\n\
Price : \t $price \n\
Quantity : \t $quantity \n"
done < $1
IFS=$OLDIFS
```

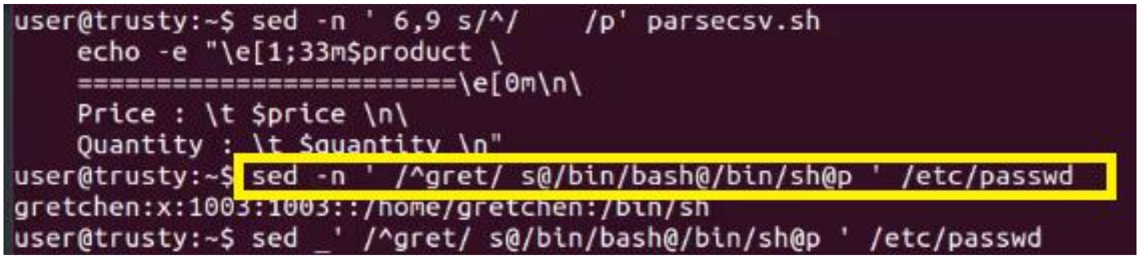
nl is used for line no



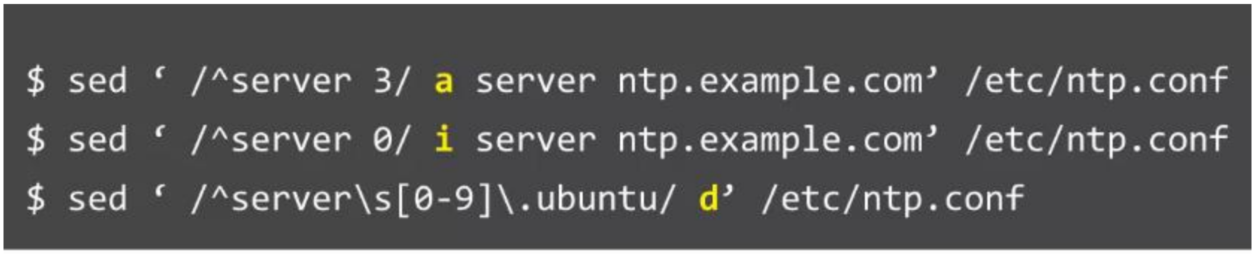
```
user@trusty:~$ nl parsecsv.sh
     1  #!/bin/bash
     2  OLDIFS=$IFS
     3  IFS=","
     4  while read product price quantity
     5  do
     6  echo -e "\e[1;33m$product \
     7  =========================\e[0m\n\
     8  Price : \t $price \n\
     9  Quantity : \t $quantity \n"
    10  done < $1
    11  IFS=$OLDIFS
```



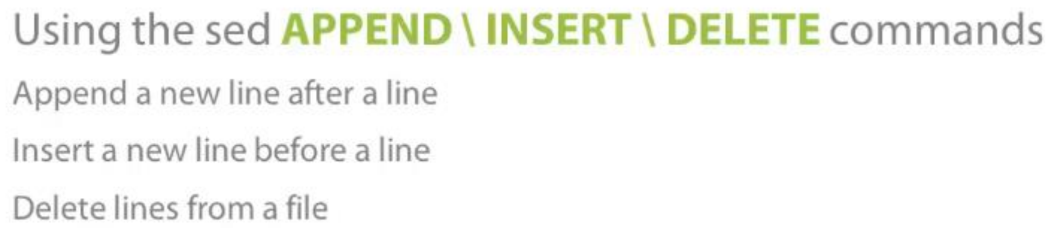
```
user@trusty:~$ nl parsecsv.sh
     1  #!/bin/bash
     2  OLDIFS=$IFS
     3  IFS=","
     4  while read product price quantity
     5  do
     6  echo -e "\e[1;33m$product \
     7  =========================\e[0m\n\
     8  Price : \t $price \n\
     9  Quantity : \t $quantity \n"
    10  done < $1
    11  IFS=$OLDIFS
user@trusty:~$ sed ' 6,9 s/^/    /g' parsecsv.sh
#!/bin/bash
OLDIFS=$IFS
IFS=","
while read product price quantity
do
    echo -e "\e[1;33m$product \
    =========================\e[0m\n\
    Price : \t $price \n\
    Quantity : \t $quantity \n"
done < $1
IFS=$OLDIFS
```



