FOSS LAB PRACTICE QUESTIONS: SET #1

Tuesday, March 31, 2020

Muzammil T

S4 CSE Roll No : 38

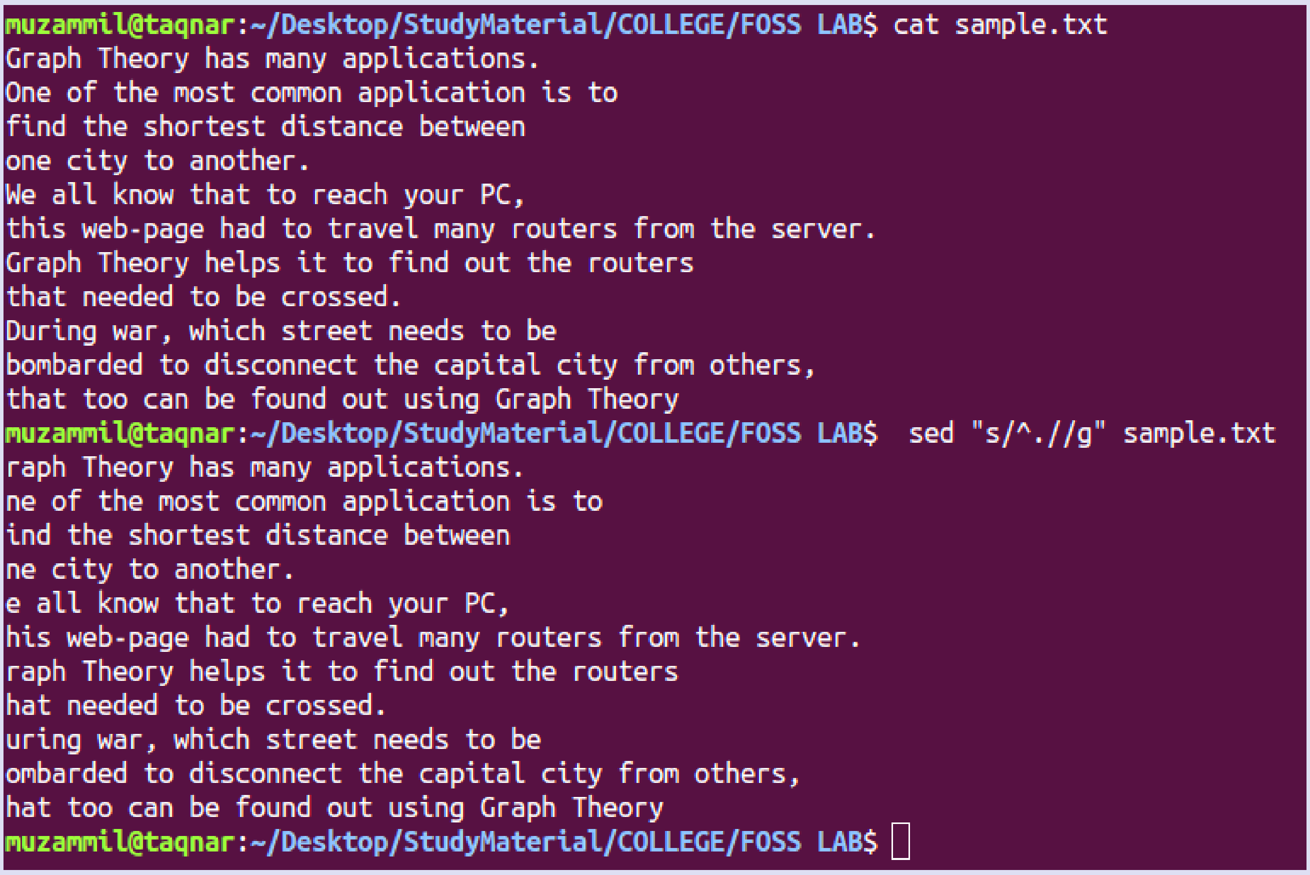
Question 1

(a) ***Write a sed command that deletes the first character in each line in a file ?***

Shell Script

```
sed "s/^.//g" sample.txt
```

Sample Output



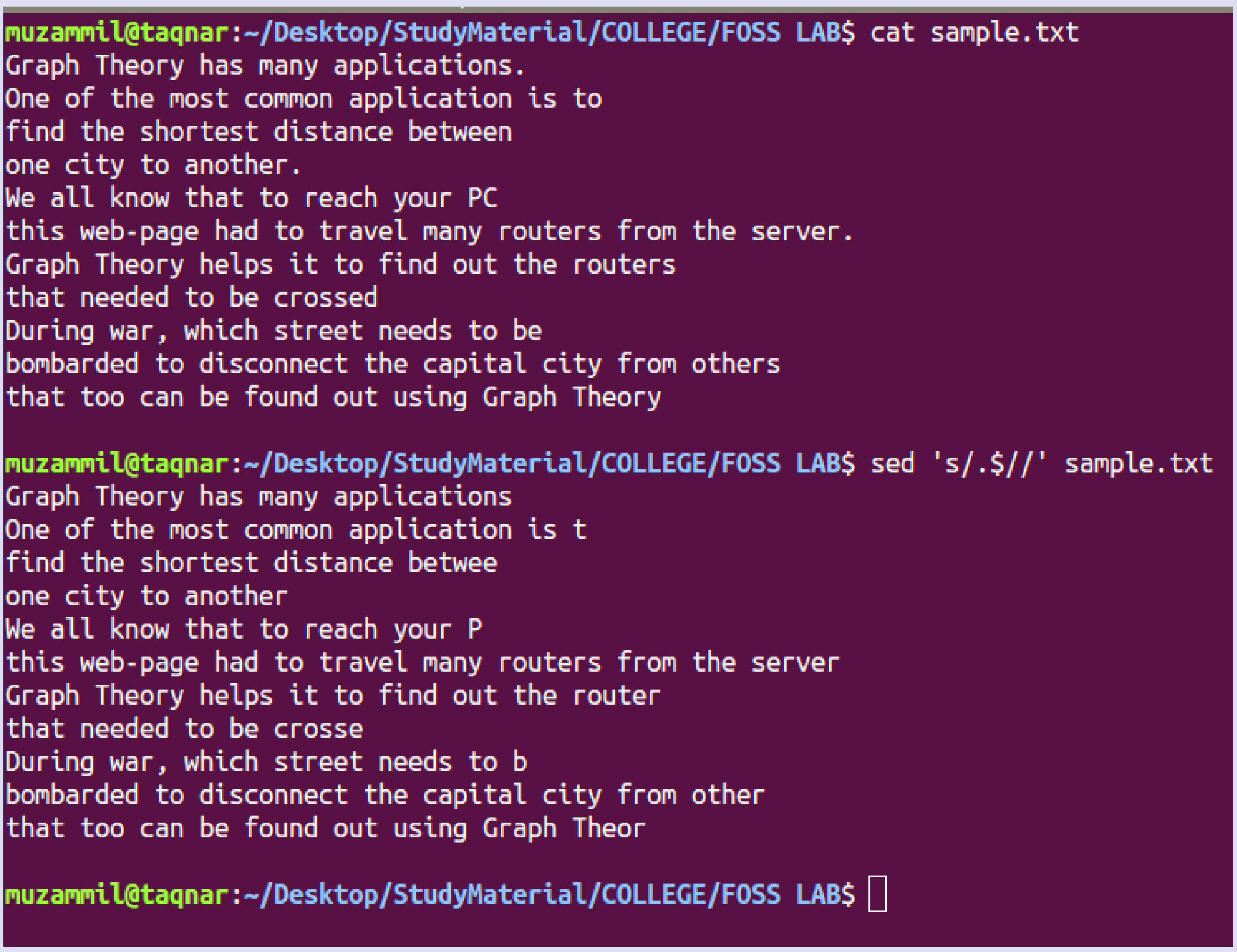
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat sample.txt
Graph Theory has many applications.
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC,
this web-page had to travel many routers from the server.
Graph Theory helps it to find out the routers
that needed to be crossed.
During war, which street needs to be
bombarded to disconnect the capital city from others,
that too can be found out using Graph Theory
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$  sed "s/^.//g" sample.txt
raph Theory has many applications.
ne of the most common application is to
ind the shortest distance between
ne city to another.
e all know that to reach your PC,
his web-page had to travel many routers from the server.
raph Theory helps it to find out the routers
hat needed to be crossed.
uring war, which street needs to be
ombarded to disconnect the capital city from others,
hat too can be found out using Graph Theory
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

