1) Illustrate the working of Linux Commands to

a) Display date with various options

ans)

1. date
2. date +"%d/%m/%y"
3. date +"%A,%B,%D"
4. date +"%A,%B,%D"

b) Display list of users currently using the server

ans)

who -l

c) Display your user name

ans)

whoami

d) List all files and directories in current working directory

ans)

ls

e) Display home directory

ans)

echo $HOME

f) Display the only the first 10 lines of a particular file

ans)

head -n 10 <filename> (replace with your filename)

g) Display last 6 lines of a particular file

ans)

tail -n 10 <filename> (replace with your filename)

2)Write shell program to get a subdirectory name from user and list the contents inside the directory. Also display how many entries of the subdirectory start with file name “ab”

ans)

```
#!/bin/bash
echo "Enter the subdirectory name:"
read subdir
if [ -d "$subdir" ]
then
echo "Contents in the subdirectory $subdir are:"
ls "$subdir"
count=$(ls "$subdir" | grep -c '^ab')
echo "Number of files that start with 'ab' are:$count"
else
echo "Subdirectory $subdir does not exist"
fi
```

3) Illustrate the working of Linux Commands to

a. Display current working directory

ans)

```
pwd
```

b. Create a sub directory in your home area and add 3 files

ans)

```
mkdir ~/subdir && touch ~/subdir/t1.txt ~/subdir/t2.txt ~/subdir/t3.txt
```

c. Change working directory

ans)

```
cd subdir
```

d. Delete the subdirectory created

ans)

```
cd ~ && rm -r subdir
```

e. Display current path settings

ans)

```
echo $PATH
```

f. Current shell

ans)

```
echo $SHELL
```

g. all shells available

ans)

```
cat /etc/shells
```

4)Write shell program to get two file names as command line arguments and perform comparison of these two files.

ans)

```
if [ "$#" -ne 2 ]
then
   echo "enter two files (file1,file2)"
   exit 1
fi
file1=$1
file2=$2
if [ ! -e "$file1" ]
then
   echo "file $file1 not exits"
   exit 1
fi
if [ ! -e "$file2" ]
then
```

```
    echo "file $file2 not exits"
    exit 1
fi
if cmp -s "$file1" "$file2"
then
    echo "files are identical"
else
    echo "files are not identical"
fi
```

5)Write a shell program to display welcome message according to time(GOOD MORNING , GOOD AFTERNOON etc)

ans)

```
hours=$(date +"%H")
if [ "$hours" -ge 5 ] && [ "$hours" -lt 12 ]
then
    echo "GOOD MORNING"
elif [ "$hours" -ge 12 ] && [ "$hours" -lt 16 ]
then
    echo "GOOD AFTERNOON"
elif [ "$hours" -ge 16 ] && [ "$hours" -lt 29 ]
then
    echo "GOOD EVENING"
else
    echo "GOOD NIGHT"
fi
```

6) Using vi Editor, Perform the following tasks

a. Create a file, write your name and address save and exit

ans)

vi <filename>

press I to insert mode

enter your name and address

Press Esc to exit insert mode.

Type :wq and press Enter to save and exit

b. Open the same file

ans)

vi <filename>

c. Copy first line and paste at the end of the file

ans)

Press gg to go to first line.

Press yy to yank (copy) the line.

Move the cursor to the last line by pressing G.

Press p to paste the copied line below the current line.

d. Delete a given word in the file

ans)

Place the cursor on the word you want to delete.

Press dw to delete the word.

e. Delete two lines

ans)

Press 2dd to delete the current line and the one below it.

f. Delete word from the current cursor position

ans)

Place the cursor at the start of the word.

Press dw to delete the word from the cursor position.

g. Exit without saving the changes

ans)

Press Esc to ensure you are in normal mode.

Type :q! and press Enter to exit without saving the changes.

7)Use Linux commands

(a) Display current working directory

ans)

pwd

(b) Change directory to the parent directory

ans)

cd ..