```
user@trusty:~$ sed -n ' 6,9 s/^/    /p' parsecsv.sh
    echo -e "\e[1;33m$product \
    ========================\e[0m\n\
    Price : \t $price \n\
    Quantity : \t $quantity \n"
user@trusty:~$ sed -n ' /^gret/ s@/bin/bash@/bin/sh@p ' /etc/passwd
gretchen:x:1003:1003::/home/gretchen:/bin/sh
user@trusty:~$ sed _' /^gret/ s@/bin/bash@/bin/sh@p ' /etc/passwd
```



```
$ sed ‘ /^server 3/ a server ntp.example.com’ /etc/ntp.conf
$ sed ‘ /^server 0/ i server ntp.example.com’ /etc/ntp.conf
$ sed ‘ /^server\s[0-9]\.ubuntu/ d’ /etc/ntp.conf
```



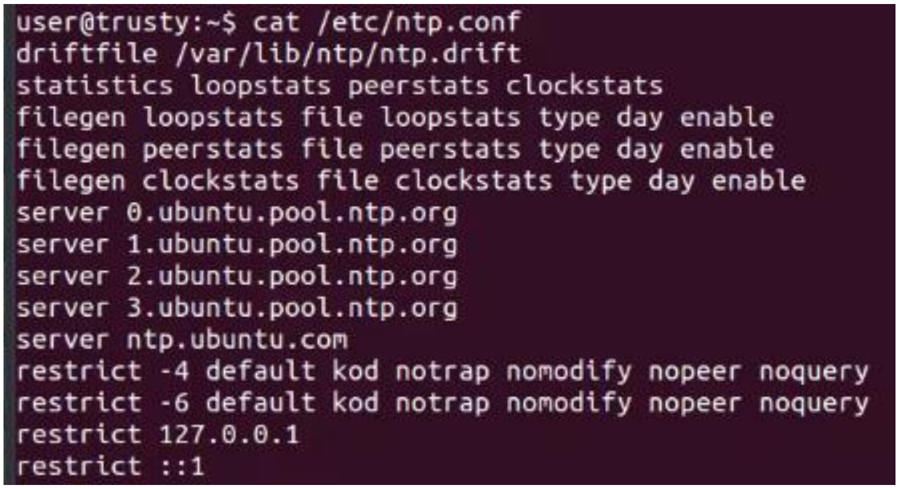
