EXPT NO : 7 **REPORT**

DATE :04/04/2019

AIM : To write the shell scripts of the given questions from shell-script set2 and verify the output

==========

1. Write a shell script that will take an input file and remove identical lines.

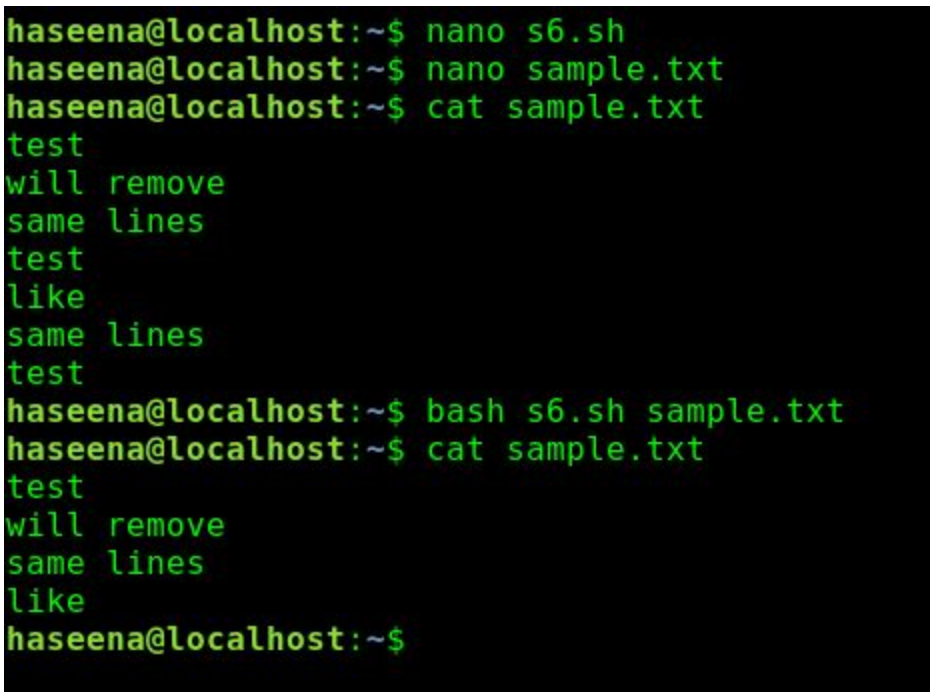
Algorithm:

1) Start
2) Read the arguments , say n.
3) If n=0, print “required a file as argument” and go to step 9.
4) If arguments are not valid print “required valid file as arguments” and exit.
5) Read contents of file and check if duplicates present
6) If present delete duplicates
7) Display the file after deletion
8) Stop

Program:

```
#! /bin/bash
if [[ $# -ne 1 ]]
then
        echo "Enter file as argument"
        exit
fi
if [[ !(-a $1) ]]
then
        echo "Enter valid file"
        exit
fi
awk '!a[$0]++' $1 >> b.temp
rm $1
mv b.temp $1
```

Output:



2. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.

Algorithm:

