BCA SEM 5 SHELL SCRIPT PROGRAM

1 Write a shell script to execute following commands

1. Sort file abc.txt and save this sorted file in xyz.txt2. Give an example of : To execute commands together without affecting result ofeach other.3. How to print “this isa three –line1. Text message”4. Which command display version of the UNIX?5. How would u get online help of cat command?

```
echo “sorting the file”

sort abc.txt > xyz.txt

echo “executing two commands”

who ; ls

echo “this is \n a three-line \n Text message” # use -e option if required

echo “The version is `uname -a`”

echo “Help of cat command”

man cat
```

2 Write a shell script to execute following commands

1. How would u display the hidden files?

2. How delete directory with files?

3. How would user can do interactive copying?

4. How would user can do interactive deletion of files?

5. Explain two functionality of “mv” command with example?

```
echo “1. How would u display the hidden files”

echo “2. How delete directory with files”

echo “3. How would user can do interactive copying”

echo “4. How would user can do interactive deletion of files”

echo “5. Explain two functionality of “mv” command with example”

echo “enter your choice”

read ch

case $ch in

1) echo “Displaying hidden files”

ls .[a-z]*  ;;

2) echo “Deleting directories with files”

rm -R dirname

3) echo “Interactive copy”

cp -i  file1 file2 ;; # file2 should be created first to check interactivity

4)  echo “Interactive Deletion”
```

```
rm -i file1 ;;
5) echo “mv command”
  mv oldfilename newfilename ;;
*) echo “Invalid choice” ;;

esac
```

5 Write a shell script to execute following commands

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

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

3. Delete the directories mydir and newdir at one shot.

4. Sort a numeric file?

5. Change the permissions for the file newtext to 666.

```
echo “1. Create a file called text and store name,age and address in it.”
echo “2. Display the contents of the file text on the screen.”
echo “3. Delete the directories mydir and newdir at one shot.”
echo “4. Sort a numeric file”
echo “5. Change the permissions for the file newtext to 666.”
echo “enter your choice”
read ch
case $ch in
1) echo “Create a file called text and store name,age and address in it.”
echo “Enter the filename”
read fn
cat > $fn  ;;
2) echo “Display the contents of the file text on the screen.”
cat $fn ;;
3) echo “Delete the directories mydir and newdir at one shot.”
rmdir mydir newdir  ;;
4)  echo “Sort a numeric file”
      sort -n filename ;;
5) echo “Change the permissions for the file newtext to 666.”
    chmod 666 newtext ;;
*) echo “Invalid choice” ;;
esac
```

6 Write shell script that accept filename and displays last modification time if file exists,

otherwise display appropriate message.

```
if [ -e $fn ]
then
ls -l $fn | cut -d “ “ -f8 #change the column number for desired output
else
echo “File does not exist”
fi
```

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

```
who | grep ^s
```

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

```
for i in *
do
if [ ! -s $i ]
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 ]
  then
    countx=`expr $countx + 1`
  fi
echo “Number of executable files are $countx”
```

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

```
d=`date +"%H"`
  if [ $d -lt 12 ]
  then
echo “Good Morning”
  elif [ $d -gt 12 -a $d -lt 14 ]
  then
```

```
echo “Good Afternoon”
    else
echo “Good Evening”
    fi
```

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

```
for i in *
do
   if [ -d $i ]
   then
      countd=`expr $countd + 1`
   fi
   if [ -f $i ]
   then
       countf=`expr $countf + 1`
  fi
done
echo “Number of directories are $countd ”
echo “Number of Ordinary files are $countf”
```

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 ]
then
echo “Its an ordinary file”
cat $1
else
echo “File does not exist or is not ordinary file”
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”

```
echo “enter the filename”
read fn

if [ -d $fn ]
then
echo “Its a directory”

ls $fn
else
echo “Its not a directory”
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.”

```
echo “enter the filename”
read fn

if [ -e $fn -a -x $fn ]
then
echo “file exists and is executable”
else
echo “file does not exist or is not executable”
fi
```

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

```
echo “List of Directories. ”
for i in *
do
  if [ -d $i ]
  then
echo $i
  fi
```

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

```
for i in *
do
    if [ -d $i ]
    then
        countd=`expr $countd + 1`
    fi
    if [ -f $i ]
    then
         countf=`expr $countf + 1`
   fi
done
echo “Number of directories are $countd ”
echo “Number of Ordinary files are $countf”
```

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.