Using the sed **APPEND \ INSERT \ DELETE** commands

Append a new line after a line

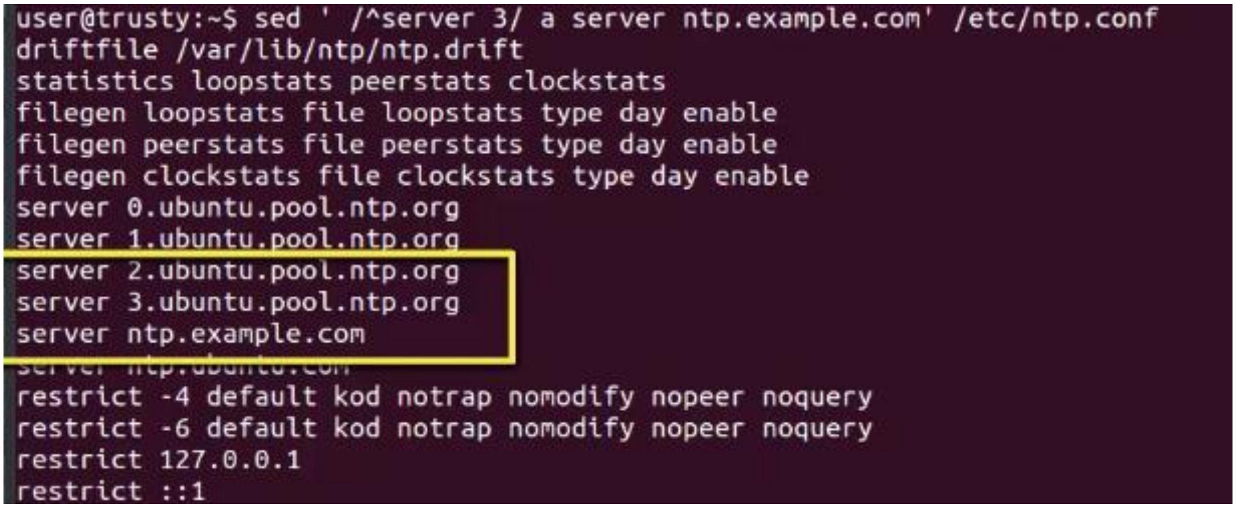
Insert a new line before a line

Delete lines from a file

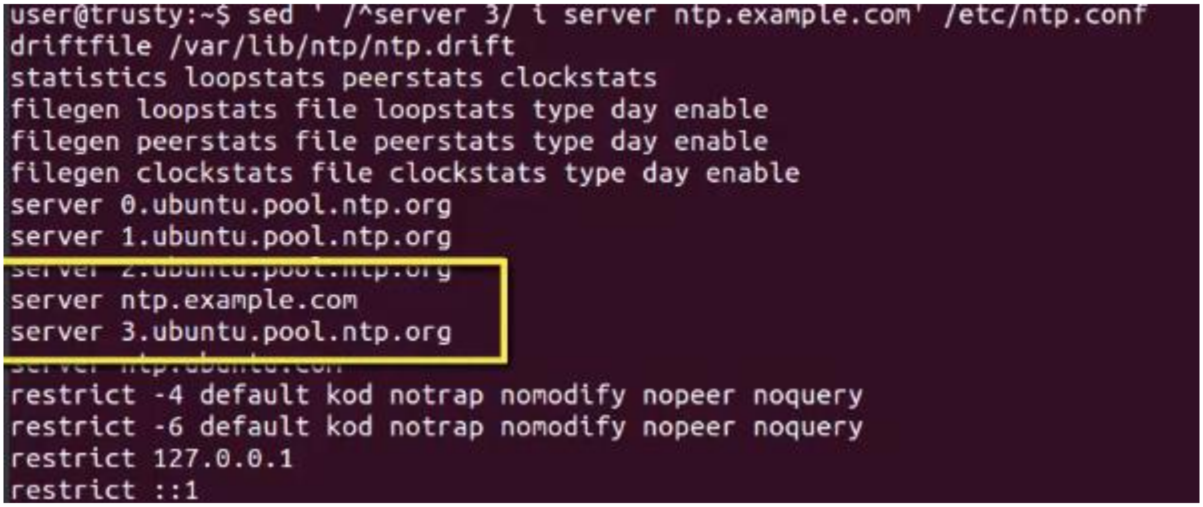
In 3 we are matching line starting with server and followed by a single space and a number and .ubuntu \ is used for escaping dot



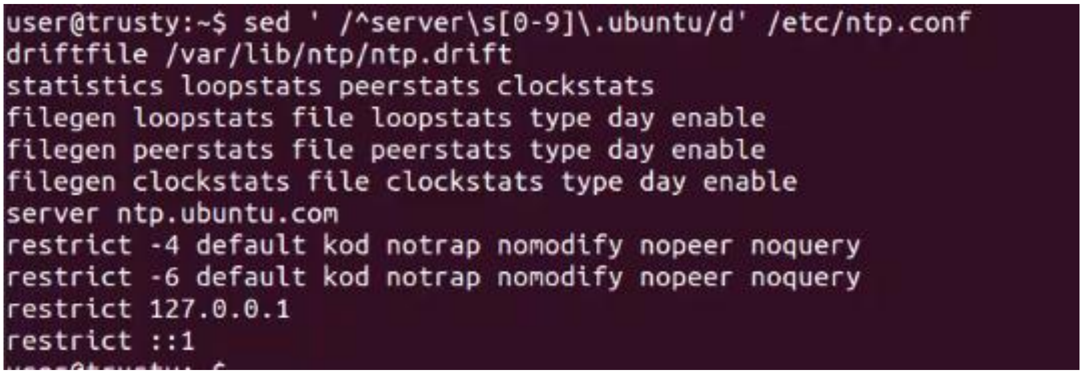
```
user@trusty:~$ cat /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
server ntp.ubuntu.com
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict ::1
```