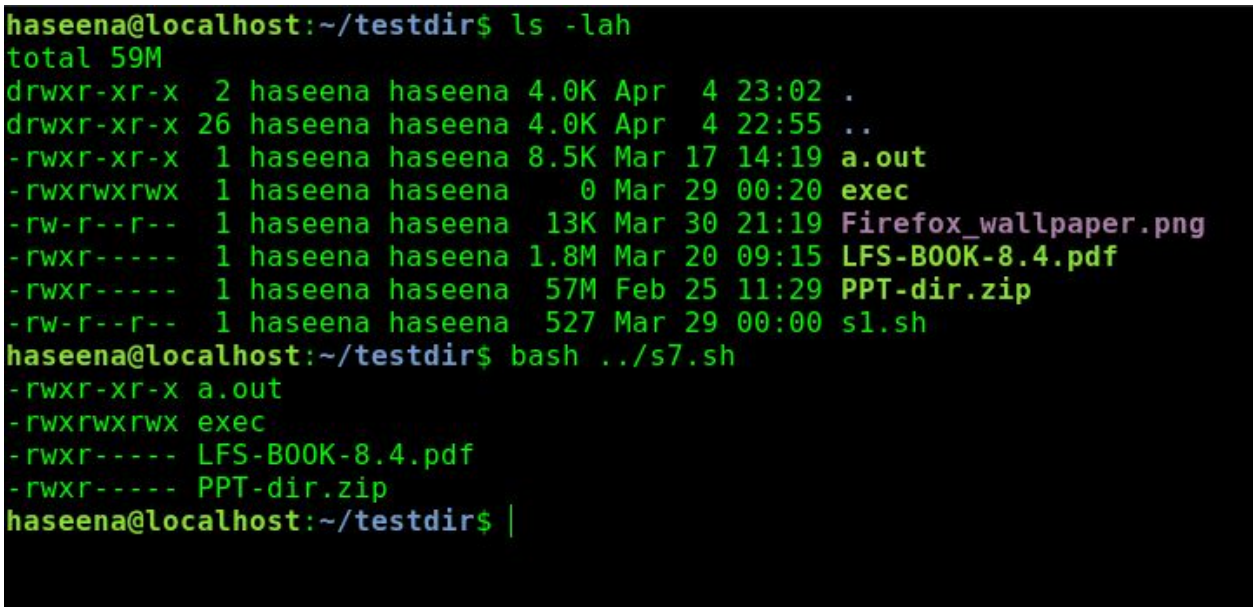
1) Start
2) Read the arguments , say n.
3) If n=0, print “required a filenames as argument” and go to step 7.
4) If arguments are not valid print “required valid files as arguments” and exit.
5) Read permissions of file and compare
6) If permission contain ‘rwx’ then print filename
7) Stop

Program:

```
#! /bin/bash
ls -lAh $PWD | awk -F " " '{print $1 " " $9}' > b.temp
while read l
do
        PERM=`echo $l | awk '{print $1}'`
        USRPERM=${PERM:1:3}
```

```
        if [[ $USRPERM == "rwx" ]]
        then
                echo "$l"
        fi
done < b.temp
rm b.temp
```

Output:



3. Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th ; a\ is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be through a text file created by the user.

Algorithm:

1) Start
2) Read the arguments , say n.
3) If n!=1, print “required filename as argument” and go to step 8.
4) If arguments are not valid print “required valid filename as arguments” and exit.
5) Read contents of files and by ignoring spaces count in each line 40 characters
6) Separate it with a ‘/’ and display balance characters in next line
7) Print it on terminal
8) stop

Program:

```
#! /bin/bash
if [[ $# -ne 1 ]]
then
        echo "Enter file as argument"
exit
fi
if [[ !(-a $1) ]]
then
        echo "Enter valid file"
exit
fi
n=`wc -l $1 | cut -d " " -f 1`
i=1
while [ $i -le $n ]
do
        line=`sed -n "$i p" $1`
        cc=`echo $line | wc -c | cut -d " " -f 1`
        while [ $cc -ge 40 ]
        do
                ext=`echo $line | cut -c 41-`
                line=`echo $line | cut -c 1-40`
                echo "$line \\"
                line=$ext
                cc=`echo $ext | wc -c | cut -d " " -f 1`
        done
        echo "$line"
        i=`expr $i + 1`
done
```

Output:

```
haseena@localhost:~$ nano s8.sh
haseena@localhost:~$ nano file
haseena@localhost:~$ cat file
ohellohellohellohellohellohellohellohellohellohellohellohellohellohellohell
ohellohellohellohellohellohellohellohellohellohello
lohellohellohellohellohellohellohellohellohellohellohell
llohellohellohellohellohellohellohellohellohellohellohellohellohello
hellohellohellohellohellohellohellohellohellohellohellohellohellohellohello
hellohellohellohellohellohellohellohellohellohellohello
haseena@localhost:~$ bash s8.sh file
ohellohellohellohellohellohellohell \
ohellohellohellohellohellohellohell \
ohellohellohellohellohellohellohell \
ohellohellohello
lohellohellohellohellohellohellohellohel \
lohellohellohellohell
llohellohellohellohellohellohellohellohellohe \
llohellohellohellohellohellohello
hellohellohellohellohellohellohellohello \
hellohellohellohellohellohellohellohello \
hellohellohellohellohellohellohellohello \
hellohellohellohello
haseena@localhost:~$
```

4. Write a shell script to delete all lines containing a specific word in one or more file supplied as argument to it.

Algorithm:

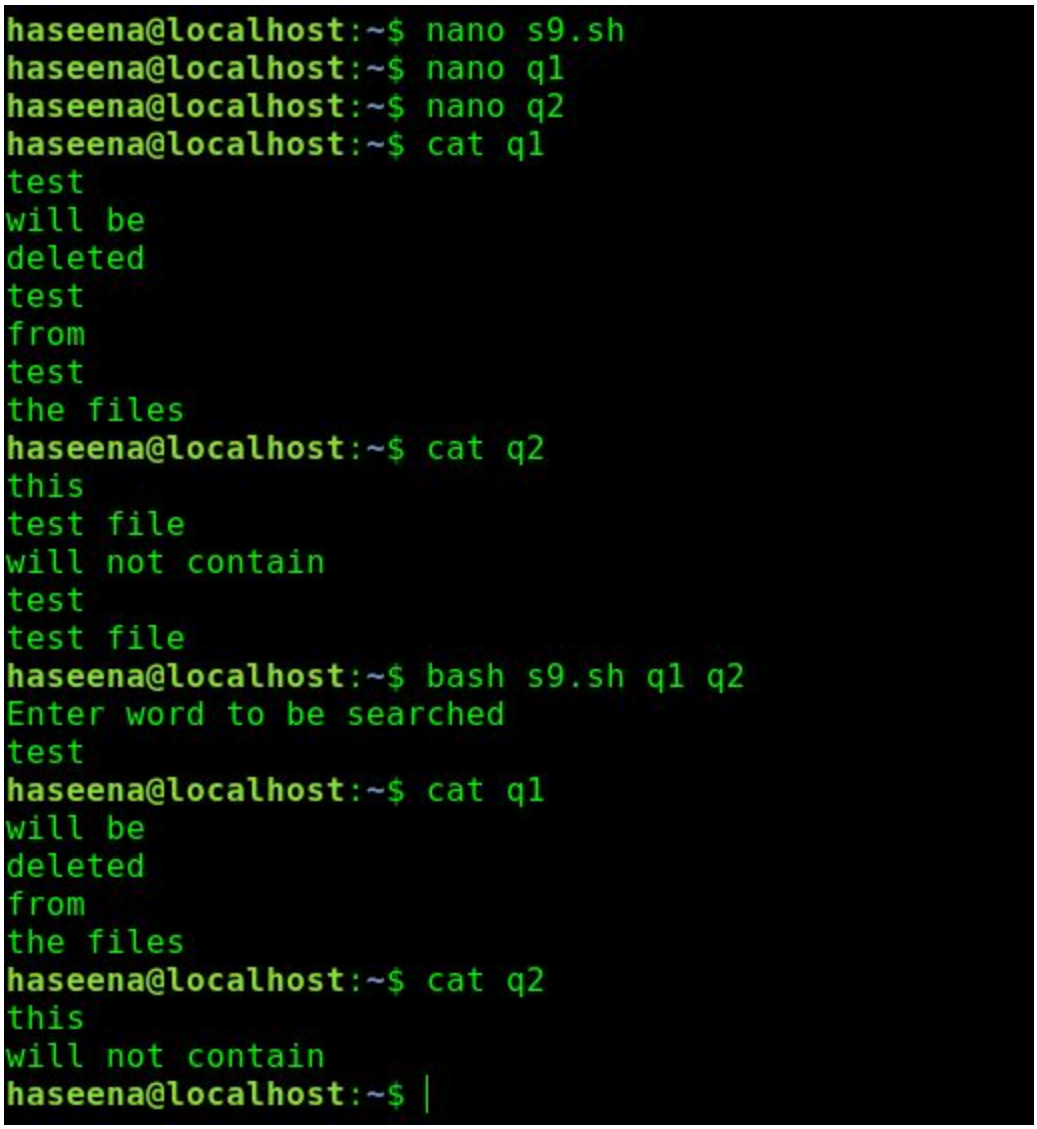
1) Start
2) Read the arguments , say n.
3) If n=0 , print “required at least one file as argument” and go to step 9.
4) Read the word to search .
5) Read contents of the file and search word containing links
6) If words present then delete that line
7) Stop

Program:

```
#! /bin/bash
if [ $# -eq 0 ]
then
        echo "Enter at least one file"
exit
fi
echo "Enter word to be searched"
```

```
read word
for file in $*
do
        sed "/$word/d" $file > b.temp
        mv b.temp $file
done
```

Output:



CONCLUSION

Verified the outputs for the above questions and improved the hold over bash shell scripting.