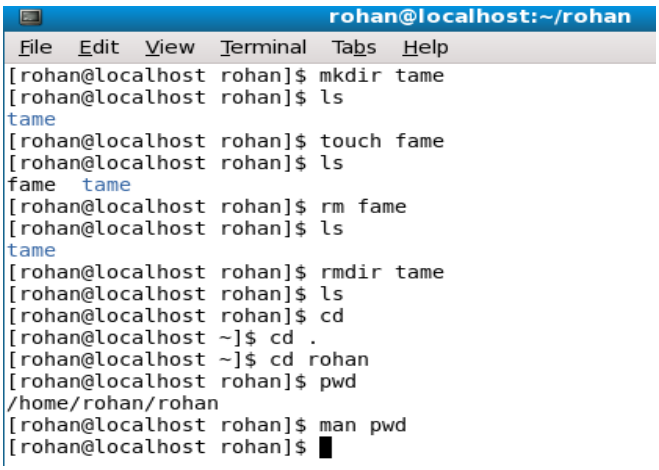
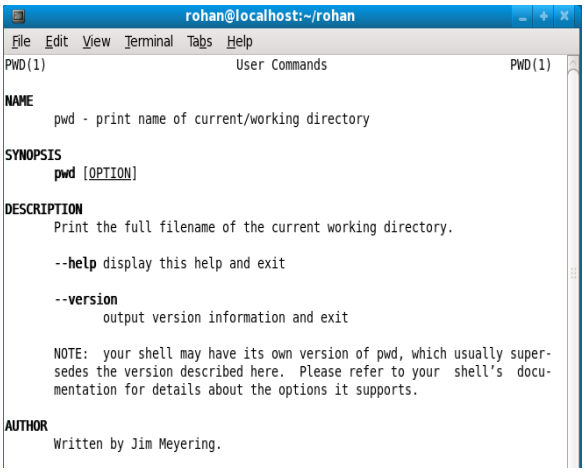
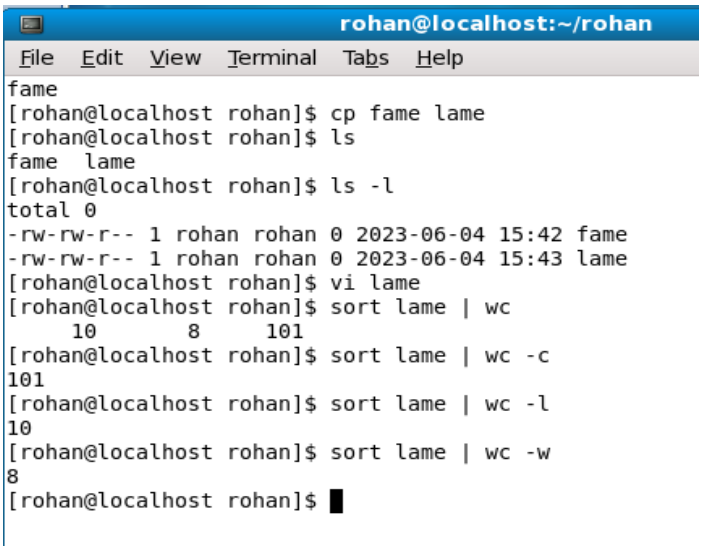
2. Explain Linux comments related to directory and files with syntax and example

