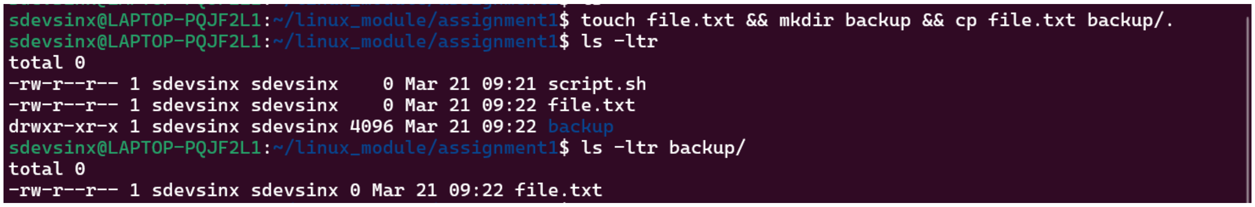
Q1 : How do you use the "cp" command to copy a file named "file.txt" from the current directory to a directory named "backup"?

```
$ touch file.txt && mkdir backup && cp file.txt backup/.
```



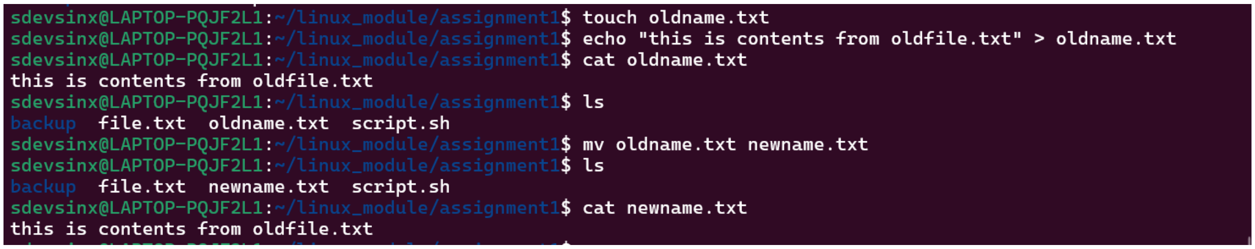
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch file.txt && mkdir backup && cp file.txt backup/.
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -ltr
total 0
-rw-r--r-- 1 sdevsinx sdevsinx    0 Mar 21 09:21 script.sh
-rw-r--r-- 1 sdevsinx sdevsinx    0 Mar 21 09:22 file.txt
drwxr-xr-x 1 sdevsinx sdevsinx 4096 Mar 21 09:22 backup
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -ltr backup/
total 0
-rw-r--r-- 1 sdevsinx sdevsinx 0 Mar 21 09:22 file.txt
```

Q2 : What is the difference between the "rm" and "rm -r" commands in Linux?

rm	rm -r
Used to delete files in one level of directory	Used to delete in directory & its sub-directories
Cannot delete directories	Can delete directories
Cannot traverse through sub-directories	Can traverse through sub-directories to delete them

Q3 : How do you use the "mv" command to rename a file named "oldname.txt" to "newname.txt"?

```
$ touch oldname.txt
$ echo “this is contents from oldfile.txt” > oldname.txt
$ cat oldname.txt
$ ls
$ mv oldname.txt newname.txt
$ ls
$ cat newname.txt
```



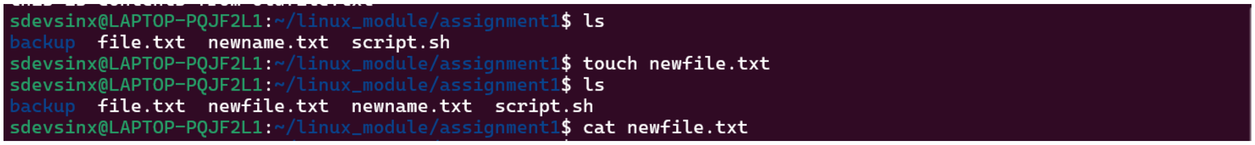
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch oldname.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ echo "this is contents from oldfile.txt" > oldname.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat oldname.txt
this is contents from oldfile.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  oldname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ mv oldname.txt newname.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat newname.txt
this is contents from oldfile.txt
```

Q4 : What does the "pwd" command do in Linux?

A: pwd command prints current working directory on the screen

Q5: How do you create a new empty file named "newfile.txt" in the current directory using the command line?

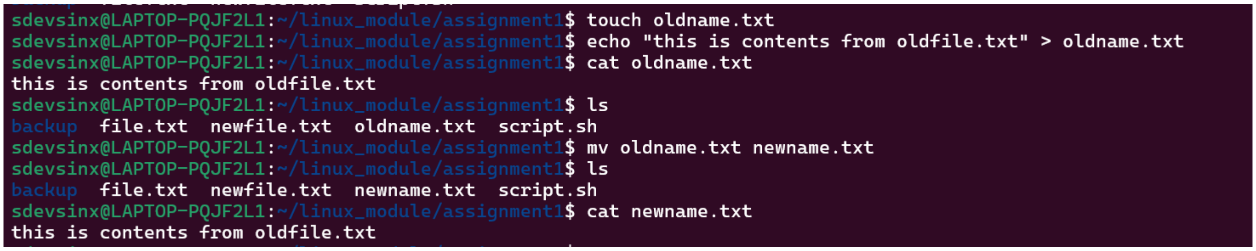
```
$ ls
$ touch newfile.txt
$ ls
$ cat newfile.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch newfile.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  newfile.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat newfile.txt
```

Q6 : How do you rename a file named "oldname.txt" to "newname.txt" using the command line?

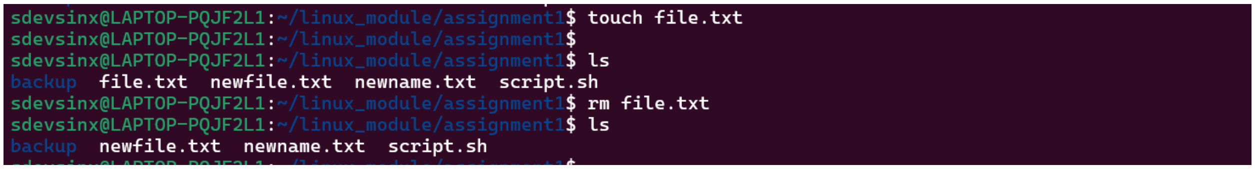
```
$ touch oldname.txt
$ echo “this is contents from oldfile.txt” > oldname.txt
$ cat oldname.txt
$ ls
$ mv oldname.txt newname.txt
$ ls
$ cat newname.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch oldname.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ echo "this is contents from oldfile.txt" > oldname.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat oldname.txt
this is contents from oldfile.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  newfile.txt  oldname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ mv oldname.txt newname.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  newfile.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat newname.txt
this is contents from oldfile.txt
```

Q7 : How do you remove a file named "file.txt" from the current directory using the command line?

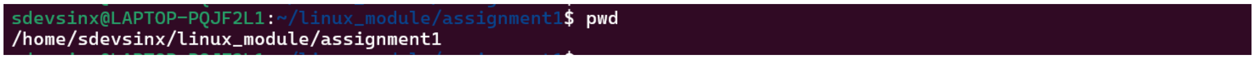
```
$ touch file.txt
$ ls
$ rm file.txt
$ ls
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  file.txt  newfile.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ rm file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  newfile.txt  newname.txt  script.sh
```

Q8 : Use a command to show the current working directory

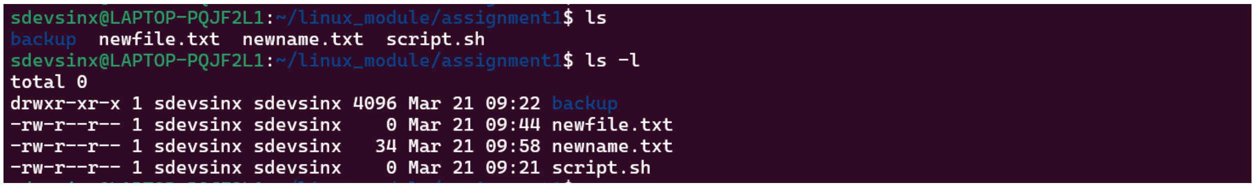
```
$ pwd
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ pwd
/home/sdevsinx/linux_module/assignment1
```

Q9 : List the directory contents in the short and long format

```
$ ls
$ ls -l
```



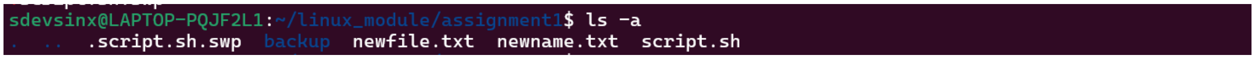
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  newfile.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l
total 0
drwxr-xr-x 1 sdevsinx sdevsinx 4096 Mar 21 09:22 backup
-rw-r--r-- 1 sdevsinx sdevsinx    0 Mar 21 09:44 newfile.txt
-rw-r--r-- 1 sdevsinx sdevsinx   34 Mar 21 09:58 newname.txt
-rw-r--r-- 1 sdevsinx sdevsinx    0 Mar 21 09:21 script.sh
```

Q10 : Explore attributes given in long format e.g. file type, file permissions, file size, file owner etc.

```
|indicates it is directory ‘d’
||owner permissions
||  |group permissions
||  |  |permissions for all other users
||  |  |   |number of links
||  |  |   | |owner of directory
||  |  |   | |        |usergroup
||  |  |   | |        |        |size in bytes
||  |  |   | |        |        |    |Date & Time
||  |  |   | |        |        |    |            |Directory name
drwxr-xr-x 1 sdevsinx sdevsinx 4096 Mar 21 09:22 backup
```

Q11 : List all files along with hidden files in the current working directory.

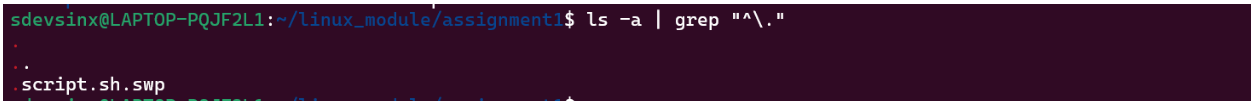
```
$ ls -a
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -a
.  ..  .script.sh.swp  backup  newfile.txt  newname.txt  script.sh
```

Q12 : list only hidden files in the directory

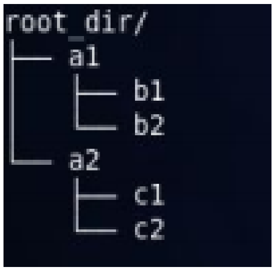
```
$ ls -a | grep “^\.”
```



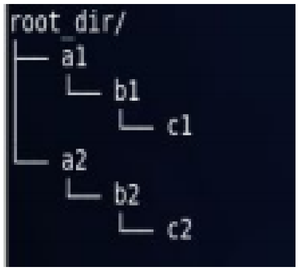
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -a | grep "^\."
.
..
.script.sh.swp
```

Q13. Make a directory and name it as **cdac-dir** and change the current working directory to the new directory.(Hint : use **mkdir,cd** commands). 3. Create following nested directories inside the current directory by invoking a single command for only one time.