(b) ***Write a sed command that deletes the last character in each line in a file ?***

Shell Script

```
sed 's/.$//' sample.txt
```

Sample Output



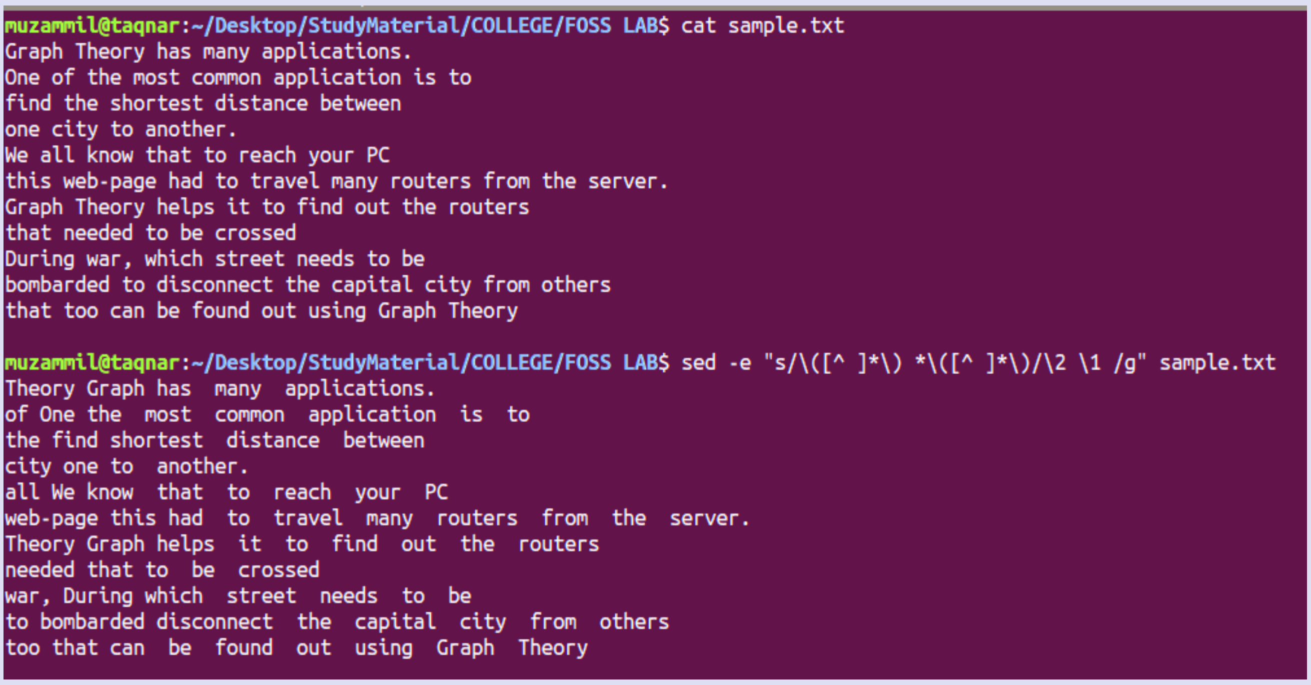
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat sample.txt
Graph Theory has many applications.
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
Graph Theory helps it to find out the routers
that needed to be crossed
During war, which street needs to be
bombarded to disconnect the capital city from others
that too can be found out using Graph Theory

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sed 's/.$//' sample.txt
Graph Theory has many applications
One of the most common application is t
find the shortest distance betwee
one city to another
We all know that to reach your P
this web-page had to travel many routers from the server
Graph Theory helps it to find out the router
that needed to be crosse
During war, which street needs to b
bombarded to disconnect the capital city from other
that too can be found out using Graph Theor

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

(c) ***Write a sed command that swaps the first and second words in each line in a file ?***

Shell Script

```
sed -e "s/\([^ ]*\) *\([^ ]*\)/\2 \1 /g" sample.txt
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat sample.txt
Graph Theory has many applications.
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
Graph Theory helps it to find out the routers
that needed to be crossed
During war, which street needs to be
bombarded to disconnect the capital city from others
that too can be found out using Graph Theory

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sed -e "s/\([^ ]*\) *\([^ ]*\)/\2 \1 /g" sample.txt
Theory Graph has  many  applications.
of One the  most  common  application  is  to
the find shortest  distance  between
city one to  another.
all We know  that  to  reach  your  PC
web-page this had  to  travel  many  routers  from  the  server.
Theory Graph helps  it  to  find  out  the  routers
needed that to  be  crossed
war, During which  street  needs  to  be
to bombarded disconnect  the  capital  city  from  others
too that can  be  found  out  using  Graph  Theory
```

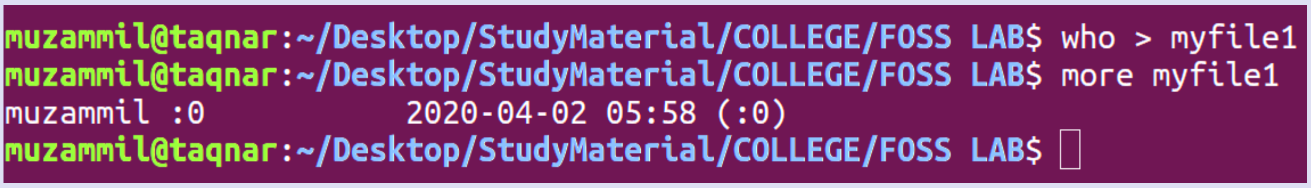
Question 2

(a) ***Use the who command and redirect the result to a file called myfile1. Use the more command to see the contents of myfile1. ?***

Shell Script

```
who > myfile1
more myfile1
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ who > myfile1
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ more myfile1
muzammil :0           2020-04-02 05:58 (:0)
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

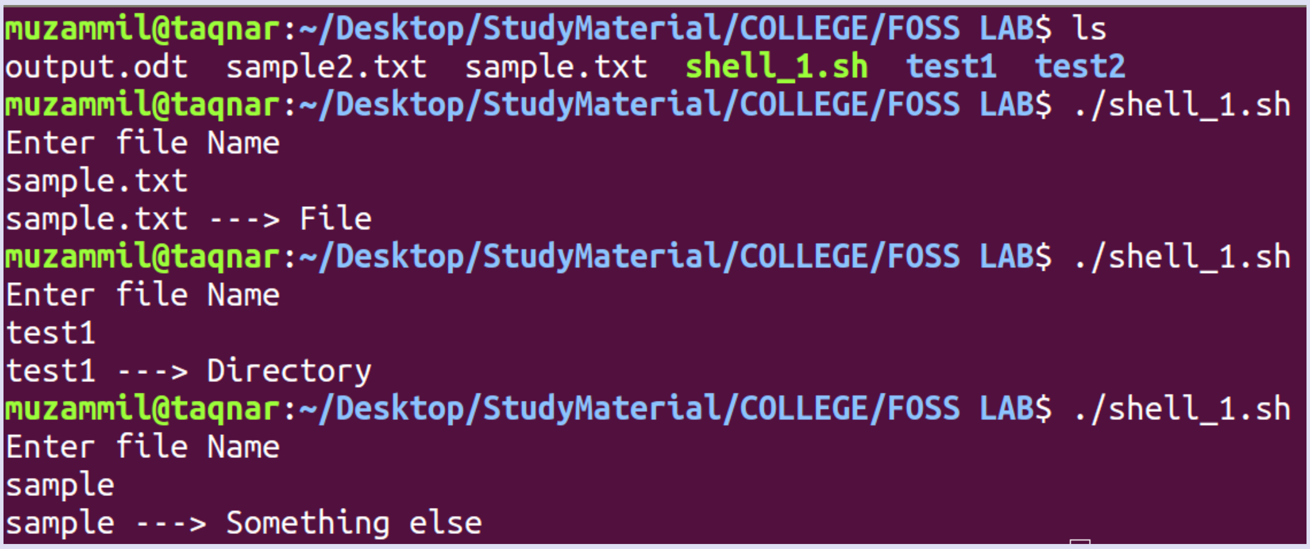
Question 3

(a) ***Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else ?***

Shell Script

