1. Check the output of the following commands:

Date – for date

Ls – to see all file and folder

Who- list all user logged in

Cal – print calender

Ps- program that are currently running

Wc- word count, use for no of lines too it has four column by default 1.lines in file 2.no of words 3.no of characters 4.name of file

Cat – create file

Uname- prints basic info about os name and system hardwere

Pwd – print working direcotry

Mkdir – create folder

Rmdir – delete folder

Cd- change current location

Cp – copy file

Rm- remove file and folder with files

Mv – rename file and move file

Diff – display difference of files by comparing line by line (cmp)

Chmod – change permission 4-read 2-write 1-exceute

Grep- search particular pattern

Sed -

Head – print top n numbers of data default 10

Tail – print last n numbers of data default 10

Cut – cut section form outputs –b(byte)-c(column)-f(field)

Paste- by default (-d delimiter) horizontal –s vertical

Sort – -r for desending

Find – use to find files and directories

Man – for help

2. Write a script to find the complete path for any file

```
read -p "enter file" i
readlink -f $i     #this command is use for file path
```

3. Write a shell script to execute following commands

- **Sort file abc.txt and save this sorted file in xyz.txt**
- **Give an example of : To execute commands together without affecting result of eachother.**
- **How to print “thisis a**

  three –line

  1. Textmessage”
- **Which command display version of the UNIX?**
- **How would you get online help of cat command?**

```
echo 1.sort file abc and store in xyz
echo 2.excute command without effect result
echo 3.three line message
echo 4.version of linux
echo 5.online help of cat command
read -p "enter choice " ch
case $ch in
1)sort a.txt>b.txt;;
2)date;cal;;
3)printf "this is
        a three line
        \t 1.text message";;
4)uname -r;; #uname -a
5)man cat;;
*)echo invalid choise
Esac
```

4. Write a shell script to execute following commands

• How would u display the hidden files?

• How delete directory with files?

• How would user can do interactive copying?

• How would user can do interactive deletion offiles?

• Explain two functionality of "mv" command with example?

```
echo 1.display hidden file
echo 2.delete directory with files
echo 3.interactive copy
echo 4.interactive deletion of file
echo 5.explain two functions of mv

read -p "coice " ch
case $ch in
1)ls .[a-z]*;; #ls -a
2)rm -r abc;;
3)cp -i abc.txt pqr.txt;;
4)rm -i pqr.txt;;
5)echo 1.rename
echo 2.move
read -p "enter choice" ch2
case $ch2 in
1)mv abc.txt pqr.txt;;
2) mv pqr.txt move/
esac;;
*)echo invalid
Esac
```

5. Write a shell script to execute following commands

• Create a file called text and store name,age and address in it.

• Display the contents of the file text on the screen.

• Delete the directories mydir and newdir at one shot.

• Sort a numeric file?

• Change the permissions for the file newtext to 666

```
echo 1.create file
echo 2.display file
echo 3.delete mydir and newdir at one shot
echo 4.sort numeric file num.txt
echo 5.change permission for newtext
read -p "choice " ch
case $ch in
1)read -p "name " name
read -p "age " age
read -p "address " add
echo "name : $name  age : $age  address : $add ">text.txt;;
2)cat text.txt;;
3)rm -r mydir newdir ;;#rmdir mydir newdir
4)sort -n num.txt;;
5)chmod 666 newtext.txt;;#4-read 2-write 1-execute
*)echo invalid choice
Esac
```

6. Write shell script that accept filename and displays last modification time if file exists, otherwise display appropriate message.

```
read -p "enter file : " f1
if [ -f $f1 ] #f for file
then
echo file time
ls -l $f1 | cut -c 45-57
stat -c %y $f1 # 2 method %y use for formating
else
echo file not exist
fi
```

7. Write a shell script to display the login names that begin with 's'

```
who | grep s* #grep for search pattern
```

8. Write a shell script to remove the zero sized file from the current directory

```
for i in * # or `ls`
do
if [ ! -s $i ] #-s for check file is greater than zero
then
rm $i
echo $i removed
fi
done
```

9. Write a shell script to display the name of all the executable file from the current directory.

```
for i in *
do
if [ -x $i ] #-x executable file
then
echo $i
fi
done
```

10. Write a shell script that will display welcome message according to time

```
d=`date +%H` #time in hour
if [ $d -lt 12 -a $d -gt 5 ]
then
echo good morning
elif [ $d -lt 17 -a $d -gt 12 ]
then
echo good afternoon
elif [ $d -gt 17 -a $d -lt 22 ]
then
echo good evening
else
echo good night
fi
```

11. Write a shell script to find number of ordinary files and directory files.

```
d=0
f=0
for i in *
do
if [ -f $i ] # -f for check file
then
f=`expr $f + 1`
fi
if [ -d $i ] #-d for check directory
then
d=`expr $d + 1`
fi
```

done

echo files $f

echo directory $d

12. Write a shell script that takes a filename from the command line and checks whether the file is an ordinary file or not.

