

OC PRACTICE ASSIGNMENT - 1

roll no:- S20180010135

1ans)

$$(1001101)_2 = 2^6(1) + 2^5(0) + 2^4(0) + 2^3(1) + 2^2(1) + 2^1(0) + 2^0(1)$$
$$==> (1001101)_2 = (77)_{10}$$

2ans)

$$(1539)_{10} = 1539/2 = 769+1$$

$$769/2 = 384 +1$$

$$384/2 = 192 +0$$

$$192/2 = 96 +0$$

$$96/2 = 48 +0$$

$$48/2 = 24 +0$$

$$24/2 = 12 +0$$

$$12/2 = 6 +0$$

$$6/2 = 3 +0$$

$$3/2 = 1 +1$$

$$==> (1539)_{10} = (11000000011)_2$$

3ans)

a) $(1111)_2 = (31)_{10}$

b) $(101101)_2 = (45)_{10}$

c) $(1100011)_2 = (99)_{10}$

d) $(101)_2 = (5)_{10}$

e) $(0.11)_2 = (0.75)_{10}$

f) $(101.11)_2 = (5.75)_{10}$

g) $(1010)_2 = (10)_{10}$

h) $(10100)_2 = (20)_{10}$

i) $(101000)_2 = (40)_{10}$

4ans)

from my observation 3 (f,g,i) we can say when we increase a zero on the right most end then the decimal value will double.

5ans)

ex :-

$$(10010)_2 = ((0001)(0010))_2$$

$$= (12)_{16}$$

$$(10010)_2 = ((010)(010))_2$$

$$= (22)_8$$

$$(10010)_2 = 2^4(1) + 2^3(0) + 2^2(0) + 2^1(1) + 2^0(0)$$

$$\Rightarrow (10010)_2 = (18)_{10}$$

binary	octal	decimal	hexadecimal
00000	0	0	0
00001	1	1	1
00010	2	2	2
00011	3	3	3
00100	4	4	4
00101	5	5	5
00110	6	6	6
00111	7	7	7
01000	10	8	8
01001	11	9	9
01010	12	10	A
01011	13	11	B
01100	14	12	C
01101	15	13	D
01110	16	14	E
01111	17	15	F
10000	20	16	10
10001	21	17	11

10010	22	18	12
-------	----	----	----

6ans)

$$(100110110101)_2 = ((1001)(1011)(0101)) = (9B5)_{16}$$

$$= ((100)(110)(110)(101)) = (4665)_8$$

binary	octal	decimal	hexadecimal
100110110101	4665	2485	9B5
111110111100	3734	2012	7DC
1010110100	1264	692	2B4

7ans)

	hexadecimal	binary	decimal
C	43	1000011	67
c	63	1100011	99
Y	59	1011001	89
y	79	1111001	121

8ans)

unit	Number of bits (n)	Largest number ($2^n - 1$)	Number of values (2^n)
bit	1	$2^1 - 1 = 1$	$2^1 = 2$
2 bit	2	3	4
nibble	4	15	16
byte	8	255	256
1k	10	1023	1024
2 bytes	16	65535	65536
3 bytes	24	16777215	16777216
Full word* (signed long)	32	4294967295	4294967296

9ans)

value	Signed binary	value	Signed binary
+7	0111	-1	1001
+6	0110	-2	1010
+5	0101	-3	1011
+4	0100	-4	1100
+3	0011	-5	1101
+2	0010	-6	1110
+1	0001	-7	1111
0	0000	-8	1000

10ans)

$$(0.78125)_{10} = 0.78125 * 2 = 1.5625 = 1 + 0.5625$$

$$0.5625 * 2 = 1 + 0.1250$$

$$0.1250 * 2 = 0 + 0.250$$

$$0.250 * 2 = 0 + 0.500$$

$$0.500 * 2 = 1 + 0.000$$

$$\Rightarrow (0.78125)_{10} = (0.11001)_2$$

$$(1.25)_{10} = (1.01)_2$$

$$(78.725)_{10} = (1001110.1011001)_2$$

OC PRACTICE ASSIGNMENT – 2

- S20180010135

1ans)

when we use **man** command we are getting all the manual on ls command in well organised way but where as when we use **--help** command we are getting all the manual form in a discrete manner.

Here we type **man ls**

(or) ls --help

2ans)

when we use **ls > output.1** it will overwrites the current information in the output.1 file with the names of all the files and directories in the current directory.

```
venkatesh@venkatesh:~/oc$ ls
baseball big_file file_a nowrite output_2 readme.1
bat     chilli   file_b output.1 output.3 sky
venkatesh@venkatesh:~/oc$ ls > output.1
venkatesh@venkatesh:~/oc$ cat output.1
baseball
bat
big_file
chilli
file_a
file_b
nowrite
output.1
output.2
output.3
readme.1
sky
venkatesh@venkatesh:~/oc$
```

3ans)

ls > output.2

will save the name of all the files and directories in the current directory in the file output.2

```
venkatesh@venkatesh:~/oc$ ls
baseball big_file file_a nowrite output.2 readme.1
bat     chilli   file_b output.1 output.3 sky
venkatesh@venkatesh:~/oc$ ls > output.2
venkatesh@venkatesh:~/oc$ cat output.2
baseball
bat
big_file
chilli
file_a
file_b
nowrite
output.1
output.2
output.3
readme.1
sky
```

who > output.2

will overwrite the current information in output.2 with the user name and the last date and time logged in.

```
venkatesh@venkatesh:~/oc$ who > output.2
venkatesh@venkatesh:~/oc$ cat output.2
venkatesh  tty7          2018-10-18 20:14 (:0)
```

ps > output.2

```
venkatesh@venkatesh:~/oc$ ps > output.2
venkatesh@venkatesh:~/oc$ cat output.2
PID TTY      TIME CMD
3713 pts/5    00:00:00 bash
4205 pts/5    00:00:00 cat
4391 pts/5    00:00:00 ps
```

it shows the list of processors that are going on and terminal type and executing time we can also see the long list of the list of processors using the command **ps -f**