Note : here root_dir is the current directory.



```
root_dir/
├── a1
│   ├── b1
│   └── b2
└── a2
    ├── c1
    └── c2
```

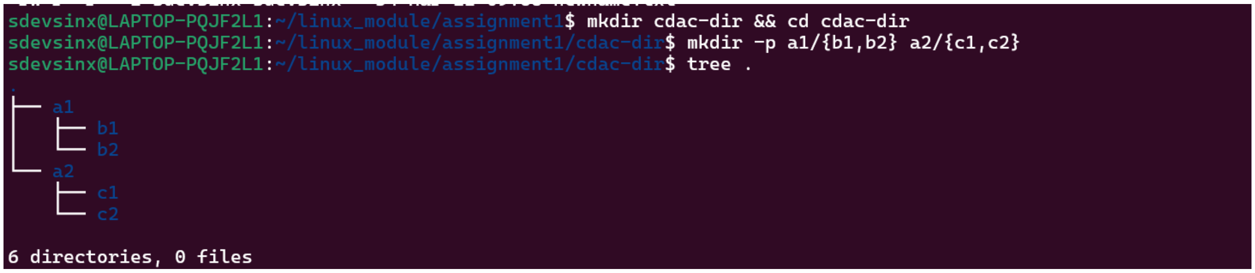


```
root_dir/
├── a1
│   └── b1
│       └── c1
└── a2
    └── b2
        └── c2
```

Directory structure 1 Directory structure 2

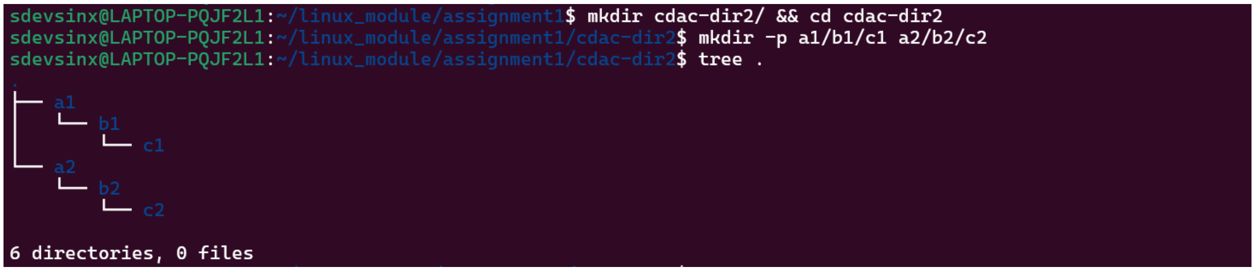
14.(Hint : explore the man page of **mkdir**).

```
$ mkdir cdac-dir && cd cdac-dir
$ mkdir -p a1/{b1,b2} a2/{c1,c2}
$ tree .
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ mkdir cdac-dir && cd cdac-dir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/cdac-dir$ mkdir -p a1/{b1,b2} a2/{c1,c2}
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/cdac-dir$ tree .
.
├── a1
│   ├── b1
│   └── b2
└── a2
    ├── c1
    └── c2

6 directories, 0 files
```

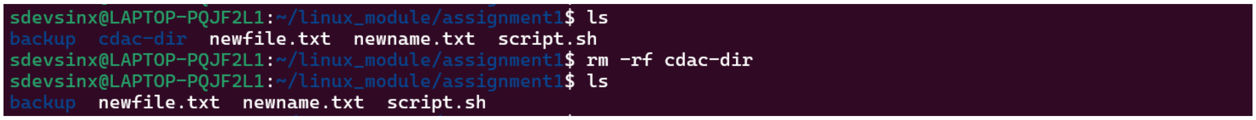
```
$ mkdir cdac-dir && cd cdac-dir
$ mkdir -p a1/b1/c1 a2/b2/c2
$ tree .
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ mkdir cdac-dir2/ && cd cdac-dir2
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/cdac-dir2$ mkdir -p a1/b1/c1 a2/b2/c2
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/cdac-dir2$ tree .
.
├── a1
│   └── b1
│       └── c1
└── a2
    └── b2
        └── c2

6 directories, 0 files
```

Q15 : List the directories(folders), then remove the cdac-dir directory and list the folders again to show that it is no longer present.(Hint : use rm, ls command

```
$ ls
$ rm -rf cdac-dir
$ ls
```

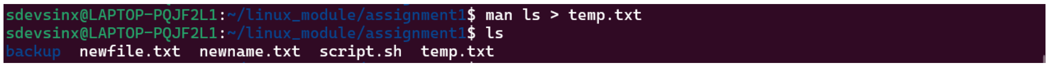


```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  cdac-dir  newfile.txt  newname.txt  script.sh
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ rm -rf cdac-dir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  newfile.txt  newname.txt  script.sh
```

16.Question-2.

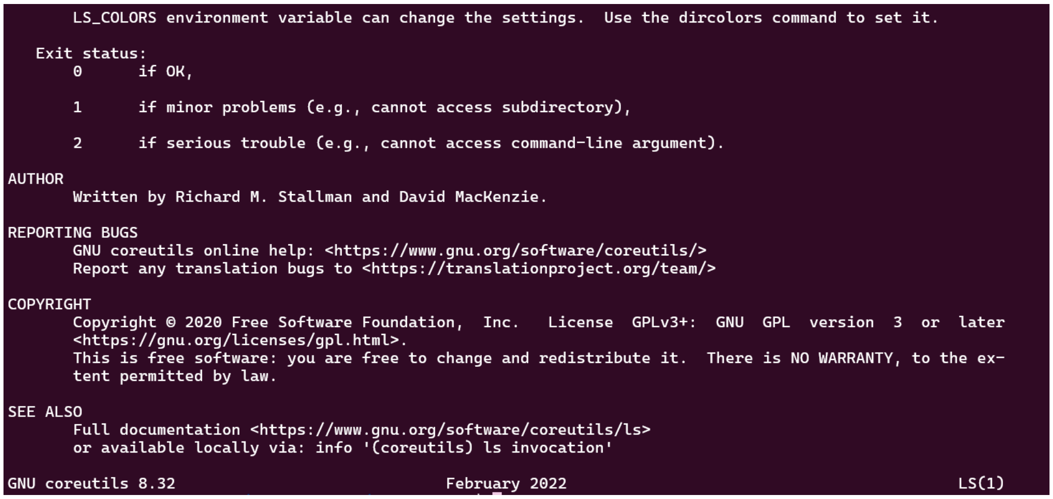
Q17: Display the man-page for ls, but redirect the output into temp.txt, then use the cat, less, and more commands to display the new file.

```
$ man ls > temp.txt
$ ls
```



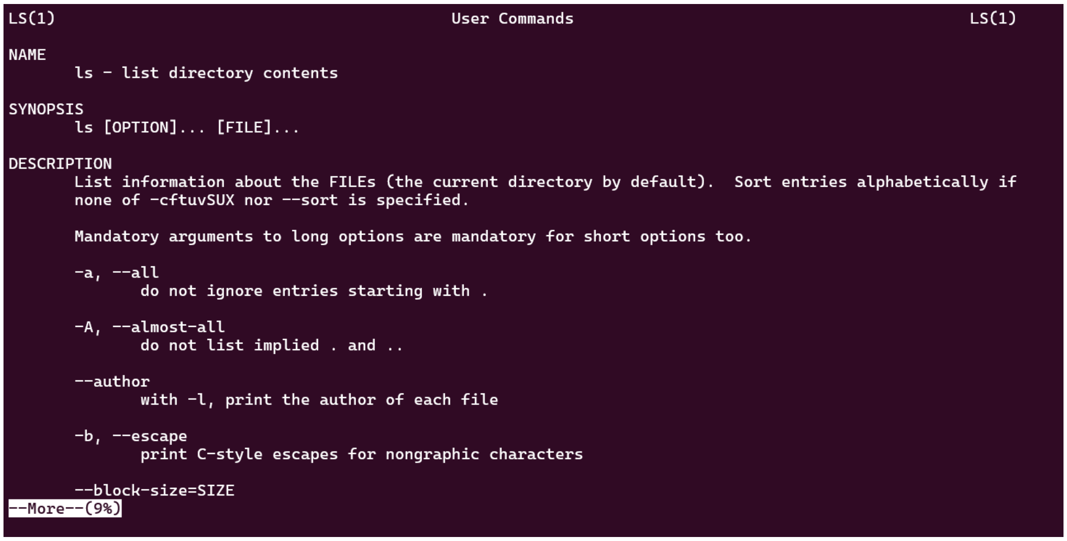
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ man ls > temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  newfile.txt  newname.txt  script.sh  temp.txt
```

```
$ cat temp.txt
```



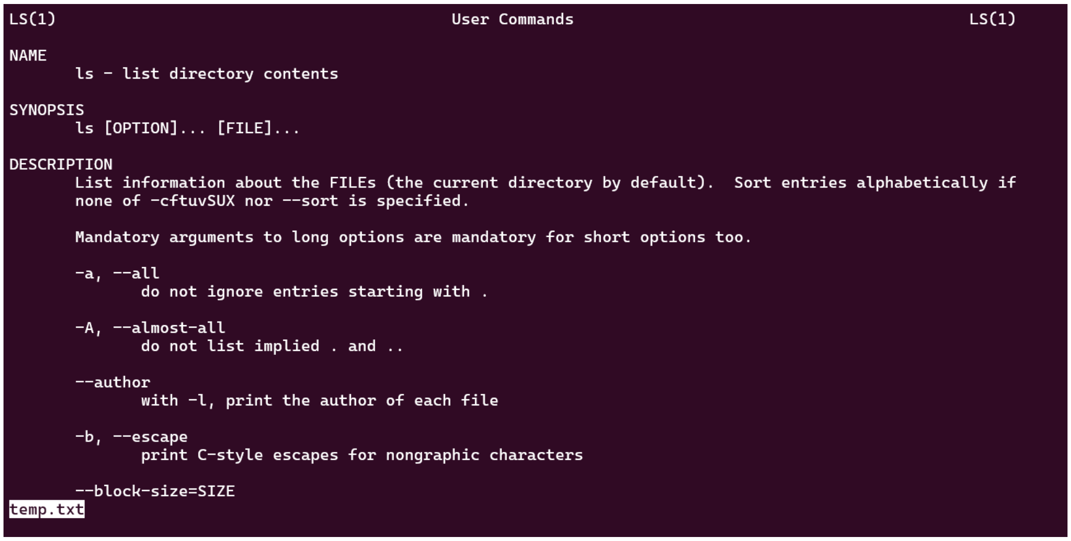
```
       LS_COLORS environment variable can change the settings.  Use the dircolors command to set it.

   Exit status:
       0      if OK,

       1      if minor problems (e.g., cannot access subdirectory),

       2      if serious trouble (e.g., cannot access command-line argument).

AUTHOR
       Written by Richard M. Stallman and David MacKenzie.

REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report any translation bugs to <https://translationproject.org/team/>

COPYRIGHT
       Copyright © 2020 Free Software Foundation,  Inc.   License  GPLv3+:  GNU  GPL  version  3  or  later
       <https://gnu.org/licenses/gpl.html>.
       This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the ex‐
       tent permitted by law.

SEE ALSO
       Full documentation <https://www.gnu.org/software/coreutils/ls>
       or available locally via: info '(coreutils) ls invocation'

GNU coreutils 8.32                              February 2022                                     LS(1)
```

```
$ more temp.txt
```



```
LS(1)                                       User Commands                                       LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alphabetically if
       none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
--More--(9%)
```

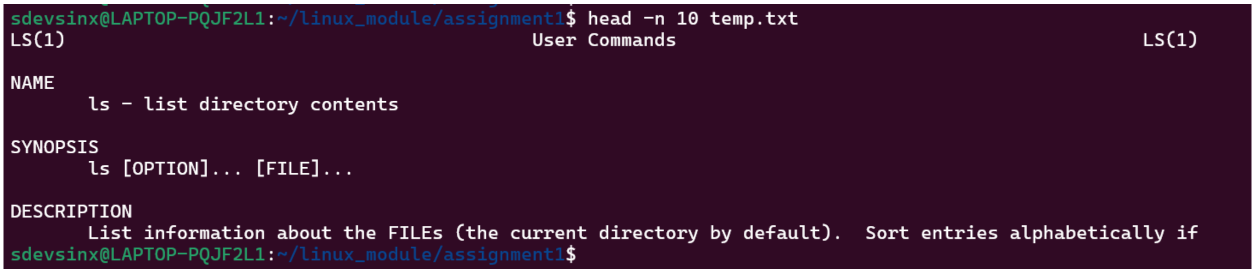
```
$ less temp.txt
```



```
LS(1)                                  User Commands                                  LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alphabetically if
       none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
temp.txt
```

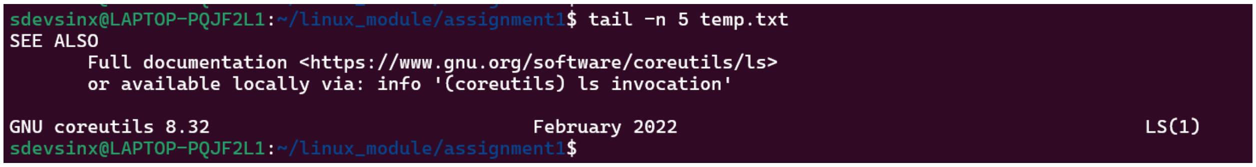
Q18 : Display the initial 10 lines and final 5 lines of temp.txt with the obvious Linux commands.(Hint: use head and tail commands).