• If it is an ordinary file then it should display the contents of the file.

• If it is not an ordinary file then script should display the message: " File does not exist or is not ordinary, cannot display."

if [-f $1] #$1 command line first argument

then

cat $1

else

echo File does not exist or is not ordinary, cannot display

fi

13. Write a shell script that takes a filename from the user and checks whether it is a directory file or not.

• If it is a directory, then the script should display the contents of the directory.

• If it is not a directory file then script should display the message: "File is not a directory file"

read -p "enter file : " f1

if [-d $f1]

then

ls $f1

else

echo File is not a directory file

fi

14. Write a shell script that takes a filename as an argument and checks if the file exists and is executable.

• If the file is executable then the shell script should display the message: “File exists”

• If the file does not exists and is not executable then the script should display the message: “File does not exist or is not executable.”

```
read -p "enter your file : " fn
if [ -e $fn -a -x $fn ] # -e exist -x executable
then
echo file exists
else
echo file does not exist or is not executable
fi
```

15. Write a shell script that displays all subdirectories in current working directory.

```
for i in *
do
if [ -d $i ]
then
echo $i
fi
done
# or one line code
# ls -F1 | grep /  (-F1 for all file vertical & grep / for file ends with /
```

16. Write a shell script that calculates the number of ordinary and directory files in your current working directory.

Same as program 11

17. Write a shell script that accepts 2 filenames and checks if both exists; if both exist then append the content of the second file into the first file.

```
read -p "first file : " f1
read -p "second file : " f2
if [ -e $f1 -a -e $f2 ]
then
cat $f2>>$f1
else
echo files not exist
fi
```

18. Write a shell script that takes the name of two files as arguments and performs the following: i. Displays the message : "Displaying the contents of file :(first argument)" and displays the contents page wise.

ii. Copies the contents of the first argument to second argument.

iii. Finally displays the message: "File copied successfully."

```
echo displaying the content of file:
cat $1
cp $1 $2
echo file copied successfully
```

19. Write a shell script to display the following menu and acts accordingly:

i. Calendar of the current month and year.

ii. Display "Good Morning/Good Afternoon/Good Evening" according to the current login time.

iii. User name, Users home directory.

iv. Terminal name, Terminal type.

v. Machine name.

vi. No. of users who are currently logged in; List of users who are currently logged in.

```
echo 1.calender
echo 2.daisplay gm/gn/ge
echo 3.user name user home directory
echo 4.terminal name,terminal type
echo 5.machine name
echo 6.no of uers currently logged in
read -p "choice : " ch
case $ch in
1)cal;;
2)d=`date +%H`
if [ $d -lt 12 ]
then
echo gm
else
echo gn
fi;;
3)echo username `uname -a | cut -d " " -f2` #or $USER
echo home directory $HOME ;;
4)tty ;;
5)uname -m ;;
6)who | wc -l ;; # who -H also used
*)echo ivalid
Esac
```

20. Write a shell script that displays the following menu and acts accordingly

1. Concatenates two strings

2. Renames a file

3. Deletes a file.

4. Copy the file to specific location

```
echo 1.concate two string
echo 2. Rename a file
echo 3. Delete a file
echo 4. copy file to specific location
read -p "choice : " ch
case $ch in
1)read -p "string1 " s1
read -p "string2 " s2
echo $s1$s2 ;;
2) read -p "file for rename " f1
read -p "new name " f2
mv $f1 $f2 ;;
3)read -p "file for delete " f1
rm $f1 ;;
4)read -p "copy file " c1
read -p "destination " d1
cp $c1 $d1/ ;;
*) echo invalid
Esac
```

21. Write a shell script to change the suffix of all your *.txt files to .dat

```
for i in *.txt
do
a=`basename $i .txt` # `echo $i| cut -d "." -f1`
mv $i $a.dat
done
```