4ans)

ls > output.3

will save the name of all the files and directories in the current directory in the file output.3

```
venkatesh@venkatesh:~/oc$ ls > output.3
venkatesh@venkatesh:~/oc$ cat output.3
oc
output.1
output.2
output.3
```

who > output.3

will overwrite the current information in output.3 with the user name and the last date and time that when we logged in

```
venkatesh@venkatesh:~/oc$ who > output.3
venkatesh@venkatesh:~/oc$ cat output.3
venkatesh tty7 2018-10-18 20:14 (:0)
```

ps > output.3

```
venkatesh@venkatesh:~/oc$ ps > output.3
venkatesh@venkatesh:~/oc$ cat output.3
PID TTY      TIME CMD
6805 pts/2    00:00:00 bash
7337 pts/2    00:00:00 ps
```

it shows the list of processors that are going on and terminal type and executing time we can also see the long list of the list of processors using the command **ps -f**

5ans)

if we use **ls -s** to get the file blocks and with **ls -h** for human readable sizes so **ls -sh** for getting the file size.

```
venkatesh@venkatesh:~/oc$ ls -s
total 16
4 oc 4 output.1 4 output.2 4 output.3
venkatesh@venkatesh:~/oc$ ls -h
oc output.1 output.2 output.3
venkatesh@venkatesh:~/oc$ ls -sh
total 16K
4.0K oc 4.0K output.1 4.0K output.2 4.0K output.3
```

6ans)

mv original copied.file

this command will renames if there is no directory named with copied.file in this current directory

7ans)

it prints the calendar of 9th month in the year 1752

```
venkatesh@venkatesh:~/oc$ cal 09 1752
September 1752
Su Mo Tu We Th Fr Sa
      1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

if we type

cal 10 2018

it prints 2018 october calendar

```
venkatesh@venkatesh:~/oc$ cal 10 2018
October 2018
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
```

8ans)

to change the login password through terminal we should type

passwd

then it will ask current password and the new password so we can change the password.

```
venkatesh@venkatesh:~/oc$ passwd
Changing password for venkatesh.
(current) UNIX password:
Enter new UNIX password: █
```

9ans)

we can see in the terminal from whose system and whose account we logged on

```
venkatesh@venkatesh:~/oc$ touch baseball bat
venkatesh@venkatesh:~/oc$ █
```

here **venkatesh@venkatesh** in which the first venkatesh is user name and the second one is system name

and also using **who** command to print the user name and when he logged on.

10ans)

the **touch** command will create new empty file and allows the terminal for reading the next command

```
venkatesh@venkatesh:~/oc$ touch sky
venkatesh@venkatesh:~/oc$ █
```

11ans) among the given three commands first one and the last one will create a file

```
sh@venkatesh:~/oc$ touch baseball bat  
sh@venkatesh:~/oc$
```

and the second one shows error message thinking that the **-b** we typed there is a command for some operation which is not in the **touch** command list

```
venkatesh@venkatesh:~/oc$ touch -bigfile-  
touch: invalid option -- 'b'  
Try 'touch --help' for more information.  
venkatesh@venkatesh:~/oc$ touch chilli&beans  
[1] 2983  
beans: command not found  
[1]+ Done touch chilli  
venkatesh@venkatesh:~/oc$
```

where as first one creates two files baseball and bat here in the last one **touch chilli&beans** create a new file named chilli and shows the linking command is not found in the touch command list so when we see the list of files present in that directory we can see

12ans)

we can create directories using **mkdir** command

```
venkatesh@venkatesh:~/oc$ ls  
baseball bat chilli oc output.1 output.2 output.3 sky  
venkatesh@venkatesh:~/oc$ mkdir Letters  
venkatesh@venkatesh:~/oc$ mkdir Programs  
venkatesh@venkatesh:~/oc$ mkdir Misc  
venkatesh@venkatesh:~/oc$ ls  
baseball chilli Misc output.1 output.3 sky  
bat Letters oc output.2 Programs  
venkatesh@venkatesh:~/oc$
```

13ans)

to change permission we should use **chmod** command and to remove the permission from user we should use **chmod u-r file name** we can see the list of permission of a file/directory using **chmod -l file/directory name**

```
venkatesh@venkatesh:~/oc$ touch readme.1  
venkatesh@venkatesh:~/oc$ ls  
baseball chilli Misc output.1 output.3 readme.1  
bat Letters oc output.2 Programs sky  
venkatesh@venkatesh:~/oc$ ls -l readme.1  
-rw-rw-r-- 1 venkatesh venkatesh 0 Oct 16 14:34 readme.1  
venkatesh@venkatesh:~/oc$ chmod u-r readme.1  
venkatesh@venkatesh:~/oc$ ls -l readme.1  
--w-rw-r-- 1 venkatesh venkatesh 0 Oct 16 14:34 readme.1  
venkatesh@venkatesh:~/oc$ cat readme.1  
cat: readme.1: Permission denied  
venkatesh@venkatesh:~/oc$
```

14ans)

- a) using **chmod u-w nowrite** we can remove the write permission of this directory will be removed so we cant add/remove any file from this directory.
- b) we can come back to the previous directory using **cd ..** and now we can add the write command to that file using **chmod u+w nowrite** now we can add? remove files from this directory.

```
venkatesh@venkatesh:~/oc$ mkdir nowrite
venkatesh@venkatesh:~/oc$ chmod u-w nowrite
venkatesh@venkatesh:~/oc$ ls -l nowrite
total 0
venkatesh@venkatesh:~/oc$ cd nowrite
venkatesh@venkatesh:~/oc/nowrite$ touch try.me
touch: cannot touch 'try.me': Permission denied
venkatesh@venkatesh:~/oc/nowrite$ cd ..
venkatesh@venkatesh:~/oc$ chmod u+w nowrite
venkatesh@venkatesh:~/oc$ cd nowrite
venkatesh@venkatesh:~/oc/nowrite$ touch try.me
venkatesh@venkatesh:~/oc/nowrite$ ls
try.me
```

15ans)

ls -l /bin > big.file will create a file named big.file and enters all the information about the permission ,metadata , user, system name, size information of all the files in this new file

- a) cat big.file shows all the content in the file at once
- b) more big.file shows all the contents page by page when we press enter
- c) more will shows page by page or line by line where as cat will shows all the content at once

16ans)

we can use **head -11** command to print the first 11 lines now we can pass the command using | and we can use **tail -1** to print the last line in the **head -11** command