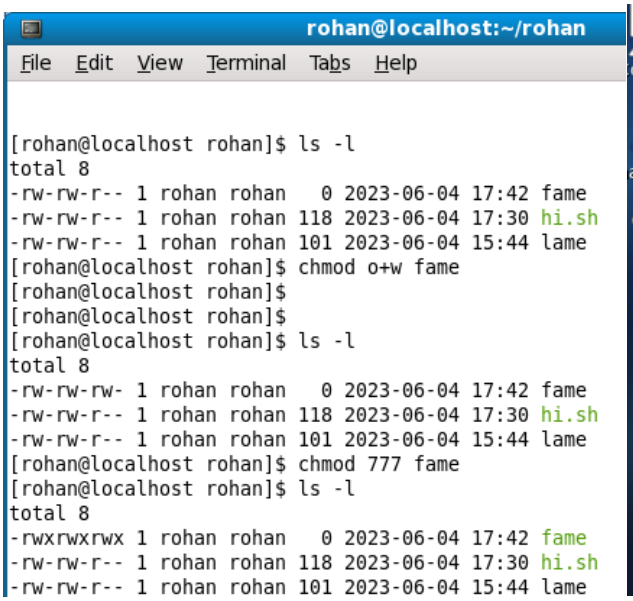


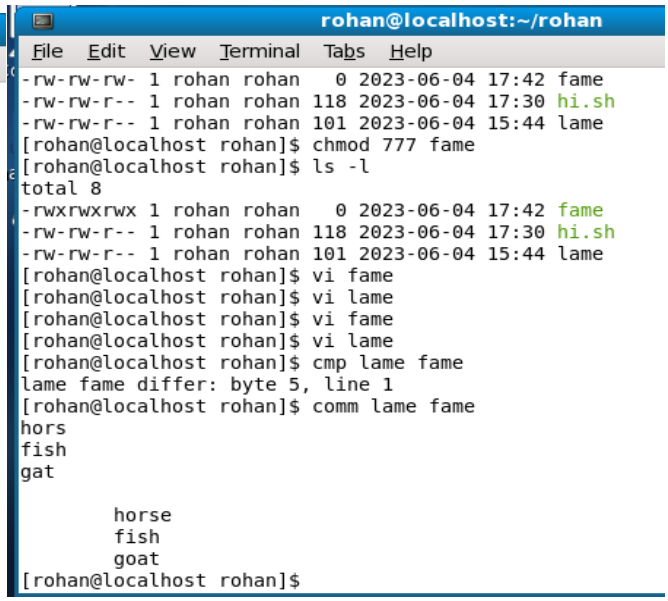


3. Explain Linux comments for copy move list the files along with usage of pipe command



4. Explain Linux Command related to change file permission and compare file





5. Explain linux commands related to text processing

```
Applications  Places  System
                                                    roha
File  Edit  View  Terminal  Tabs  Help
[rohan@localhost rohan]$ ls
fame  hi.sh  lame
[rohan@localhost rohan]$ cat lame
horse
fish
goat
horse
horse
horse
fish
goat
fish
goat
fish
goat
[rohan@localhost rohan]$ sort lame
fish
fish
fish
fish
goat
goat
goat
goat
horse
horse
horse
horse
[rohan@localhost rohan]$
```

```
[rohan@localhost rohan]$ uniq lame
horse
fish
goat
horse
fish
goat
fish
goat
fish
goat
[rohan@localhost rohan]$ cut -c 1-3 lame
hor
fis
goa
hor
hor
hor
fis
goa
fis
goa
fis
goa
[rohan@localhost rohan]$
```

```
[kvtcs@localhost rohan]$ paste rohan fame
asif    charan
rohan   chethan
juned   darshan
[kvtcs@localhost rohan]$
```

```
[rohan@localhost rohan]$ head lame
horse
fish
goat
horse
horse
horse
fish
goat
fish
goat
[rohan@localhost rohan]$ tail lame
goat
horse
horse
horse
fish
goat
fish
goat
fish
goat
[rohan@localhost rohan]$ S
rohan@localhost:~/r...
```

6. Explain linux commands related processs management

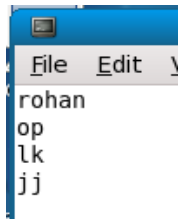
```
[kvtcs@localhost rohan]$ ps
  PID TTY          TIME CMD
 2490 pts/1    00:00:00 bash
 2900 pts/1    00:00:00 cat
 2904 pts/1    00:00:00 ps
[kvtcs@localhost rohan]$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     16) SIGSTKFLT
17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU
25) SIGXFSZ     26) SIGVTALRM   27) SIGPROF     28) SIGWINCH
29) SIGIO       30) SIGPWR      31) SIGSYS      34) SIGRTMIN
35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3  38) SIGRTMIN+4
39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6
59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