22. Write a shell script to accept a directory-name and display its contents.

If input is not given then HOME directory's contents should be listed. (Make use of command line argument)

```
if [ $1 ]
then
ls $1
else
ls $HOME
fi
```

23. Write a shell script to get all files of home directory and rename them if their names start with c. Newname = oldname111

```
cd $HOME
for i in c*
do
j="${i}111"
mv $i $j
done
```

24. Write a shell script that takes two filename as arguments. It should check whether the contents of two files are same or not, if they are same then second file should be deleted.

```
read -p "file 1 " f1
read -p "file 2 " f2
if cmp $f1 $f2
then
echo same
rm $f2
else
echo diff
fi
```

25. Write a shell script that accepts two directory names from the command line and copies all the files of one directory to another. The script should do the following

▪ If the source directory does not exist, flash a error message

▪ If destination directory does not exist create it

▪ Once both exist copy all the files from source directory to destination directory

```
if [ ! -d $1 ]
then
echo source not exist
else
if [ ! -d $2 ]
then
mkdir $2
fi
cp -R $1 $2
echo done
fi
```

26. Write a shell script that displays the following menu

▪ List home directory

▪ Date

▪ Print working directory

▪ Users loggedin

```
echo 1.list home dir
echo 2.date
echo 3.print working dir
echo 4.user logged in
read -p "choice : " ch
case $ch in
1)echo $HOME;;
2)date +"%d %h %Y";;
3)pwd ;;
4)who ;; # or echo $USER
```

*)echo invlid

Esac

27. Write a shell script that displays all hidden files in current directory.

ls .[a-z]* # ls –a

28. Write a shell script that Combine two files in the third file horizontally and vertically.

read -p "file 1 " f1

read -p "file 2 " f2

echo horizo

paste $f1 $f2

echo verti

paste -s $f1 $f2

29. Write a shell script to delete all the spaces from a given file

read -p "file name " f1

cat $f1 | tr -d " "#'[: :]'tr for translate charactor -d for delete

30. Write a shell script to find a given date fall on a weekday or a weekend.

if [$# -ne 1]

then

echo date in yyyy-mm-dd format

else

d1=`date -d"$1" +%u`

if [$d1 -eq 6 -o $d1 -eq 7]

then

echo weekend

else

echo week day

fi

fi

31. Write a shell script to search for a given word in all the files given as the arguments on the command line

```
read -p "enter word for search : " wr
for i in $@ #no of given comand line argument
do
grep $wr $i
c=`echo $? `
if [ $c = 0 ]
then
echo $i
fi
done
```

32. Write a shell script that display last modified file in the current directory.

```
ls -lt | head -2| tail -1
```

33. Write a script to display the permissions of the particular file.

```
read -p "enter file " f1
ls -l $f1|cut -c 1-10
```

35. Write a shell script to display the following menu for a particular file :

i. Display all the words of a file in ascending order.

ii. Display a file in descending order.

iii. Toggle all the characters in the file.

Iv. Display type of the file

```
read -p "enter file " f1
echo 1.asencding order
echo 2.desending order
echo 3.toggle
echo 4.types of file
read -p "enter choice " ch
case $ch in
1)sort $f1;;
2)sort -r $f1;;
3)cat $f1| tr "[a-z] [A-Z]" "[A-Z] [a-z]";;
4)file $f1;;
*)echo invalid
Esac
```

37. Write a shell script to display the following menu for a particular file:

i. Display all the words of a file in ascending order.

ii. Display a file in descending order.

iii. Display a file in reerse order.

iv. Toggle all the characters in the file

v. Display type of the file.

```
echo reerse order
rev filename
```

38. Write a shell script to find total no. Of users and finds out how many of them are currently logged in.

```
echo total user

cat /etc/passwd | wc -l

echo currnetly log in

who | wc –l
```

39. Write a shell script that displays the directory information in the following format Filename Size Date Protection Owner

```
echo -e "filename\tsize\tdate\tprotection\towner"

for i in *

do

fn=`ls -l $i | cut -d ' ' -f 22`

s=`ls -l $i | cut -d ' ' -f 18`

d1=`ls -l $i | cut -d ' ' -f 19`

d2=`ls -l $i | cut -d ' ' -f 20`

p=`ls -l $i | cut -d ' ' -f 1`

o=`ls -l $i | cut -d ' ' -f 6`

echo -e "$fn\t$s\t$d1 $d2\t$p\t$o"

done
```

40. Write a shell script to display five largest files from the current directory

```
ls -S| head -5
```