General Commands - LAB 1

Page No. 1 | Date: 31/10/22 | YOUVA

① :- $ cal Cal - Displays Calculator

```
      October 2022
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 .  .  .  .  .  .  .
30 31
```

② :- $ date date - displays Current date

Monday 31 October 2022 11:27:42 AM IST

③ :- $ echo "hello unix" echo - Outputs the given text within " "

hello unix

④ :- $ echo "bc 10+10" | bc bc - basic Calculator

20

⑤ man - displays manual syn:- $ man man

:- $ man

⑥ ls - Lists the name of files in a particular unix directory

:- $ ls

o/p: area.h first.h . . .
big.sh first.sh . . .
comb.sh for2.sh . . .

⑦ Mkdir - It is used to create new directories

:- $ mkdir -p linux/asba

$ ls -R

o/p .:
linux:
.linux/asba

(8) cd - can be used to change into a subdirectory, move back into the parent directory, move all the way to the root directory or move to any given dir

:~ $ cd linux/asba
:~ /linux/asba $

(9) Cat - Concatenate files and print on the standard output (to view file contents)
$ Cat filename

:~ Cat asba
Is a directory

(10) pwd - print name of current/working dir

:~ $ pwd
/home/bmsce/linux/asba

(11) tty - print the file name of the terminal connected to standard i/p.

:~ $ tty
/dev/pts/0 ← terminal file name is 0

(12) Uname - print system information
:~ $ uname
Linux

(13) who - show who is logged on
:~ $ who
bmsce :0 2022-10-31 11:22 (:0)

(14) echo $PATH - display path
$ echo $PATH
/usr/local/bin:/usr/local/bin:/usr/sbin

(15) echo $SHELL

:~ $ echo $SHELL
/bin/bash

(16) wc - print newline, word, and byte counts for each file

:~ $ wc [filename]
:~ $ wc asba
wc: asba: Is a directory
0 0 0 asba

(17) rmdir - remove empty directories
rmdir dir

:~ rmdir ananya
ls
asba

mkdir ananya
ananya asba
rmdir ananya
ls
asba

(18) rm - remove files or directories
:~ rm

(19) PS - report a snapshot of the current processes
:~ $ ps

PID	TTY	TIME	CMD
2267	pts/0	00:00:00	bash
:			
6032	pts/0	00:00:00	ps

(20) type - used to describe how its argument would be translated if used as command.

∴ type -t pwd
builtin

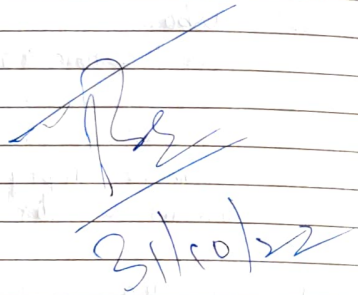
-t - display single word o/p

who am i

passwd - Assigns a login password

Clear - Clears terminal screen

mailx - is a universal mailer to send or receive mail



LAB 2

1) stty - change and print terminal line settings

$ stty
speed 38400 baud; line = 0;
-brkint -imaxbel iutf8

2) stty -a - print all current settings in human-readable form
$ stty -a

speed 38400 baud; row 24; columns 80;
line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U;
eof = ^D; eol = <undef>; . . .

3) script - make typescript of terminal session
syn: script [options] [file]
↑ filename

$ script myty1
script started, file is myty1

$ script
type all commands
script started, file is typescript
↳ at end type exit (keep logs of cmd typed)

4) ls -l :- -dereference
when showing file information for a symbolic link, show information for the file the link reference rather than for the link itself.
↑ filename

$ ls -l area
-rw-rw-r-- 1 bmsce bmsce 136 Oct 25 2021 area

$ ls -l
total 340
-rwxrwxr-x 1 bmsce bmsce 16768 Jan 3 2022 17

Change of stty

$ stty -a
intr = ^A ; quit = ^\ ;

$ stty intr ^C
$ stty -a
intr = ^C ;