```
echo "Enter file Name"
read file
if [ -f "${file}" ];
then
        echo  $file "---> File"
elif [ -d "${file}" ];
then
        echo $file "---> Directory"
else
        echo $file "---> Something else"
fi
```

Sample Output



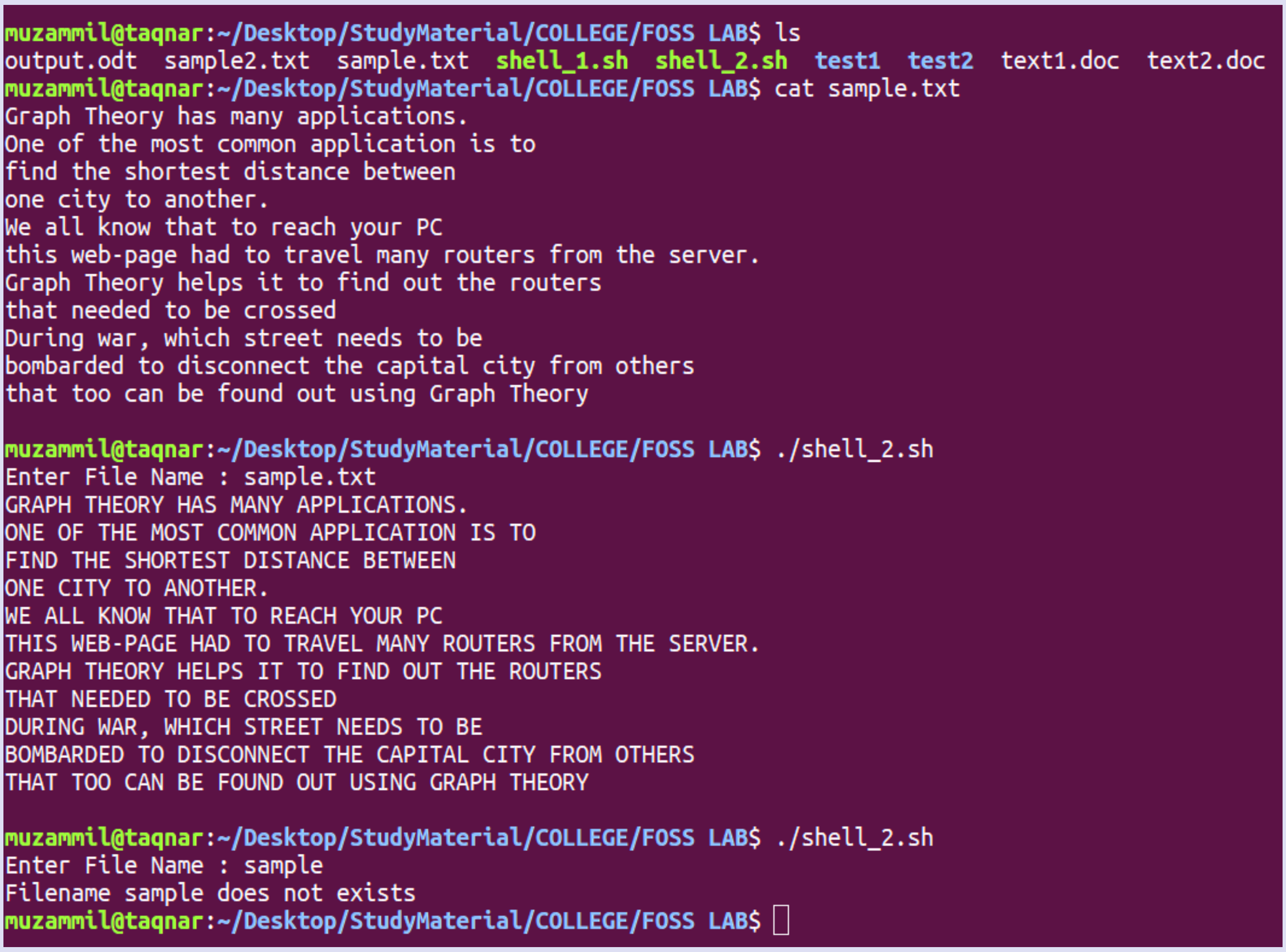
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ls
output.odt  sample2.txt  sample.txt  shell_1.sh  test1  test2
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_1.sh
Enter file Name
sample.txt
sample.txt ---> File
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_1.sh
Enter file Name
test1
test1 ---> Directory
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_1.sh
Enter file Name
sample
sample ---> Something else
```

(b) ***Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory ?***

Shell Script

```
echo -n "Enter File Name : "
read fileName
if [ ! -f "${fileName}" ];
then
        echo "Filename $fileName does not exists"
        exit 1
fi
tr '[a-z]' '[A-Z]' < "${fileName}"
```

Sample Output



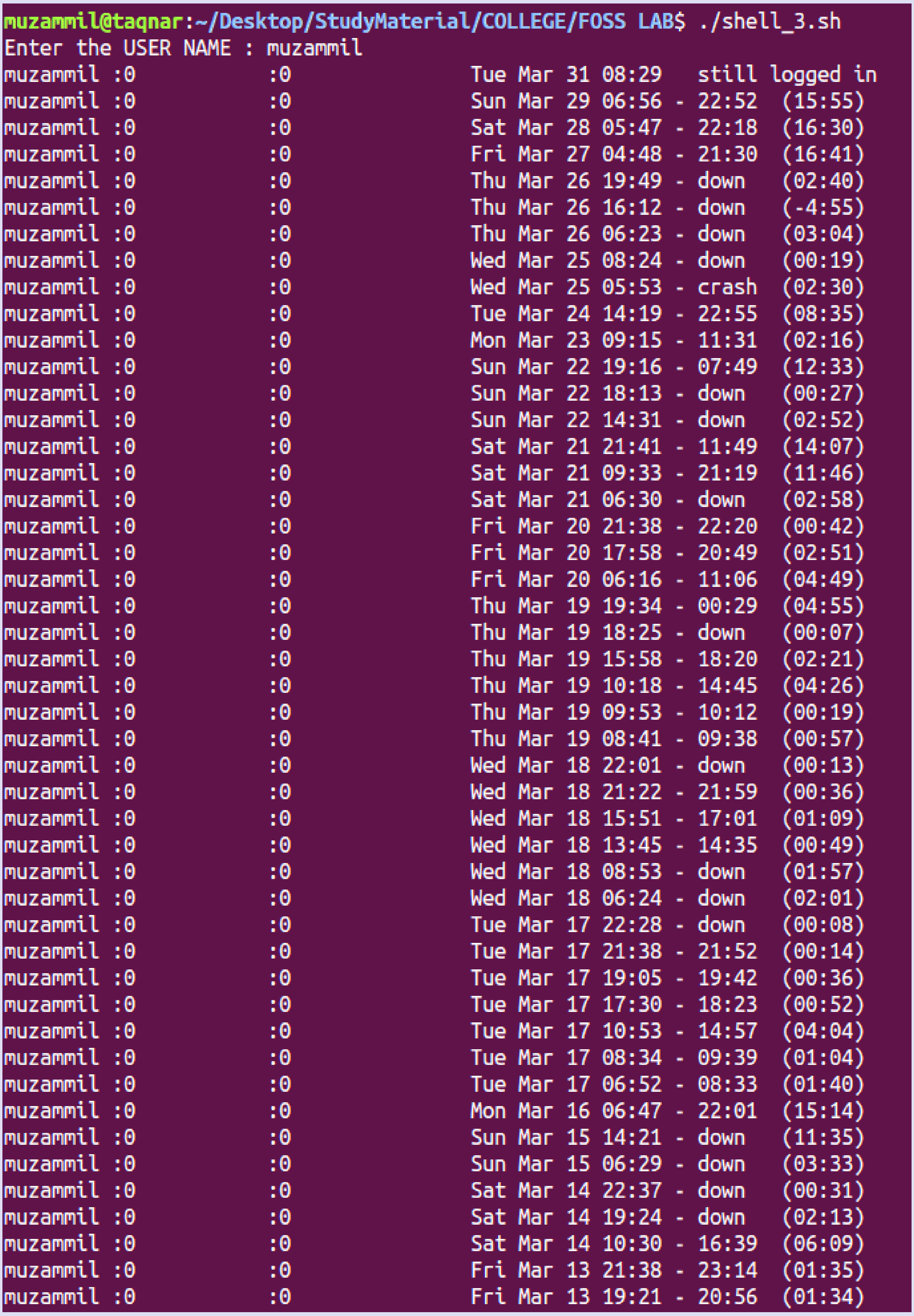
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ls
output.odt  sample2.txt  sample.txt  shell_1.sh  shell_2.sh  test1  test2  text1.doc  text2.doc
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat sample.txt
Graph Theory has many applications.
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
Graph Theory helps it to find out the routers
that needed to be crossed
During war, which street needs to be
bombarded to disconnect the capital city from others
that too can be found out using Graph Theory