```
$ head -n 10 temp.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ head -n 10 temp.txt
LS(1)                                  User Commands                                  LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alphabetically if
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

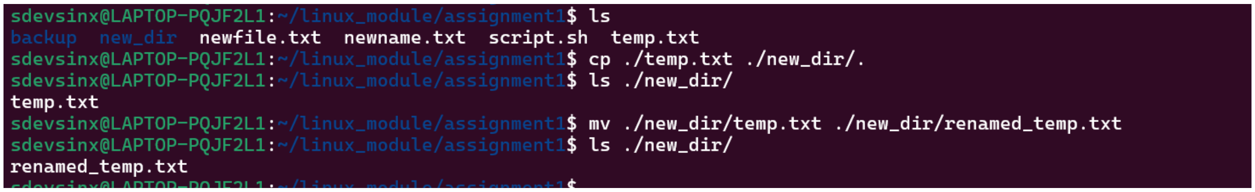
```
$ tail -n 5 temp.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ tail -n 5 temp.txt
SEE ALSO
       Full documentation <https://www.gnu.org/software/coreutils/ls>
       or available locally via: info '(coreutils) ls invocation'

GNU coreutils 8.32                     February 2022                                  LS(1)
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q19 : Copy temp.txt to another directory and rename it there.
(Hint: use cp to copy and mv command to rename).

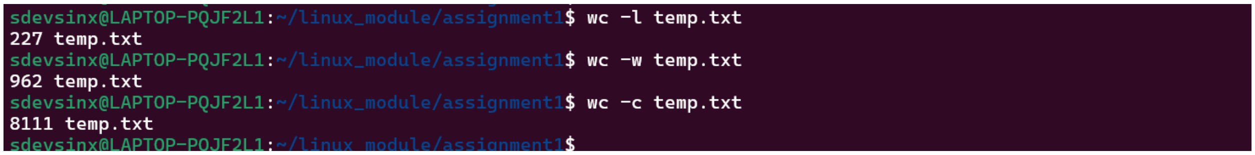
```
$ ls
$ cp ./temp.txt ./new_dir/.
$ ls ./new_dir/
$ mv ./new_dir/temp.txt ./new_dir/renamed_temp.txt
$ ls ./new_dir/
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  new_dir  newfile.txt  newname.txt  script.sh  temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cp ./temp.txt ./new_dir/.
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls ./new_dir/
temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ mv ./new_dir/temp.txt ./new_dir/renamed_temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls ./new_dir/
renamed_temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q20 : Display the number of lines, words and characters in file using Linux command (Hint: use wc command).

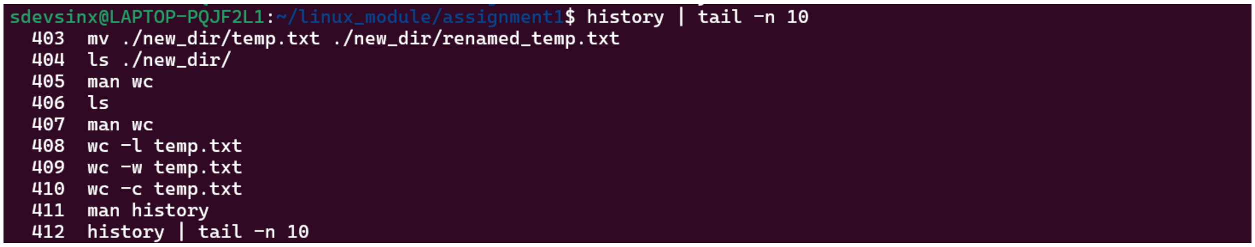
```
$ wc -l temp.txt
$ wc -w temp.txt
$ wc -c temp.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ wc -l temp.txt
227 temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ wc -w temp.txt
962 temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ wc -c temp.txt
8111 temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q21 : Use history command to display the last 10 commands used. (Hint: use history command).

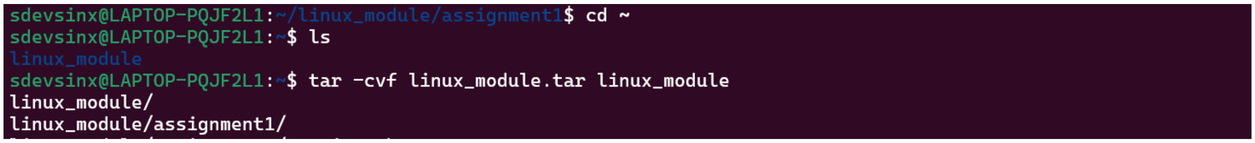
$ history | tail -n 10



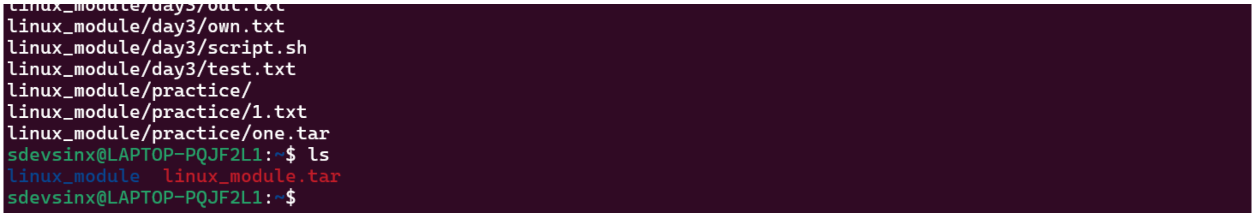
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ history | tail -n 10
  403  mv ./new_dir/temp.txt ./new_dir/renamed_temp.txt
  404  ls ./new_dir/
  405  man wc
  406  ls
  407  man wc
  408  wc -l temp.txt
  409  wc -w temp.txt
  410  wc -c temp.txt
  411  man history
  412  history | tail -n 10
```

Q22 : Create a tar archive file of any directory present in your home directory. (Hint: use tar command)

```
$ cd ~
$ ls
$ tar -cvf linux_module.tar linux_module
$ ls
```



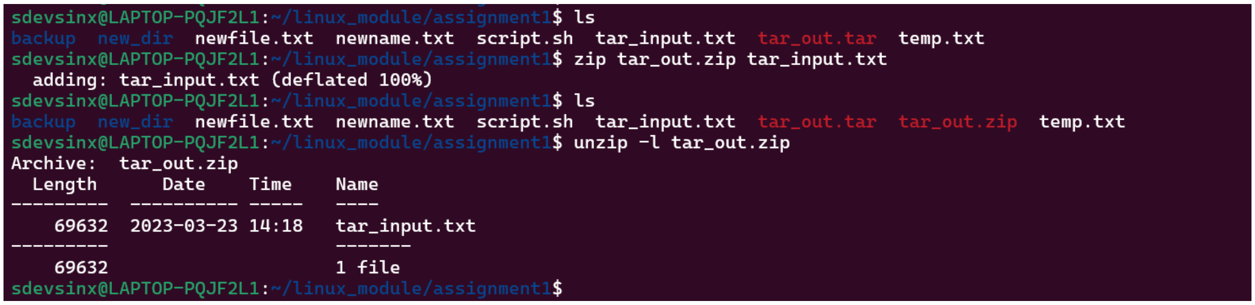
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cd ~
sdevsinx@LAPTOP-PQJF2L1:~$ ls
linux_module
sdevsinx@LAPTOP-PQJF2L1:~$ tar -cvf linux_module.tar linux_module
linux_module/
linux_module/assignment1/
```



```
linux_module/day3/out.txt
linux_module/day3/own.txt
linux_module/day3/script.sh
linux_module/day3/test.txt
linux_module/practice/
linux_module/practice/1.txt
linux_module/practice/one.tar
sdevsinx@LAPTOP-PQJF2L1:~$ ls
linux_module  linux_module.tar
sdevsinx@LAPTOP-PQJF2L1:~$
```

Q23: Create a zip file of another directory. (Hint: use zip command) - list the contents of the zip file without extracting.

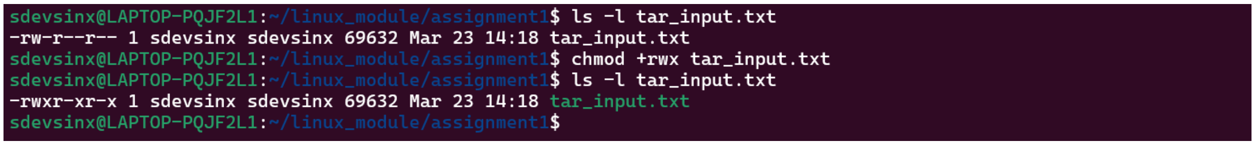
```
$ ls
$ zip tar_out.zip tar_input.txt
$ ls
$ unzip -l tar_out.zip
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  new_dir  newfile.txt  newname.txt  script.sh  tar_input.txt  tar_out.tar  temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ zip tar_out.zip tar_input.txt
  adding: tar_input.txt (deflated 100%)
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup  new_dir  newfile.txt  newname.txt  script.sh  tar_input.txt  tar_out.tar  tar_out.zip  temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ unzip -l tar_out.zip
Archive:  tar_out.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
    69632  2023-03-23 14:18   tar_input.txt
---------                     -------
    69632                     1 file
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q24 : Give read, write & execute permissions to your file. (Hint: use chmod command)