To work with nano editor

$ mkdir Ananya
$ pwd
$ Ananya ~ $ nano Ananya.sh

GNU nano 4.8 Ananya.sh
#! /bin/sh
echo "Hello Unix"
echo "Ananya first Unix program"

ctrl+o, ctrl+x

To view file: sh Ananya.sh ✓ ⇒ Positional parameters / command arguments
Hello Unix
$0 $1 Ananya first Unix program

To view of: ./Ananya.sh $* => it stores all $0, $1
Hello Unix as single string
Ananya first Unix Program

To view of: sh Ananya.sh
Hello Unix
Ananya first Unix Program

#! /bin/bash
$ nano
echo "printing text with newline"
echo -n "printing text without newline"
echo -e "\n Removing \t backslash \t characters \n"

ctrl+o

filename: Ananya1.sh

ctrl+x

$ sh Ananya1.sh

printing text with newline
printing text without newline -e
Removing backslash Characters

role of printf

nano
Sum=$((30+30))
echo $Sum
$ sh Sum3.sh
60

nano
area=$(2*3)
echo $area
$ sh area2.sh
6

nano
echo "Enter your name"
read name
echo "Welcome $name to Linuxhint"
sh wel1.sh
$ $#
=> 2

Enter your name
Ananya
welcome Ananya to Linuxhint

print the no of arg

$@
$?
$$
$!

0 - pgm success 1 - pgm at sue. /errors
PID of current shell
PID of last background job

LAB 3

PROGRAM 1: LEAP YEAR OR NOT PROGRAM

Date: 14/11/22

```
#!/bin/sh

echo "Enter the year"
read y
if [ `expr $y % 4` -eq 0 ]
then
echo "the year is leap"
else
echo "the year is not leap"
fi
```

-eq
-ne
-ge
-gt
-lt
-le

```
~$ sh pg1leap.sh
Enter the year
2008
the year is leap
```

⇒ Area of Circle

```
echo "Enter radius"
read r
area=`echo 3.14\*$r\*$r |bc`
echo "$area"
```

```
~$ sh areacir.sh
Enter radius
2
12.56
```

⇒ Area of ~~Square~~ Rectangle

```
echo "Enter length"
read l
echo "Enter breadth"
read b
area=`echo $l\* $b |bc`
echo "$area"
```

```
sh area rect.sh
Enter length
4
Enter breadth
6
24
```

```
$ y=7;
$ y=9
$ x=5 ; y=7 ; z=7.2
```

⇒ Area of Square

```
echo "Enter S1"
read s1
echo "Enter S2"
read s2
area=`echo $s1\* $s2 |bc`
echo "$area"
```

```
~$ sh areasq.sh
Enter S1
2
Enter S2
2
4
```

⇒ Creating & renaming file

```
echo "Enter directory to be Created"
read nd
`mkdir $nd`
```

```
$ sh newdir.sh
Enter directory to be Created
MYdir
$ ls
MYdir
```

```
echo "Renaming File"
echo "Enter old file"
read of
echo "Enter new file"
read nf
`mv $of $nf`
echo "Renamed"
```

```
sh rename.sh
Enter old file
MYdir
Enter new file
mydir
Renamed
```

14/11/22

⇒ Positional Parameter Programs

```
echo "Positional parameters execution"
echo '$0 = '$0
echo '$1 = '$1
echo '$2 = '$2
echo '$3 = '$3
```

command line args

```
sh positional.sh 45 78 56
Positional parameters execution
$0 = positional.sh
$1 = 45
$2 = 78
$3 = 56
```

command line args

⇒ Copying source file to destination file

```
#!/bin/sh
echo "Enter Source and Target Filenames"
read source target
if cp $source $target
then
echo "File copied successfully"
else
echo "Copying of file failed"
fi
```

LAB 4 PROGRAMS

1. Write a shell program to check for leap year for all 3 conditions like divisible by 4, divisible by 400, not divisible by 100

```
echo "leap year pgm"
echo "enter the year"
read y
if [ `expr $y %4` -eq 0 ] && [ `expr $y %100` -eq 0 ] || [ `expr $y %400` -eq 0 ]
then
    echo "It is leap year"
else
    echo "Not a leap year"
fi
```

```
Enter the year
2008
It is leap year
```