(c) Change directory to a subdirectory of current working directory

ans)

cd subdir

(d) Display all files in your directory with “.c” extension.

ans)

ls *.c

(e) Create a file “student.txt” and store name of five students

ans)

cat >student.txt

(enter 5 names and press cntrl d)

(f) Add two more names to “student.txt”

ans)

echo -e "nithin\njoel" >> student.txt

(g) Give write permission to others for “student.txt”

ans)

chmod o+w student.txt

8) Use Linux commands

a. Create file “student” in your current working directory

ans)

```
touch student
```

b. Rename the file as “names.txt”

ans)

```
mv student names.txt
```

c. Create a folder “bca”

ans)

```
mkdir bca
```

d. Copy names.txt as names.dat in “bca” and Delete the file names.txt.

ans)

```
cp names.txt bca/names.dat

rm names.txt
```

e. Display contents of directory “bca” with various options without moving to “bca” subdirectory

ans)

```
ls -a bca

ls -l bca
```

f. Move to the directory “bca”

ans)

```
cd bca
```

g. Create another file and give read, write and execute permission for user and only execute permission for group and other.

ans)

```
touch file2
chmod 711 file2
```

9)Write shell program to perform menu driven program to check for A) file existence, B) file readable or not C) file writeable or not.

ans)

```
while true
do
```

```
        echo "1.check file exits"
        echo "2.check file readable or not"
        echo "3.check file writeable or not."
        echo "4.Exit"
        echo -n "enter your choice : "
        read choice

case $choice in

1)
        echo -n "enter the file name : "
        read file
        if [ -e "$file" ];then
                echo "$file exits"
        else
                echo "$file not exits"
        fi
        ;;

2)
        echo -n "enter the file name : "
        read file
        if [ -r "$file" ];then
                echo "$file is readable"
        else
                echo "$file is not readable"
        fi
        ;;
3)
        echo -n "enter the file name : "
        read file
        if [ -w "$file" ];then
                echo "$file is writable"
        else
                echo "$file is not writable"
        fi
        ;;

4)
        echo -n "exiting"
        exit 1
        ;;
```

```
*)
        echo -n "enter a valid choice"
        ;;

esac
done
```

10)Write shell program to perform menu driven program to check for A) file existence, B) file readable or not C) file writeable or not.

ans)

```
for file in *
do
        if [ -f "$file" ]
        then
        chmod u+rwx "$file"
        echo -n "set read write execute permission set for users for all files"
        chmod go+x "$file"
        echo -n "set execute permission for group and others for all the files"
        fi

done
```

11)Write a shell program to check whether two strings are equal or not, length is zero or not and concatenate the two strings.

ans)

```
echo -n "enter the first string :"
read string1
echo -n "enter the second string : "
read string2

if [ "$string1" = "$string2" ]; then
        echo "strings are equal"
else
        echo "strings are not equal"
fi
```

12)Write a menu driven shell program to copy, edit, rename and delete a file.

ans)

```
while true;
do
        echo "1.edit"
        echo "2.copy"
        echo "3.rename"
        echo "4.delete"
        echo "5.exit"
        echo -n "enter your choice : "
        read choice

case $choice in

1)

        echo -n "enter the file name you want to edit : "
        read file
        nano "$file"
        echo "file edited sucessfully"
        ;;
2)
        echo -n "enter the souce file :"
        read source
        echo -n "enter the destination file :"
        read des
        cp "$source" "$des"
        echo "copied sucessfully"
        ;;
3)
        echo -n "eneter the current rename : "
        read current
        echo -n "enter the new name :"
        read new
        mv "$current" "$new"
        echo "$delete $current renamed to $new  sucessfully"
        ;;

4)
        echo -n "enter the file name you want to delete :"
        read delete
        if [ -e "$delete" ];then
```

```
        rm "$delete"
        echo "$delete deleted sucessfully"
        else
        echo -n "file not exits"
        fi
        ;;

5)
        echo -n "existing..."
        exit 1
        ;;
*)
        echo "invalid choice"
        ;;

esac

done
```