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_2.sh
Enter File Name : sample.txt
GRAPH THEORY HAS MANY APPLICATIONS.
ONE OF THE MOST COMMON APPLICATION IS TO
FIND THE SHORTEST DISTANCE BETWEEN
ONE CITY TO ANOTHER.
WE ALL KNOW THAT TO REACH YOUR PC
THIS WEB-PAGE HAD TO TRAVEL MANY ROUTERS FROM THE SERVER.
GRAPH THEORY HELPS IT TO FIND OUT THE ROUTERS
THAT NEEDED TO BE CROSSED
DURING WAR, WHICH STREET NEEDS TO BE
BOMBARDED TO DISCONNECT THE CAPITAL CITY FROM OTHERS
THAT TOO CAN BE FOUND OUT USING GRAPH THEORY

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_2.sh
Enter File Name : sample
Filename sample does not exists
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

(c) ***Write a shell script that determines the period for which a specified user is working on the system ?***

Shell Script

```
echo -n  "Enter the USER NAME : "
read  user
last  $user
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_3.sh
Enter the USER NAME : muzammil
muzammil :0          :0               Tue Mar 31 08:29   still logged in
muzammil :0          :0               Sun Mar 29 06:56 - 22:52  (15:55)
muzammil :0          :0               Sat Mar 28 05:47 - 22:18  (16:30)
muzammil :0          :0               Fri Mar 27 04:48 - 21:30  (16:41)
muzammil :0          :0               Thu Mar 26 19:49 - down   (02:40)
muzammil :0          :0               Thu Mar 26 16:12 - down   (-4:55)
muzammil :0          :0               Thu Mar 26 06:23 - down   (03:04)
muzammil :0          :0               Wed Mar 25 08:24 - down   (00:19)
muzammil :0          :0               Wed Mar 25 05:53 - crash  (02:30)
muzammil :0          :0               Tue Mar 24 14:19 - 22:55  (08:35)
muzammil :0          :0               Mon Mar 23 09:15 - 11:31  (02:16)
muzammil :0          :0               Sun Mar 22 19:16 - 07:49  (12:33)
muzammil :0          :0               Sun Mar 22 18:13 - down   (00:27)
muzammil :0          :0               Sun Mar 22 14:31 - down   (02:52)
muzammil :0          :0               Sat Mar 21 21:41 - 11:49  (14:07)
muzammil :0          :0               Sat Mar 21 09:33 - 21:19  (11:46)
muzammil :0          :0               Sat Mar 21 06:30 - down   (02:58)
muzammil :0          :0               Fri Mar 20 21:38 - 22:20  (00:42)
muzammil :0          :0               Fri Mar 20 17:58 - 20:49  (02:51)
muzammil :0          :0               Fri Mar 20 06:16 - 11:06  (04:49)
muzammil :0          :0               Thu Mar 19 19:34 - 00:29  (04:55)
muzammil :0          :0               Thu Mar 19 18:25 - down   (00:07)
muzammil :0          :0               Thu Mar 19 15:58 - 18:20  (02:21)
muzammil :0          :0               Thu Mar 19 10:18 - 14:45  (04:26)
muzammil :0          :0               Thu Mar 19 09:53 - 10:12  (00:19)
muzammil :0          :0               Thu Mar 19 08:41 - 09:38  (00:57)
muzammil :0          :0               Wed Mar 18 22:01 - down   (00:13)
muzammil :0          :0               Wed Mar 18 21:22 - 21:59  (00:36)
muzammil :0          :0               Wed Mar 18 15:51 - 17:01  (01:09)
muzammil :0          :0               Wed Mar 18 13:45 - 14:35  (00:49)
muzammil :0          :0               Wed Mar 18 08:53 - down   (01:57)
muzammil :0          :0               Wed Mar 18 06:24 - down   (02:01)
muzammil :0          :0               Tue Mar 17 22:28 - down   (00:08)
muzammil :0          :0               Tue Mar 17 21:38 - 21:52  (00:14)
muzammil :0          :0               Tue Mar 17 19:05 - 19:42  (00:36)
muzammil :0          :0               Tue Mar 17 17:30 - 18:23  (00:52)
muzammil :0          :0               Tue Mar 17 10:53 - 14:57  (04:04)
muzammil :0          :0               Tue Mar 17 08:34 - 09:39  (01:04)
muzammil :0          :0               Tue Mar 17 06:52 - 08:33  (01:40)
muzammil :0          :0               Mon Mar 16 06:47 - 22:01  (15:14)
muzammil :0          :0               Sun Mar 15 14:21 - down   (11:35)
muzammil :0          :0               Sun Mar 15 06:29 - down   (03:33)
muzammil :0          :0               Sat Mar 14 22:37 - down   (00:31)
muzammil :0          :0               Sat Mar 14 19:24 - down   (02:13)
muzammil :0          :0               Sat Mar 14 10:30 - 16:39  (06:09)
muzammil :0          :0               Fri Mar 13 21:38 - 23:14  (01:35)
muzammil :0          :0               Fri Mar 13 19:21 - 20:56  (01:34)
```