```
venkatesh@venkatesh:~/oc$ head -11 big.file | tail -1
-rw-r-xr-x 1 root root 3642 May 20 2015 bzgrep
```

17ans)

we can create a file using **touch** command and a file using **mkdir** for moving a file **mv filename directory_where_it_should_move** and to delete a directory we can use **rmdir** but to remove a directory containing files we should use **rm -R direc3toryname** it do a recursive operation until all the files and the folder deletes

```
venkatesh@venkatesh:~/oc$ touch UG1
venkatesh@venkatesh:~/oc$ mkdir IIITS
venkatesh@venkatesh:~/oc$ mv UG1 UG1.IIITS
venkatesh@venkatesh:~/oc$ ls
baseball big.file IIITS output.1 output.3 sky
bat chilli nowrite output.2 readme.1 UG1.IIITS
venkatesh@venkatesh:~/oc$ mv UG1 IIITS
mv: cannot stat 'UG1': No such file or directory
venkatesh@venkatesh:~/oc$ mv UG1.IIITS IIITS
venkatesh@venkatesh:~/oc$ ls
baseball big.file IIITS output.1 output.3 sky
bat chilli nowrite output.2 readme.1
venkatesh@venkatesh:~/oc$ mkdir IIITS1
venkatesh@venkatesh:~/oc$ ls
baseball big.file IIITS nowrite output.2 readme.1
bat chilli IIITS1 output.1 output.3 sky
venkatesh@venkatesh:~/oc$ rmdir IIITS1
venkatesh@venkatesh:~/oc$ rm -R IIITS
venkatesh@venkatesh:~/oc$ ls
baseball big.file nowrite output.2 readme.1
bat chilli output.1 output.3 sky
venkatesh@venkatesh:~/oc$
```

18ans)

ls -l gives long list of files and directories where as **ls -dl** gives the longlist of the . and .. files

```
venkatesh@venkatesh:~/oc$ ls -d
.
venkatesh@venkatesh:~/oc$ ls -l
total 28
-rw-rw-r-- 1 venkatesh venkatesh    0 Oct 16 12:23 baseball
-rw-rw-r-- 1 venkatesh venkatesh    0 Oct 16 12:23 bat
-rw-rw-r-- 1 venkatesh venkatesh 8588 Oct 16 15:24 big.file
-rw-rw-r-- 1 venkatesh venkatesh    0 Oct 16 12:21 chilli
drwxrwxr-x 2 venkatesh venkatesh 4096 Oct 16 15:04 nowrite
-rw-rw-r-- 1 venkatesh venkatesh   30 Oct 16 12:15 output.1
-rw-rw-r-- 1 venkatesh venkatesh   30 Oct 16 00:44 output.2
-rw-rw-r-- 1 venkatesh venkatesh   45 Oct 16 00:54 output.3
-rw-rw-r-- 1 venkatesh venkatesh    0 Oct 16 14:34 readme.1
-rw-rw-r-- 1 venkatesh venkatesh    0 Oct 16 12:19 sky
venkatesh@venkatesh:~/oc$ ls -ld
drwxrwxr-x 3 venkatesh venkatesh 4096 Oct 16 21:10 .
venkatesh@venkatesh:~/oc$
```

19ans)

here we are using **wc -w** command to get the words count of the file

```
venkatesh@venkatesh:~/oc$ ls
baseball big.file file_a nowrite output.2 readme.1
bat      chilli   file_b output.1 output.3 sky
venkatesh@venkatesh:~/oc$ wc -w file_a file_b
4 file_a
5 file_b
9 total
```

20ans)

a)& b) & c) we can create a file using **touch filename** command and a directory using **mkdir dirname** and to enter a folder using **cd dirname** command

```
Venkatesh@venkatesh:~/ITWS1/dir_practice
venkatesh@venkatesh:~$ mkdir ITWS1
venkatesh@venkatesh:~$ cd ITWS1
venkatesh@venkatesh:~/ITWS1$ mkdir dir_practice
venkatesh@venkatesh:~/ITWS1$ cd dir_practice
venkatesh@venkatesh:~/ITWS1/dir_practice$ mkdir stuff
venkatesh@venkatesh:~/ITWS1/dir_practice$ cd stuff
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff$ mkdir morestuff
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff$ cd morestuff
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff$ mkdir stillmore
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff$ cd stillmore
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff/stillmore$ touch filec
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff/stillmore$ cd ..
cd... command not found
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff/stillmore$ cd ..
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff$ touch fileb
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff$ cd ..
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff$ touch filea
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff$ cd ..
venkatesh@venkatesh:~/ITWS1/dir_practice$
```

- d) **ls -R** undergoes recursive program and prints all the files and folders present that are linking to the home directory
- e) we can enter text in a file by creating using **cat >> filename**

```
venkatesh@venkatesh:~$ cd ITWS1
venkatesh@venkatesh:~/ITWS1$ cd dir_practice
venkatesh@venkatesh:~/ITWS1/dir_practice$ ls
stuff
venkatesh@venkatesh:~/ITWS1/dir_practice$ cd stuff
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff$ cat >> mystuff
If you can read the message,
you have accessed my account
Congratulations...
^Z
[1]+  Stopped                  cat >> mystuff
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff$ 
```

- f) we can rename a file using the command **mv filename renamefile**

```
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff$ mv filec stillmore
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff$ cd stillmore
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff/stillmore$ ls
filec
venkatesh@venkatesh:~/ITWS1/dir_practice/stuff/morestuff/stillmore$ 
```

- g)

```
venkatesh@venkatesh:~$ cp ITWS1/dir_practice/stuff/filex.bak filex.bak
venkatesh@venkatesh:~$ cat filex.bak
venkatesh@venkatesh:~$ 
```

- h)

-i asks the user whether to do that work or not

```
venkatesh@venkatesh:~/ITWS1/dir_practice
venkatesh@venkatesh:~$ cd ITWS1/dir_practice
venkatesh@venkatesh:~/ITWS1/dir_practice$ ls
stuff
venkatesh@venkatesh:~/ITWS1/dir_practice$ rm -i -R stuff
rm: descend into directory 'stuff'? Y
rm: remove regular empty file 'stuff/filex.bak'? Y
rm: remove regular file 'stuff/mystuff'? Y
rm: descend into directory 'stuff/morestuff'? Y
rm: descend into directory 'stuff/morestuff/stillmore'? Y
rm: remove regular empty file 'stuff/morestuff/stillmore/filec'? Y
rm: remove directory 'stuff/morestuff/stillmore'? Y
rm: remove directory 'stuff/morestuff'? Y
rm: remove directory 'stuff'? Y
venkatesh@venkatesh:~/ITWS1/dir_practice$ 
```

OC PRACTICE ASSIGNMENT – 3

-S20180010135

1Ans)

to print all the files and directories that ends with sh i will use the command ls *sh

```
venkatesh@venkatesh:~$ cd ../../bin
venkatesh@venkatesh:/bin$ ls *sh
bash dash rbash sh static-sh
venkatesh@venkatesh:/bin$
```

2Ans)

we can use ls with a [] meta characters and the * selects all the files that includes none.