13)Write a Shell program to get two file names from the user and check whether they are same or not. If both the files are same delete the second one.

ans)

```
echo -n "enter the first file name : "
read file1
echo -n "enetr the second file name : "
read file2
if cmp "$file1" "$file2" ; then
echo -n "$file1 and $file2 are same..."
rm "$file2"
echo -n "$file2 has been deleted.."
else
echo "$file1 and $file2 are different.."
fi
```

14)Write a shell script to get two strings from the user and check whether the two strings are equal or not, length is 0 or not and then concatenate the two strings.

ans)

```
echo -n "enter the first string : "
read string1
echo -n "enetr the second string : "
read string2

if [ "$string1" = "$string2" ];then
echo "string are equal..."
else
echo "strings are not equal..."
fi
if [ -z "$string1" ]
then
echo "lenght of the first string is zero"
else
echo "lenght of the first string is not zero"
fi
if [ -z "$string2" ]
then
echo "lenght of the second string is zero"
else
echo "lenght of the second string is not zero"
fi
concat="$string1 $string2"
echo "concatendated string is "$concat""
```

15)Illustrate the working of Linux commands to

(a) Display current working directory

ans)

```
pwd
```

(b) List all running processes

ans)

```
ps aux
```

(c) Display current shell

ans)

```
echo $SHELL
```

(d) Change directory to a subdirectory of current working directory

ans)

```
cd subdir
```

(e) Display all files in your directory with “.c” extension.

ans)

```
ls *.c
```

(f) Display primary and secondary prompt

ans)

```
echo $PS1
echo $PS2
```

(g) Display information related to terminal

ans)

```
tty
```

16)Perform the following actions using vi Editor

a. Open vi editor and type 8 line of text

ans)

```
vi <filename>
enter five lines of text
```

b. Copy and paste first three lines

ans)

```
Go to the first line by pressing gg.
press 3yy
```

press p to paste

c. Delete a line from cursor position and also 5 lines from the beginning

ans)

Delete the current line where the cursor is positioned by typing: dd
Go to the beginning of the file by pressing gg.
Delete the first five lines by typing: 5dd

d. Delete a word from cursor position

ans)

Move the cursor to the beginning of the word you want to delete.
Delete the word by typing: dw

e. Search for a specific word and delete it

ans)

Search for a specific word by typing / followed by the word and pressing Enter
eg: /searchword
Move the cursor to the beginning of the word and delete it using dw. You can repeat the search and delete process by pressing n to move to the next occurrence and then typing dw again.

f. Save file with file name and exit

ans)

Save the file and exit by typing
:wq filename

g. Open the file make changes and quit without saving

ans)

vi filename

Summary of vi Commands

- i: Enter insert mode.

- Esc: Exit insert mode.
- gg: Go to the beginning of the file.
- 3yy: Yank (copy) three lines.
- p: Paste below the cursor.
- P: Paste above the cursor.
- dd: Delete the current line.
- 5dd: Delete five lines.
- dw: Delete the word from the cursor position.
- /word: Search for "word".
- n: Move to the next occurrence of the search term.
- :wq filename: Save and quit with the specified filename.
- :q!: Quit without saving.

17)Write menu driven shell program to display today's date, current user and current working directory.

ans)

```
#!/bin/bash

while true; do
   echo "1. Display today's date"
   echo "2. Current user"
   echo "3. Current working directory"
   echo "4. Exit..."
   echo -n "Enter your choice: "

   read choice

   case $choice in
      1)
       echo "Today's date is: $(date +"%A, %B %d, %Y")"
         ;;
      2)
        echo "Current user is: $(whoami)"
         ;;
      3)
       echo "The current working directory is: $(pwd)"
         ;;
      4)
         echo "Exiting..."
         exit 0
```

```
        ;;
    *)
        echo "Enter a valid option...."
        ;;
  esac
done
```

