1) The apropos command allows to search the system's manual pages for commands that match a specific task or keyword. This is particularly useful when you need to find a command that performs a specific task but you don't know the name of the command.

(i) apropos "search for files"
(ii) apropos "list directory contents". or
apropos "^list.*directory" The ^ denotes that there is no other character before the word 'list' in the search.
(iii) apropos "file compress"
(iv) apropos "file.*copy$" The $ denotes that there is no character to be searched after the word 'copy'.

```
ibab@IBAB-RA-Comp203:~$ apropos 'search for files'
find (1)              - search for files in a directory hierarchy
suspicious-source (1) - search for files that do not meet the GPL...
ibab@IBAB-RA-Comp203:~$ apropos 'list directory contents'
dir (1)              - list directory contents
ls (1)               - list directory contents
ntfsls (8)           - list directory contents on an NTFS filesystem
vdir (1)             - list directory contents
ibab@IBAB-RA-Comp203:~$ apropos '^list.*directory
> ^C
ibab@IBAB-RA-Comp203:~$ apropos '^list.*directory'
dir (1)              - list directory contents
ls (1)               - list directory contents
ntfsls (8)           - list directory contents on an NTFS filesystem
vdir (1)             - list directory contents
ibab@IBAB-RA-Comp203:~$ apropos 'file compress'
bunzip2 (1)          - a block-sorting file compressor, v1.0.8
bzip2 (1)            - a block-sorting file compressor, v1.0.8
Dpkg::Compression::FileHandle (3perl) - class dealing transparent...
ibab@IBAB-RA-Comp203:~$ apropos 'file.*copy$'
scp (1)              - OpenSSH secure file copy
```

We are able to create more such examples using the apropos command.

```
ibab@IBAB-RA-Comp203:~$ apropos 'copy files'
cp (1)               - copy files and directories
cpio (1)             - copy files to and from archives
git-checkout-index (1) - Copy files from the index to the working...
install (1)          - copy files and set attributes
ibab@IBAB-RA-Comp203:~$ apropos 'rename files'
rename.ul (1)        - rename files
zipnote (1)          - write the comments in zipfile to stdout, e...
ibab@IBAB-RA-Comp203:~$ apropos 'unzip files'
unzip files: nothing appropriate.
ibab@IBAB-RA-Comp203:~$ apropos 'delete files'
delete files: nothing appropriate.
ibab@IBAB-RA-Comp203:~$ apropos 'remove files'
git-rm (1)           - Remove files from the working tree and fro...
rm (1)               - remove files or directories
ibab@IBAB-RA-Comp203:~$ 
```

2) i.
We create a new file using the command touch. We then list all the files and hidden files using ls -al.
The permissions for newfile are -rw-rw-r-- :
User: rw- read/write
Group: rw- read/write
Others: r-- read only

ii. umask gives us the default permission that is removed from the full permission.

Full permission for files is 666 (read/write for all). Hence after removing the default,
666 - 002 = 664 which results in rw-rw-r-- for files.
For directories: 777 - 002 = 775 → rwxr-rwxr-rx

iii. The current permissions for the new_file are : -rw-rw-r--.
The permissions match the default permissions subtracted from the full permissions.
The permissions for newfile are -rw-rw-r-- :
User: rw- read/write
Group: rw- read/write
Others: r-- read only

```
ibab@IBAB-RA-Comp203:~$ mkdir Lab7
ibab@IBAB-RA-Comp203:~$ cd Lab7
ibab@IBAB-RA-Comp203:~/Lab7$ touch new_file
ibab@IBAB-RA-Comp203:~/Lab7$ ls -l
total 0
-rw-rw-r-- 1 ibab ibab 0 Aug  8 15:53 new_file
ibab@IBAB-RA-Comp203:~/Lab7$ 
```

iv. The current permissions for the file_perm is drwxrwxr – x.
The permissions match the default permissions subtracted from the full permissions.
The permissions for file_perm is drwxrwxr – x :
User: rwx read/write/execute
Group: rwx read/write/ execute
Others: r-x read and execute

```
ibab@IBAB-RA-Comp203:~/Lab7$ mkdir file_perm
ibab@IBAB-RA-Comp203:~/Lab7$ ls
file_perm   new_file
ibab@IBAB-RA-Comp203:~/Lab7$ ls -l
total 4
drwxrwxr-x 2 ibab ibab 4096 Aug  8 16:07 file_perm
-rw-rw-r-- 1 ibab ibab    0 Aug  8 15:53 new_file
ibab@IBAB-RA-Comp203:~/Lab7$ uname
Linux
ibab@IBAB-RA-Comp203:~/Lab7$ umask
0002
ibab@IBAB-RA-Comp203:~/Lab7$ 
```

v. Using the cp command we can say that, File retains original permissions and Copying does not change permissions.

```
ibab@IBAB-RA-Comp203:~/Lab7$ cp new_file file_perm
ibab@IBAB-RA-Comp203:~/Lab7$ ls -l
total 4
drwxrwxr-x 2 ibab ibab 4096 Aug  8 16:15 file_perm
-rw-rw-r-- 1 ibab ibab    0 Aug  8 15:53 new_file
ibab@IBAB-RA-Comp203:~/Lab7$ 
```

vi. To change the permission to readable, writable and executable by user and the group and only readable by everyone else, the code is 774 . Therefore we use the command chmod 774 <dirname> The files inside the directory don't change but remain unchanged. Therefore we conclude it only changes the directory,

```
ibab@IBAB-RA-Comp203:~/Lab7$ ls -l
total 4
drw----r-- 2 ibab ibab 4096 Aug  8 16:15 file_perm
-rw-rw-r-- 1 ibab ibab    0 Aug  8 15:53 new_file
ibab@IBAB-RA-Comp203:~/Lab7$ chmod 774 fileperm
chmod: cannot access 'fileperm': No such file or directory
ibab@IBAB-RA-Comp203:~/Lab7$ chmod 774 file_perm
ibab@IBAB-RA-Comp203:~/Lab7$ ls -l
total 4
drwxrwxr-- 2 ibab ibab 4096 Aug  8 16:15 file_perm
-rw-rw-r-- 1 ibab ibab    0 Aug  8 15:53 new_file
ibab@IBAB-RA-Comp203:~/Lab7$ 
```

vii. The moving action does not change the permissions for the previous directory.

```
ibab@IBAB-RA-Comp203:~$ cd Lab7
ibab@IBAB-RA-Comp203:~/Lab7$ mkdir lab7dir
ibab@IBAB-RA-Comp203:~/Lab7$ mv file_perm lab7fir
ibab@IBAB-RA-Comp203:~/Lab7$ ls -l
total 8
drwxrwxr-x 2 ibab ibab 4096 Aug  8 17:04 lab7dir
drwxrwxr-- 2 ibab ibab 4096 Aug  8 16:15 lab7fir
-rw-rw-r-- 1 ibab ibab    0 Aug  8 15:53 new_file
ibab@IBAB-RA-Comp203:~/Lab7$ cd lab7dir
ibab@IBAB-RA-Comp203:~/Lab7/lab7dir$ ls -l
total 0
ibab@IBAB-RA-Comp203:~/Lab7/lab7dir$ 
```