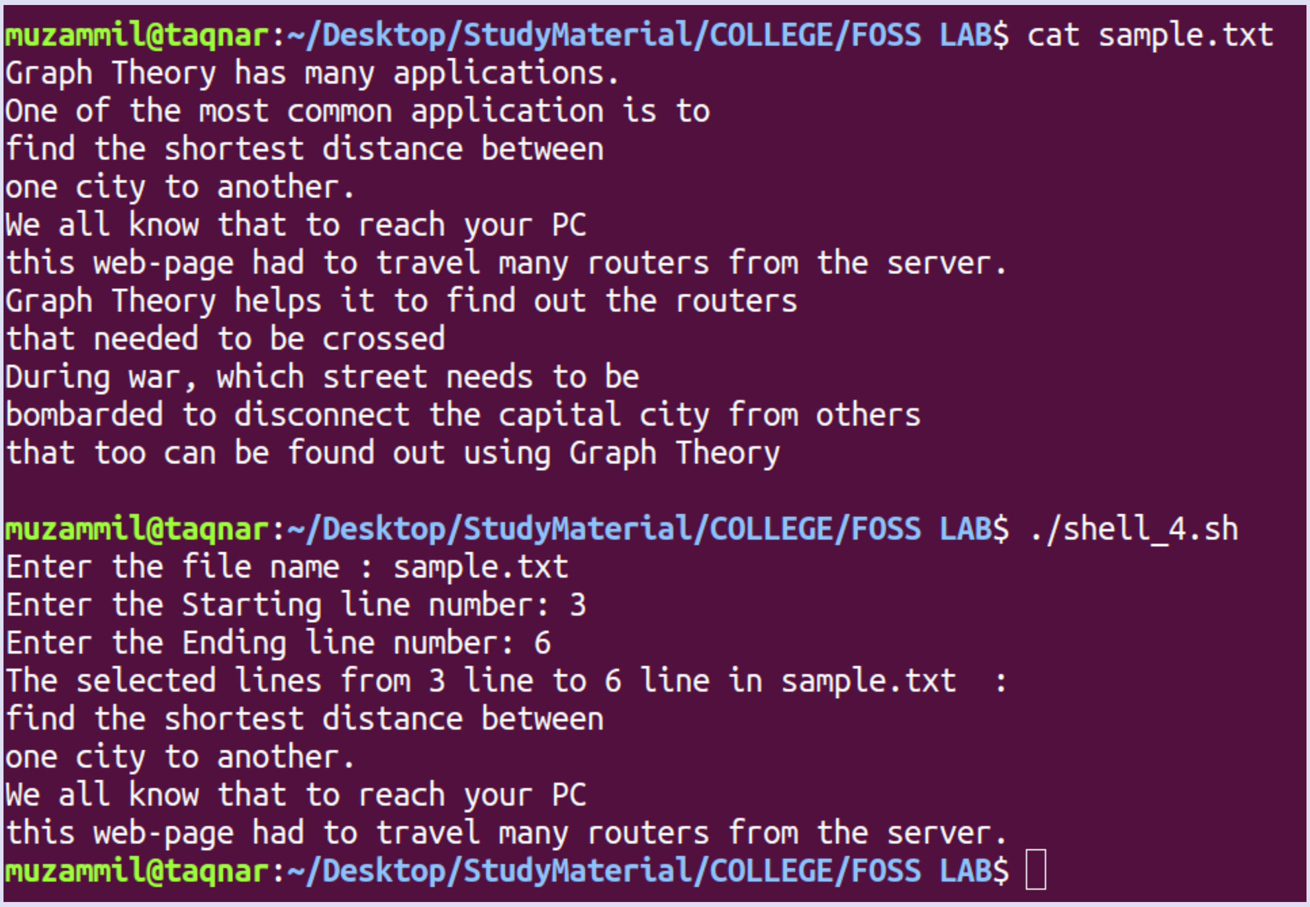
Question 4

(a) ***Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers. ?***

Shell Script

```
echo -n "Enter the file name : "
read  file
if [ -f "${file}" ];
then
        echo -n "Enter the Starting line number: "
        read  starting_num
        echo -n "Enter the Ending line number: "
        read  ending_num
        if  [ "${starting_num}" -lt  "${ending_num}" ];
        then
                echo  "The selected lines from $starting_num line to $ending_num
                ↪  line in $file  :"
                sed -n "$starting_num,$ending_num p" $file
        else
                echo  "Enter proper starting & ending line numbers."
        fi
else
        echo  "The file ' $file ' doesn't exists. "
fi
```

Sample Output



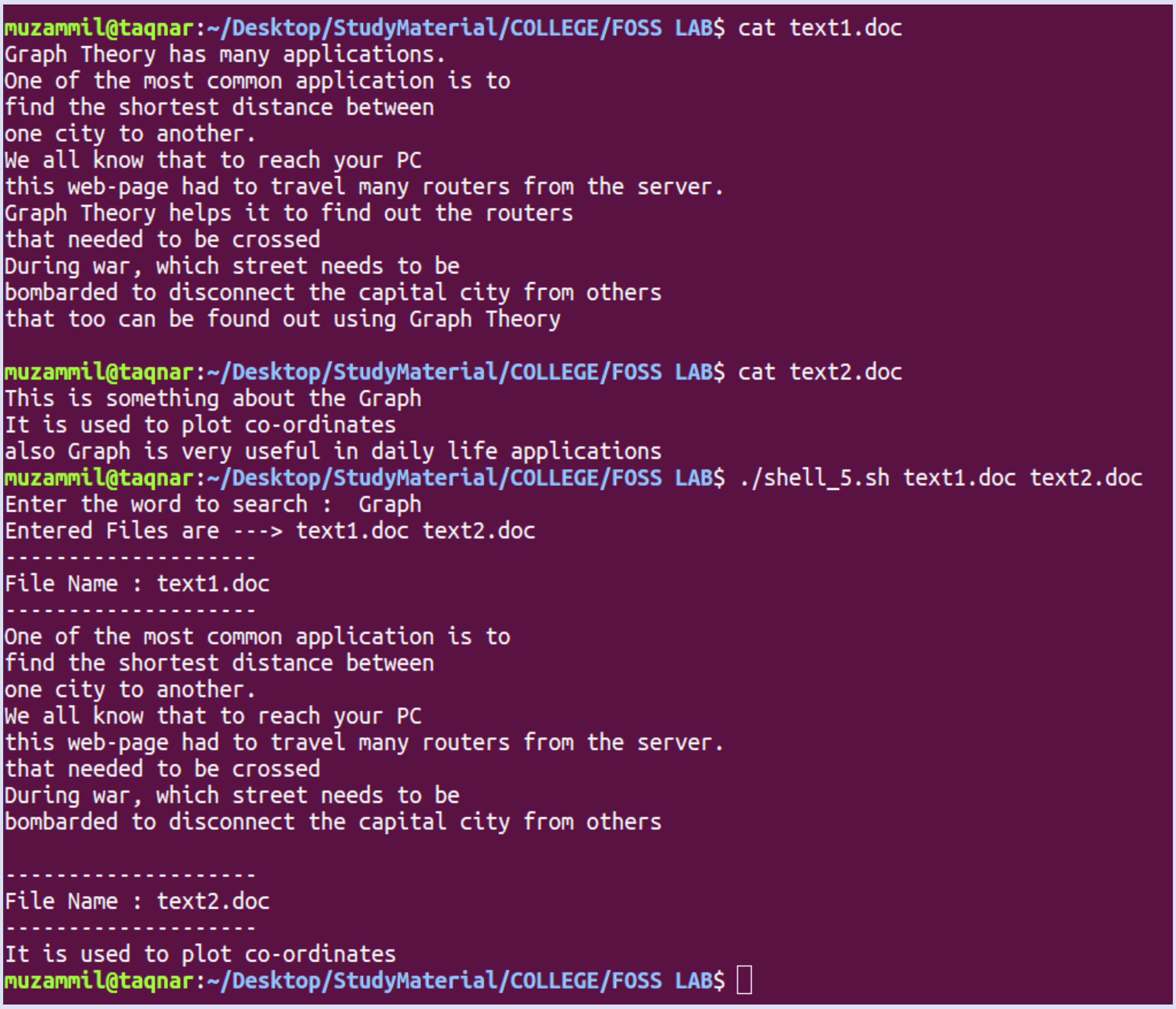
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat sample.txt
Graph Theory has many applications.
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
Graph Theory helps it to find out the routers
that needed to be crossed
During war, which street needs to be
bombarded to disconnect the capital city from others
that too can be found out using Graph Theory

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_4.sh
Enter the file name : sample.txt
Enter the Starting line number: 3
Enter the Ending line number: 6
The selected lines from 3 line to 6 line in sample.txt  :
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

(b) ***Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it ?***

Shell Script

```
echo -n "Enter the word to search :  "
read  word
echo  "Entered Files are ---> $*"
for i  in  $*
do
        echo -e  "--------------------\nFile Name : $i\n--------------------"
        grep -v $word $i
done
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat text1.doc
Graph Theory has many applications.
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
Graph Theory helps it to find out the routers
that needed to be crossed
During war, which street needs to be
bombarded to disconnect the capital city from others
that too can be found out using Graph Theory

muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat text2.doc
This is something about the Graph
It is used to plot co-ordinates
also Graph is very useful in daily life applications
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_5.sh text1.doc text2.doc
Enter the word to search :  Graph
Entered Files are ---> text1.doc text2.doc
--------------------
File Name : text1.doc
--------------------
One of the most common application is to
find the shortest distance between
one city to another.
We all know that to reach your PC
this web-page had to travel many routers from the server.
that needed to be crossed
During war, which street needs to be
bombarded to disconnect the capital city from others

--------------------
File Name : text2.doc
--------------------
It is used to plot co-ordinates
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

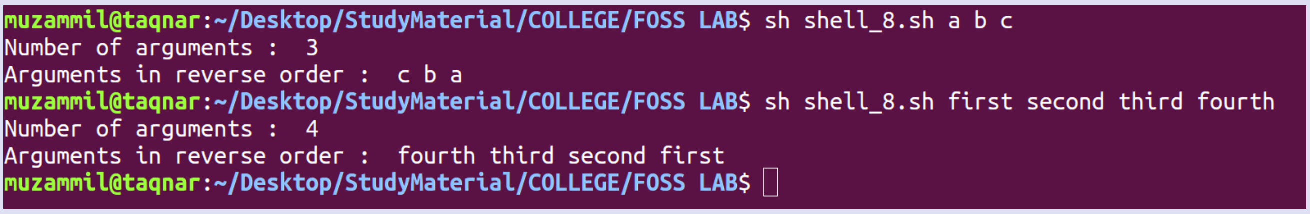
Question 5

(a) ***Write a shell script which accepts any number of arguments and prints them in the reverse order ?***

Shell Script