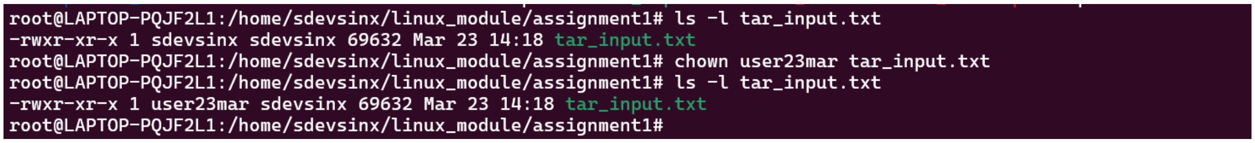
```
$ ls -l tar_input.txt
$ chmod +rwx tar_input.txt
$ ls -l tar_input.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l tar_input.txt
-rw-r--r-- 1 sdevsinx sdevsinx 69632 Mar 23 14:18 tar_input.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ chmod +rwx tar_input.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l tar_input.txt
-rwxr-xr-x 1 sdevsinx sdevsinx 69632 Mar 23 14:18 tar_input.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q25 : Change ownership of that file.(Hint: use chown command)

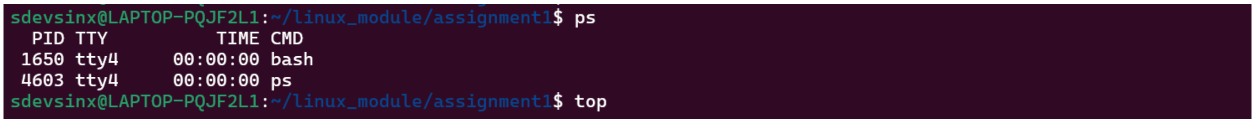
```
# ls -l tar_input.txt
# chown user23mar tar_input.txt
# ls -l tar_input.txt
```



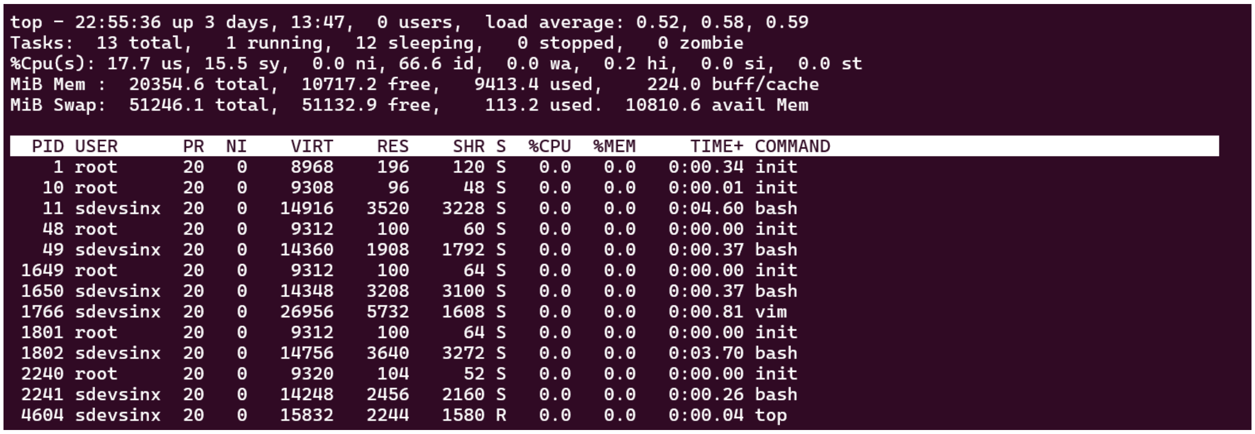
```
root@LAPTOP-PQJF2L1:/home/sdevsinx/linux_module/assignment1# ls -l tar_input.txt
-rwxr-xr-x 1 sdevsinx sdevsinx 69632 Mar 23 14:18 tar_input.txt
root@LAPTOP-PQJF2L1:/home/sdevsinx/linux_module/assignment1# chown user23mar tar_input.txt
root@LAPTOP-PQJF2L1:/home/sdevsinx/linux_module/assignment1# ls -l tar_input.txt
-rwxr-xr-x 1 user23mar sdevsinx 69632 Mar 23 14:18 tar_input.txt
root@LAPTOP-PQJF2L1:/home/sdevsinx/linux_module/assignment1#
```

Q26 : List processes running in shell, all running processes(Hint: use man page of ps command) and show top processes in decreasing order of their resource utilization.(Hint: use top command).

```
$ ps
$ top
```



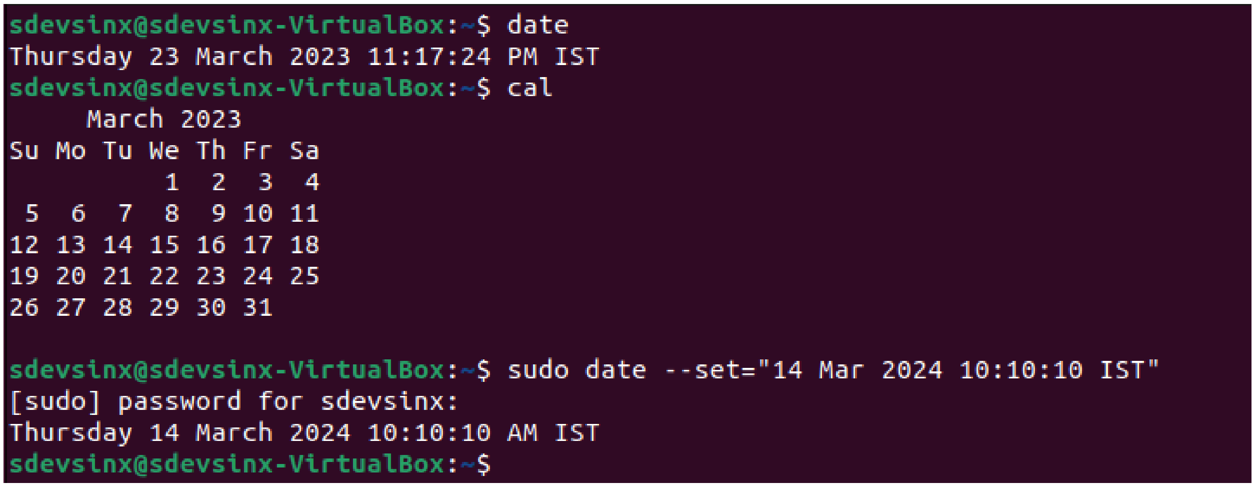
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ps
  PID TTY          TIME CMD
 1650 tty4     00:00:00 bash
 4603 tty4     00:00:00 ps
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ top
```



```
top - 22:55:36 up 3 days, 13:47,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:  13 total,   1 running,  12 sleeping,   0 stopped,   0 zombie
%Cpu(s): 17.7 us, 15.5 sy,  0.0 ni, 66.6 id,  0.0 wa,  0.2 hi,  0.0 si,  0.0 st
MiB Mem :  20354.6 total,  10717.2 free,   9413.4 used,    224.0 buff/cache
MiB Swap:  51246.1 total,  51132.9 free,    113.2 used.  10810.6 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0    8968    196    120 S   0.0   0.0   0:00.34 init
   10 root      20   0    9308     96     48 S   0.0   0.0   0:00.01 init
   11 sdevsinx  20   0   14916   3520   3228 S   0.0   0.0   0:04.60 bash
   48 root      20   0    9312    100     60 S   0.0   0.0   0:00.00 init
   49 sdevsinx  20   0   14360   1908   1792 S   0.0   0.0   0:00.37 bash
 1649 root      20   0    9312    100     64 S   0.0   0.0   0:00.00 init
 1650 sdevsinx  20   0   14348   3208   3100 S   0.0   0.0   0:00.37 bash
 1766 sdevsinx  20   0   26956   5732   1608 S   0.0   0.0   0:00.81 vim
 1801 root      20   0    9312    100     64 S   0.0   0.0   0:00.00 init
 1802 sdevsinx  20   0   14756   3640   3272 S   0.0   0.0   0:03.70 bash
 2240 root      20   0    9320    104     52 S   0.0   0.0   0:00.00 init
 2241 sdevsinx  20   0   14248   2456   2160 S   0.0   0.0   0:00.26 bash
 4604 sdevsinx  20   0   15832   2244   1580 R   0.0   0.0   0:00.04 top
```

Q27 : Display current time and calendar (Hint: use date, cal commands) 2. Change the current date and time of the system to following 14th March 2024, 10:10 AM

```
$ date
$ cal
$ date --set=”14 Mar 2024 10:10:10 IST”
```

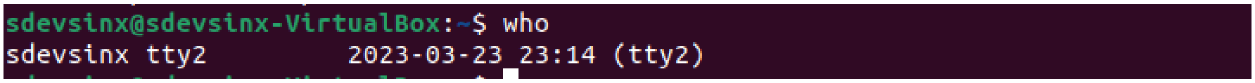


```
sdevsinx@sdevsinx-VirtualBox:~$ date
Thursday 23 March 2023 11:17:24 PM IST
sdevsinx@sdevsinx-VirtualBox:~$ cal
     March 2023
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

sdevsinx@sdevsinx-VirtualBox:~$ sudo date --set="14 Mar 2024 10:10:10 IST"
[sudo] password for sdevsinx:
Thursday 14 March 2024 10:10:10 AM IST
sdevsinx@sdevsinx-VirtualBox:~$
```

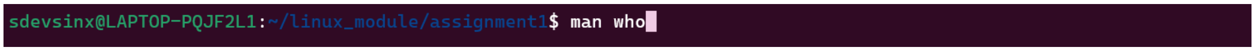
Q28 : Explore following commands

29.who, whoami, whatis, whereis, (Hint: use man pages).

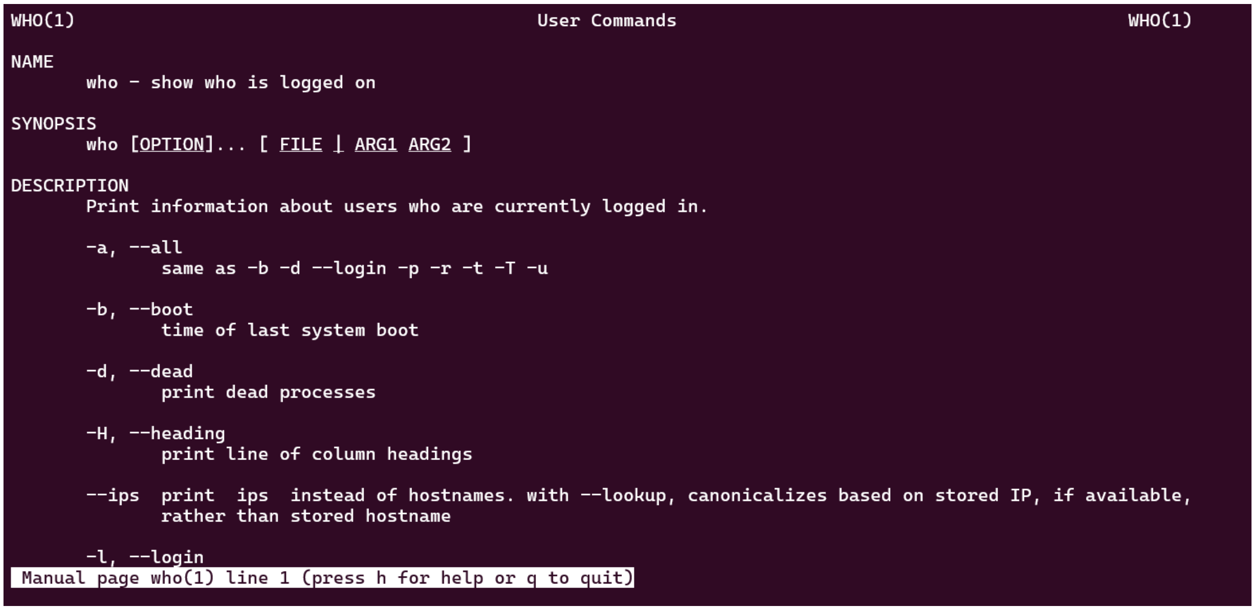
```
$ man who
```



```
sdevsinx@sdevsinx-VirtualBox:~$ who
sdevsinx tty2         2023-03-23 23:14 (tty2)
```

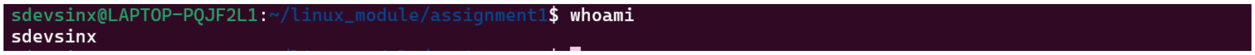


```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ man who
```

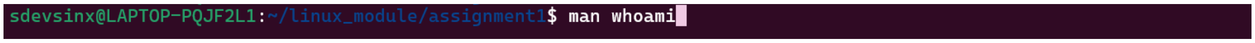


```
WHO(1)                                User Commands                               WHO(1)

NAME
       who - show who is logged on

SYNOPSIS
       who [OPTION]... [ FILE | ARG1 ARG2 ]

DESCRIPTION
       Print information about users who are currently logged in.

       -a, --all
              same as -b -d --login -p -r -t -T -u

       -b, --boot
              time of last system boot

       -d, --dead
              print dead processes

       -H, --heading
              print line of column headings

       --ips  print  ips  instead of hostnames. with --lookup, canonicalizes based on stored IP, if available,
              rather than stored hostname

       -l, --login
 Manual page who(1) line 1 (press h for help or q to quit)
```