2. SS to check whether the given number is +ve, -ve or zero.

```
echo "pos, neg or zero"
echo "Enter the number"
read n
if [ $n -gt 0 ]
then
    echo "Number is positive"
elif [ $n -lt 0 ]
then
    echo "num is negative"
else
    echo "num is zero"
fi
```

Pos, neg or zero
enter the number
1
Num is positive

enter the number
-1
num is negative

enter the number
0
num is zero

3. SS to print largest of given 3 numbers

```
echo "largest of 3 num"
echo "enter n1"
read n1
echo "enter n2"
read n2
echo "enter n3"
read n3
if [ $n1 -gt $n2 ]&& [ $n1 -gt $n3 ]
then
    echo "the largest number is $n1"
elif [ $n2 -gt $n1 ]&& [ $n2 -gt $n3 ]
then
    echo "the largest is $n2"
else
    echo "the largest is $n3"
fi
```

largest of 3 num
enter n1
10
enter n2
20
enter n3
30
the largest is 30

4)* 3 numbers using Command line arguments

```
echo "largest of 3 numbers cmd line arg"
echo '$0n1 = '$n1
echo '$n2 = '$n2
echo '$n3 = '$n3


if [ $n1 -gt $n2 ]&& [ $n1 -gt $n3 ]
then
    echo "the largest number is $n1"
elif [ $n2 -gt $n1 ]&& [ $n2 -gt $n3 ]
then
    echo "the largest is $n2"
else
    echo "the largest is $n3"
fi
```

⇒ largest of 3 numbers cmd 10 20 30
the largest number is 30

*5) Compare two strings using test

```
echo "enter String S1"
read S1
echo "enter S2"
read S2
if [ "$S1" = $S2 ]
  then
      echo "String is equal"
else
  then
      echo "not equal"
```

enter S1
aa
enter S2
aa
Strings are equal

*4 b) echo "Largest of 3 num using cmd line arg"
if [$# -ne 3]
then
echo "Cmd line arg is missing"
exit 1
fi
if [$1 -eq $2 -a $1 -eq $3]
then
echo "All num are equal"
elif [$1 -gt $2 -a $1 -gt $3]
then
echo [$2 -gt $1 -a $2 -gt $3]
then
echo "$2 is the biggest"
else
fi echo "$3 is the biggest"

o/p

Sh largnumcmd.sh 10 20 30
Largest of 3 num using cmd line arg
30 is the biggest

*8) echo "To find num is even or odd"
echo "Enter the number"
read n
if [$((n%2)) -eq 0]
then
echo "Num is even"
else
echo "Num is odd"
fi

o/p
Enter the number
13
Num is odd

LAB-3

Date 28/11/22 YOUVA

1. echo "SS to find the factorial of a number"
echo "factorial of number"
echo "Enter the number"
read n
fact=1
while [$n -gt 1]
do
fact=$((fact*n))
n=$((n-1))
done
echo $fact

O/P
SS Factorial of number
Enter the number
4
24

2. echo "SS to compute the gross salary of an employee"
(given basic, Hra 20% of basic, da 10% of basic)
echo "Enter the Sal of employee"
read Sal
~~hr~~='~~echo~~ hr='echo 0.2*$Sal|bc'
da='echo 0.1*$Sal|bc'
gross='echo $Sal+$hr+$da|bc'
echo $gross

O/P
Enter the Salary of employee
1200
1560.0

3. echo "SS to Convert the temperature Fahrenheit to Celsius"
echo "Cel to Fahar : F=(C*9/5)+32"
echo "Fhar to Cel : C=(F-32)*5/9"
echo "Enter the temperature in Fahrenheit"
read tf
t='echo tf-32|bc'
C='echo $t|*5|9|bc'
echo $C

Enter the temperature in Fahrenheit
52
11

4. echo "SS to find the Sum of even numbers upto n"
echo "Enter the upper limit"
read n
i=2
while test $i -lt $n
do
Sum=$((Sum+i))
i=$((i+2))
done
echo $sum