```
[kvtcs@localhost rohan]$ kill -9 2900
[kvtcs@localhost rohan]$ ps
  PID TTY          TIME CMD
 2490 pts/1    00:00:00 bash
 3267 pts/1    00:00:00 ps
[1]+  Killed                  cat > tame
[kvtcs@localhost rohan]$ pkill -9 3267
[kvtcs@localhost rohan]$ ps
  PID TTY          TIME CMD
 2490 pts/1    00:00:00 bash
 4795 pts/1    00:00:00 ps
[kvtcs@localhost rohan]$
```

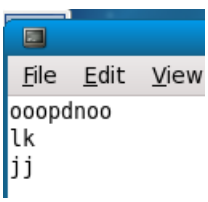
7. Explain the different modes of execution of visual editor (vi) used in linux and command used in command mode (a , i , esc , dd, yy , p , :w , :q , :wq, :q! , :wq!)

Insert (i)

esc

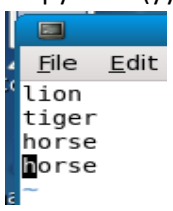


Delete a line(dd)

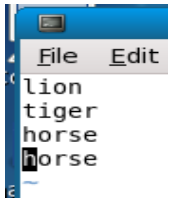


append (a)

copy a line(yy)



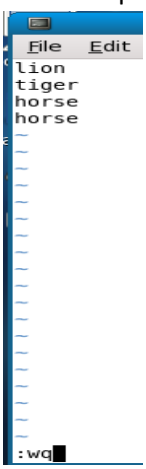
Paste(p)



write (:w)

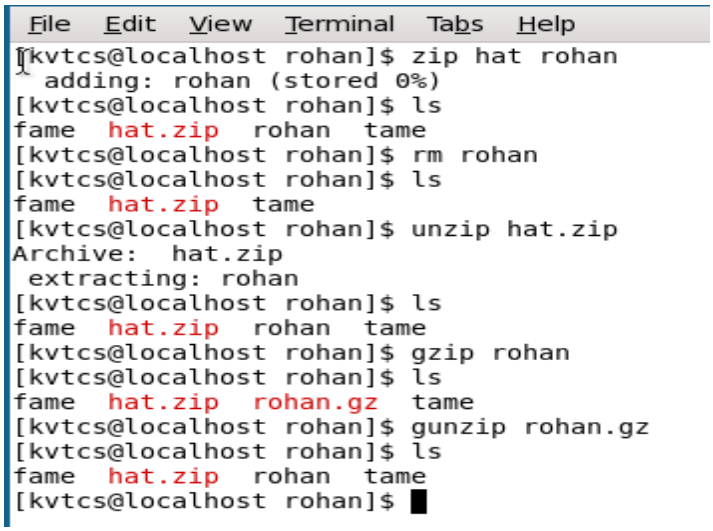
quit (:q)

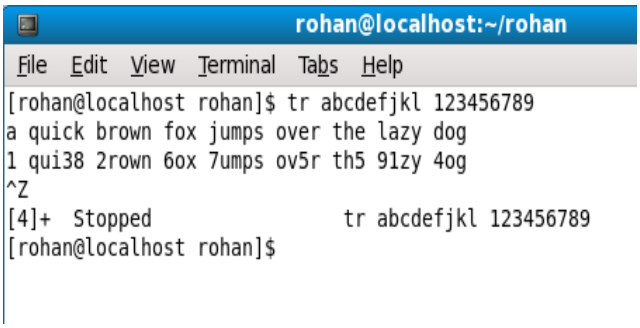
write & quit (:wq)



Override Quit (q!)

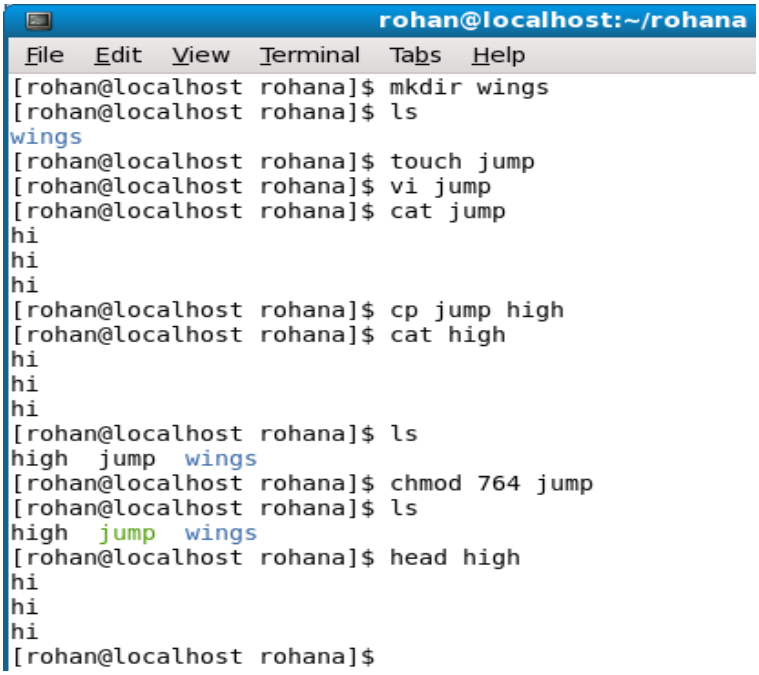
8. Explain linux commands related to compression and decompression and translation commands searching commands





9. Explain the following commands with syntax and example

i) mkdir ii) ls iii) cp iv) chmood v)head



10. Explain the following commands with syntax and example

rmdir, uniq, cut,tail,ps