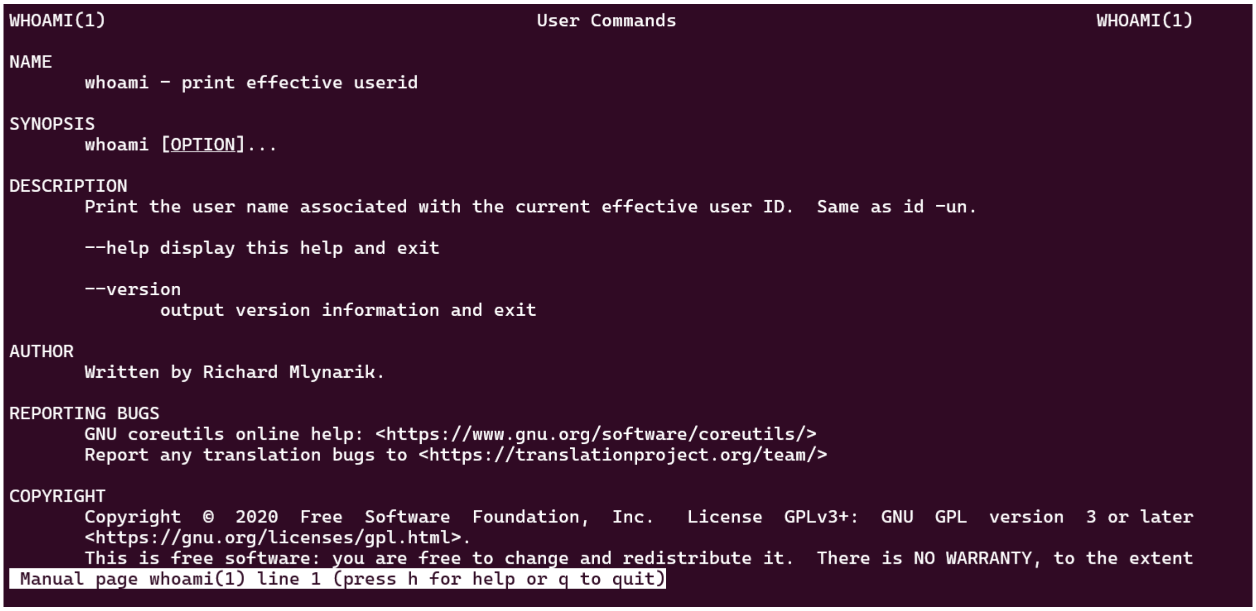
$ man whoami



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ whoami
sdevsinx
```

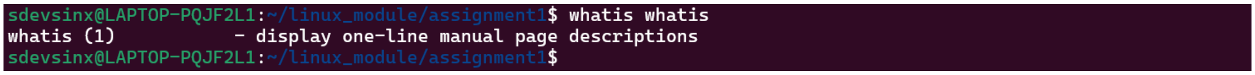


```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ man whoami
```

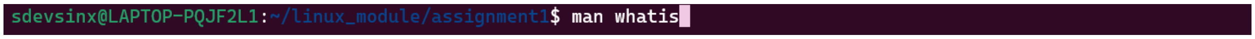


```
WHOAMI(1)                             User Commands                            WHOAMI(1)

NAME
       whoami - print effective userid

SYNOPSIS
       whoami [OPTION]...

DESCRIPTION
       Print the user name associated with the current effective user ID.  Same as id -un.

       --help display this help and exit

       --version
              output version information and exit

AUTHOR
       Written by Richard Mlynarik.

REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report any translation bugs to <https://translationproject.org/team/>

COPYRIGHT
       Copyright  ©  2020  Free  Software  Foundation,  Inc.   License  GPLv3+:  GNU  GPL  version  3 or later
       <https://gnu.org/licenses/gpl.html>.
       This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent
 Manual page whoami(1) line 1 (press h for help or q to quit)
```

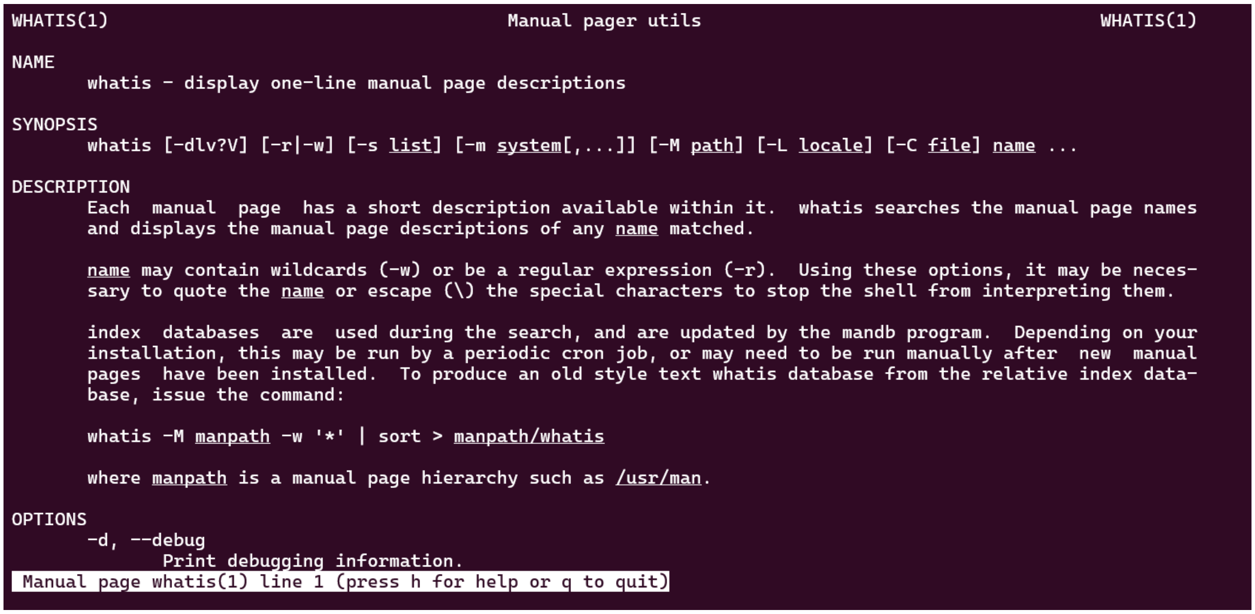
$ man whatis



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ whatis whatis
whatis (1)           - display one-line manual page descriptions
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

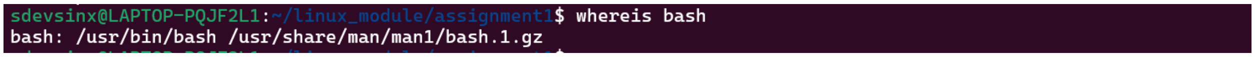


```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ man whatis
```

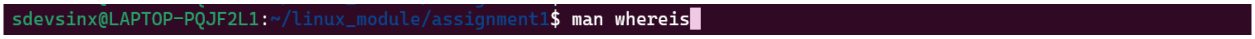


```
WHATIS(1)                          Manual pager utils                         WHATIS(1)

NAME
       whatis - display one-line manual page descriptions

SYNOPSIS
       whatis [-dlv?V] [-r|-w] [-s list] [-m system[,...]] [-M path] [-L locale] [-C file] name ...

DESCRIPTION
       Each  manual  page  has a short description available within it.  whatis searches the manual page names
       and displays the manual page descriptions of any name matched.

       name may contain wildcards (-w) or be a regular expression (-r).  Using these options, it may be neces-
       sary to quote the name or escape (\) the special characters to stop the shell from interpreting them.

       index  databases  are  used during the search, and are updated by the mandb program.  Depending on your
       installation, this may be run by a periodic cron job, or may need to be run manually after  new  manual
       pages  have been installed.  To produce an old style text whatis database from the relative index data-
       base, issue the command:

       whatis -M manpath -w '*' | sort > manpath/whatis

       where manpath is a manual page hierarchy such as /usr/man.

OPTIONS
       -d, --debug
              Print debugging information.
 Manual page whatis(1) line 1 (press h for help or q to quit)
```

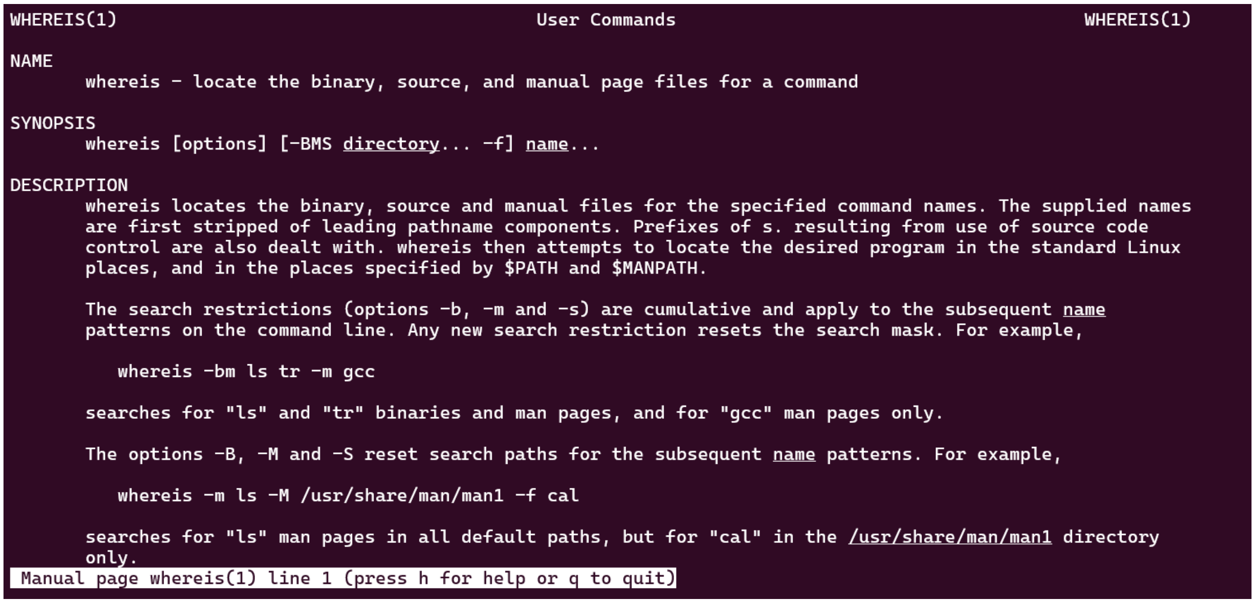
```
$ man whereis
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ whereis bash
bash: /usr/bin/bash /usr/share/man/man1/bash.1.gz
```



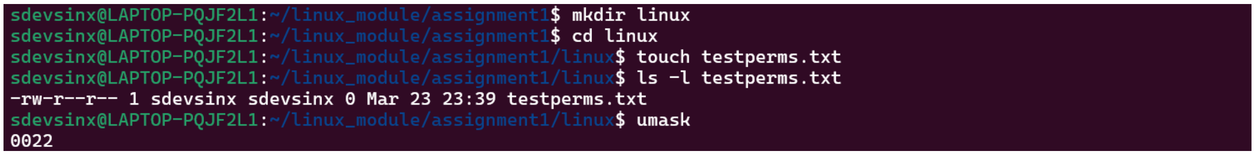
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ man whereis
```



```
WHEREIS(1)                           User Commands                           WHEREIS(1)

NAME
       whereis - locate the binary, source, and manual page files for a command

SYNOPSIS
       whereis [options] [-BMS directory... -f] name...

DESCRIPTION
       whereis locates the binary, source and manual files for the specified command names. The supplied names
       are first stripped of leading pathname components. Prefixes of s. resulting from use of source code
       control are also dealt with. whereis then attempts to locate the desired program in the standard Linux
       places, and in the places specified by $PATH and $MANPATH.

       The search restrictions (options -b, -m and -s) are cumulative and apply to the subsequent name
       patterns on the command line. Any new search restriction resets the search mask. For example,

          whereis -bm ls tr -m gcc

       searches for "ls" and "tr" binaries and man pages, and for "gcc" man pages only.

       The options -B, -M and -S reset search paths for the subsequent name patterns. For example,

          whereis -m ls -M /usr/share/man/man1 -f cal

       searches for "ls" man pages in all default paths, but for "cal" in the /usr/share/man/man1 directory
       only.
 Manual page whereis(1) line 1 (press h for help or q to quit)
```