O/P
SS to find the sum of even numbers upto
Enter the upper limit
10
20

4. echo " ss to perform arithmetic operation on given two numbers using case "
```
echo "Enter Num1"
read a
echo "Enter Num2"
read b
echo "Enter operation"
read op
case $op in
+) Case='expr $a + $b' ;;
-) Case='expr $a - $b' ;;
*) Case='expr $a * $b' ;;
*) echo "Invalid ;;
esac
echo $case
```

2 pending

```
Enter num1
2
Enter num2
2
Enter the operation
+
4
```

7. echo " ss to find the sum of n natural nos"
```
echo "Enter the number"
read n
i=1
sum=0
echo "Digits are: "
while [ $i -le $n ]
do
    echo "$i"
    Sum=$(( $sum + $i ))
    i=$(( $i + 1 ))
done
echo "Sum = $Sum"
```

o/p
```
sh Sumofn.sh
Enter the number
2
Digits are:
1
2
Sum = 3
```

6. echo " ss to find the power of a number"
```
echo "Enter the base number"
read x
echo "Enter the power number"
read n
echo "$x ^ $n" | bc
echo "$pow"
```

o/p
```
Enter the base number
2
Enter the power
8
base numbers
3
Enter the power
3
27
```

LAB - 6

1. echo "SS to print the combination of numbers"
echo "Combination of 1 2 3"
i=1
j=2
k=3
for i in 1 2 3
do
for j in 1 2 3
do
for k in 1 2 3
do
echo $i $j $k
done
done
done

O/P Combination of 1 2 3

1 1 1
1 1 2
1 1 3
1 2 1
1 2 2
1 2 3
1 3 1
1 3 2
1 3 3
2 1 1
2 1 2
2 1 3
2 2 1
2 2 2
2 2 3
2 3 1
2 3 2
2 3 3
3 1 1
3 1 2
3 1 3
3 2 1
3 2 2
3 2 3
3 3 1
3 3 2
3 3 3

2. echo "SS to display the pass class of Student"
read echo "Enter the Student per"
read per
if [$per -ge 75]
then
echo "Distinction"
elif [$per -ge 60]
then
echo "First Class"
elif [$per -ge 35]
then
echo "Second Class"
elif [$per -lt 35]
then
echo "Fail"
fi

O/P
Enter the Student per
75
Distinction
Enter the Student per
60
First Class
Enter the Student per
40
Second Class
Enter the Student per
20
Fail.

3) echo "Shell Script to find the Fibonacci Series up to n"
echo "Enter the number"
read n
a=0
b=1
fib=0
echo "Fibonacci Series is :"
while [$n -gt 0]
do
echo "$a"
fib=$(($a + $b))
a=$b
b=$fib
n=$((n-1))
echo "$fib"
done

o/p
Enter the number
2
Fibonacci Series is :
0
1
1
2

o/p for 2 or 5

(a gr) Set, Shift, trap, (chmod, chown, chgrp) → Shift, trap

LAB 7

I) echo "SS for Set Command"
echo "Set"
Set -n → doesn't execute the next commands
echo "hi"
Set a b c d
echo $1
echo $2
echo $3
echo $4
→ to set positional parameters using Set cmd
→ to unset Set -
echo $5 $6 echo "hi"

o/p
+ Set a b c d
+ a b c d

II) echo "Shift"
echo "Enter the arguments : $*"
echo "Enter first parameter $1"
Shift 2
echo "First arg after Shift 2 : $1"
Shift
echo "the first arg after Shift : $1"

o/p
Sh Shift.sh a1 a2 a3 a4
Shift
Enter the arguments : a1 a2 a3 a4
Enter the first parameter a1
First arg after Shift 2 : a3
The First arg after Shift : a4