```
[rohan@localhost rohana]$ ls
high  jump  wings
[rohan@localhost rohana]$ rmdir wings
[rohan@localhost rohana]$ ls
high  jump
[rohan@localhost rohana]$ cat jump
hi
hi
h
i
hlo
hlo
hlohlo
[rohan@localhost rohana]$ uniq jump
hi
h
i
hlo
hlohlo
```

```
[rohan@localhost rohana]$ cut -c 1-2 jump
hi
hi
h
i
hl
hl
hl
[rohan@localhost rohana]$ tail jump
hi
hi
h
i
hlo
hlo
hlohlo
[rohan@localhost rohana]$ ps
  PID TTY          TIME CMD
 2455 pts/1    00:00:00 bash
 3095 pts/1    00:00:00 tr
 3208 pts/1    00:00:00 tr
 3234 pts/1    00:00:00 tr
 3541 pts/1    00:00:00 cut
 3570 pts/1    00:00:00 ps
[rohan@localhost rohana]$
```

11. Explain the following commands with syntax and example

i)touch ii) ls-l iii) mv iv) cmp v)sort

```
File  Edit  View  Terminal  Tabs  Help
[rohan@localhost rohana]$ ls
high  jump  king
[rohan@localhost rohana]$ touch rohan
[rohan@localhost rohana]$ ls
high  jump  king  rohan
[rohan@localhost rohana]$ ls -l
total 8
-rw-rw-r-- 1 rohan rohan  9 2023-06-14 19:51 high
-rwxrw-r-- 1 rohan rohan 25 2023-06-14 19:59 jump
-rw-rw-r-- 1 rohan rohan  0 2023-06-14 20:11 king
-rw-rw-r-- 1 rohan rohan  0 2023-06-14 20:12 rohan
[rohan@localhost rohana]$ mv king queen
[rohan@localhost rohana]$ ls
high  jump  queen  rohan
[rohan@localhost rohana]$ cmp high jump
high jump differ: byte 8, line 3
[rohan@localhost rohana]$ sort jump
h
hi
hi
hlo
hlo
hlohlo
i
[rohan@localhost rohana]$
```

12. Explain the following commands with syntax and example
rm ,man, tr, date, who am i

```
rohan@localhost:~/rohana
File  Edit  View  Terminal  Tabs  Help
[rohan@localhost rohana]$ ls
high  jump  queen  rohan
[rohan@localhost rohana]$ rm queen
[rohan@localhost rohana]$ ls
high  jump  rohan
[rohan@localhost rohana]$ man kill
[rohan@localhost rohana]$
[rohan@localhost rohana]$ tr abcdefyio 1233456789
a quick brown fox jumps over the lazy dog
1 qu73k 2r8wn 58x jumps 8v4r th4 l1z6 38g
^Z
[6]+  Stopped                 tr abcdefyio 1233456789
[rohan@localhost rohana]$ date
Wed Jun 14 20:16:38 IST 2023
[rohan@localhost rohana]$ who am i
rohan    pts/1        2023-06-14 18:53 (:0.0)
[rohan@localhost rohana]$
```

13. write a shell script to illustrate if……fi statement

```
kvtcs@l
File  Edit  View  Terminal  Tabs  Help
#if fi statement
echo "enter any number"
read a
echo "enter another number"
read b
if [ $a = $b ]
then
echo "the numbers are equal"
else
echo "the number are not equal"
fi
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Output:-

```
kvtcs@
File  Edit  View  Terminal  Tabs  Help
[kvtcs@localhost ~]$ vi rohan1.sh
[kvtcs@localhost ~]$ sh rohan1.sh
enter any number
1
enter another number
2
the number are not equal
[kvtcs@localhost ~]$ sh rohan1.sh
enter any number
2
enter another number
2
the numbers are equal
[kvtcs@localhost ~]$
```

14. write a shell script to check to given number is odd or even if…….fi statement