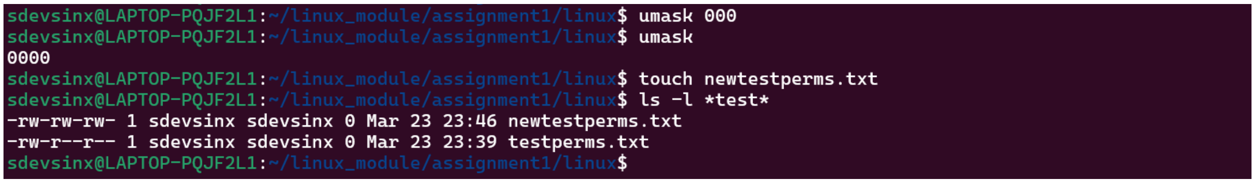
Q30 : Create one directory named linux. cd to that directory and create one file named testperms.txt. Check permissions of that file. Check value of umask. Change the value of umask and create one new file newtestperms.txt and check its permissions. Note down the difference.(Hint: use umask, ls command)

```
$ mkdir linux
$ cd linux
$ touch testperms.txt
$ ls -l testperms.txt
$ umask
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ mkdir linux
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cd linux
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ touch testperms.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ ls -l testperms.txt
-rw-r--r-- 1 sdevsinx sdevsinx 0 Mar 23 23:39 testperms.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ umask
0022
```

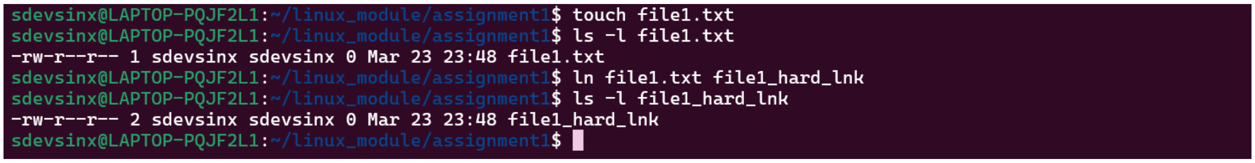
```
$ umask 000
$ umask
$ touch newtestperms.txt
$ ls -l *test*
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ umask 000
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ umask
0000
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ touch newtestperms.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$ ls -l *test*
-rw-rw-rw- 1 sdevsinx sdevsinx 0 Mar 23 23:46 newtestperms.txt
-rw-r--r-- 1 sdevsinx sdevsinx 0 Mar 23 23:39 testperms.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/linux$
```

Q31 : Create a file and name it as file1.txt and create a hardlink to this file. (Hint use ln command).

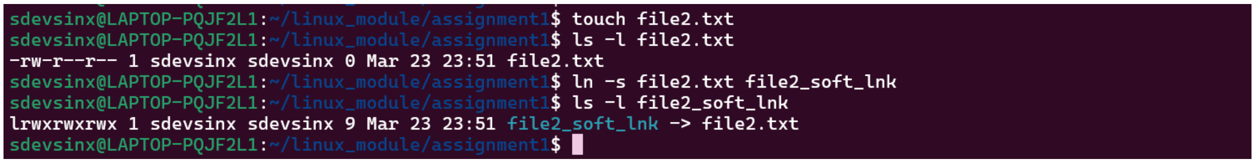
```
$ touch file1.txt
$ ls -l file1.txt
$ ln file1.txt file1_hard_lnk
$ ls -l file1_har_lnk
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch file1.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l file1.txt
-rw-r--r-- 1 sdevsinx sdevsinx 0 Mar 23 23:48 file1.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ln file1.txt file1_hard_lnk
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l file1_hard_lnk
-rw-r--r-- 2 sdevsinx sdevsinx 0 Mar 23 23:48 file1_hard_lnk
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q32 : Create a file and name it as file2.txt and create a softlink to this file. (Hint use ln command).

```
$ touch file2.txt
$ ls -l file2.txt
$ ln -s file2.txt file2_soft_lnk
$ ls -l file2_soft_lnk
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch file2.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l file2.txt
-rw-r--r-- 1 sdevsinx sdevsinx 0 Mar 23 23:51 file2.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ln -s file2.txt file2_soft_lnk
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l file2_soft_lnk
lrwxrwxrwx 1 sdevsinx sdevsinx 9 Mar 23 23:51 file2_soft_lnk -> file2.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Hard Link	Soft link
Hard lng with being a link.	Soft link is a link which indicates path to

	its parent file.
Since it is a file by itself, if parent file is deleted, child file remains as it was previously.	Since its a path, if parent file is deleted, child doesn't point tp proper path and becomes a zombie file.
Can be used for creating backup files.	Can be used as a shortcut.
Syntax is: In parentfile.ext childfile.ext	Syntax is: In -s parentfile.ext Q childfile.ext (here, -s indicates soft link)

Q33 : Use ssh to connect to your friends shell by specifying port number in the ssh command. use exit command to come out of your friends shell.
(Hint: use ssh command)

```
$
$
```

Q34 : Use scp using your friend's credentials to copy file into a directory owned by your friend, inside his home directory, specify port number in scp command.

```
$
$
```

Q35 : Use scp using your friend's credentials to copy directory into a directory owned by you, inside your home directory, specify port number in scp command

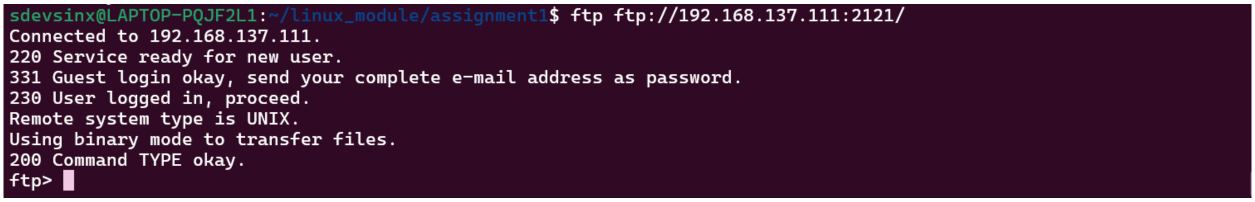
```
$
$
```

Q36 : Use scp using your friend's credentials to copy directory into a directory owned by you, inside your home directory, specify port number in scp command

```
$
$
```

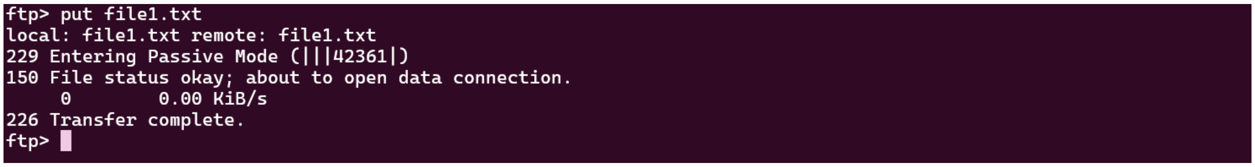
Q37 : Connect to any publicly available ftp server from terminal and try to download, upload and delete files. If you get error in any process (connect, upload, download or delete), justify the reasons behind them.(Hint: use ftp command) Example:
Try to access ftp.netbsd.org
username : anonymous
password : anonymous

```
$ ftp ftp://192.168.137.11:2121/
```



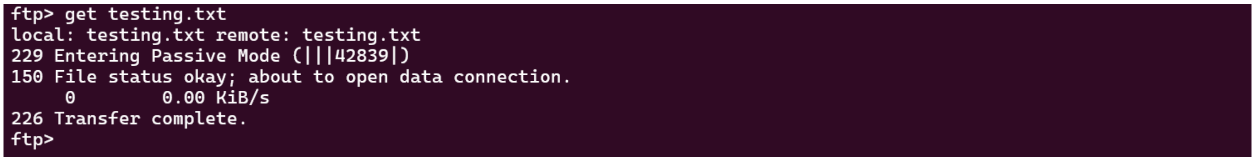
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ftp ftp://192.168.137.111:2121/
Connected to 192.168.137.111.
220 Service ready for new user.
331 Guest login okay, send your complete e-mail address as password.
230 User logged in, proceed.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Command TYPE okay.
ftp>
```

```
>put file1.txt
```



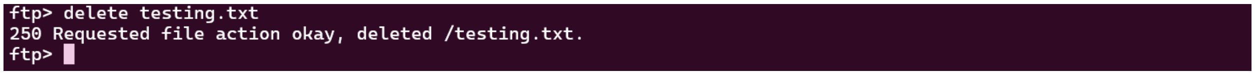
```
ftp> put file1.txt
local: file1.txt remote: file1.txt
229 Entering Passive Mode (|||42361|)
150 File status okay; about to open data connection.
     0        0.00 KiB/s
226 Transfer complete.
ftp>
```

```
>get testing.txt
```



```
ftp> get testing.txt
local: testing.txt remote: testing.txt
229 Entering Passive Mode (|||42839|)
150 File status okay; about to open data connection.
     0        0.00 KiB/s
226 Transfer complete.
ftp>
```

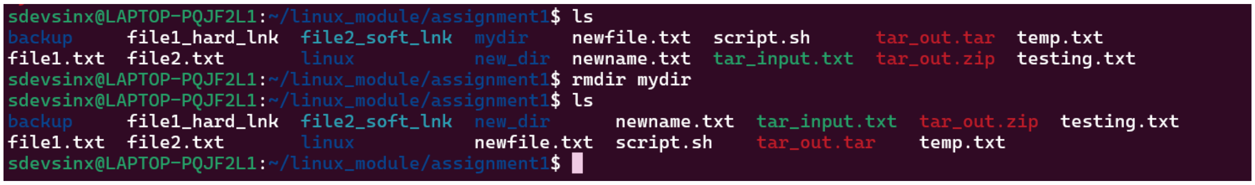
```
> delete testing.txt
```



```
ftp> delete testing.txt
250 Requested file action okay, deleted /testing.txt.
ftp>
```

Q38: How do you remove a directory named "mydir" and all of its contents using the command line?

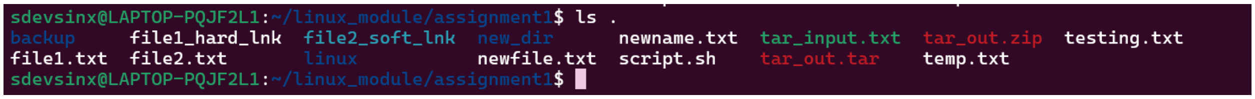
```
$ ls
$ rmdir mydir
$ ls
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup     file1_hard_lnk  file2_soft_lnk  mydir    newfile.txt  script.sh      tar_out.tar  temp.txt
file1.txt  file2.txt       linux           new_dir  newname.txt  tar_input.txt  tar_out.zip  testing.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ rmdir mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup     file1_hard_lnk  file2_soft_lnk  new_dir      newname.txt  tar_input.txt  tar_out.zip  testing.txt
file1.txt  file2.txt       linux           newfile.txt  script.sh    tar_out.tar    temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q39 : How do you use the "ls" command to list all files and directories in the current directory?