III) echo "Trap"
trap "echo The script is terminated ; exit"
while true
do
echo Test
Sleep 1
done

o/p
Test
Test
Test
^C The script is terminated

2) echo "SS to count the no of Vowels of a String"
echo "Enter the String"
read str
l = `expr length $str`
Vowel=0
while [$l -gt 0]
do
temp = `expr $str | cut -c $l`
case $temp in
a|A) Vowel = `expr $ Vowel + 1` ;;
e|E) Vowel = `expr $Vowel + 1` ;;
i|I) Vowel = `expr $Vowel +1` ;;
o|O) Vowel = `expr $Vowel +1` ;;
u|U) Vowel = `expr $Vowel +1` ;;
esac
l = `expr $l - 1`
done
echo "The String has $Vovel Vowel"

O/P
SS to count number of Vowels of a String
Enter the String
aeiou
The String has 5 Vowel
Enter the String
aa
The String has 2 vowel

3) echo "SS to Check number of lines, words, Characters in a file"
echo "Enter the file name"
read file
if [-f $file]
then
echo "File exists"
echo "Number of lines"
wc -l $file
echo "No. of words"
wc -w $file
else
echo "file does not exists"
fi

O/P
Enter the file name
hi.sh
File exists
Number of lines
1 hi.sh
No of words
2 hi.sh
Enter the file name
ab.sh
File does not exists

hi.sh
echo "hi"

O/P 2m 12/12/22

I
echo "Chmod Win"
Chown Ananya first.sh
ls -l first.txt
-rw -rw -r -- 1 bms Ananya 0 Dec 12 12:29 first.txt

I)
Chmod
Chmod +x first.txt
Chmod : slow ownership
Chown roots # Change the owner of root

II)
Chmod (relative)
Chmod 755 myfile

III
chgrp (absolut)
Chgrp abc duf

Chmod

→ Relative

$ Chmod u+x pg1

$ ls -l pg1

-rwxr--r-- 1 Ananya 2022 Dec 12-12-22 pg1

→ Absolute

$ Chmod 644 pg2

$ ls -l pg2

-rw-rw-rw 1 Ananya 2022 Dec 12-12-22 pg2

Chown - Changing File owner

Sy: Chown options owner [:group] files(s)

$ id

uid = 101 (user1)

gid = 101 (group1) $ cp f1 /tmp/user2 LA

$ Chown user2 /f1

ls -l /tmp /user2 /f1

-rw-r----- 1 usr2 clas 3966 Dec 12-12 pg1

chgrp

$ ls -l ananya.lst

-rw-r--r-- 1 Ananya 13673 Dec ananya.lst

$ chgrp Setty ananya.lst

$ ls -l ananya.lst

-rw-r--r-- 1 Ananya Setty 13675 Dec ananya.lst

Page No.

Date 19/12/22

1. #ss to create hard and soft link files

```
echo "hard link"
ln ter areasq.sh areasquare.sh
echo "Soft link"
ln -s aeiou.sh vowel.sh
```

Hard link

Soft link

$ ls -l

lrwxrwxrwx 1 bms6 bms6 9 Dec 19 11:36 areasquare.sh → areaSq.sh

$ ls -l aeiou.sh vowel.sh

lrwxrwxrwx 1 bms6 bms6 8 Dec 19 11:47 vowel.sh → aeiou.sh

-rw-rw-r-- 1 bms6 bms6 410 Dec 12 12:01 aeiou.sh

2) # ss for GCD and LCM

```
echo "Enter the first number:"
read a
echo "Enter the second number:"
read b
if [$a -gt $b]
then
num=$a
den=$b
else
num=$b
den=$a
fi
r='expr $num % $den'
while [$r -ne 0]
do
num=$den
den=$r
r='expr $num % $den'
done
```

bash ...

```
while [ $i -le 5 ];
do
    echo $i
    i=$((i+1))
```

```
gcd=$dem
lcm=`expr $a \* $b / $gcd`
echo "The LCM of $a and $b is: $lcm"
echo "The GCD of $a and $b is: $gcd"
```

o/p

```
gcd and lcm
Enter num1
10
Enter num2
20
LCM of 10 and 20 is: 20
GCD of 10 and 20 is: 10
```