```
user@trusty:~$ sed ' /^server 3/ a server ntp.example.com' /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
server ntp.example.com
server ntp.ubuntu.com
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict ::1
```



```
user@trusty:~$ sed ' /^server 3/ i server ntp.example.com' /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server ntp.example.com
server 3.ubuntu.pool.ntp.org
server ntp.ubuntu.com
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict ::1
```



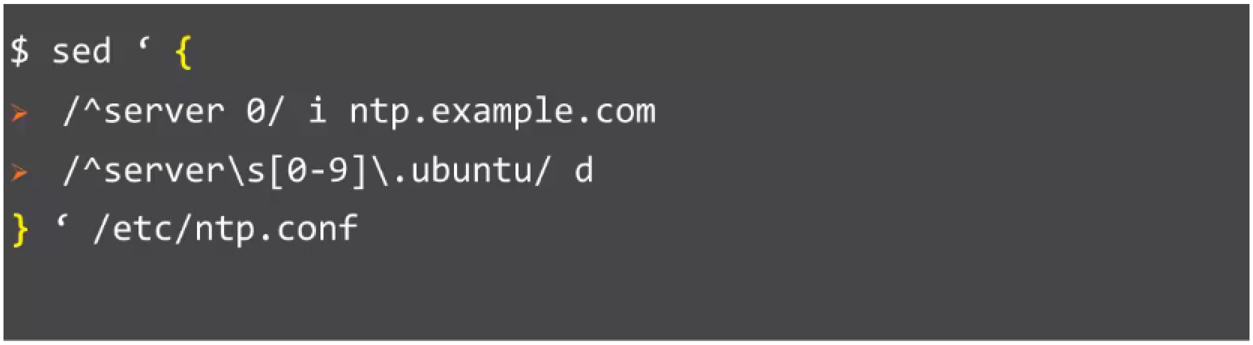
```
user@trusty:~$ sed ' /^server\s[0-9]\.ubuntu/d' /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
server ntp.ubuntu.com
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict ::1
user@trusty:~$
```

Multiline sed expression

2 ways

1.braces

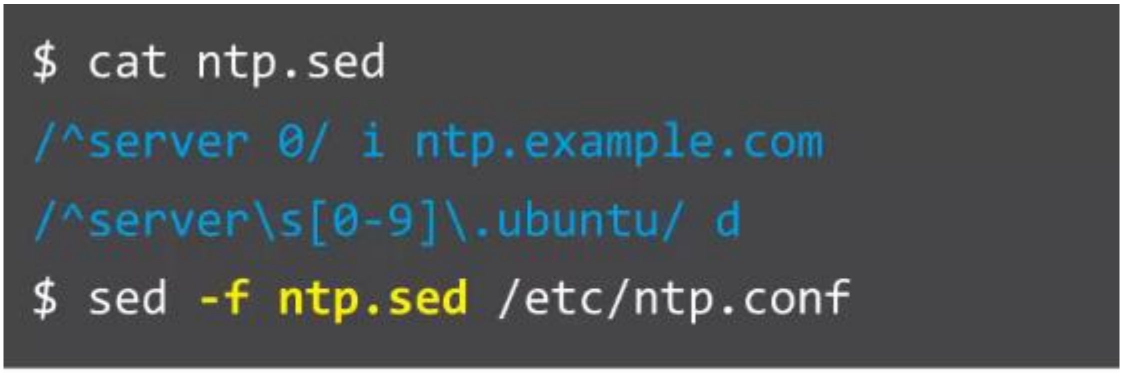
2.file