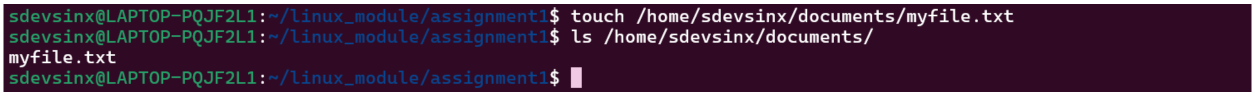
```
$ ls .
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls .
backup     file1_hard_lnk  file2_soft_lnk  new_dir      newname.txt  tar_input.txt  tar_out.zip  testing.txt
file1.txt  file2.txt       linux           newfile.txt  script.sh    tar_out.tar    temp.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q40 : How do you create a new file named "myfile.txt" in the directory "/home/user/documents" using the command line?

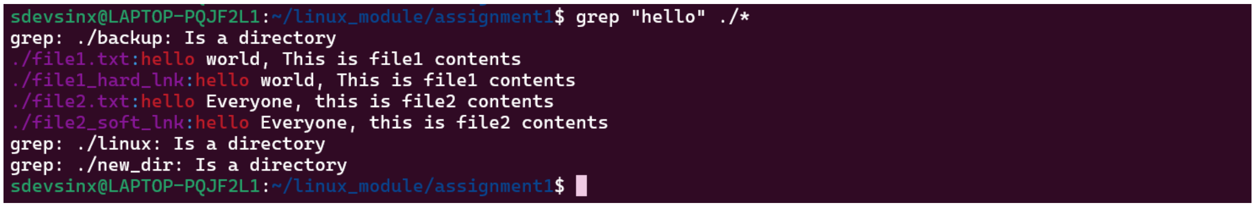
```
$ touch  /home/sdevsinx/documents/myfile.txt
$ ls /home/sdevsinx/documents
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ touch /home/sdevsinx/documents/myfile.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls /home/sdevsinx/documents/
myfile.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ 
```

Q41 : How do you use the "grep" command to search for a specific word or phrase in multiple files at once?

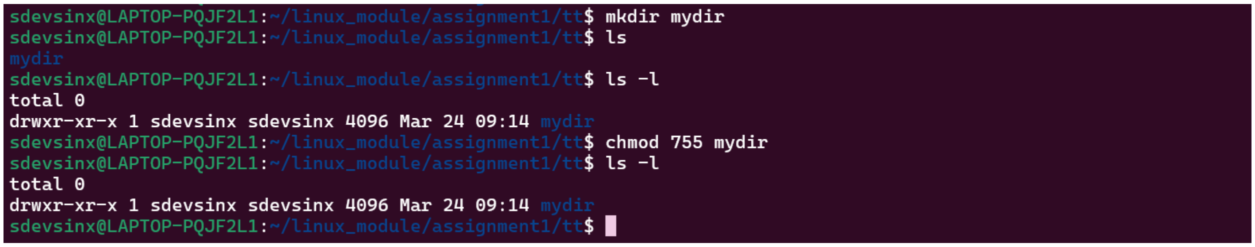
```
$ grep “hello”  ./*
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ grep "hello" ./*
grep: ./backup: Is a directory
./file1.txt:hello world, This is file1 contents
./file1_hard_lnk:hello world, This is file1 contents
./file2.txt:hello Everyone, this is file2 contents
./file2_soft_lnk:hello Everyone, this is file2 contents
grep: ./linux: Is a directory
grep: ./new_dir: Is a directory
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ 
```

Q42 : How do you create a new directory named "mydir" and set its permissions to read, write, and execute for the owner and read and execute for everyone else?

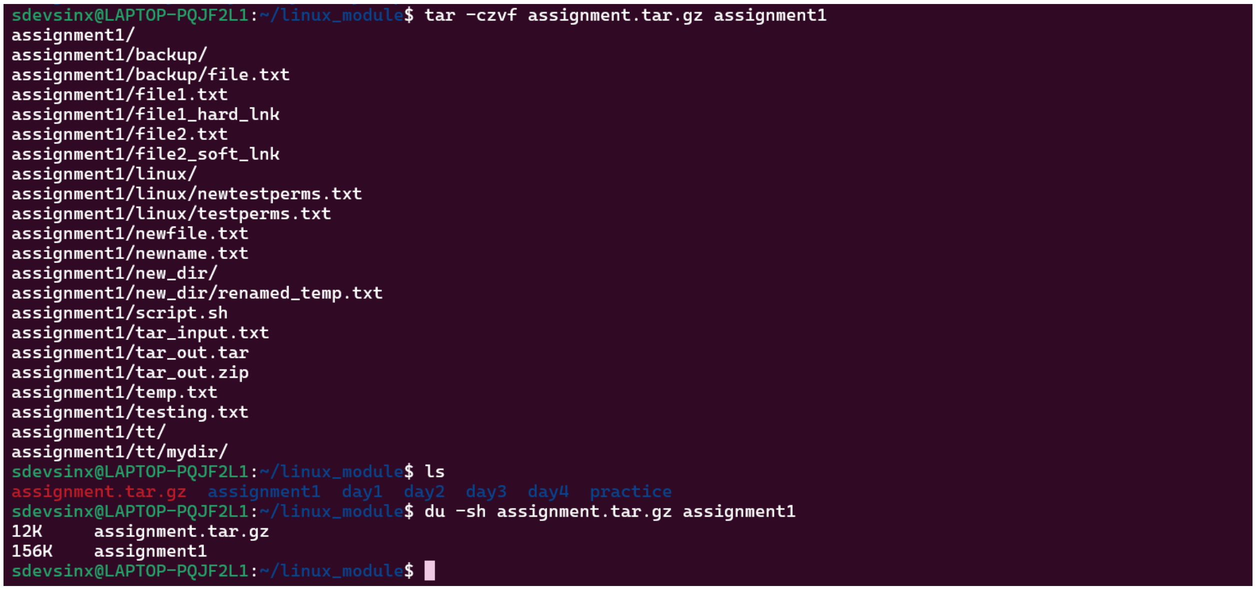
```
$ mkdir mydir
$ ls -l
$ chmod 755 mydir
$ ls -l
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/tt$ mkdir mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/tt$ ls
mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/tt$ ls -l
total 0
drwxr-xr-x 1 sdevsinx sdevsinx 4096 Mar 24 09:14 mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/tt$ chmod 755 mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/tt$ ls -l
total 0
drwxr-xr-x 1 sdevsinx sdevsinx 4096 Mar 24 09:14 mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1/tt$ 
```

Q43 : How do you use the "tar" command to create a compressed archive of all files in the current directory and its subdirectories?

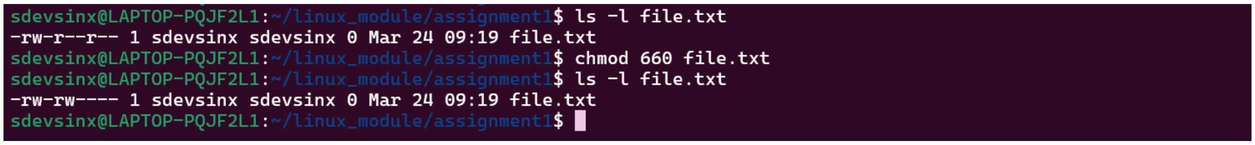
```
$ tar -czvf assignment.tar.gz assignment1
$ du -sh assignment.tar.gz assignment1
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module$ tar -czvf assignment.tar.gz assignment1
assignment1/
assignment1/backup/
assignment1/backup/file.txt
assignment1/file1.txt
assignment1/file1_hard_lnk
assignment1/file2.txt
assignment1/file2_soft_lnk
assignment1/linux/
assignment1/linux/newtestperms.txt
assignment1/linux/testperms.txt
assignment1/newfile.txt
assignment1/newname.txt
assignment1/new_dir/
assignment1/new_dir/renamed_temp.txt
assignment1/script.sh
assignment1/tar_input.txt
assignment1/tar_out.tar
assignment1/tar_out.zip
assignment1/temp.txt
assignment1/testing.txt
assignment1/tt/
assignment1/tt/mydir/
sdevsinx@LAPTOP-PQJF2L1:~/linux_module$ ls
assignment.tar.gz  assignment1  day1  day2  day3  day4  practice
sdevsinx@LAPTOP-PQJF2L1:~/linux_module$ du -sh assignment.tar.gz assignment1
12K     assignment.tar.gz
156K    assignment1
sdevsinx@LAPTOP-PQJF2L1:~/linux_module$ 
```

Q44 : How do you use the "chmod" command to give read and write permissions to the owner and group for a file named "file.txt"?

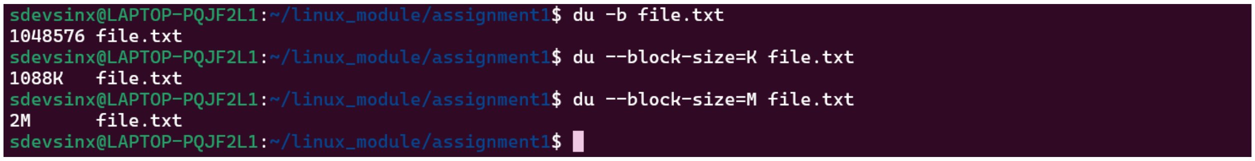
```
$ ls -l file.txt
$ chmod 660 file.txt
$ ls -l file.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l file.txt
-rw-r--r-- 1 sdevsinx sdevsinx 0 Mar 24 09:19 file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ chmod 660 file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l file.txt
-rw-rw---- 1 sdevsinx sdevsinx 0 Mar 24 09:19 file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ 
```

Q45 : How do you find the size of a file named "file.txt" in bytes, kilobytes, and megabytes using the command line?

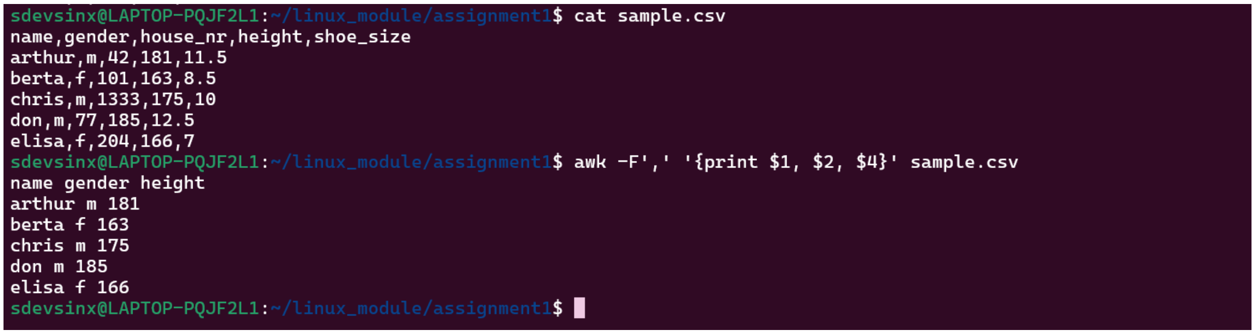
```
$ du -b file.txt
$ du --block-size=K file.txt
$ du --block-size=M file.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ du -b file.txt
1048576 file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ du --block-size=K file.txt
1088K   file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ du --block-size=M file.txt
2M      file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ 
```

Q46 : How do you use the "awk" command to extract a specific column from a comma-separated value (CSV) file and sort it in reverse order?

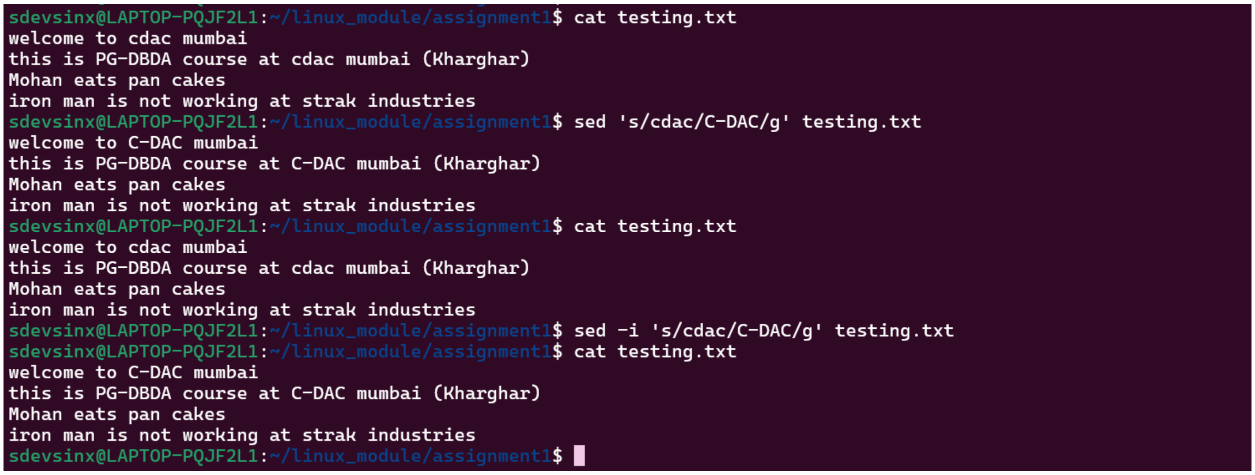
```
$ cat sample.csv
$ awk -F’,’ ‘{print $1, $2, $4}’ sample.csv
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat sample.csv
name,gender,house_nr,height,shoe_size
arthur,m,42,181,11.5
berta,f,101,163,8.5
chris,m,1333,175,10
don,m,77,185,12.5
elisa,f,204,166,7
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ awk -F',' '{print $1, $2, $4}' sample.csv
name gender height
arthur m 181
berta f 163
chris m 175
don m 185
elisa f 166
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q47 : How do you use the "sed" command to replace all occurrences of a word or phrase in a file with a different word or phrase?

