Samarth Nasit

- Linux test -2 - 16/01/2019

1. Write a shell script that adds an extension ".new" to all the files in the directory.
Ans:

```
#! /bin/bash

 rename()
 {
        for file in "$1"/*
        do
                mv $file $file.new
        done

 }

 rename "."
```

2. Delete file which has special characters. " -", "—", "*", "$"
Ans:

```
#! /bin/bash

 delete()
 {
        for file in "$1"/*
        do
                if [ -f "$file" ]; then
                temp=`echo "$file" | cut -c 3-`

                        if [ $temp == *-* ]; then
                        rm $temp

                        elif [ $temp == *_* ]; then
```

```
                        rm $temp

                        elif [ $temp == *\** ]; then
                        rm $temp

                        elif [ $temp == *\$* ]; then
                        rm $temp

                        fi
                fi

        done

}

delete "."
```

3. Write a shell script that take two input numbers from user at runtime and display arithmetic operation on that numbers, find out max, & min number from them, find weather that numbers negative or positive.

Ans:

```
#! /bin/bash

read -p "Enter 1st number :" num1
read -p "Enter 2nd number :" num2

echo "Arithmetic operations ....."
echo "Sum of two number is : $(($num1 + $num2))"
echo "Multiplication of twq number is : $(($num1 * $num2))"
echo "Subtraction of two number is : $(($num1 - $num2))"
echo "Division of two number is : $(($num1 / $num2))"

echo "Min-Max of numbers...\n"

if [ $num1 -eq $num2 ]; then
        echo "Both are equal..."

elif [ $num -gt $num2 ]; then
        echo "$num1 is max"
        echo "$num2 is min"
```

```
else
        echo "$num2 is max"
        echo "$num1 is min"

fi

echo "Positivity and negativity of number...\n"

if [ $num1 -gt 0 ]; then
        echo "$num1 is positive"
else
        echo "$num1 is negative"
fi

if [ $num2 -gt 0 ]; then
        echo "$num2 is positive"
else
        echo "$num2 is negative"
fi
```

4. Write a shell script that take one input number from user and print 1 to n number using three loops (For, while, until). (N = entered number)

Ans:

```
#! /bin/bash

read -p "enter number :" num1
echo " Using for loop...."

for i in $(seq 1 $num1); do
        echo $i;
done

echo " Using while loop...."
temp=1
while true; do
        if [ $temp -le $num1 ]; then
                echo $temp
        else
                break
```

```
        fi
temp=$(($temp + 1))
done

echo " Using until loop..."
a=1
until [ $a -gt $num1 ]
do
        echo $a
        a=$(($a + 1))
done
```

5. Write a shell script to display the last updated file of the newest file in a directory.
Ans:

```
#! /bin/bash

 echo `ls -pt | grep -v / | head -1`
```

6. Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across
files present in subdirectories.
Ans:

```
#! /bin/bash

match=$(grep -ro "a" * | wc -l)
echo "total count: $match"
```

7. Write a shell script that copy all the directories, subdirectories and files from one location to
another specific location.
Ans:

```
#! /bin/bash
cp * $1
```

8. Display specific number of lines as follow:

1. **Display first and last 10 lines of file contains**
2. **Display line no. 3 to 8 from file contains.**

3. Display 7 lines and start from second last line in reverse manner.

Ans:

```
#! /bin/bash

if [ -f $1 ]; then
        echo `head $1`
        echo " "
        echo `head -n 8 $1 | tail -5`
        echo "  "
        echo `tac $1 | head -n 8 | tail -7`
fi
```

9. Perform following task:

1. Add two new users and two groups
2. Login as one user and then create new file
3. Send created file from one user to another user
4. Login as second user and copy that file from user2 to user1(in same system)

Ans:

10. Ex. 10 Task to find all files from folder where file contains string 'abc'

ANS:

```
#! /bin/bash

filecontainstring()
{
        for file in "$1"/*; do
                if [ -f $file ]; then
                temp=`echo "$file" | cut -c 3-`
                if [ $temp == *abc* ]; then
                        echo "$file"
                fi
                fi
        done

}

filecontainstring "."
```