Output:-

```
# numbers are odd or even
echo "enter the number"
read n
if [ $((n%2)) == 0 ]
then
echo "the number is even"
else
echo "the number is odd"
fi
```

```
[kvtcs@localhost ~]$ vi rohan2.sh
[kvtcs@localhost ~]$ sh rohan2.sh
enter the number
2
the number is even
[kvtcs@localhost ~]$ sh rohan2.sh
enter the number
3
the number is odd
[kvtcs@localhost ~]$
```

15. write a shell script to check the given character is vowel or not using case ……..esac statement

```
# the charaters vowel or not
echo "enter any charather"
read ch
case $ch in
"a") echo $ch is a vowel;;
"e") echo $ch is a vowel;;
"i") echo $ch is a vowel;;
"o") echo $ch is a vowel;;
"u") echo $ch is a vowel;;
*) echo $ch is not a vowel;;
esac
~
~
```

output:-

```
[kvtcs@localhost ~]$ vi rohan4.sh
[kvtcs@localhost ~]$ sh rohan4.sh
enter any charather
a
a is a vowel
[kvtcs@localhost ~]$ sh rohan4.sh
enter any charather
b
b is not a vowel
[kvtcs@localhost ~]$
```

16. write a shell script to check the two strings are equal or not

```
# string  statement
echo "enter first string"
read s1
echo "enter second string"
read s2
if [ $s1 == $s2 ]
then
echo "$s1 and $s2 are equal"
else
echo "$s1 and $s2 are not equal"
fi
~
```

output:-

```
[kvtcs@localhost ~]$ vi rohan3.sh
[kvtcs@localhost ~]$ sh rohan3.sh
enter first string
rohan
enter second string
rohan
rohan and rohan are equal
[kvtcs@localhost ~]$ sh rohan3.sh
enter first string
tame
enter second string
fame
tame and fame are not equal
[kvtcs@localhost ~]$
```

17. write a shell script to find the biggest of three numbers using if…….elif……..fi….. stamement

```
# greater of three numbers
echo "enter the 1st number"
read a
echo "enter the 2nd number"
read b
echo "enter the 3rd number"
read c
if [ $a -gt $b -a $a -gt $c ]
then
echo "$a is greater"
elif [ $b -gt $a -a $b -gt $c ]
then
echo "$b is greater"
else
echo "$c is greater"
fi
~
~
```

output:-

```
[kvtcs@localhost ~]$ vi rohan5.sh
[kvtcs@localhost ~]$ sh rohan5.sh
enter the 1st number
1
enter the 2nd number
5
enter the 3rd number
7
7 is greater
[kvtcs@localhost ~]$
```

18. write a shell script to implement menu driven program to perfrom all arithematic operation using case statement

```
echo "enter first number"
read a
echo "enter second number"
read b
echo -e "1 for add , 2 for sub, 3 for mul, 4 for div 5 for modulus "
read ch
case $ch in
"1") echo "sum= $(expr $a + $b)";;
"2") echo "sub= $(expr $a - $b)";;
"3") echo "mul= $(expr $a \* $b)";;
"4") echo "div=$(expr $a / $b)";;
"5") echo "rem= $(expr $a % $b)";;
*) echo "enter valid number";;
esac
```

output:-

```
[kvtcs@localhost ~]$ vi rohan6.sh
[kvtcs@localhost ~]$ sh rohan6.sh
enter first number
1
enter second number
5
1 for add , 2 for sub, 3 for mul, 4 for div 5 for modulus
3
mul= 5
[kvtcs@localhost ~]$ sh rohan6.sh
enter first number
2
enter second number
8
1 for add , 2 for sub, 3 for mul, 4 for div 5 for modulus
2
sub= -6
[kvtcs@localhost ~]$
```

19. write a shell script to print 0 to 9 using for loop

```
# 0 to 9 serirs
for var in 0 1 2 3 4 5 6 7 8 9
do
echo $var
done
~
```

Output:-

```
[kvtcs@localhost ~]$ vi rohan7.sh
[kvtcs@localhost ~]$ sh rohan7.sh
0
1
2
3
4
5
6
7
8
9
[kvtcs@localhost ~]$
```

20. write a shell script to find factorial of numbers using loop

```
# factriol of given numbers
echo "enter the numbers"
read n
fact=1
while [ $n -gt 1 ]
do
fact=$((fact*n))
n=$((n-1))
done
echo $fact
~
```

output:-

```
[kvtcs@localhost ~]$ vi rohan8.sh
[kvtcs@localhost ~]$ sh rohan8.sh
enter the numbers
5
120
[kvtcs@localhost ~]$
```