```
venkatesh@venkatesh:~$ ls [ABCDEFGHIJKLMNOPQRSTUVWXYZ]*
Desktop:
cse os

Documents:

Downloads:
codeblocks_17.12-1_amd64_stable.tar.xz sublime_text_3_build_3176_x64.tar.bz2
comm.png

Music:
audios

Pictures:
2que.png
format speci-1.png
format specifiers.png
Screenshot_2018-10-25-22-49-15-828_com.miui.gallery.png
Screenshot from 2018-10-11 09-52-24.png
Screenshot from 2018-10-16 00-45-04.png
Wallpapers
Webcam

Public:

Templates:

Videos:
```

3ans)

we can use ls with the meta characters [] which gives options selecting .c and .h files and the * selects all the files that includes none.

```
venkatesh@venkatesh:~$ touch 1.c
venkatesh@venkatesh:~$ touch 2.c
venkatesh@venkatesh:~$ touch 3.c
venkatesh@venkatesh:~$ touch 3.h
venkatesh@venkatesh:~$ touch 1.
venkatesh@venkatesh:~$ touch 1.h
venkatesh@venkatesh:~$ touch 2.h
venkatesh@venkatesh:~$ ls
1. 2.h a.out      Desktop      format strings  linux.odt  Public
1.c 3.c assignment.c  Documents   git           logisim    sai
1.h 3.h assignments  Downloads   github        Music       Templates
2.c a.c boot.tar.gz examples.desktop html        Pictures   Videos
venkatesh@venkatesh:~$ ls *[.c,.h]
1. 1.c 1.h 2.c 2.h 3.c 3.h a.c assignment.c
```

4ans)

we can get all the files with r or g or i in the third place using metacharacters [] and using ? Which will show all the files with atleast one character

```
venkatesh@venkatesh:~$ ls ??[r,g,i]*
format strings
logisim:
sai:
SAI KUMAR OC 2.odt  SAI KUMAR OC 3.odt  SAI KUMAR OC.odt
```

5ans)

we can copy all the files that have character string into the Misc directory using the command

cp *[notes,misc]* Misc

```
venkatesh@venkatesh:~/oc$ ls
1misc1 1notes1 Misc misc12 notes11
venkatesh@venkatesh:~/oc$ cp *[notes,misc]* Misc
cp: omitting directory 'Misc'
venkatesh@venkatesh:~/oc$ cd Misc
venkatesh@venkatesh:~/oc/Misc$ ls
1misc1 1notes1 misc12 notes11
venkatesh@venkatesh:~/oc/Misc$
```

6ans)

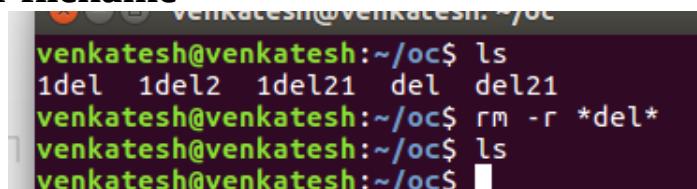
to move file we can use **mv source destiny**

to copy file we can use **cp source destiny**

```
venkatesh@venkatesh:~/oc$ ls
copy.me copy.me1 copy.me12 move.me move.me1 move.me12 UStoreIt
venkatesh@venkatesh:~/oc$ cp copy.me* UStoreIt
venkatesh@venkatesh:~/oc$ ls
copy.me copy.me1 copy.me12 move.me move.me1 move.me12 UStoreIt
venkatesh@venkatesh:~/oc$ cd UStoreIt
venkatesh@venkatesh:~/oc/UStoreIt$ ls
copy.me copy.me1 copy.me12
venkatesh@venkatesh:~/oc/UStoreIt$ cd ..
venkatesh@venkatesh:~/oc$ mv move.me* UStoreIt
venkatesh@venkatesh:~/oc$ ls
copy.me copy.me1 copy.me12 UStoreIt
venkatesh@venkatesh:~/oc$ cd UStoreIt
venkatesh@venkatesh:~/oc/UStoreIt$ ls
copy.me copy.me1 copy.me12 move.me move.me1 move.me12
venkatesh@venkatesh:~/oc/UStoreIt$
```

7ans)

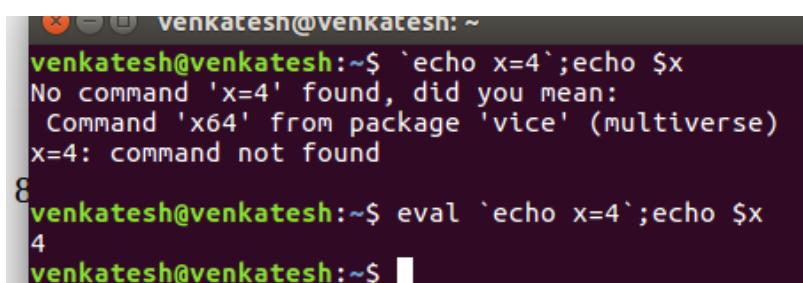
we can use **rm filename** to delete the file to do recursively we can use **rm -r filename**



