**Quiz 5**

All Shell scripts should display usage function, they should also validate input arguments to be correct, use functions and recursion as much as possible.

1. Write a shell script to find the number of lines in a list of files using sed.

You should write a for loop to go through all files in a directory and then count the number of lines in each file, display it as:

./script <Full path to directory>

File1 has 45 lines

File2 has 20 lines

#!/bin/bash

FILENAME=$<Full path to directory>

FILEBYTES=$(stat -c%s "$FILENAME")

FILEWORDS=$(stat -w%s "$FILENAME")

FILELINES=$(stat -l%s "$FILENAME")

2. Write a shell script to substitute one pattern for another in a text file.

./script.sh oldpattern newpattern

while read a ; do echo ${a//abc/XYZ} ; done < /tmp/file.txt > /tmp/file.txt.t ; mv /tmp/file.txt{.t,}

3. Write a shell script to print complete pathname associated with pid. User has to pass the PID from command line.

./script.sh PID

#/bin/bash

echo "Please enter the processor name"

read processname

ps -ef | grep $processname | grep -v grep | awk '{print $2}'

4. Write a shell script to print all users on system using awk.

Hint: Learn what is /etc/passwd file in Linux

#!/bin/bash

VAR1=$(cut -d: -f1 /etc/passwd)

echo -e “The users in the system as fallows..n $VAR1 n Completed listing users..!”

echo “Total number of users are $(cat /etc/passwd | wc -l)”

5. Write a shell script to list the frequency of words used in a file.

Hint: Sort and uniq commands will help

#/bin/bash

echo " Please enter the file name "

read filename

tr -c '[:alnum:]' '[\n\*]' < $filename | sort | uniq -c | sort -nr | head -10

6. Write a script to take backup of files changed in last 24 hours and archive them.

Hint: Read the Find command tutorial in Linux folder. We typically take backups of a folder by “tar”-ring the entire folders.

7. Write a shell script to determine if a particular service is active or not. For eg: if SSH service is active it should display yes and vice versa. Use netstat , ps commands etc

#/bin/bash

echo " Enter the processor name"

read process

systemctl status $process

ps -ef |grep $process

8. Write a shell script to remove spaces from filenames and replace it with underscore

Hint: you can use mv command to re-name files

find /tmp/ -depth -name "\* \*" -execdir rename 's/ /\_/g' "{}" \;

9. Write a shell script which prints the df output in more formatted way as below

Filesystem Size Used Avail Capacity Mounted

/dev/sda1 446.71G 18.11G 405.88G 5% /

udev 10M 0 10M 0% /dev

tmpfs 1.14G 9.16M 1.13G 1% /run

df -sh

10. Write a shell script to summarize available disk space and present in a logical and readable fashion

#!/bin/sh

# diskspace - summarize available disk space and present in a logical

# and readable fashion

tempfile="/tmp/available.$$"

trap "rm -f $tempfile" EXIT

cat << 'EOF' > $tempfile

{ sum += $4 }

END { mb = sum / 1024

gb = mb / 1024

printf "%.0f MB (%.2fGB) of available disk space\n", mb, gb

}

EOF

df -k | awk -f $tempfile

exit 0

11. Write a shell function to rename .txt files to .text

files=`ls -1 \*.txt`

for x in $files

do

mv $x "`basename $files .txt`.text"

done