```
$ sed ‘ {
> /^server 0/ i ntp.example.com
> /^server\s[0-9]\.ubuntu/ d
} ‘ /etc/ntp.conf
```

Multiple sed **expressions**

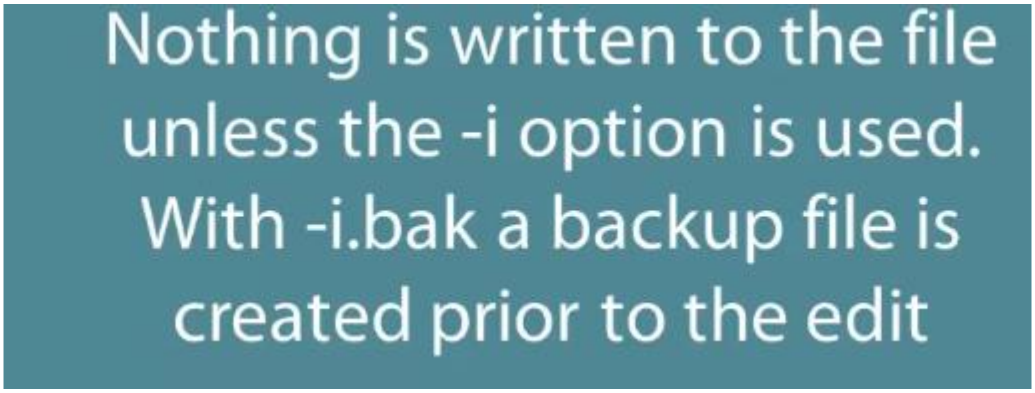
Multiple expressions can be written on the command line by including brace brackets within the quoted sed instructions



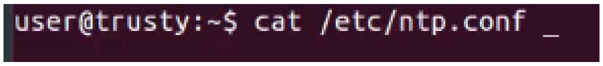
```
$ cat ntp.sed
/^server 0/ i ntp.example.com
/^server\s[0-9]\.ubuntu/ d
$ sed -f ntp.sed /etc/ntp.conf
```

For **code-reuse** implement sed files

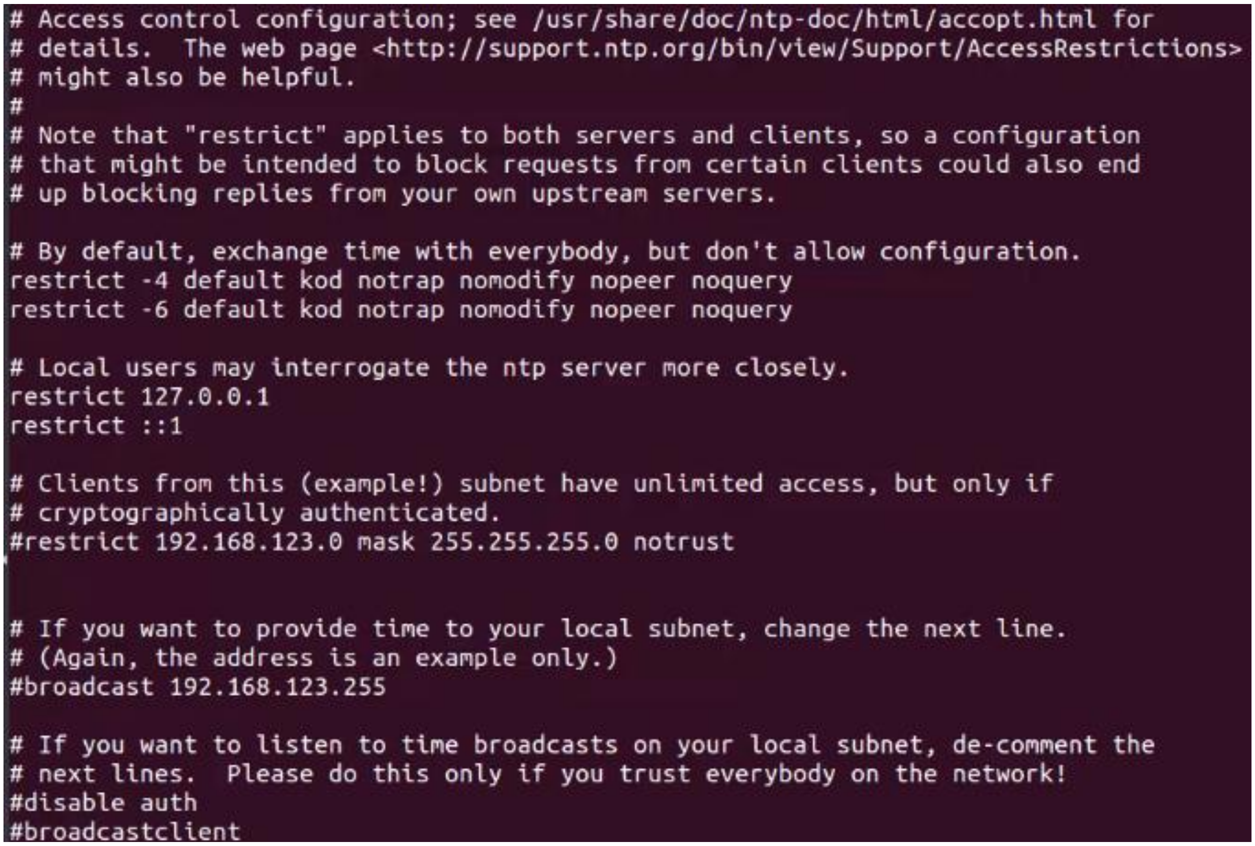
The sed file can be referenced with the **-f** option



Nothing is written to the file unless the -i option is used. With -i.bak a backup file is created prior to the edit



```
user@trusty:~$ cat /etc/ntp.conf _
```



```
# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
# details.  The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1

# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust


# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255

# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines.  Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient
```