```
venkatesh@venkatesh:~/oc$ ls
1del 1del2 1del21 del del21
venkatesh@venkatesh:~/oc$ rm -r *del*
venkatesh@venkatesh:~/oc$ ls
venkatesh@venkatesh:~/oc$
```

8ans)

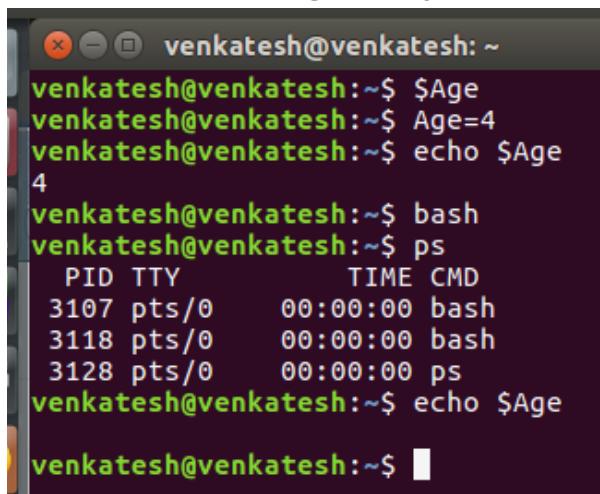
in (a) it treats as a command that is in `` so it prints no command found but in the (b) we use **eval** which assigns the value of x=4 so when we prints x next time it prints x=4
; --> used to do the series of commands



```
venkatesh@venkatesh: ~
venkatesh@venkatesh:~$ `echo x=4`;echo $x
No command 'x=4' found, did you mean:
  Command 'x64' from package 'vice' (multiverse)
x=4: command not found
8
venkatesh@venkatesh:~$ eval `echo x=4`;echo $x
4
venkatesh@venkatesh:~$
```

9ans)

here first we are defining the variable Age and assigning the value 4 so it will print this shell but if we call for another shell again it will show null because it is not defined globally.



```
venkatesh@venkatesh: ~
venkatesh@venkatesh:~$ $Age
venkatesh@venkatesh:~$ Age=4
venkatesh@venkatesh:~$ echo $Age
4
venkatesh@venkatesh:~$ bash
venkatesh@venkatesh:~$ ps
  PID TTY          TIME CMD
 3107 pts/0    00:00:00 bash
 3118 pts/0    00:00:00 bash
 3128 pts/0    00:00:00 ps
venkatesh@venkatesh:~$ echo $Age

venkatesh@venkatesh:~$
```

10ans)

if we call for a new shell and enter a directory on exiting it, in parent shell it will again comes to the home directory

```
venkatesh@venkatesh: ~
venkatesh@venkatesh:~$ bash
venkatesh@venkatesh:~$ cd Documents
venkatesh@venkatesh:~/Documents$ exit
exit
venkatesh@venkatesh:~$
```

11ans)

- A. we can get to home directory using **cd ~** command

```
venkatesh@venkatesh: ~
venkatesh@venkatesh:~$ cd Documents
venkatesh@venkatesh:~/Documents$ cd ~
venkatesh@venkatesh:~$
```

- B. Make public_html a directory named

```
venkatesh@venkatesh:~/oc
venkatesh@venkatesh:~/oc$ mkdir public_html
venkatesh@venkatesh:~/oc$ ls
fruits  public_html  trfile
venkatesh@venkatesh:~/oc$
```

- C. Allow group and others to be able to read and execute on your homedirectory

```
venkatesh@venkatesh:~/oc
venkatesh@venkatesh:~/oc$ ls -l
total 12
-rw-rw-r-- 1 venkatesh venkatesh 280 Oct 30 16:34 fruits
drwxrwxr-x 2 venkatesh venkatesh 4096 Nov  1 19:35 public_html
-rw-rw-r-- 1 venkatesh venkatesh 38 Oct 30 14:46 trfile
venkatesh@venkatesh:~/oc$ chmod go+r *
venkatesh@venkatesh:~/oc$ ls -l
total 12
-rw-rw-r-x 1 venkatesh venkatesh 280 Oct 30 16:34 fruits
drwxrwxr-x 2 venkatesh venkatesh 4096 Nov  1 19:35 public_html
-rw-rw-r-x 1 venkatesh venkatesh 38 Oct 30 14:46 trfile
venkatesh@venkatesh:~/oc$
```

- D. Allow group and others to be able to read and execute on the public_html directory

```
venkatesh@venkatesh:~/oc
venkatesh@venkatesh:~/oc$ chmod go+r public_html
venkatesh@venkatesh:~/oc$ ls -l
total 12
-rw-rw-r-x 1 venkatesh venkatesh 280 Oct 30 16:34 fruits
drwxrwxr-x 2 venkatesh venkatesh 4096 Nov  1 19:35 public_html
-rw-rw-r-x 1 venkatesh venkatesh 38 Oct 30 14:46 trfile
venkatesh@venkatesh:~/oc$
```

- E. permissions for public_html directory

```
venkatesh@venkatesh:~/oc$ ls -l
total 12
-rw-rw-r-x 1 venkatesh venkatesh 280 Oct 30 16:34 fruits
drwxrwxr-x 2 venkatesh venkatesh 4096 Nov  1 19:35 public_html
-rw-rw-r-x 1 venkatesh venkatesh 38 Oct 30 14:46 trfile
venkatesh@venkatesh:~/oc$
```

permissions for home directory

```
venkatesh@venkatesh:/$ ls -l home
total 4
drwxr-xr-x 23 venkatesh venkatesh 4096 Nov  1 19:07 venkatesh
venkatesh@venkatesh:/$
```

we can observe both the permissions are same
F, G.