```
a=$#
echo "Number of arguments : " $a
x=$*
c=$a
res=''
while [ 1 -le $c ]
do
        c=`expr $c - 1`
        shift $c
        res=$res' '$1
        set $x
done
echo "Arguments in reverse order : " $res
```

Sample Output



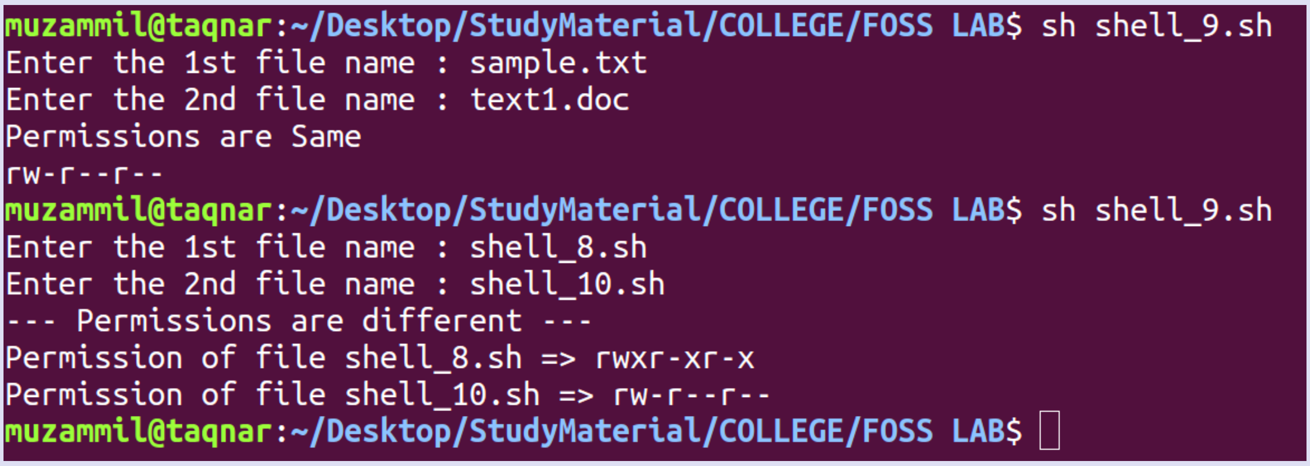
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_8.sh a b c
Number of arguments :  3
Arguments in reverse order :  c b a
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_8.sh first second third fourth
Number of arguments :  4
Arguments in reverse order :  fourth third second first
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

(b) ***Write a shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each file name followed by its permissions. ?***

Shell Script

```
echo -n "Enter the 1st file name : "
read f1
echo -n "Enter the 2nd file name : "
read f2
p1=`ls -l $f1 | cut -c 2-10`
p2=`ls -l $f2 | cut -c 2-10`
if [ $p1 = $p2 ];
then
        echo "Permissions are Same"
        echo $p1
else
        echo "--- Permissions are different ---"
        echo "Permission of file "${f1}" => "${p1}""
        echo "Permission of file "${f2}" => "${p2}""
fi
```

Sample Output



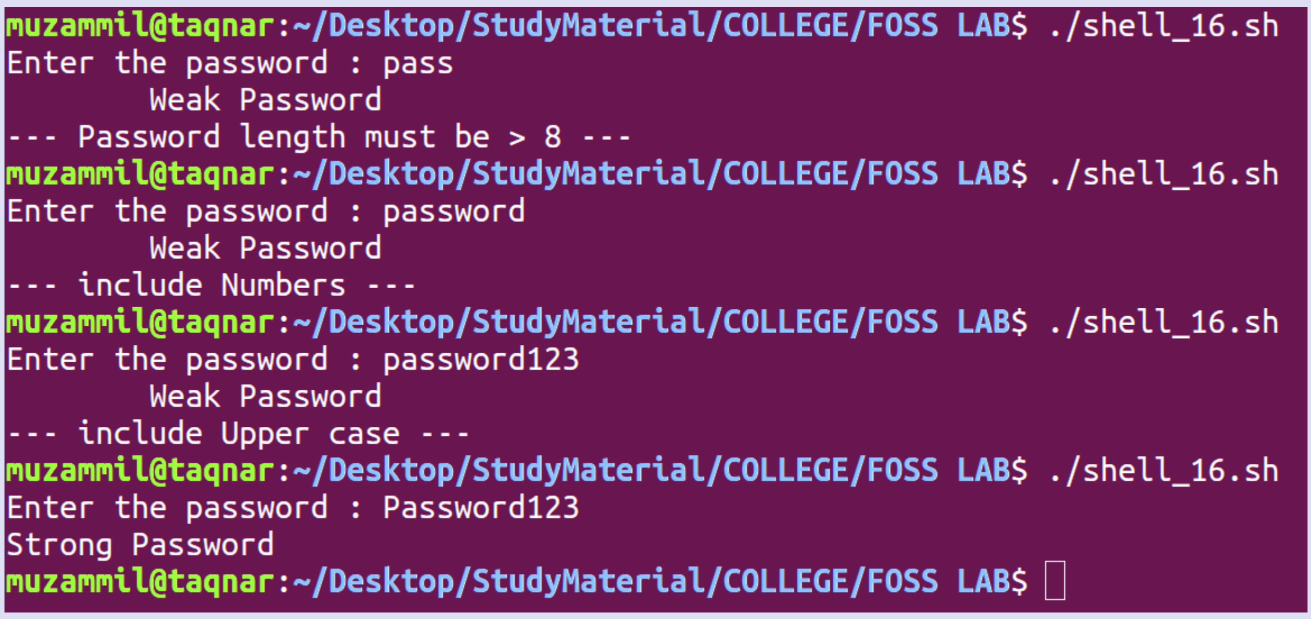
```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_9.sh
Enter the 1st file name : sample.txt
Enter the 2nd file name : text1.doc
Permissions are Same
rw-r--r--
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_9.sh
Enter the 1st file name : shell_8.sh
Enter the 2nd file name : shell_10.sh
--- Permissions are different ---
Permission of file shell_8.sh => rwxr-xr-x
Permission of file shell_10.sh => rw-r--r--
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

(c) ***Write a shell script to validate password strength. Here are a few assumptions for the password string.***
** Length – minimum of 8 characters.*
** Contain both alphabet and number.*
** Include both the small and capital case letters.*
If the password doesn't comply with any of the above conditions, then the script should report it as a <Weak Password>