3) # SS to print the below pattern

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
#!/bin/bash
num=1
rows=5
for (( i=1; i<=rows; i++))
do
    for (( j=1; j<=i; j++))
    do
        echo -n "$num"
        num=$((num+1))
    done
num=1
echo
done
```

```
i=1
while [ $i -le
do
    echo $i
    i=$((i
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
find -name "*.html" | grep -e
find -name '*.html'
```

4) # SS to locate & print from home dir # with files with .html & dir 666 per

```
find -regex '*.html'
find $dir1. -perm 666
```

o/p

```
Skill.html
Ananya [rep.sh
Ananya [vovel.sh
:
.
.
.
```

o/p ... 17/2/22

LAB-9

Date 26/12/22

3b i)
```
echo "Menu driven SS"
echo "Enter the choice"
read choice
case $choice in
1) echo "list of files 'ls'" ;;
2) echo "machine Char 'stty'" ;;
3) echo "todays date 'date'" ;;
4) echo "name of terminal $SHELL" ;;
*) echo 'exit' ;;
esac
```

O/P
```
Menu driven Shell Script
Enter the choice
1
list of files 3bi.sh
unf
aeiou.sh
:
:
Enter the choice
3
todays date Monday 26 December 2022 11:52:21 AM
Enter the choice
4
name of terminal /bin/bash
Enter the choice
*
$
```