```
echo “enter the first filename”
read fn1
echo “enter the second filename”
read fn2
if [ -f $fn1 -a -f $fn2 ]
then
echo “Both file exists”
cat $fn2 >> $fn1
else
echo “Files does 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 contents of file $1”
cat $1
echo “Displaying the contents page wise”
cat $1 | more
echo “Copying the files”
cp $1 $2
c=`echo $?`
if [ $c -eq 0 ]
then
echo “File copied successfully”
else
echo “Files not copied successfully”
fi
```

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. Calendar of the current month and year.”
echo “2. Display “Good Morning/Good Afternoon/Good Evening” according to the current login time.”
echo “3. User name, Users home directory.”
echo “4. Terminal name, Terminal type.”
echo “5 Machine name.”
echo “6. No. of users who are currently logged in; List of users who are currently logged in.”
echo “enter your choice”
read ch
case $ch in
1) echo “Calendar of current month is”
   cal ;;
2) d=`date +"%H"`
    if [ $d -lt 12 ]
```

```
     then
echo “Good Morning”
     elif [ $d -gt 12 -a $d -lt 16 ]
     then
echo “Good Afternoon”
     else
echo “Good Evening”
     fi
3) echo “Username is $USER”
     echo “Users Home directory is $HOME”  ;;
4)  echo “Terminal details”
     tty;;
5)  echo “Machine name is”
      uname -m ;;
6) echo “The number of users logged in are”
     who | wc -l
*) echo “Invalid choice”
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. Concatenates two strings ”
echo “2. Renames a file”
echo “3. Deletes a file.”
echo “4. Copy the file to specific location”
echo “enter your choice”
read ch
case $ch in
1) echo “enter first string”
    read str1
    echo “enter second string”
    read str2
    echo  “The concated strings are $str1$str2” ;;
```

```
2) echo “enter the old filename”
     read ofn
     echo “enter the new filename”
     read nfn
     mv $ofn $nfn
    echo “file renamed”  ;;
3) echo “enter the filename”
    read fn
    rm $fn
    echo “file deleted”  ;;
4) echo “enter the filename”
     read fn
     cp $fn \usr\home\dir\$fn     #you can change the specific path
    echo “file copied”  ;;
*)  echo “invalid choice” ;;
esac
```

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

```
for file in *.txt
do mv $file `echo $file | sed 's/ \( .*\.  \)  txt/\1dat/'` ;
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 [ $# ]
then
ls $1
else
ls $HOME
fi
```

sh filename.sh dir1

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

Newname = oldname111

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.

```
echo “enter the first filename”
read fn1
echo “enter the second filename”
read fn2
cmp $fn1 $fn2
c=`echo $?`
if [ $c -eq 0 ]
then
echo “both files are same”
rm  $fn2
else
echo “both files are not same”
fi
```

```
sh fn.sh dir1 dir2
```

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 [ $# ]
then
if [ -d $1 ]
then
if [ -d $2 ]
then
cp -R $1 $2
else
mkdir $2
echo "Directory created $2"
cp -R $1 $2
fi
```

```
else
echo “source directory does not exist”
fi
else
echo "Please provide command line arguments"
fi
```

26 Write a shell script that displays the following menu

- List home directory
- Date
- Print working directory
- Users logged in

Read the proper choice. Execute corresponding command. Check for invalid choice.

```
echo “1.List home directory”
echo “2.Date”
echo “3. Print working directory”
echo “4. Users logged in”
echo “enter your choice”
read ch
case $ch in
1) echo “Home directory is $HOME” ;;
2) echo “Todays date is `date` ” ;;
3) echo “Present working directory is `pwd` ” ;;
4) echo “ No of users logged in are”
   who | wc -l  ;;
*) echo “Invalid choice” ;;
esac
```

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

```
ls .[a-z]*
```

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

```
echo “enter the first filename”
read fn1
echo “enter the second filename”
read fn2

echo “Combining two files horizontally”
```

```
cat $fn2 >> $fn1


echo “Combining two files vertically”
paste $fn1 $fn2
```

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

```
echo “enter the filename”
read datafile
cat $datafile | tr -d '[:space:]' > newfile
```

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

```
d=`date   +”%u”`
if   [   $d   –eq    7   ]
then
echo “it is weekend”
else
echo “it is a weekday”
fi
```

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

```
echo  "Enter the word"
read w
for i in $@
do
grep $w $i
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.

```
echo “enter the filename”
read fn


ls -l $fn | cut -c 2-10
```

34 Write a shell script to display the calendar in the following manner:

i. Display the calendar of months m1 and m2 by ‘CAL m1, m2’ command file.

ii. Display the calendar of the months from m1 to m2 by ‘CAL m1-m2’ command file.

```
# run the shell script as  sh filename.sh 1 3
```

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.

```
echo “1. Display all the words of a file in ascending order.”
echo “2. Display a file in descending order.”
echo “3. Toggle all the characters in the file.”
echo “4. Display type of the file.”
echo “enter your choice”
read ch
echo “enter the filename”
read fn
case $ch in
1) sort $fn;;
2) sort -r $fn;;
3) cat $fn | tr “[a-z][A-Z]” “[A-Z][a-z]”
4) file $fn;;
*) echo “invalid choice”
esac
```