Shell Script

```
echo -n "Enter the password : "
read password
len="${#password}"
if test $len -ge 8 ; then
        echo "$password" | grep -q [0-9]
        if test $? -eq 0 ; then
                echo "$password" | grep -q [A-Z]
                if test $? -eq 0 ; then
                        echo "$password" | grep -q [a-z]
                        if test $? -eq 0 ; then
                                echo "Strong Password"
                        else
                                echo -e "\tWeak Password\n--- include lower case ---"
                        fi
                else
                        echo -e "\tWeak Password\n--- include Upper case ---"
                fi
        else
                echo -e "\tWeak Password\n--- include Numbers ---"
        fi
else
        echo -e "\tWeak Password\n--- Password length must be > 8 ---"
fi
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_16.sh
Enter the password : pass
        Weak Password
--- Password length must be > 8 ---
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_16.sh
Enter the password : password
        Weak Password
--- include Numbers ---
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_16.sh
Enter the password : password123
        Weak Password
--- include Upper case ---
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_16.sh
Enter the password : Password123
Strong Password
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

FOSS LAB PRACTICE QUESTIONS : SET #2

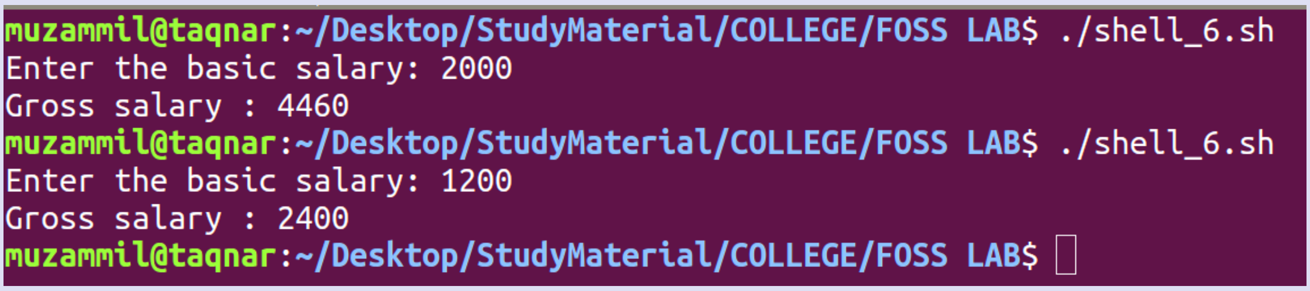
Question 1

Write a shell script that computes the gross salary of a employee according to the following rules
i) if basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.
ii) If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic..

Shell Script

```
echo -n "Enter the basic salary: "
read basic_salary
if [ $basic_salary -lt 1500 ];
then
        # gross salary = basic + basic * %(HRA) + basic * %(DA))
        gross_salary=$(( basic_salary + ((basic_salary/100)*10) +
        ↪  (basic_salary/100)*90 ))
        echo "Gross salary : $gross_salary"
fi
if [ $basic_salary -ge 1500 ];
then
        # gross salary = basic + HRA(in rs) + basic * %(DA))
        gross_salary=$(((basic_salary+500)+(basic_salary/100)*98))
        echo "Gross salary : $gross_salary"
fi
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_6.sh
Enter the basic salary: 2000
Gross salary : 4460
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_6.sh
Enter the basic salary: 1200
Gross salary : 2400
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

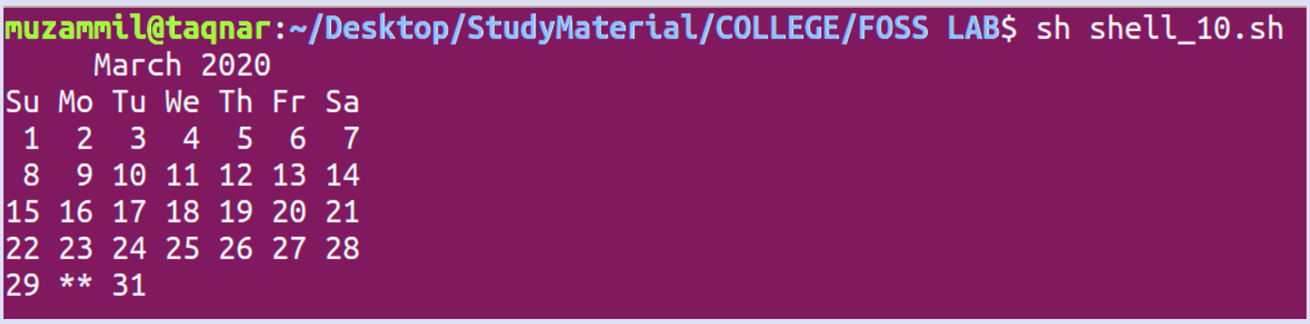
Question 2

*Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one digit or two digits.*

Shell Script

```
d=`date +%d`
cal > cal1
if [ $d -le 9 ]
then
        sed 's/'$d'/*/' cal1
        exit
fi
sed 's/'$d'/**/' cal1
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_10.sh
     March 2020
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 ** 31
```

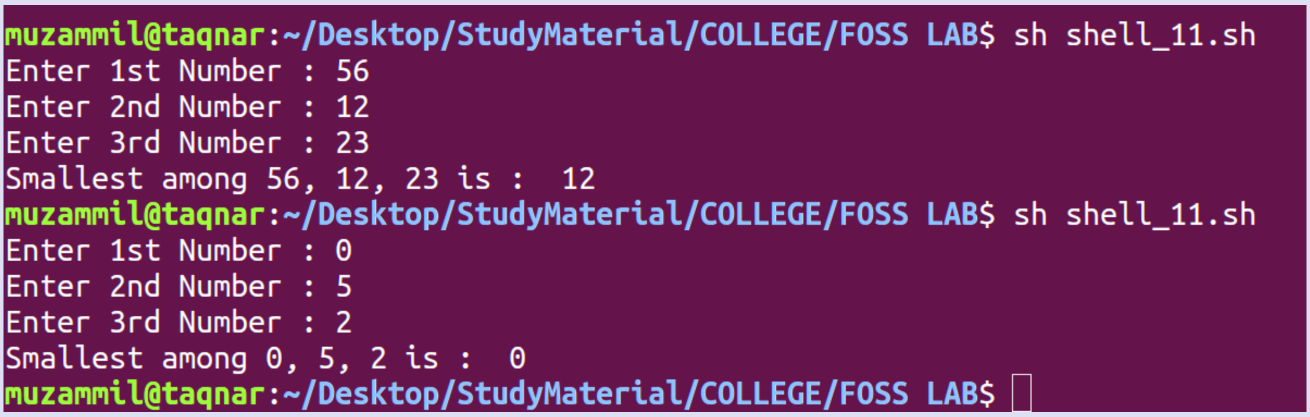
Question 3

Write a shell script to find smallest of 3 numbers that are read from keyboard ?

Shell Script

```
echo -n "Enter 1st Number : "
read a
echo -n "Enter 2nd Number : "
read b
echo -n "Enter 3rd Number : "
read c
if [ $a -eq $b -a $b -eq $c ];
then
        echo "All Numbers are Equal"
        exit
fi
if [ $a -lt $b ];
then
        s1=$a
        s2=$b
else
        s1=$b
        s2=$a
fi
if [ $s1 -gt $c ];
then
        s2=$s1
        s1=$c
fi
echo "Smallest among $a, $b, $c is : " $s1
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_11.sh
Enter 1st Number : 56
Enter 2nd Number : 12
Enter 3rd Number : 23
Smallest among 56, 12, 23 is :  12
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_11.sh
Enter 1st Number : 0
Enter 2nd Number : 5
Enter 3rd Number : 2
Smallest among 0, 5, 2 is :  0
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

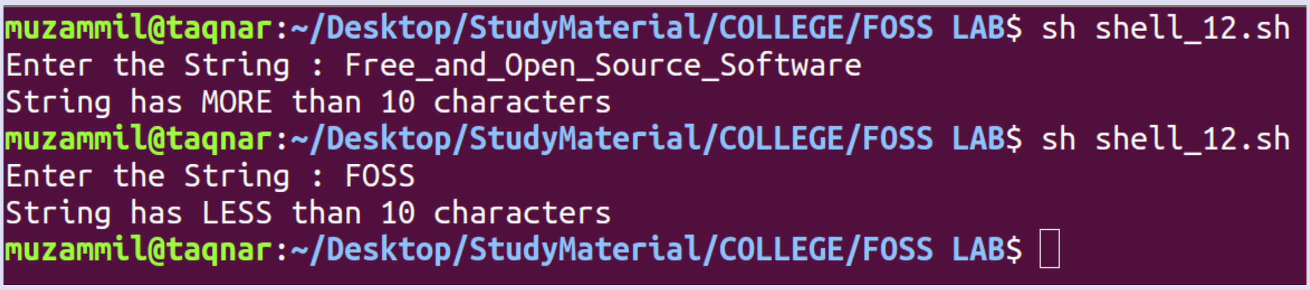
Question 4

Write a shell script using expr command to read in a string and display a suitable message if it does not have at least 10 characters.

Shell Script