```
Enter the choice
2
machine char speed 38400
baud; line = 0;
-brkint -imaxbel iutf8
```

3b ii)
```
echo "To generate electricity bill"
echo "Enter cname"
read cname
echo "Enter cid"
read cid
echho "Enter the units consumed"
read units ; charge=0
unit= $ if [$units -le 199]
then
  charge = $(expr $unit \* 1.50)
else if [$ units -eq 200] && [$units -g 400]
then
  charge = ($unit \* 1.50)
else if [$units -eq 400] && [$units -le 600]
then
  echo charge = ($units \* 1.80)
then
bill $ (($bill \* 0.15 + bill))
if [$ bill -lt 100]
then
  bill = 100
fi
```

O/P
```
Enter units = 200
bill : 300
```

4b)
```
echo "enter m and n"
read m n
for a in $(seq $m $n)
do
  k=0
  for i in $(seq 2 $(expr $a - 1))
  do
    if [ $(expr $a % $i) -eq 0 ]
    then
      k=1
      break
    fi
  done
  if [$k -eq 0]
  then
    echo $a
```

O/P
```
Enter m and n
3 10
3
5
7
```

Page No.
Date

(ii) echo "Enter the decimal number"
read dn
b = $(echo "obase=2 ; $dn" | bc)
echo binary $b

Op Enter the decimal number
12
binary 1100
4
binary 100

O/p
26/12/22

LAB - 10

M T W T F S S
Page No.:
Date: 2/1/23
YOUVA

$ Cat > emp.lst
2233 | a.k Shukla | g.m | Sales | 12 | 12 | 52 | 6000
9876 | jai Sharma | director | production | 03 | 12 | 50 |
5678 | Sunil Chakrabarty | d.g.m | marketing | 04 | 9 | 8 | 6000
2365 | barun Sengupta | director | personnel | 05 | 11 | 47 | 7800
:
.
.
.
.
2345 | j.b Saxena | g.m | marketing | 03 | 12 | 45 | 8000
0110 | v.k agrawal | g.m | marketing | 12 | 31 | 40 | 9000

1) to display header and footer of a file
$ head emp.lst
2233 | a.k Shukla | g.m | Sales | 12 | 12 | 52 | 6000
9876 | jai Sharma | director | production | 03 | 12 | 50 | 7000
:

$ head -n 3 emp.lst
2233 | a.k Shukla | g.m | Sales | 12 | 12 | 52 | 6000
9876 | jai Sharma | director | production | 03 | 12 | 50 | 7000
5678 | Sunil Chakrabarty | d.g.m | marketing | 04 | 19 | 43 | 6000

$ tail -n 3 emp.lst
2345 | j.b Saxena | g.m | marketing | 03 | 12 | 45 | 8000
0110 | v.k agrawal | g.m | marketing | 12 | 31 | 40 | 9000

$ tail emp.lst
6213 | Kareena ganguly | g.m | accounts | 06 | 05 | 62 | 6300
:
0110 | v.k agrawal | g.m | marketing | 12 | 31 | 40 | 9000

2) to display first 5 lines & last 5 lines of file
$ head -n 5 emp.lst
2233 | a.k Shukla | g.m | Sales | 12 | 12 | 52 | 6000
9876 | jai Sharma | director | production | 03 | 12 | 50 | 7000

5678	Sumit Chakrobarty	d.g.m	marketing	01	19	63	6000
2365	barun Sengupta	director	personnel	05	11	47	7800
5423	n.k gupta	Chairman	admin	08	30	56	5400

$ tail -n 5 emp.lst

3212	shyam saksena	d.g.m	accounts	12	12	55	6000
:							
0110	v.k agrawal	g.m	marketing	12	31	40	9000

3) grep cmd with any 5 basic regular expression operations on input file.

⇒ as pattern matching

$ grep "Sales" emp.lst

2233	a.k Shukla	g.m	Sales	12	12	52	6000
1006	Chanchal Sanghvi	director	Sales	09	03	38	6700
1265	S.n dasgupta	manager	Sales	09	12	63	56000
2476	anil aggarwal	manager	Sales	05	01	59	5000

$ grep -i 'Agarwal' emp.lst

3564	Sudhir Agarwal	executive	personnel	07	06	47	8000

$ grep -i 'aggarwal' emp.lst

2476	anil aggarwal	manager	Sales	05	01	59	5000

i- ignoring cases

⇒ grep -v '' emp.lst - 'v' displays except ''

$ grep -v 'Sales' emp.lst

9876	jai Sharma	director	production	03	12	45	7000

grep -v 'production' emp.lst

2233	a.k Shukla	g.m	Sales	12	12	52	6000

grep -v 'Sharma' emp.lst
2233 | a.k shukla | g.m | Sales | 12 | 12 | 52 | 6000

=) grep -n 'pattern' emp.lst
-n displays the line no's containing the pattern, along with the lines.

$ grep -n 'marketing' emp.lst
3 : 5678 | Sumit Chakrabarty | d.g.m | marketing | 04 | 19 | 43 | 6000
11 : 6521 | lalit | dir | marketing | 09 | 26 | 45 | 8200
14 : 2345 | j.b Saxena | g.m | marketing | 03 | 12 | 45 | 8000
15 : 0110 | v.k agrawal | g.m | marketing | 12 | 31 | 40 | 9000

$ grep -n 'Sales' emp.lst
$ grep -n 'Sumit' emp.lst
3 : 5678 | Sumit Chakrabarty | d.g.m | marketing | 04 | 19 | 43 | 6000

=> grep -c 'Pattern' emp.lst
-c counts the pattern with only that line
$ grep -c 'Sumit' emp.lst
1
$ grep -c 'marketing' emp.lst
4

=> grep -l ' ' *.lst :- displays the patterns which matches with in the multiple files

cat > des.lst
2233 | a.k shukla | g.m | Sales | 12 | 12 | 52 | 6000

$ grep -l 'Sales' *.lst
des.lst
emp.lst

=> grep -e :- matches with multiple patterns

$ grep -e 'agrwal' -e 'aggarwal' -e 'Agarwal' emp.lst

2476 | anil aggarwal | manager | ...
3564 | Sudhir Agarwal | executive | ...

4) demo of cut cmd columns and field wise both

$ cut -c 6-22, 24-32 emp.lst
a.k Shukla | g.m | ales | 12 ||
jai Sharma | diratr | product

$ cut -d\| -f 2,3 emp.lst
a.k Shukla | g.m
jai Sharma | director

$ cut -d "|" -f 1,4 emp.lst > empstore.lst
$ cat empstore.lst
2233 | Sales
:
0110 | marketing

O/p screen
2/1/23

To compile C program in unix
$> cc filename.c
$> ./a.out
executable file