```
venkatesh@venkatesh:~/oc/public_html
venkatesh@venkatesh:~/oc$ cd public_html
venkatesh@venkatesh:~/oc/public_html$ touch index.html
venkatesh@venkatesh:~/oc/public_html$ ls
index.html
venkatesh@venkatesh:~/oc/public_html$ chmod go+r *
venkatesh@venkatesh:~/oc/public_html$ ls -l
total 0
-rw-rw-r-- 1 venkatesh venkatesh 0 Nov 1 19:49 index.html
venkatesh@venkatesh:~/oc/public_html$
```

H. permission of desktop files and public_html files

```
venkatesh@venkatesh:~/oc/public_html
venkatesh@venkatesh:~/oc/public_html$ ls -l
total 0
-rw-rw-r-- 1 venkatesh venkatesh 0 Nov 1 19:49 index.html
venkatesh@venkatesh:~/oc/public_html$ cd ../..
venkatesh@venkatesh:~$ ls -l
total 10596
-rwxrwxrwx 1 venkatesh venkatesh 825 Oct 25 21:53 a.c
-rwxrwxr-x 1 venkatesh venkatesh 8856 Oct 31 19:32 a.out
-rwxrwxrwx 1 venkatesh venkatesh 927 Oct 26 21:55 assignment.c
drwxrwxr-x 2 venkatesh venkatesh 4096 Nov 1 19:07 assignments
-rwxrwxrwx 1 venkatesh venkatesh 390 Sep 20 19:55 boot.tar.gz
drwxr-xr-x 4 venkatesh venkatesh 4096 Nov 1 08:28 Desktop
drwxr-xr-x 2 venkatesh venkatesh 4096 Oct 24 23:35 Documents
drwxr-xr-x 2 venkatesh venkatesh 4096 Oct 29 22:09 Downloads
-rw-r--r-- 1 venkatesh venkatesh 8980 Oct 24 23:28 examples.desktop
-rwxrwxrwx 1 venkatesh venkatesh 2380 Sep 21 09:46 format strings
drwxrwxr-x 4 venkatesh venkatesh 4096 Oct 28 12:21 git
drwxrwxr-x 4 venkatesh venkatesh 4096 Oct 30 12:39 iota
-rw-rw-r-- 1 venkatesh venkatesh 8393371 Oct 25 17:24 linux.odt
drwxr-xr-x 3 venkatesh venkatesh 4096 Oct 25 07:29 Music
drwxrwxr-x 3 venkatesh venkatesh 4096 Nov 1 19:35 oc
drwxr-xr-x 4 venkatesh venkatesh 4096 Oct 25 22:49 Pictures
drwxr-xr-x 2 venkatesh venkatesh 4096 Oct 24 23:35 Public
-rw-rw-r-- 1 venkatesh venkatesh 2361099 Oct 27 22:16 S20180010135_practice assignment 3.odt
drwxr-xr-x 2 venkatesh venkatesh 4096 Oct 24 23:35 Templates
drwxr-xr-x 2 venkatesh venkatesh 4096 Nov 1 18:48 Videos
venkatesh@venkatesh:~$
```

12ans)

```
venkatesh@venkatesh:~/oc/reg_exp
venkatesh@venkatesh:~/oc$ mkdir reg_exp
venkatesh@venkatesh:~/oc$ cd reg_exp
venkatesh@venkatesh:~/oc/reg_exp$ cat > file1
^Z
[1]+  Stopped                  cat > file1
venkatesh@venkatesh:~/oc/reg_exp$ cat file1
Rather than teach your gadgets what to do,
Intel researchers say that in the not-too-distant future
they will learn about you on their own.
That means where you are, how you're feeling, and what you want.
It's actually not as creepy as it sounds.
Intel Chief Technology Officer and Director of Intel Labs Justin Rattner
took the stage Wednesday at the annual Intel Developer Forum
here to talk about the future of "context-aware computing,"
what Intel is doing about it, and how gadgets
can make life easier for their owners,
but in a way that the owners can control.
Context-aware computing is Intel's term for devices
that anticipate what people need or want and guide them accordingly.
The context is gathered through a combination of
"hard sensors"--cameras that detect movement and
GPS-based location information--and "soft sensors"--
such as calendar information or pieces of data you input into a device.
venkatesh@venkatesh:~/oc/reg_exp$
```

12.1 to 12.8)

```
venkatesh@venkatesh:~/oc/reg_exp$ egrep '^T' file1
That means where you are, how you're feeling, and what you want.
The context is gathered through a combination of
venkatesh@venkatesh:~/oc/reg_exp$ egrep '^t' file1
they will learn about you on their own.
took the stage Wednesday at the annual Intel Developer Forum
that anticipate what people need or want and guide them accordingly.
venkatesh@venkatesh:~/oc/reg_exp$ egrep '^G' file1
GPS-based location information--and "soft sensors"--
venkatesh@venkatesh:~/oc/reg_exp$ grep -i '^d' file1
venkatesh@venkatesh:~/oc/reg_exp$ egrep '?$' file1
Rather than teach your gadgets what to do,
Intel researchers say that in the not-too-distant future
they will learn about you on their own.
That means where you are, how you're feeling, and what you want.
It's actually not as creepy as it sounds.
Intel Chief Technology Officer and Director of Intel Labs Justin Rattner
took the stage Wednesday at the annual Intel Developer Forum
here to talk about the future of "context-aware computing,"
what Intel is doing about it, and how gadgets
can make life easier for their owners,
but in a way that the owners can control.
Context-aware computing is Intel's term for devices
that anticipate what people need or want and guide them accordingly.
The context is gathered through a combination of
"hard sensors"--cameras that detect movement and
GPS-based location information--and "soft sensors"--
such as calendar information or pieces of data you input into a device.
venkatesh@venkatesh:~/oc/reg_exp$ egrep '!' file1
venkatesh@venkatesh:~/oc/reg_exp$ egrep '#' file1
venkatesh@venkatesh:~/oc/reg_exp$ egrep '\<the\>' file1
Intel researchers say that in the not-too-distant future
took the stage Wednesday at the annual Intel Developer Forum
here to talk about the future of "context-aware computing,"
but in a way that the owners can control.
venkatesh@venkatesh:~/oc/reg_exp$
```

12.9 to 12.15)