```
echo -n "Enter the String : "
read str
l=`expr length $str`
if [ $l -gt 10 ];
then
        echo "String has MORE than 10 characters"
else
        echo "String has LESS than 10 characters"
fi
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_12.sh
Enter the String : Free_and_Open_Source_Software
String has MORE than 10 characters
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_12.sh
Enter the String : FOSS
String has LESS than 10 characters
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

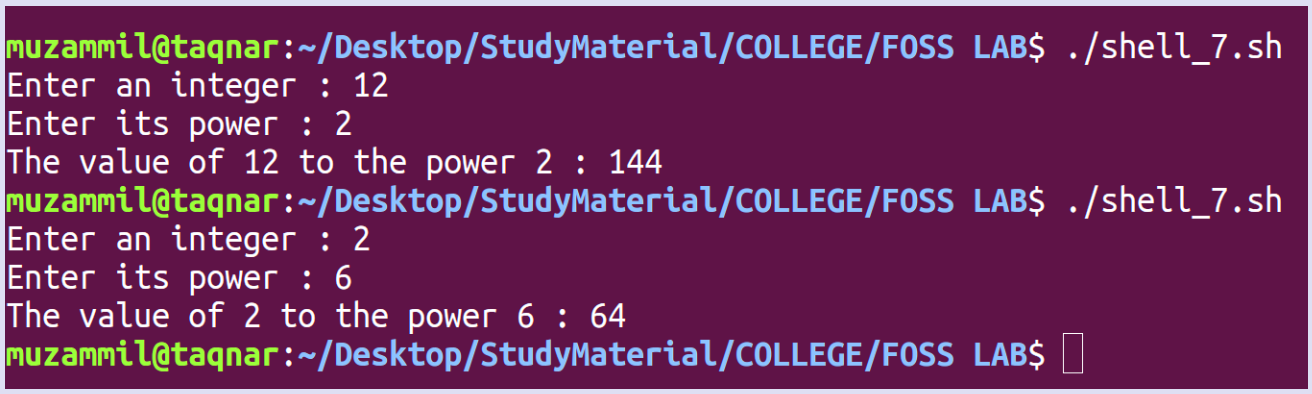
Question 5

Write a shell script that accepts two integers as its arguments and computes the value of first number raised to the power of the second number.

Shell Script

```
echo -n "Enter an integer : "
read num
echo -n "Enter its power : "
read pow
result=$num
i=1
while [ $i -lt $pow ]
do
        result=`expr $result \* $num`
        i=`expr $i + 1 `
done
echo "The value of $num to the power $pow : $result"
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_7.sh
Enter an integer : 12
Enter its power : 2
The value of 12 to the power 2 : 144
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_7.sh
Enter an integer : 2
Enter its power : 6
The value of 2 to the power 6 : 64
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

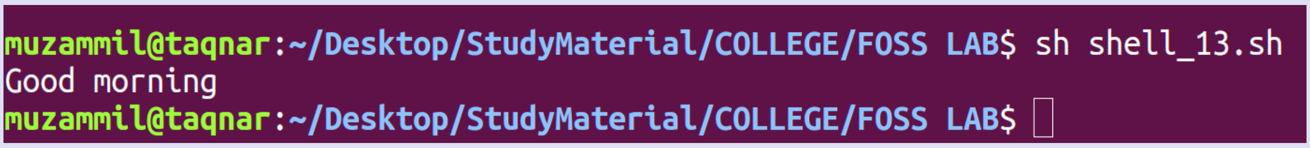
Question 6

Write a shell script that gets executed displays the message either "Good Morning" or "Good Afternoon " or "Good Evening" depending upon the time at which user logs in.

Shell Script

```
hournow=`date +%H`
if [ $hournow -ge 06 -a $hournow -le 12 ]
then
        echo "Good morning"      # 6:00am to 12:00pm
elif [ $hournow -ge 12 -a $hournow -le 17 ]
then
        echo "Good afternoon"    # 12:00pm to 5:00pm
else
        echo "Good evening"      # 5:00pm to 6:00am
fi
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_13.sh
Good morning
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

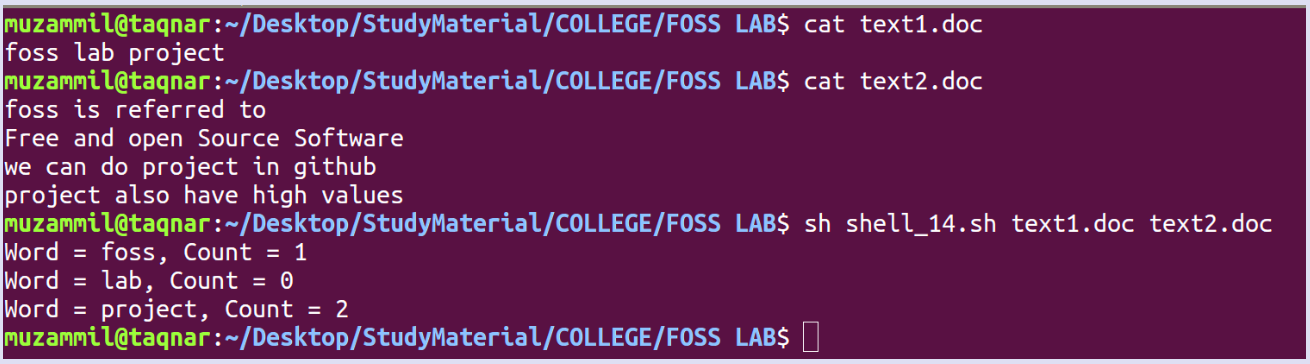
Question 7

A shell script that accepts a list of filenames as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files ?

Shell Script

```
if [ $# -ne 2 ];
then
        echo "Error : Invalid number of arguments."
        exit
fi
str=`cat $1 | tr '\n' ' '`
for a in $str
do
        echo "Word = $a, Count = `grep -c "$a" $2`"
done
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat text1.doc
foss lab project
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ cat text2.doc
foss is referred to
Free and open Source Software
we can do project in github
project also have high values
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ sh shell_14.sh text1.doc text2.doc
Word = foss, Count = 1
Word = lab, Count = 0
Word = project, Count = 2
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$
```

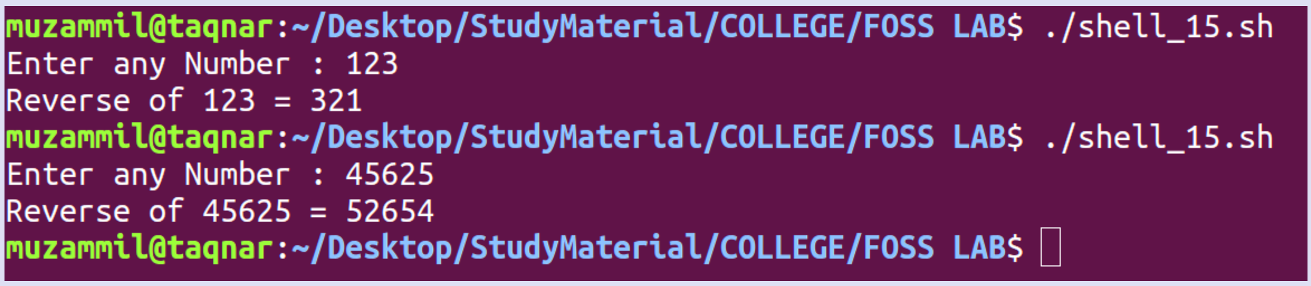
Question 8

Write a shell script to print a number in reverse order.

Shell Script

```
echo -n "Enter any Number : "
read num
digit=0
rev=0
orginal=$num
while [ $num -gt 0 ]
do
    digit=$(( $num % 10 ))          # Getting the last digit
    rev=$(( $rev * 10 + $digit )) #reversing by multiplying by weight
    num=$(( $num / 10 ))            # num is modifying
done

echo "Reverse of $orginal = $rev"
```

Sample Output



```
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_15.sh
Enter any Number : 123
Reverse of 123 = 321
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ ./shell_15.sh
Enter any Number : 45625
Reverse of 45625 = 52654
muzammil@taqnar:~/Desktop/StudyMaterial/COLLEGE/FOSS LAB$ 
```

Result

The Shell programs are done as per the given questions and observed the Results.