18)Illustrate the working of basic Linux commands to

a. Check the present working directory

ans)

```
pwd
```

b. Display current path settings

ans)

```
echo $PATH
```

c. Include current working directory in path settings

ans)

```
echo export PAATH=$PATH:$(pwd)
```

d. List the contents of a directory using wild cards *, ?

ans)

```
ls *
ls a*
ls a?
```

e. To create and delete multiple sub directories

ans)

```
mkdir dir1 dir2 dir3
rmdir dir1 dir2 dir3
```

f. Create a directory hierarchy “bca/exam/internal”

ans)

```
mkdir bca1/exam1/external1
```

g. Change primary prompt to current date

ans)

```
export PS1='\d $ '
```

19)Write menu driven shell program to display today's date, current user and current working directory.

ans)

```
echo -n "enter the filename :"
read file

if [ -e "$file" ];then
        echo "file exits"
                if [ -f "$file" ];then
                        echo "$file is an ordinary file.."
                else
                        echo "$file is not an ordinary file.."
                fi
                if [ -d "$file" ];then
                        echo "$file is an directory file.."
                else
                        echo "$file is not an directory file.."
                fi
                if [ -r "$file" ];then
                        echo "$file is an readable file.."
                else
                        echo "$file is not an a readable file.."
                fi
                if [ -w "$file" ];then
                        echo "$file is an writable file.."
                else
                        echo "$file is not an a writable file.."
                fi
                if [ -x "$file" ];then
                        echo "$file is an executable file.."
                else
                        echo "$file is not an a executable file.."
                fi
else
        echo "file $file doesnot exits"
fi
```

20)Write a shell script to get three file names and a directory name as command line argument and create the three files and the directory in in the current working directory. Display appropriate message if command line arguments are less than four.

ans)

```
if [ "$#" -ne 4 ]
then
        echo "give three file names and  a directory name"
        echo "file1 file2 file3 dir1 (give like this)"
else
        file1=$1
        file2=$2
        file3=$3
        dir=$4
        touch "$file1" "$file2" "$file3"
        echo ""$file1" ,"$file2","$file3" created sucessfully"
        mkdir "$dir"
        echo ""$dir" created sucessfully"
fi
```

21 prgm to read n numbers in an array

ans)

```
#!/bin/bash
declare -a ar
echo"enter the number of elements"
read n
echo"enter the elements"
for((i=0;i<n;i++))
do
read a[i]
done
echo"array elements are:"
echo ${a[@]}
```

22) pgrm to find the sum of elements in array

ans)

```
#!/bin/bash
```

```
declare -a ar
sum =0
echo"enter the number of elements"
read n
echo"enter the elements"
for((i=0;i<n;i++))
do
read a[i]
sum='expr $num +${a[i]}'
done
echo"array elements are:"
echo ${a[@]}
echo "sum is:$sum"
```

23)pgrm to sort elements in array

ans)

```
#!/bin/bash
declare -a ar
echo"enter the number of elements"
read n
echo"enter the elements"
for((i=0;i<n;i++))
do
read a[i]
done
echo"numbrs before sorting:"
echo ${a[@]}
for((i=0;i<n;i++))
do
for((j=0;j<n;j++))
do
if[${a[i]} -gt ${a[j]}]
then
t=${a[i]}
a[i]=${a[j]}
a[j]=$t
```

24) TO PERFORM ARITHMETIC OPERATION

ans)

```
#bin/bash
```

```
echo "enter two numbers"
read a
read b
sum=$((a+b))
diff=$((a-b))
pro=$(($a*$b))
let div=$a/$b
echo "sum is $sum"
echo "difference is $diff"
echo "product is $pro"
echo "quotient is $div
```

25)TO ADD TWO NUMBERS

ans)

```
#!bin/bash
echo “Enter first number”
read num1
echo “Enter second number”
read num2
sum= $((num1+num2))
echo “sum of the entered numbers:$sum”
```