21. write a shell script to print multiplication table of given numbers using while statement

```
echo "enter a number"
read n
echo "table"
i=1
while [ $i -le 10 ]
do
m=$(expr $n \* $i )
echo "$n * $i = $m"
i=$((i+1))
done
```

output:-

```
[kvtcs@localhost ~]$ vi rohan9.sh
[kvtcs@localhost ~]$ sh rohan9.sh
enter a number
9
table
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
[kvtcs@localhost ~]$
```

22. write a shell script to generate Fibonacci series

```
echo "enter a number of terms "
read n
echo "fibonacci series "
f1=0
f2=1
echo $f1
echo $f2
i=2
while [ $i -lt $n ]
do
f3=$(( $f1 + $f2 ))
echo $f3
f1=$f2
f2=$f3
i=$(( i + 1 ))
done
~
```

output:-

```
[kvtcs@localhost ~]$ vi rohan10.sh
[kvtcs@localhost ~]$ sh rohan10.sh
enter a number of terms
5
fibonacci series
0
1
1
2
3
[kvtcs@localhost ~]$
```

23. write a shell script to perfrom file operations (copy,rename,remove,exit)

```
# shell script for file operations
while :
do
echo "Menu"
echo "1.copy a file"
echo "2.rename a file"
echo "3.delete a file"
echo "4.exit"
echo "enter your choice"
read ch
case $ch in
1) echo "enter file name to create copy"
read f1
echo "enter name for copy"
read f2
if [ -f $f1 ]
then
cp $f1 $f2
echo "$f1 copied $f2"
else
echo "$f1 not exist"
fi
;;
2) echo "enter file name to rename file"
read f1
echo "enter name for rename"
read f2
if [ -f $f1 ]
then
mv $f1 $f2
echo "$f1 renamed $f2"
else
echo "$f1 not exist"
fi
;;
3) echo "enter file name to delete file"
read f1
if [ -f $f1 ]
then
rm -i $f1
echo "$f1 deleted"
else
echo "$f1 not exist"
fi
;;
esac
done
```

output:-

```
[kvtcs@localhost ~]$ sh asifm12.sh
Menu
1.copy a file
2.rename a file
3.delete a file
4.exit
enter your choice
1
enter file name to create copy
asifm
enter name for copy
rohan
asifm copied rohan
Menu
1.copy a file
2.rename a file
3.delete a file
4.exit
enter your choice
2
enter file name to rename file
rohan
enter name for rename
asif
rohan renamed asif
Menu
1.copy a file
2.rename a file
3.delete a file
4.exit
enter your choice
3
enter file name to delete file
asif
rm: remove regular file `asif'?
asif deleted
Menu
1.copy a file
2.rename a file
3.delete a file
4.exit
enter your choice
4
exit
[kvtcs@localhost ~]$
```

24. write a shell script to perfrom file operations (find the length ,concatenate of two strings, compare two strings, exit)

```
# shell script for string operations
while :
do
echo "Menu"
echo "1.Length of string"
echo "2.concatinate of two stings"
echo "3.compare two strings"
echo "4.exit"
echo "enter your choice:"
read ch
case $ch in
1) echo "enter your string"
read s1
let len=`expr length "$s1"`
echo "lenth of '$s1'is $len"
;;
2) echo "enter first string"
read s1
echo "enter second string"
read s2
echo "concate of two strings $s1$s2"
;;
3) echo "enter first string"
read s1
echo "enter second string"
read s2
if [ $s1 == $s2 ]
then
echo "$s1 and $s2 are equal"
else
echo "$s1 and $s2 are not equal"
fi
;;
4) echo "exit"
exit
;;
esac
done
```

output:-

```
[kvtcs@localhost ~]$ sh asifm11.sh
Menu
1.Length of string
2.concatinate of two stings
3.compare two strings
4.exit
enter your choice:
1
enter your string
rohan
lenth of 'rohan'is 5
Menu
1.Length of string
2.concatinate of two stings
3.compare two strings
4.exit
enter your choice:
2
enter first string
rohan
enter second string
a
concate of two strings rohana
Menu
1.Length of string
2.concatinate of two stings
3.compare two strings
4.exit
enter your choice:
3
enter first string
rohan
enter second string
rohana
rohan and rohana are not equal
Menu
1.Length of string
2.concatinate of two stings
3.compare two strings
4.exit
enter your choice:
4
exit
[kvtcs@localhost ~]$
```