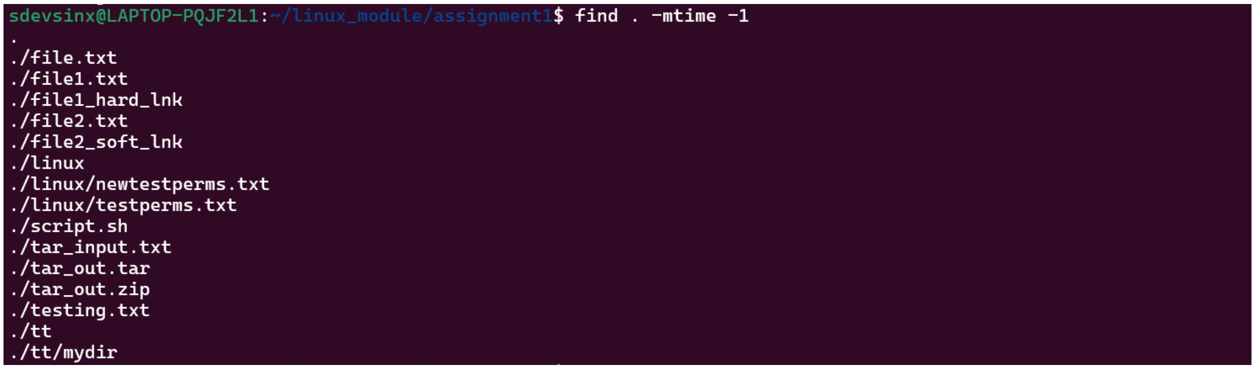
```
$ cat testing.txt
$ sed ‘s/cdac/C-DAC/g’testing.txt
$ cat testing.txt
$ sed -i ‘s/cdac/C-DAC/g’testing.txt
$ cat testing.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat testing.txt
welcome to cdac mumbai
this is PG-DBDA course at cdac mumbai (Kharghar)
Mohan eats pan cakes
iron man is not working at strak industries
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ sed 's/cdac/C-DAC/g' testing.txt
welcome to C-DAC mumbai
this is PG-DBDA course at C-DAC mumbai (Kharghar)
Mohan eats pan cakes
iron man is not working at strak industries
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat testing.txt
welcome to cdac mumbai
this is PG-DBDA course at cdac mumbai (Kharghar)
Mohan eats pan cakes
iron man is not working at strak industries
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ sed -i 's/cdac/C-DAC/g' testing.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat testing.txt
welcome to C-DAC mumbai
this is PG-DBDA course at C-DAC mumbai (Kharghar)
Mohan eats pan cakes
iron man is not working at strak industries
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

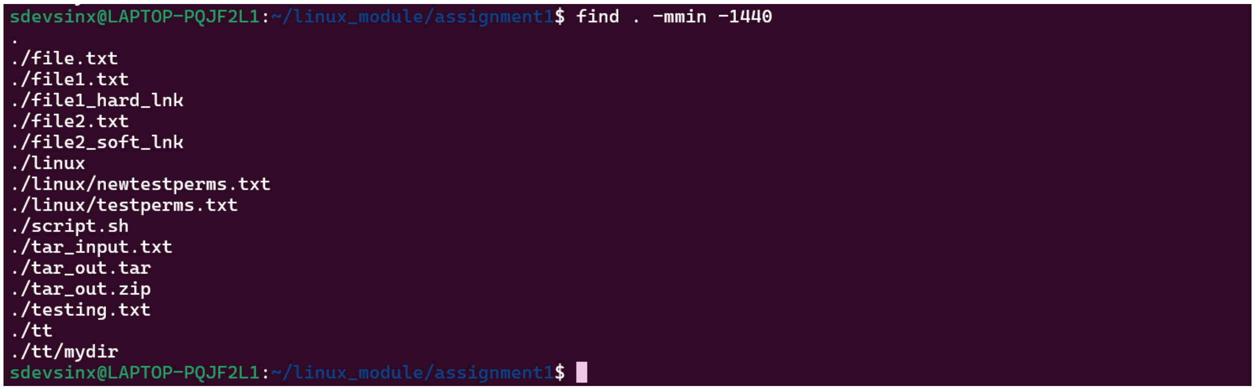
Q48 : How do you use the "find" command to search for all files in a directory and its subdirectories that were modified within the last 24 hours?

```
$ find . -mtime -1
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ find . -mtime -1
.
./file.txt
./file1.txt
./file1_hard_lnk
./file2.txt
./file2_soft_lnk
./linux
./linux/newtestperms.txt
./linux/testperms.txt
./script.sh
./tar_input.txt
./tar_out.tar
./tar_out.zip
./testing.txt
./tt
./tt/mydir
```

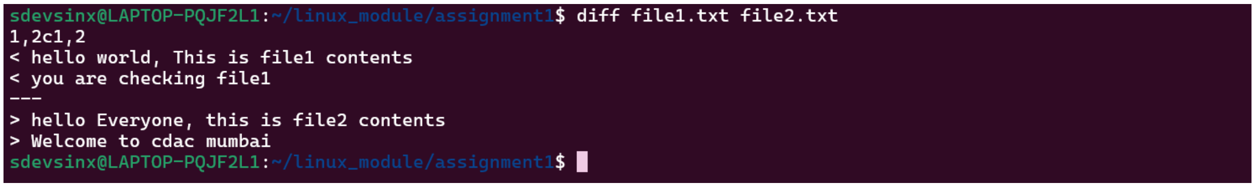
```
$ find . -mmin -1440
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ find . -mmin -1440
.
./file.txt
./file1.txt
./file1_hard_lnk
./file2.txt
./file2_soft_lnk
./linux
./linux/newtestperms.txt
./linux/testperms.txt
./script.sh
./tar_input.txt
./tar_out.tar
./tar_out.zip
./testing.txt
./tt
./tt/mydir
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q49 : How do you use the "diff" command to compare two files and show only the lines that are different between them?

```
$ diff file1.txt file2.txt
```



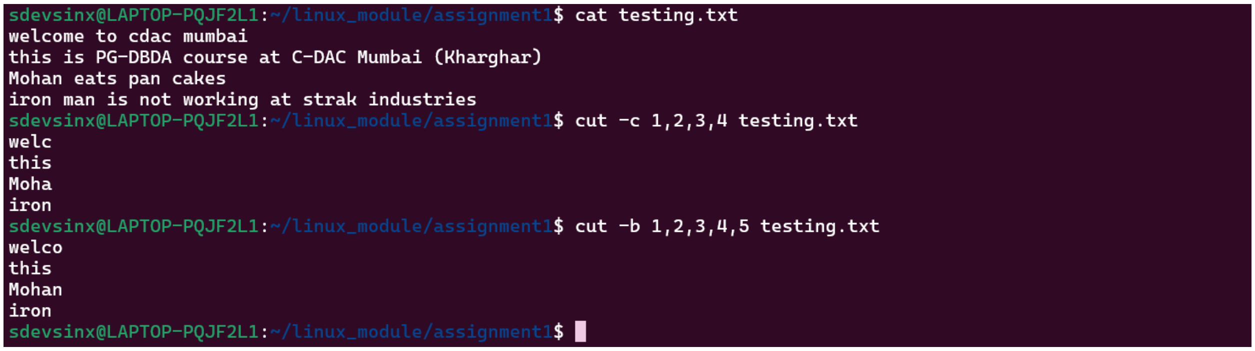
```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ diff file1.txt file2.txt
1,2c1,2
< hello world, This is file1 contents
< you are checking file1
---
> hello Everyone, this is file2 contents
> Welcome to cdac mumbai
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q50 ; How do you use the "rsync" command to synchronize the contents of two directories, including all subdirectories and files, while preserving file permissions and ownerships?

```
$
$
```

Q51 : How do you use the "cut" command to extract a specific range of characters or bytes from a file?

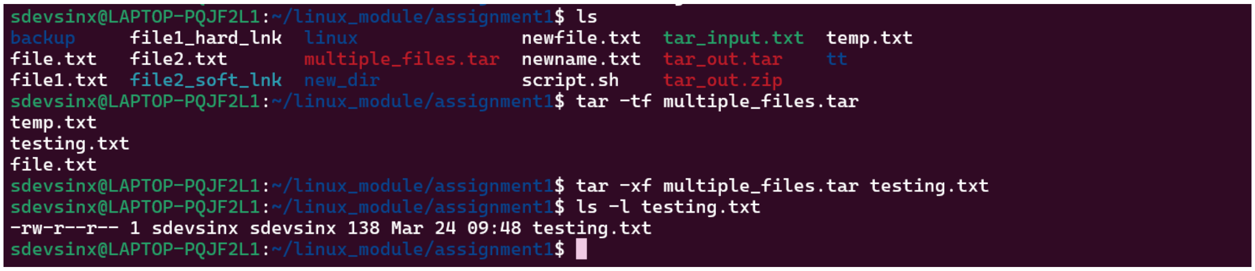
```
$ cat testing.txt
$ cut -c 1,2,3,4 testing.txt
$ cut -b 1,2,3,4,5 testing.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cat testing.txt
welcome to cdac mumbai
this is PG-DBDA course at C-DAC Mumbai (Kharghar)
Mohan eats pan cakes
iron man is not working at strak industries
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cut -c 1,2,3,4 testing.txt
welc
this
Moha
iron
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ cut -b 1,2,3,4,5 testing.txt
welco
this
Mohan
iron
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$
```

Q52 : How do you use the "tar" command to extract a specific file or directory from a compressed archive without extracting the entire archive?

```
$ ls
$ tar -tf multiple_files.tar
$ tar -xf multiple_files.tar testing.txt
$ ls -l testing.txt
```



```
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls
backup     file1_hard_lnk  linux               newfile.txt  tar_input.txt  temp.txt
file.txt   file2.txt       multiple_files.tar  newname.txt  tar_out.tar    tt
file1.txt  file2_soft_lnk  new_dir             script.sh    tar_out.zip
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ tar -tf multiple_files.tar
temp.txt
testing.txt
file.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ tar -xf multiple_files.tar testing.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ ls -l testing.txt
-rw-r--r-- 1 sdevsinx sdevsinx 138 Mar 24 09:48 testing.txt
sdevsinx@LAPTOP-PQJF2L1:~/linux_module/assignment1$ 
```

Q53 : How do you use the "awk" command to count the number of occurrences of a specific word or phrase in a file?

```
$
$
```