```
venkatesh@venkatesh:~/oc/reg_exp$ egrep '.not' file1
Intel researchers say that in the not-too-distant future
It's actually not as creepy as it sounds.
venkatesh@venkatesh:~/oc/reg_exp$ egrep '[?!"]' file1
venkatesh@venkatesh:~/oc/reg_exp$ egrep '$[?!"]' file1
venkatesh@venkatesh:~/oc/reg_exp$ egrep '[?!"$]' file1
venkatesh@venkatesh:~/oc/reg_exp$ egrep '[^@!#%.A-Za-z]' file1
Rather than teach your gadgets what to do,
Intel researchers say that in the not-too-distant future
they will learn about you on their own.
That means where you are, how you're feeling, and what you want.
It's actually not as creepy as it sounds.
Intel Chief Technology Officer and Director of Intel Labs Justin Rattner
took the stage Wednesday at the annual Intel Developer Forum
here to talk about the future of "context-aware computing,"
what Intel is doing about it, and how gadgets
can make life easier for their owners,
but in a way that the owners can control.
Context-aware computing is Intel's term for devices
that anticipate what people need or want and guide them accordingly.
The context is gathered through a combination of
"hard sensors"--cameras that detect movement and
GPS-based location information--and "soft sensors"--
such as calendar information or pieces of data you input into a device.
venkatesh@venkatesh:~/oc/reg_exp$ egrep -i 'in a' file1
but in a way that the owners can control.
venkatesh@venkatesh:~/oc/reg_exp$ egrep 'e{2}' file1
That means where you are, how you're feeling, and what you want.
It's actually not as creepy as it sounds.
that anticipate what people need or want and guide them accordingly.
venkatesh@venkatesh:~/oc/reg_exp$
```

‘^T’ used to show the content of the file starts with T
 ‘^t’ used to show the content of the file starts with t
 ‘^G’ used to show the content of the file starts with G
 -i ‘^d’ used to show the content of the file starts with d without case distinction
 ‘.not’ used to show all the content of the file highlighting (not)
 ‘\<the\>’ used to highlights the content of the file (the)
 ‘e{2}’ used to show all the content of the file highlighting (ee)

13ans)

compress the file using **zip** command and to uncompress the file using **gunzip** command

```

4 big.file
venkatesh@venkatesh:~/oc$ ls -l big.file
-rw-rw-r-- 1 venkatesh venkatesh 443 Nov  1 21:31 big.file
venkatesh@venkatesh:~/oc$ gzip big.file
venkatesh@venkatesh:~/oc$ ls -l big.file
ls: cannot access 'big.file': No such file or directory
venkatesh@venkatesh:~/oc$ ls -l big.file.gz
-rw-rw-r-- 1 venkatesh venkatesh 227 Nov  1 21:31 big.file.gz
venkatesh@venkatesh:~/oc$ unzip big.file.gz
Archive: big.file.gz
End-of-central-directory signature not found. Either this file is not
a zipfile, or it constitutes one disk of a multi-part archive. In the
latter case the central directory and zipfile comment will be found on
the last disk(s) of this archive.
unzip: cannot find zipfile directory in one of big.file.gz or
      big.file.gz.zip, and cannot find big.file.gz.ZIP, period.
venkatesh@venkatesh:~/oc$ gunzip big.file.gz
venkatesh@venkatesh:~/oc$ ls -l big.file
-rw-rw-r-- 1 venkatesh venkatesh 443 Nov  1 21:31 big.file
venkatesh@venkatesh:~/oc$ 
```

14ans)

Here we must insert the files into a file and then again we must unzip them. So as we must do for multiple files we must use tar. For unzip use must use the command **tar -xf(extracting them)**

```

venkatesh@venkatesh:~/oc
venkatesh@venkatesh:~/oc$ tar -cf venky.tar big.file public_html reg_exp
venkatesh@venkatesh:~/oc$ ls
big.file  public_html  reg_exp  trfile  venky.tar
venkatesh@venkatesh:~/oc$ tar +xf venky.tar
tar: invalid option -- '+'
Try 'tar --help' or 'tar --usage' for more information.
venkatesh@venkatesh:~/oc$ tar -xf venky.tar
venkatesh@venkatesh:~/oc$ ls
big.file  public_html  reg_exp  trfile  venky.tar
venkatesh@venkatesh:~/oc$ 
```

15ans)

- command i.e.. ps -fe is
shows all the process infirmation and their parent processor id

UID	PID	PPID	C	S	TIME	TTY	TIME	CMD
root	1	0	0	20:40	?		00:00:01	/sbin/init splash
root	2	0	0	20:40	?		00:00:00	[kthreadd]
root	3	2	0	20:40	?		00:00:00	[ksoftirqd/0]
root	5	2	0	20:40	?		00:00:00	[kworker/0:0H]
root	7	2	0	20:40	?		00:00:06	[rcu_sched]
root	8	2	0	20:40	?		00:00:00	[rcu_bh]
root	9	2	0	20:40	?		00:00:00	[migration/0]
root	10	2	0	20:40	?		00:00:00	[watchdog/0]
root	11	2	0	20:40	?		00:00:00	[watchdog/1]
root	12	2	0	20:40	?		00:00:00	[migration/1]
root	13	2	0	20:40	?		00:00:00	[ksoftirqd/1]
root	15	2	0	20:40	?		00:00:00	[kworker/1:0H]
root	16	2	0	20:40	?		00:00:00	[watchdog/2]
root	17	2	0	20:40	?		00:00:00	[migration/2]
root	18	2	0	20:40	?		00:00:00	[ksoftirqd/2]
root	20	2	0	20:40	?		00:00:00	[kworker/2:0H]
root	21	2	0	20:40	?		00:00:00	[watchdog/3]
root	22	2	0	20:40	?		00:00:00	[migration/3]
root	23	2	0	20:40	?		00:00:00	[ksoftirqd/3]
root	25	2	0	20:40	?		00:00:00	[kworker/3:0H]
root	26	2	0	20:40	?		00:00:00	[kdevtmpfs]
root	27	2	0	20:40	?		00:00:00	[netns]

b) ps -fe | more shows page by page

UID	PID	PPID	C	S	TIME	TTY	TIME	CMD
root	1	0	0	20:40	?		00:00:01	/sbin/init splash
root	2	0	0	20:40	?		00:00:00	[kthreadd]
root	3	2	0	20:40	?		00:00:00	[ksoftirqd/0]
root	5	2	0	20:40	?		00:00:00	[kworker/0:0H]
root	7	2	0	20:40	?		00:00:06	[rcu_sched]
root	8	2	0	20:40	?		00:00:00	[rcu_bh]
root	9	2	0	20:40	?		00:00:00	[migration/0]
root	10	2	0	20:40	?		00:00:00	[watchdog/0]
root	11	2	0	20:40	?		00:00:00	[watchdog/1]
root	12	2	0	20:40	?		00:00:00	[migration/1]
root	13	2	0	20:40	?		00:00:00	[ksoftirqd/1]
root	15	2	0	20:40	?		00:00:00	[kworker/1:0H]
root	16	2	0	20:40	?		00:00:00	[watchdog/2]
root	17	2	0	20:40	?		00:00:00	[migration/2]
root	18	2	0	20:40	?		00:00:00	[ksoftirqd/2]
root	20	2	0	20:40	?		00:00:00	[kworker/2:0H]
root	21	2	0	20:40	?		00:00:00	[watchdog/3]
root	22	2	0	20:40	?		00:00:00	[migration/3]
root	23	2	0	20:40	?		00:00:00	[ksoftirqd/3]
root	25	2	0	20:40	?		00:00:00	[kworker/3:0H]
root	26	2	0	20:40	?		00:00:00	[kdevtmpfs]
root	27	2	0	20:40	?		00:00:00	[netns]

--More--

c & d) here the output of first command will be sent to first command.