26)program to accept the name of the file from the standard input and then performs the following
operation :enter 5 values in afile ,sort file ,and list unsorted and sorted file

ans)

```
#!bin/bash
echo -n "Enter the file name:"
read fname
echo "Enter 5 values in the file $fname and press ^d at the end "
cat>$fname
sort -n $fname>sortfile
echo
echo "UNSORTED LIST"
cat$fname
echo
echo "SORTED LIST"
cat sortfile
```

27)program to read an integer and display it

ans)

```
#!/bin/bash
 echo “enter a number”
read num
echo “entered number is $num”
```

28)Prgm to find the largest among two numbers

ans)

```
#!/bin/bash
echo ”Enter two numbers”
read a b
if[ $a-gt $b ]
then
echo ” numbers are $a and $b”
echo “largest number is $a”
elif[  $a-it $b ]
then
echo “Numbers are $a and $b”
echo “largest number is $b”
else
echo “Numbers are $a and $b and are equal”
fi
```

29)Pgrm to read a student register number,name and four subjects marks and print whether he is passed or fail.

ans)

```
#!/bin/bash
echo -n "Enter the register number"
read reg
echo -n "Enter the name"
read name
echo -n "Enter four marks one by one"
read m1 m2 m3 m4
if[ $m1 -gt 40 -a $m2 -ge 40 -a $m3 -ge 40 -a $m4 -ge 40 ]
then
```

```
total=’expr $m1+$m2+$m3+$m4’
echo “total mark is : $total”
echo”Result :PASS”
else
echo “result : Fail”
fi
```

30)program to find the largest among three numbers

ans)

```
#!/bin/bash
echo ”Enter three numbers”
read a b c
if[ $a -gt $b ]
then
if[ $a -gt $c ]
then
echo ”Largest number is $a”
else
echo “Largest number is $c”
fi
else
if[ $b -gt $c ]
then
echo “Largest number is $b”
else
echo “Largest number is $c”
fi
fi
```

31.program for menu driven calculator

ans)

```
#!/bin/bash
echo “Enter two numbers”
read  a b
echo “Enter an operator”
read op
case $op in
+)
```

```
res=’expr $a +$b’
echo “sum is $res”
;; -)
res=’expr $a - $b’
echo “difference is $res”
;;
\*)
res =’expr $a\*$b’
echo “product is $res”
;;
/)
res =’expr $a/$b’
echo “quotient is $res”
;;
*)
echo “invalid choice”
esac
```

32)Program to print first n numbers using while loop.

ans)

```
#!/bin/bash
#first n using while
echo "Enter a number"
read n
echo "First $n numbers are:"
i=1
while [ $i -le $n ]
do
echo $i
i=$(($i + 1))
done
```

33)Program to print the first n numbers using UNTIL loop.

ans)

```
#!/bin/bash
echo "Enter a number"
read n
```

```
echo "First $n numbers are:"
i=1
until [ $i -gt $n ]
do
echo $i
i=$(($i +1))
done
```

34).Program to enter n numbers and find the sum.

ans)

```
#!/bin/bash
sum=0
i=1
echo "Enter the number of elements"
read n
echo "Enter the elements"
while [ $i -le $n ]
do
read num
sum=$(( $sum + $num))
i=$(( $i + 1))
done
echo "Sum of above $n numbers are : $sum"
```

35).Program to find the sum of digits of a number.

ans)

```
#!/bin/bash

sum=0

echo "Enter a number"
read num

while [ $num -ne 0 ]
```

```
do
  r=$(($num % 10))
  sum=$(($sum + $r))
  num=$(($num / 10))
done

echo "Sum of digits is: $sum"
```

36)Program to find the sum of even digits and an average of odd digits of a given number.

ans)

```
#!/bin/bash

sumeven=0
sumodd=0

echo -n "Enter a number: "
read num

while [ $num -ne 0 ]
do
  rem=$(($num % 10))
  if [ $(($rem % 2)) -eq 0 ]
  then
    sumeven=$(($sumeven + $rem))
  else
    sumodd=$(($sumodd + $rem))
  fi
  num=$(($num / 10))
done

echo "Sum of even digits is: $sumeven"
echo "Sum of odd digits is: $sumodd"
```