36 Write a shell script to check whether the named user is currently logged in or not.

```
echo “enter the username”
read un
c=`who | grep -c $un`
if [ $c -gt 0 ]
then
    echo “User is currently logged in ”
else
   echo “User is not currently logged in”
fi
```

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 “1.Display all the words of a file in ascending order.”
echo “2.Display a file in descending order.”
echo “3.Display a file in reerse order.”
echo “4.Toggle all the characters in the file”
echo “5.Display type of the file.”
echo “Enter your choice”
read ch
echo “enter the file name”
read fn
case $ch in
1) sort  $fn ;;
2) sort -r $fn ;;
3) rev $fn
4) cat $fn | tr “[a-z][A-Z]” “[A-Z][a-z]”
5) file $fn
*) echo “Invalid choice”
esac
```

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

```
echo “The number of users in the system are”
cat etc/passwd | wc -l
echo “The number of uses currently logged in are “
who | wc -l
```

39 Write a shell script that displays the directory information in the following format-

Filename Size Date Protection Owner

```
echo “Enter the filename”
read fn

echo “ Filename    Size    Date      Protection       Owner”
echo  “`ls -l $fn | cut -d ' ' -f3` ` ls -l $fn | cut -d ' ' -f5`    `ls -l $fn | cut -d ' ' -f6,7`  `ls -l $fn | cut -d ' ' -f1`  `ls -l $fn | cut -d ' ' -f4`   “
```

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

```
ls -lS |  head -6 | tail -1
```

41 Write a shell script that toggles contents of the file

```
echo “Enter the filename”
```

```
read fn
cat $fn | tr “[a-z][A-Z]” “[A-Z][a-z]”
```

42 Write a shell script that report whether your friend has currently logged in or not.

```
echo “Enter the username”
read un

c=`who | grep -c $un`

if [ $c -gt 0 ]
then
     echo “User is currently logged in ”
else
    echo “User is not currently logged in”
fi
```

44 Write a shell script to accept any character using command line and list all the files starting with that character in the current directory

```
ls | grep ^$1

# run the shell script as:   sh  filename.sh a
# it will list files starting with a
```

45 Create a file called student containing roll-no, name and marks.

a. Display the contents of the file sorted by marks in descending order

b. Display the names of students in alphabetical order ignoring the case.

c. Display students according to their roll nos.

d. Sort file according to the second field and save it to file ‘names’.

e. Display the list of students who scored between 70 and 80.

```
echo “enter the filename”
read fn
cat > $fn

# enter   roll-no |  name |  marks  of students and press ctlr+d

echo “1. Display the contents of the file sorted by marks in descending order”
echo “2. Display the names of students in alphabetical order ignoring the case.”
echo “3. Display students according to their roll nos.”
echo “4. Sort file according to the second field and save it to file ‘names’.”

echo “5. Display the list of students who scored between 70 and 80”


echo “enter your choice”

read ch


case $ch in

1) sort -k5 -r $fn  ;;

2) sort -k3 -i $fn ;;

3) sort $fn ;;

4) sort -k3 $fn > names ;;

5) awk ‘{ if( $5 > 70 && $5 < 80 ) print $5 }’ $fn ;;

*) echo “Invalid Choice”

esac
```

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

Newname = oldname111

```
clear
for f in c*;
do
mv $f r
done
```

Q.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.

```
clear
if test -e $1 -a -e $2
then
same=`diff $1 $2`
if test -z $same
then
rm $2
```

```
echo File 2 removed because contents of both the
files were same
else
echo Both files are not same
fi
else
echo File does not exist
fi
```

Q.26 Write a shell script that displays the following menu
_ List home directory
_ Date
_ Print working directory
_ Users logged in
Read the proper choice. Execute corresponding command. Check for invalid choice.

```
clear
select=
until [ "$select" = "5" ];
do
echo PROGRAM MENU
echo 1 List home directory
echo 2 Date
echo 3 Print working directory
echo 4 Users logged in
echo 5 Exit
echo Enter choice
read a
case $a in
1) ls ~
read z
clear;;
2) date
read g
clear;;
3) pwd
read j
clear;;
4) who
read v
clear;;
```

```
5) exit;;
*) echo Please enter valid option
read h
clear;;
esac
done
```

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

```
ls -a
```

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

```
clear
cat $1 | tr -d " \t"
```

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

```
clear
ls -lt | head -2 | tail -1 | cut -c 32-50
```

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

```
clear
ls -l $1 | more | cut -c 1-11,42-46
```

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

```
clear
tot=`cat /etc/passwd | wc -l`
cur=`who | wc -l`
echo Total users $tot
echo Currently logged in users $cur
```

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

```
clear
du -a | sort -n -r | head -5
ls -lSh | head -6 | tail -5
```