```
venkatesh@venkatesh:~/oc$ ps -fe | grep "/broot/b" | more
venkatesh@venkatesh:~/oc$ cd ..
venkatesh@venkatesh:~$ ps -fe | grep "/broot/b" | more
venkatesh 5516 5248 0 22:18 pts/0    00:00:00 grep --color=auto /broot/b
venkatesh@venkatesh:~$ ps -fe | grep "/broot/b" | sort | more
venkatesh 5526 5248 0 22:19 pts/0    00:00:00 grep --color=auto /broot/b
venkatesh@venkatesh:~$
```

16ans)

- a) cat > command to store the content
- b) use cat command to verify the content
- c) use file command to find the file type
- d)use the command “tail -4 stuff| head -3|cat > stuff.new “. this is the single line command to copy required content into stuff.new .

- e) use again cat command to verify the content.
- f) use mv -i command to rename the file so that the permission for overwriting is asked.
- g) Now to transfer the lines which contain the word cow and doesn't contain cow use grep command and insert the output of first command into the cat command using pipe metacharacter.
- h) -I is used for the whole pattern insertion and use cat to store the content.i)-v is used for invert insertion and use cat command to store the content.
- j) From this we can understand that we must remove two lines which contain cow in that line to make stuff equal to cows.

```

venkatesh@venkatesh:~/oc$ mkdir ITWS1
venkatesh@venkatesh:~/oc$ cd ITWS1
venkatesh@venkatesh:~/oc/ITWS1$ cd UNIX
bash: cd: UNIX: No such file or directory
venkatesh@venkatesh:~/oc/ITWS1$ mkdir UNIX
venkatesh@venkatesh:~/oc/ITWS1$ cd UNIX
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ cat > stuff
The cow is mad
I tell you
I think the cow is looking at me
Terrible News!
I think the cow is looking at me
^Z
[1]+  Stopped                  cat > stuff
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ tail -4 stuff | head -3 | cat > stuff.new
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ cat stuff.new
I tell you
I think the cow is looking at me
Terrible News!
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ mv -i stuff.new stuff
mv: overwrite 'stuff'? Y
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ grep -i "cow" stuff | cat > cows
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ grep -v "cow" stuff | cat > not_cows
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ cat cows
I think the cow is looking at me
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ cat not_cows
I tell you
Terrible News!
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ diff cows not_cows
1c1,2
< I think the cow is looking at me
---
> I tell you
> Terrible News!
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ diff stuff cows
1d0
< I tell you
3d1
< Terrible News!
venkatesh@venkatesh:~/oc/ITWS1/UNIX$ 
```

17.ans) Make the directory using the given command.

Then again use the another given command to change the directory.

Now use touch command to create the given files.

Then execute the commands one by one.

? shows a single character

* shows multiple characters

[] is a option character

! is a negation option shows except it

```
venkatesh@venkatesh:~/oc/ITWS1/UNIX\$ cd ..
venkatesh@venkatesh:~/oc/ITWS1\$ mkdir file_practice
venkatesh@venkatesh:~/oc/ITWS1\$ cd file_practice
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ touch a1 a11 a12 a23 a123 a1234 a5
43 a321 bab2 cab5 12abc12 1a1 abc1 abcdef1 1 12 123
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls
1 123 1a1 a11 a123 a23 a543 abcdef1 cab5
12 12abc12 a1 a12 a1234 a321 abc1 bab2
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls *
1 123 1a1 a11 a123 a23 a543 abcdef1 cab5
12 12abc12 a1 a12 a1234 a321 abc1 bab2
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls *1*
1 12 123 12abc12 1a1 a1 a11 a12 a123 a1234 a321 abc1 abcdef1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls ?
1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls ???
12 a1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls a[1]
a1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls a[123456789]
a1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls a[123456789]*
a1 a11 a12 a123 a1234 a23 a321 a543
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls [a][1][2]*
a12 a123 a1234
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls [a-z]*3
a123 a23 a543
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls a*[1]
a1 a11 a321 abc1 abcdef1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls a?*[1]
a11 a321 abc1 abcdef1
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ ls *[!1]
12 123 12abc12 a12 a123 a1234 a23 a543 bab2 cab5
venkatesh@venkatesh:~/oc/ITWS1/file_practice\$ file [a-z]*3
a123: empty
a23: empty
a543: empty
venkatesh@venkatesh:~/oc/ITWS1/file_practice$
```

18ans)

- a)use **gg** command to jump to the first line
- b)**ctrl + f** to scroll down by one page
- c) first we have to use **60gg** and then go to the 5th character and press **x** to delete the character
- d) use **60gg** and then press **o** and input text **previous changed**.
- e) press **esc** and then **:wq tmp**
- f) use **gg** command and then press **I** in a new line and add **WARNING:**
- g) press **w** in a new line
- h) press **dw** and then **changed** to write **modified**
- i) **\<kill\>** command finds kill and then **5dd** to delete 5lines from there
- j) **G** to go to last line
- k) use **gg** command at first and then place cursor at the point then press **15x**
- l) use **/bash** to find bash

- m) cursor at **s** then press ~
- n) **50gg** and **10yy** to copy all the lines between 50 to 60 and then press **p** to paste it
- o) **G** to move to end of the file and then press **p** to paste it
- p) press **G** to go to last line and then press **9dk** to delete the last 10 lines of the file
- q) **:wq** to save and quit
- r) to open tmp use command **vi tmp**
- s) keeping cursor at **S** press ~
- t) **:w** to save the file.