37)Program to find the reverse of a number:

ans)

```
#!/bin/bash
```

```
rev=0
echo "Enter a number"
read num
while [ $num -ne 0 ]
do
  r=$((num % 10))
  rev=$((rev * 10 + r))
  num=$((num / 10))
done
echo "Reverse is: $rev"
```

37)Program to check whether the entered number is palindrome or not:

ans)

```
#!/bin/bash
sum=0
rev=0
echo "Enter a number"
read num
n=$num
while [ $num -ne 0 ]
do
  r=$((num % 10))
  rev=$((rev * 10 + r))
  num=$((num / 10))
done
if [ $n -eq $rev ]
then
  echo "Entered number is Palindrome"
else
  echo "Entered number is Not Palindrome"
fi
```

38)Program to find the factorial of a number:

ans)

```
#!/bin/bash
fact=1
echo "Enter a number"
```

```
read num
for ((i=1;i<=num;i++))
do
  fact=$((fact * i))
done
echo "Factorial of $num is $fact"
```

39)Program to check whether the entered number is prime or not:

ans)

```
#!/bin/bash
flag=0
echo "Enter a number"
read num
for ((i=2;i<=num/2;i++))
do
  if [ $((num % i)) -eq 0 ]
  then
    flag=1
    break
  fi
done
if [ $flag -eq 0 ]
then
  echo "Entered number is Prime"
else
  echo "Entered number is Not Prime"
fi
```

40)Program to print all prime numbers between two ranges:

ans)

```
#!/bin/bash
echo "Enter the starting number"
read st
echo "Enter the ending number"
read end
echo "Prime numbers between $st and $end are:"
for ((num=st;num<=end;num++))
```

```
do
 flag=0
 for ((i=2;i<=num/2;i++))
 do
  if [ $((num % i)) -eq 0 ]
  then
   flag=1
   break
  fi
 done
 if [ $flag -eq 0 ]
 then
  echo $num
 fi
done
```

41)Program to check whether the entered number is Armstrong or not:

ans)

```
#!/bin/bash
a=0
sum=0
echo "Enter a number"
read num
temp=$num
while [ $temp -ne 0 ]
do
 a=$((temp % 10))
 sum=$((sum + a * a * a))
 temp=$((temp / 10))
done
if [ $num -eq $sum ]
then
 echo "$num is an Armstrong number"
else
 echo "$num is not an Armstrong number"
fi
```

42)Program to print the first n Fibonacci series:

ans)

```
#!/bin/bash
a=0
b=1
i=3
echo "Enter the value of n"
read n
echo "First $n Fibonacci Series"
if [ $n -eq 1 ]
then
  echo $a
elif [ $n -eq 2 ]
then
  echo $a
  echo $b
else
  echo $a
  echo $b
  while [ $i -le $n ]
  do
    c=$((a + b))
    echo $c
    a=$b
    b=$c
    i=$((i + 1))
  done
fi
```

43)Program to find the GCD of two numbers:

ans)

```
#!/bin/bash
echo "Enter two numbers"
read a b
if [ $a -gt $b ]
then
  m=$a
else
  m=$b
```

```
fi
for ((i=1;i<=m;i++))
do
  x=$((a % i))
  y=$((b % i))
  if [ $x -eq 0 ] && [ $y -eq 0 ]
  then
    gcd=$i
  fi
done
echo "GCD of $a and $b is $gcd"
```

44)Program to find the value of nCr:

ans)

```
#!/bin/bash
echo "Enter the values of n and r"
read n r
nfact=1
rfact=1
nrfact=1
# Finding n!
for ((i=1;i<=n;i++))
do
  nfact=$((nfact * i))
done
# Finding r!
for ((i=1;i<=r;i++))
do
  rfact=$((rfact * i))
done
# Finding (n-r)!
nr=$((n-r))
for ((i=1;i<=nr;i++))
do
  nrfact=$((nrfact * i))
done
res=$((nfact / (rfact * nrfact)))
echo "Result is: $res"
```